

Volumen I de la Documentación de Blender - Guía de Usuario

Last modified December 07 2004 S68

Claudio Andaur

Manuel Bastioni

Alejandro Conty Estevez

Karsten Dambekalns

Florian Findeiss

Alex Heizer

Tim van Howe

Wouter van Heyst

Joeri Kassenaar

Martin Kleppman

Lyubomir Kovacev

Johnny Matthews

Reevan McKay

Kent Mein

Martin Middleton

Jason Oppel

Willem-Paul van Overbruggen

Ton Roosendaal

Bastian Samela

Stefano Selleri

Bart Veldhuizen

Chris Williamson

Carsten Wartmann

Copyright © 2002 Stichting Blender Foundation. Se concede el permiso para copiar, distribuir y/o modificar este documento bajo los términos de la licencia de contenido abierto (Open Content License). Una copia de la licencia se incluye en el apéndice.

Esta es Guía oficial del proyecto de documentación de Blender. Sientase libre para añadir o modificar la guía y enviar sus cambios a la lista de correo del proyecto de documentación de Blender (bf-docboard@blender.org).

Table of Contents

I. [Introducción a Blender](#)

1. [Introducción](#)

1.1. [¿Qué es Blender?](#)

1.2. [La historia de Blender](#)

1.3. [Sobre el Software Libre y la licencia GPL](#)

1.4. [Consigue ayuda - la comunidad de usuarios de Blender](#)

2. [Instalación](#)

2.1. [Descarga y ejecución de Blender](#)

2.2. [Instalando Blender a partir del código fuente](#)

3. [Comprendiendo la interfaz](#)

3.1. [Conceptos de la Interface de Blender](#)

3.2. [Navegando en el Espacio 3D](#)

3.3. [Funciones fundamentales](#)

4. [Su primera animación en 30 + 30 minutos](#)

4.1. [Calentando](#)

4.2. [Construyendo el cuerpo](#)

4.3. [Veamos como se ve Gus](#)

4.4. [Materiales y Texturas](#)

4.5. [Construyendo un esqueleto \(Rigging\)](#)

4.6. [Asignación del esqueleto \(Skinning\)](#)

4.7. [Posando](#)

4.8. [¡Gus anda!](#)

II. [Modelado, Materiales e Iluminación](#)

5. [Modo Objeto](#)

5.1. [Selección de objetos](#)

5.2. [Movimiento \(traslación\) de objetos](#)

5.3. [Rotación de objetos](#)

5.4. [Cambio de tamaño/imagen especular de objetos](#)

5.5. [Panel de propiedades de transformación](#)

5.6. [Duplicación](#)

5.7. [Emparentar \(Agrupar\)](#)

5.8. [Seguimiento \(Tracking\)](#)

5.9. [Otras Acciones](#)

5.10. [Operaciones booleanas](#)

6. [Modelado de Malla básico](#)

6.1. [Objetos Básicos](#)

6.2. [Modo Edición](#)

6.3. [Suavizando \(Smoothing\)](#)

6.4. [Extrudado](#)

6.5. [Giro \(Spin\) y Giro duplicado \(SpinDup\)](#)

6.6. [Rosca \(Screw\)](#)

6.7. [Herramienta Curvar \(Warp\)](#)

6.8. [Garfios de Objeto \(Object Hooks\)](#)

7. [Modelado de Malla avanzado](#)

7.1. [Subdivisión de Superficies Catmull-Clark](#)

7.2. [Pliegues pesados para superficies de subdivision](#)

7.3. [Herramientas de Arista](#)

7.4. [Herramientas de sollevado \(Bevel\)](#)

7.5. [Modelado Simétrico](#)

7.6. [Herramienta de Edición Proporcional \(PET\)](#)

- 7.7. [Ruido \(Noise\)](#)
- 7.8. [Herramienta Diezmador \(Decimator\)](#)
- 8. [Los Meta Objects](#)
- 9. [Curvas y Superficies](#)
 - 9.1. [Curvas](#)
 - 9.2. [Superficies](#)
 - 9.3. [Texto](#)
 - 9.4. [Extruir a lo largo de una trayectoria](#)
 - 9.5. [Curva Taper](#)
 - 9.6. [Curva Deformar](#)
 - 9.7. [Skinning](#)
- 10. [Materiales](#)
 - 10.1. [Difusión](#)
 - 10.2. [Reflexión especular](#)
 - 10.3. [Materiales en la práctica](#)
 - 10.4. [Sombreadores de Rampa \(Ramp Shaders\)](#)
 - 10.5. [Reflexiones con el Trazador de rayos](#)
 - 10.6. [Transparencias con el trazador de rayos](#)
 - 10.7. [Materiales Múltiples](#)
 - 10.8. [Materiales Especiales](#)
- 11. [Texturas](#)
 - 11.1. [Texturas](#)
 - 11.2. [Mapas de Ambiente \(Environment Maps\)](#)
 - 11.3. [Mapas de Desplazamiento](#)
 - 11.4. [Cristal Solido y Hueco](#)
 - 11.5. [EditorUV y Selección de Caras](#)
 - 11.6. [Desenrollando a Suzanne](#)
 - 11.7. [Plugins de Textura](#)
- 12. [Iluminación](#)
 - 12.1. [Introducción](#)
 - 12.2. [Tipos de luces](#)
 - 12.3. [Sombras con el trazador de rayos](#)
 - 12.4. [Sombras con el buffer](#)
 - 12.5. [Luz Volumétrica](#)
 - 12.6. [Ajustando la iluminación](#)
- 13. [El Mundo y el Universo](#)
 - 13.1. [El Fondo del Mundo](#)
 - 13.2. [Exposición](#)

- 13.3. [Niebla \(Mist\)](#)
- 13.4. [Estrellas](#)
- 13.5. [Oclusión Ambiental](#)

III. [Animación](#)

- 14. [Animación de Objetos No Deformables](#)
 - 14.1. [El bloque IPO](#)
 - 14.2. [Cuadros clave](#)
 - 14.3. [Las Curvas IPO](#)
 - 14.4. [Curvas IPO y claves IPO](#)
 - 14.5. [Otras aplicaciones de las Curvas IPO](#)
 - 14.6. [La Ipo del tiempo](#)
 - 14.7. [Trayectoria de la animación](#)
- 15. [Animación de Deformaciones](#)
 - 15.1. [Claves de Vértices Absolutas](#)
 - 15.2. [Claves de Vértices Relativas](#)
 - 15.3. [Animación de Retículas](#)
- 16. [Animación de Personajes](#)
 - 16.1. [Introducción: Luces, Cámaras y... ACCION !](#)
 - 16.2. [Herramientas Generales](#)
 - 16.3. [El Objeto Armature](#)
 - 16.4. [Skinning](#)
 - 16.5. [Modo Pose](#)
 - 16.6. [Ventana de Acciones](#)
 - 16.7. [Animación No Lineal](#)
 - 16.8. [Restricciones \(Constraints\)](#)
 - 16.9. [Rigging de Mano y Pie](#)
 - 16.10. [Rigging de elementos mecánicos](#)
 - 16.11. [Cómo hacer un ciclo de andar usando NLA](#)

IV. [Renderizado](#)

- 17. [Renderizado](#)
 - 17.1. [Renderizando por Partes](#)
 - 17.2. [Renders panorámicos](#)
 - 17.3. [Antialiasing](#)
 - 17.4. [Formatos de salida](#)
 - 17.5. [Renderizando animaciones](#)
 - 17.6. [Desenfoco de Movimiento \(Motion Blur\)](#)
 - 17.7. [Profundidad de Campo](#)
 - 17.8. [Bordes de Caricatura](#)

17.9. [El Renderizador Unificado \(Unified Renderer\)](#)

17.10. [Preparando su trabajo para vídeo](#)

18. [Radiosidad](#)

18.1. [El método de Radiosidad en Blender](#)

18.2. [Renderizando la Radiosidad](#)

18.3. [Radiosidad como herramienta de modelado](#)

18.4. [Un jugoso ejemplo de radiosidad](#)

19. [Trazado de rayos \(Raytracing\)](#)

V. [Herramientas Avanzadas](#)

20. [Efectos](#)

20.1. [Introducción](#)

20.2. [Partículas sencillas](#)

20.3. [Interacción entre Partículas](#)

21. [Otros Efectos](#)

21.1. [Introducción](#)

21.2. [Efecto Build](#)

21.3. [Efecto Onda](#)

22. [Técnicas especiales de modelado](#)

22.1. [Introducción](#)

22.2. [DupliVerts](#)

22.3. [DupliFrames](#)

22.4. [Modelando con lattices](#)

23. [Efectos Volumétricos](#)

24. [Cuerpos blandos](#)

24.1. [Introducción](#)

24.2. [El sistema SoftBody de Blender](#)

25. [Editor de secuencia](#)

25.1. [Conociendo el editor de secuencia](#)

25.2. [Editor de Secuencia de sonidos](#)

25.3. [Plugins del editor de secuencia](#)

VI. [Extendiendo Blender](#)

26. [Python](#)

26.1. [Un ejemplo funcional de Python](#)

26.2. [Referencia de Python](#)

26.3. [Scripts de Python](#)

27. [Sistema de plugins de Blender](#)

27.1. [Escribiendo un plugin de Textura](#)

27.2. [Especificación](#)

- 27.3. [Un plugin de textura genérico](#)
- 27.4. [Nuestras modificaciones](#)
- 27.5. [Compilando:](#)
- 27.6. [Escribiendo un plugin de Secuencia](#)

VII. [Más alla de Blender](#)

- 28. [Yafray como motor de render integrado en Blender](#)
 - 28.1. [Parte 1](#)
 - 28.2. [Parte 2](#)
 - 28.3. [Parte 3 \(Aún por traducir\)](#)
 - 28.4. [Glosario para los geeks \(Aún por traducir\)](#)
[Glosario para los geeks \(Aún por traducir\)](#)
- 29. [De Blender a Yafray usando YableX](#)
 - 29.1. [¿Qué es Yable?](#)
 - 29.2. [Instalando el script](#)
 - 29.3. [La interfaz](#)
 - 29.4. [Ejemplo de utilización de Yable](#)
- 30. [YafRay](#)
 - 30.1. [Introducción](#)
 - 30.2. [Instalación](#)
 - 30.3. [Repaso del Lenguaje de Descripción de Escenas](#)
 - 30.4. [Sombreadores](#)
 - 30.5. [Objectos renderizables](#)
 - 30.6. [Luces](#)
 - 30.7. [Fondo](#)
 - 30.8. [Cámara](#)
 - 30.9. [Renderizado](#)
 - 30.10. [Filtros](#)

[Glossary](#)

List of Tables

- 2-1. [Versión mínima de las librerías](#)
- 2-2. [Requerimientos de las librerías dependiendo del sistema utilizado](#)
- 2-3. [Mínima versión de las herramientas](#)
- 29-1. [IORs de ejemplo](#)

List of Figures

- 3-1. [La escena por defecto de Blender.](#)
- 3-2. [El menu de Split para crear ventanas nuevas.](#)
- 3-3. [El menú de selección del tipo de ventana.](#)

- 3-4. [Contextos y Sub-Contextos](#)
- 3-5. [Menu de la Ventana de Botones.](#)
- 3-6. [Panel con fichas.](#)
- 3-7. [Un botón de operación](#)
- 3-8. [Botones alternantes](#)
- 3-9. [Botones numéricos](#)
- 3-10. [Botones de enlace de DataBlocks](#)
- 3-11. [La Caja de Herramientas](#)
- 3-12. [Selectores de Pantalla y Escena](#)
- 3-13. [El menú de vistas de una ventana 3D.](#)
- 3-14. [Ortográfica \(izquierda\) y perspectiva \(derecha\).](#)
- 3-15. [Demostración de vista de Cámara.](#)
- 3-16. [Un botón de modo de dibujado de una vista 3D.](#)
- 3-17. [Los botones de Capas de una vista 3D.](#)
- 3-18. [Ventana de Selección de Archivos - cargando.](#)
- 3-19. [Ventana de Selección de Archivos - guardando.](#)
- 3-20. [Opciones de Renderizado en los Botones de Render.](#)
- 3-21. [Ventana de Preferencias del Usuario.](#)
- 4-1. [Blender recién iniciado.](#)
- 4-2. [Caja de herramientas para el control de capas.](#)
- 4-3. [Controles de visibilidad de las capas.](#)
- 4-4. [Nuestro cubo en Modo Edición, todos los vértices seleccionados.](#)
- 4-5. [El botón de la ventana de Edición.](#)
- 4-6. [Ventana de Edición para una malla.](#)
- 4-7. [El cubo, subdividido una vez.](#)
- 4-8. [La secuencia de seleccionar con una caja un grupo de vértices.](#)
- 4-9. [El menú emergente para borrar \(**Tecla X**\) en acción.](#)
- 4-10. [Extruyendo los brazos en dos pasos.](#)
- 4-11. [Medio cuerpo.](#)
- 4-12. [La ventana de botones de edición.](#)
- 4-13. [Configure el centro de referencia a el cursor.](#)
- 4-14. [Voltee la copia de la mitad del cuerpo para obtener un cuerpo completo.](#)
- 4-15. [La secuencia de añadir una cabeza.](#)
- 4-16. [La ventana de botones de Edición.](#)
- 4-17. [Haciendo a Gus más suave.](#)
- 4-18. [Adelgazando a Gus utilizando el escalado restringido.](#)
- 4-19. [Haciendo visibles las capas 1 y 10.](#)
- 4-20. [El panel para la entrada numérica de la posición/rotación del objeto.](#)

- 4-21. [La posición de la cámara con respecto a Gus.](#)
- 4-22. [Seleccione el centro de referencia al centro del objeto.](#)
- 4-23. [Insertando un lámpara.](#)
- 4-24. [El botón para el panel de lámparas.](#)
- 4-25. [Configuración de una luz Spot.](#)
- 4-26. [La configuración de la lámpara Hemi](#)
- 4-27. [El panel de botones de renderizado.](#)
- 4-28. [La ventana de botones de renderizado](#)
- 4-29. [Su primer renderizado. ¡Felicidades!](#)
- 4-30. [El menú para guardar.](#)
- 4-31. [El botón de la ventana de botones de materiales.](#)
- 4-32. [El botón en el menú de los materiales.](#)
- 4-33. [La ventana de botones del material y un primer material de gengibre.](#)
- 4-34. [Panel texturas en la ventana de materiales](#)
- 4-35. [El botón de la ventana de texturas.](#)
- 4-36. [La ventana de los botones de textura con una textura stucci.](#)
- 4-37. [Configuración para la textura Stucci en la ventana de botones del material.](#)
- 4-38. [Configuración para una textura adicional de ruido en el canal 2.](#)
- 4-39. [Un material muy simple para el suelo.](#)
- 4-40. [Los botones de visibilidad de las capas en la barra de herramientas.](#)
- 4-41. [Secuencia de creación de los ojos.](#)
- 4-42. [Creando una boca con herramientas de spinning \(revolución\).](#)
- 4-43. [Los botones de la herramienta Spin en la ventana de botones de Edición.](#)
- 4-44. [¡Gus completado!](#)
- 4-45. [Algunos materiales de caramelo.](#)
- 4-46. [Seleccionando un material existente en el menú de los materiales.](#)
- 4-47. [Renderizado estático de Gus una vez terminado.](#)
- 4-48. [Menú de selección del tipo de fichero en la ventana de los botones de renderizado.](#)
- 4-49. [Añadiendo el primer hueso del brazo.](#)
- 4-50. [Añadiendo el segundo y el tercer hueso, una cadena de huesos en la pierna.](#)
- 4-51. [El esqueleto completo después de duplicarlo y voltearlo.](#)
- 4-52. [la ventana de botones de edición para una armadura.](#)
- 4-53. [La ventana emerge cuando encajamos un Objeto a una esqueleto.](#)
- 4-54. [Opciones de Skinning automático.](#)
- 4-55. [Los botones del grupo de vértices en la ventana de botones de edición de una malla.](#)
- 4-56. [El menú con los grupos de vértices automaticamente creados en el proceso de skinning.](#)
- 4-57. [Gus en Modo Edición con todos los vértices del grupo Arm.R seleccionados.](#)
- 4-58. [Los seis grupos de vértices.](#)

- 4-59. [El botón para cambiar al modo posado en la barra de herramientas de la ventana 3D.](#)
- 4-60. [¡Ahora está en modo posado!](#)
- 4-61. [El frame actual en la barra de herramientas de la ventana de botones.](#)
- 4-62. [Nuestra primera pose.](#)
- 4-63. [Asignando la pose al frame.](#)
- 4-64. [Nuestra segunda pose.](#)
- 4-65. [Copiando la pose al buffer.](#)
- 4-66. [Pegar la copia como una nueva, y volteada, pose.](#)
- 4-67. [Configurando los botones de renderizado para una animación.](#)
- 5-1. [Los botones de selección de punto de rotación](#)
- 5-2. [El cuadro de diálogo numérico](#)
- 5-3. [Selección de grupo](#)
- 5-4. [Liberación de hijos](#)
- 5-5. [Configuración del eje de seguimiento](#)
- 5-6. [Selección de enlaces.](#)
- 5-7. [Opciones para operaciones booleanas](#)
- 5-8. [Objetos resultantes para: intersección, unión, diferencia \(de arriba a abajo\).](#)
- 6-1. [Objetos Basicos o Primitivas](#)
- 6-2. [Dos piramides, una en ModoEdición \(izquierda\) y una en Modo Objeto \(derecha\).](#)
- 6-3. [Cubo con los vértices seleccionados en amarillo.](#)
- 6-4. [Pirámide truncada](#)
- 6-5. [Botones de Edición](#)
- 6-6. [Menú Specials](#)
- 6-7. [Deshacer y Rehacer](#)
- 6-8. [Undo Menu](#)
- 6-9. [Objeto de prueba simple sin suavizar](#)
- 6-10. [Botones Set Smooth y Set Solid de la ventana de botones de edición](#)
- 6-11. [El mismo objeto pero completamente suavizado con 'Set Smooth'](#)
- 6-12. [Objeto en ModoEdición con algunas caras seleccionadas.](#)
- 6-13. [Grupo de botones AutoSmooth en la ventana de Botones de Edición.](#)
- 6-14. [La misma prueba con AutoSuavizado activado](#)
- 6-15. [Circulo deformado para generar la sección transversal de la hoja.](#)
- 6-16. [Botón Extrude en la ventana de Botones de Edición](#)
- 6-17. [Caja de confirmación de Extrude.](#)
- 6-18. [La Hoja](#)
- 6-19. [Menú Edición de Malla](#)
- 6-20. [La hoja completa](#)
- 6-21. [La UV sphere para la empuñadura: los vértices se eliminan](#)

- 6-22. [Primera extrusión de la empuñadura](#)
- 6-23. [Empuñadura completa](#)
- 6-24. [Guarnición Completa](#)
- 6-25. [Espada terminada, con texturas y materiales](#)
- 6-26. [Perfil del vaso](#)
- 6-27. [Botones de Spin](#)
- 6-28. [Perfil del vaso, vista superior en Modo Edición, justo antes de realizar el barrido.](#)
- 6-29. [Datos de malla - numero de caras y vertices.](#)
- 6-30. [Perfil barrido](#)
- 6-31. [Selección de vertices en la costura](#)
- 6-32. [Cuenta de vertices después de eliminar dobles.](#)
- 6-33. [El menú Merge](#)
- 6-34. [Render final de los vasos.](#)
- 6-35. [Marca de la hora indicada por la flecha](#)
- 6-36. [Botones Spin Dup](#)
- 6-37. [Malla seleccionada y lista para ser duplicada](#)
- 6-38. [Selección de la Vista para Spin Dup.](#)
- 6-39. [Eliminación de objetos duplicados](#)
- 6-40. [Render Final del Reloj.](#)
- 6-41. [Como hacer un resorte: antes \(izquierda\) y después \(derecha\) de aplicar el comando Screw.](#)
- 6-42. [Rosca creciente \(derecha\) obtenida con el perfil de la izquierda.](#)
- 6-43. [Parámetros del Texto](#)
- 6-44. [Top view of text and sphere](#)
- 6-45. [Texto curvado](#)
- 6-46. [Render Final](#)
- 6-47. [Menú de garfios](#)
- 6-48. [Menú extendido de garfios](#)
- 6-49. [Panel de Garfios](#)
- 7-1. [Botones SubSurf](#)
- 7-2. [SubSurfed Suzanne.](#)
- 7-3. [SubSurf de un cuadrado simple y caras rectangulares..](#)
- 7-4. [Vista lateral de mallas subdivididas. Con normales al azar \(arriba\) y con normales coherentes \(coherent normals\) \(abajo\)](#)
- 7-5. [Vista de Sólido de mallas Subdivididas con normales inconsistentes \(inconsistent normals\) \(arriba\) y con normales consistentes \(consistent normals\) \(abajo\).](#)
- 7-6. [Una malla "Non-Manifold"](#)
- 7-7. [Malla base de una Gárgola \(izquierda\) y su correspondiente malla SubSurf con nivel 2 \(derecha\) .](#)
- 7-8. [Vista de sólido \(izquierda\) y render final \(derecha\) de la Gárgola.](#)

- 7-9. [Herramientas de Mallas 1 panel](#)
- 7-10. [Afilado de un borde alrededor de 0.5](#)
- 7-11. [Panel de Mallas](#)
- 7-12. [Menu Seleccion de Modo](#)
- 7-13. [Todos los bordes estan seleccionados](#)
- 7-14. [Afilado de Bordes](#)
- 7-15. [Dos bordes seleccionados](#)
- 7-16. [Resultado](#)
- 7-17. [Cuchillo "Center" con poligonal \(arriba\); Cuchillo "Exact" con un único segmento \(medio\) and Cuchillo "Exact" en modo mano alzada \(abajo\).](#)
- 7-18. [Un bucle de caras abierto \(izquierda\) y dos cerrados \(centro y derecha\).](#)
- 7-19. [Dividiendo un bucle de caras.](#)
- 7-20. [Solevando un cubo.](#)
- 7-21. [Un plano.](#)
- 7-22. [Mitad derecha.](#)
- 7-23. [Rebatiendo especularmente el duplicado vinculado.](#)
- 7-24. [Editando una mitad.](#)
- 7-25. [Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.](#)
- 7-26. [Una malla plana densa.](#)
- 7-27. [Una malla plana densa con solo un vértice seleccionado.](#)
- 7-28. [Icono de Edición proporcional](#)
- 7-29. [Menú Falloff.](#)
- 7-30. [Constante - Sin Falloff.](#)
- 7-31. [Falloff lineal.](#)
- 7-32. [Falloff Afilado.](#)
- 7-33. [Falloff Cuadrático.](#)
- 7-34. [Falloff esférico.](#)
- 7-35. [Falloff suave.](#)
- 7-36. [Círculo de influencia.](#)
- 7-37. [Un paisaje obtenido con Edición Proporcional](#)
- 7-38. [Paisaje final renderizado](#)
- 7-39. [Herramienta Subdivide \(Subdividir\)](#)
- 7-40. [Botón Ruido en Botones de Edición](#)
- 7-41. [El proceso de aplicación de Ruido](#)
- 7-42. [Paisaje generado con Ruido](#)
- 7-43. [Botones del Diezmador.](#)
- 7-44. [El Diezmador "diezmando".](#)
- 7-45. [Paisaje diezmado, arriba: original; medio: ligeramente diezmado; abajo: fuertemente diezmado.](#)

- 8-1. [Dos Metaballs](#)
- 9-1. [Tipos de agarraderas para curvas Bézier](#)
- 9-2. [Configurando la resolución de la curva Bézier.](#)
- 9-3. [Botones de control de las curvas Nurbs.](#)
- 9-4. [Configuración de las Nurbs para controlar el polígono y los pesos.](#)
- 9-5. [El logotipo](#)
- 9-6. [Configuración de la ventana 3D.](#)
- 9-7. [Dibujo del logo cargado de fondo](#)
- 9-8. [Agarraderas Bézier](#)
- 9-9. [Añadiendo un punto de control.](#)
- 9-10. [Agarraderas Vector \(en verde\).](#)
- 9-11. [Agarraderas libres \(en negro\).](#)
- 9-12. [El final de linea acabado.](#)
- 9-13. [Logotipo sombreado.](#)
- 9-14. [Añadiendo un círculo.](#)
- 9-15. [Definiendo un ojo.](#)
- 9-16. [Definiendo las alas.](#)
- 9-17. [Colocación del círculo en la cola.](#)
- 9-18. [Configuración del Bevel](#)
- 9-19. [Renderizado final.](#)
- 9-20. [El menú añadir superficie.](#)
- 9-21. [Una superficie esférica](#)
- 9-22. [Pesos preconfigurados](#)
- 9-23. [Ejemplos de texto.](#)
- 9-24. [Botones de edición de texto.](#)
- 9-25. [Perfil \(a la izquierda\) y trayectoria \(a la derecha\).](#)
- 9-26. [Perfil modificado \(a la izquierda\) y trayectoria \(a la derecha\).](#)
- 9-27. [Botón 3D.](#)
- 9-28. [Nombre del perfil.](#)
- 9-29. [Especifique el perfil que se utilizará en la trayectoria.](#)
- 9-30. [Resultado de la extrusión.](#)
- 9-31. [Plano local de la trayectoria.](#)
- 9-32. [Problemas con la extrusión debido a la restricción del eje y.](#)
- 9-33. [Panel de curvas y superficies](#)
- 9-34. [Ejemplo 1 de Taper](#)
- 9-35. [Ejemplo 2 de Taper](#)
- 9-36. [Ejemplo 3 de Taper](#)
- 9-37. [El menú que aparece cuando se emparenta.](#)

- 9-38. [La configuración del Panel Anim.](#)
- 9-39. [Panel de Curvas y superficies.](#)
- 9-40. [¡Añada un mono!](#)
- 9-41. [Añada una curva.](#)
- 9-42. [Editando la curva.](#)
- 9-43. [Mono en la curva.](#)
- 9-44. [Deformaciones del mono.](#)
- 9-45. [Una "curva superficie" para realizar skinning.](#)
- 9-46. [Perfil del barco.](#)
- 9-47. [Varios perfiles a lo largo del eje del barco.](#)
- 9-48. [Varios perfiles de la forma correcta.](#)
- 9-49. [Perfiles unidos.](#)
- 9-50. [Superficie en modo edición.](#)
- 9-51. [El casco finalizado.](#)
- 10-1. [Principio básico del motor de render.](#)
- 10-2. [Luz re-irradiada con fenómeno de difusión.](#)
- 10-3. [Reflexión especular.](#)
- 10-4. [Agregar nuevo material.](#)
- 10-5. [Botones de Material.](#)
- 10-6. [Previsualización del Material, plano \(izquierda\), esfera \(centro\) y cubo \(derecha\).](#)
- 10-7. [Botones de Color de Material.](#)
- 10-8. [Botones del Sombreador de Material \(Material Shader buttons\).](#)
- 10-9. [Sombreadores de Difusión del Material \(Material Diffuse shaders\).](#)
- 10-10. [Sombreadores de Especularidad del Material \(Material Specular shaders\).](#)
- 10-11. [Descripción de Sombreadores \(Shader overview\).](#)
- 10-12. [Deslizadores adicionales de Material \(Additional material sliders\).](#)
- 10-13. [Material común \(izquierda\), material con Alpha < 1 \(centro\) y material con Emit > 0 \(derecha\).](#)
- 10-14. [Botones especiales de Material.](#)
- 10-15. [Panel Ramps.](#)
- 10-16. [Panel Colorband de las Rampas.](#)
- 10-17. [Menú emergente Input.](#)
- 10-18. [Menú emergente Método.](#)
- 10-19. [Deslizador Factor.](#)
- 10-20. [Valores del Sombreador.](#)
- 10-21. [Valores del Sombreador de Rampa.](#)
- 10-22. [Color 0 del Sombreador de Rampa Especular.](#)
- 10-23. [Color 1 del Sombreador de Rampa Especular.](#)
- 10-24. [Ningún Sombreador de Rampa.](#)

- 10-25. [Rampa de Color.](#)
- 10-26. [Rampas de Color y Especular.](#)
- 10-27. [Malla de una Seta](#)
- 10-28. [Seta con un material](#)
- 10-29. [Seta con los vertices del sombrero seleccionados.](#)
- 10-30. [Añadiendo un nuevo material a la malla.](#)
- 10-31. [Material con usuarios múltiples](#)
- 10-32. [Seta con dos materiales.](#)
- 10-33. [Botones Halo](#)
- 10-34. [Resultados del Halo](#)
- 10-35. [Textura de la Matriz de Puntos.](#)
- 10-36. [Visualización de la Matriz de Puntos \(Dot Matrix display\)](#)
- 10-37. [Configuración de Lens Flare](#)
- 10-38. [Lens Flare](#)
- 11-1. [Canales de Textura](#)
- 11-2. [Bloque de selección de Textura](#)
- 11-3. [Coordenadas de entrada del Material](#)
- 11-4. [Mapeado de Textura \(Texture mapping\)](#)
- 11-5. [Transformación de Coordenadas](#)
- 11-6. [Factores de Offset y Escala de coordenadas](#)
- 11-7. [Entradas de Textura \(Texture Inputs\)](#)
- 11-8. [Salidas de Textura \(Texture Outputs\)](#)
- 11-9. [Configuración de Salida \(Output settings\)](#)
- 11-10. [Botones de Textura](#)
- 11-11. [Banda de Color de Textura \(Texture Colorband\).](#)
- 11-12. [Copiando y Pegando Texturas](#)
- 11-13. [Primera textura del anillo de la Madera](#)
- 11-14. [Segunda textura del anillo de la Madera](#)
- 11-15. [Textura Nubes \(Clouds texture\)](#)
- 11-16. [Resultado final](#)
- 11-17. [Mapeado Plano \(Flat Mapping\).](#)
- 11-18. [Mapeado Cúbico \(Cube Mapping\).](#)
- 11-19. [Mapeado Tubular \(Tube Mapping\).](#)
- 11-20. [Mapeado Esférico \(Sphere Mapping\).](#)
- 11-21. [Ejemplo del uso del Mapa de Ambiente](#)
- 11-22. [Material del plano Reflejante.](#)
- 11-23. [Configuración EnvMap del plano reflejante.](#)
- 11-24. [Esfera sobre una superficie reflejante.](#)

- 11-25. [Esfera reflejante sobre una superficie reflejante.](#)
- 11-26. [Esfera reflejante sobre superficie reflejante, con múltiples reflejos.](#)
- 11-27. [Configuración adicional de la textura para Mapeado de Relieves \(BumpMapping\).](#)
- 11-28. [Esfera reflejante sobre agua reflejante, con múltiples reflejos.](#)
- 11-29. [El EnvMap como queda al ser salvado.](#)
- 11-30. [Diferencia entre Mapas de NOR y de Desplazamiento.](#)
- 11-31. [Map To panel.](#)
- 11-32. [Valores para Disp y Nor.](#)
- 11-33. [Desplazamiento Positivo seleccionado.](#)
- 11-34. [Desplazamiento Negativo seleccionado.](#)
- 11-35. [Niveles de Subsurf 0, 1, 2, \(arriba\) 3, 4, 5 \(medio\) y 6 \(abajo\).](#)
- 11-36. [.](#)
- 11-37. [Ajustes de los materiales](#)
- 11-38. [Mapa de Reflexión](#)
- 11-39. [Toques Finales](#)
- 11-40. [Renderizando](#)
- 11-41. [Fijar la transparencia.](#)
- 11-42. [Fijar Reflexiones](#)
- 11-43. [Cristal Hueco](#)
- 11-44. [Entrando al Modo Seleccion de Caras.](#)
- 11-45. [Modo Selección de Caras.](#)
- 11-46. [Panel Cálculo de UV.](#)
- 11-47. [El Editor UV.](#)
- 11-48. [UV pre-sets.](#)
- 11-49. [Menú de transformaciones UV.](#)
- 11-50. ["Stitch" and "Limit Stitch".](#)
- 11-51. [Los colores de vertice modulan la textura.](#)
- 11-52. [Método de despliegue LSCM.](#)
- 11-53. [La herramienta "Paint" en acción.](#)
- 11-54. [Suzanne con las costuras marcadas.](#)
- 11-55. [Fijando UVs.](#)
- 11-56. [Accessing occluded UVs.](#)
- 11-57. [LSCM in action.](#)
- 11-58. [Grupo frontal mapeado con "From Window".](#)
- 11-59. [Dos islas en el mapa.](#)
- 11-60. [Cosiendo.](#)
- 11-61. [Stitching.](#)
- 11-62. [Costuras revisadas.](#)

- 11-63. [Y otro desenrollado LSCM ...](#)
- 11-64. [Desenrollando las orejas.](#)
- 11-65. [Las etapas de desenrollado. Notar los UV que fueron fijados.](#)
- 11-66. [Mapa UV exportado final.](#)
- 11-67. [Gimp es un programa de retoque 2D que soporta capas transparentes.](#)
- 11-68. [Usando el mapa UV como referencia para mapas de textura.](#)
- 11-69. [Los tres mapas de textura.](#)
- 11-70. [Ajustando el mapa de color.](#)
- 11-71. [Render final. Qué texturas más feas... debería pintarlas de nuevo...](#)
- 12-1. [Botones de luces.](#)
- 12-2. [Botones generales de iluminación.](#)
- 12-3. [Luz solar.](#)
- 12-4. [Ejemplo de luz solar.](#)
- 12-5. [Ejemplo de luz solar.](#)
- 12-6. [Esquema conceptual de la luz Hemi.](#)
- 12-7. [Ejemplo de luz hemi.](#)
- 12-8. [ejemplo de iluminación exterior. Energía de la luz solar=1 RGB=\(1.,0.95,0.8\) La dirección del sol en un sistema de referencia polar es \(135°,135°\). Energía de la luz Hemi=0.5 RGB=\(0.64,0.78,1.\) dirigida hacia abajo.](#)
- 12-9. [Botones de luz lámpara.](#)
- 12-10. [Ejemplo de luz lámpara. En los ejemplos Quad Quad1=0, Quad2=1.](#)
- 12-11. [Atenuación de la luz: a\) Lineal, por defecto en Blender; b\) Cuadrática, por defecto en Blender con Quad1=0, Quad2=1; c\) Cuadrática, con Quad1=Quad2=0.5; d\) cuadrática, con Quad1=Quad2=0. También se muestran en el gráfico las mismas curvas, en los mismos colores, pero con el botón esfera activado.](#)
- 12-12. [Esquema de luz focal.](#)
- 12-13. [Los botones de opciones de lámpara](#)
- 12-14. [Botones de luz focal.](#)
- 12-15. [Ejemplos de luz focal para SpotSi=45°](#)
- 12-16. [Botones de sombra de la luz Spot.](#)
- 12-17. [Ejemplos de luz Spot con sombra.](#)
- 12-18. [Botón halo de la lámpara tipo Spot.](#)
- 12-19. [Ajustes de la lámpara tipo Spot](#)
- 12-20. [Renderizado con halo.](#)
- 12-21. [Deslizador de intensidad de halo.](#)
- 12-22. [Botón numérico Halo Step.](#)
- 12-23. [Halo con sombra volumétrica, Halo Step = 1](#)
- 12-24. [Halo con sombra volumétrica, Halo Step = 12](#)

- 12-25. [Ajustando la iluminación de Cornelius.](#)
- 12-26. [Configuración de un simple spot.](#)
- 12-27. [Configuración de un simple Spot con truncamiento apropiado.](#)
- 12-28. [Ajustes de truncamiento de la luz Spot. Izquierda: ClipSta demasiado alto; Centro: Bien; Derecha: ClipEnd demasiado bajo.](#)
- 12-29. [Configuración de luz trasera.](#)
- 12-30. [Solo luz clave \(Izquierda\), Solo luz trasera \(centro\), y ambas \(derecha\).](#)
- 12-31. [Configuración de la luz de relleno.](#)
- 12-32. [Solo luz Clave y Trasera \(izquierda\), Solo luz de Relleno \(centro\), y las tres \(derecha\).](#)
- 12-33. [Configuración de la luz difusa del suelo.](#)
- 12-34. [Configuración de cuatro puntos de luz.](#)
- 12-35. [Configuración de luces Sun y Hemi para exteriores.](#)
- 12-36. [Render de luces Sun y Hemi para exteriores.](#)
- 12-37. [Render de exteriores.](#)
- 12-38. [La luz de Area y su sombra.](#)
- 12-39. [La luz de Area y su sombra 2](#)
- 12-40. [La luz Spot y su sombra](#)
- 12-41. [Configuración de la rejilla](#)
- 12-42. [Luces Spot y sus DupliVerts](#)
- 12-43. [Luz de Area trucada con multiples Spots.](#)
- 12-44. [Luz de Area trucada con multiples Spots suavizados.](#)
- 12-45. [Cornelius bajo una luz de Area.](#)
- 12-46. [Escena de Iluminación Global.](#)
- 12-47. [Bóveda celeste.](#)
- 12-48. [Bóveda celeste con Spots duplicados.](#)
- 12-49. [Invirtiendo las normales.](#)
- 12-50. [Bóveda celeste correcta y luces Spot DupliVerteadas.](#)
- 12-51. [Configuración de luz Spot.](#)
- 12-52. [Configuración de luz Spot.](#)
- 12-53. [Haciendo más densos los Spots en un area.](#)
- 12-54. [Render de la bóveda celeste Anisotrópica.](#)
- 13-1. [Botones del mundo](#)
- 13-2. [Background colors](#)
- 13-3. [Botones de textura](#)
- 13-4. [Botones de niebla](#)
- 13-5. [Configuración de la niebla](#)
- 13-6. [Generación sin niebla \(izquierda\) y con niebla \(derecha\).](#)
- 13-7. [Configuración de Mundo.](#)

- 13-8. [Botones de Estrella](#)
- 13-9. [Generación de estrellas.](#)
- 13-10. [Configuración de las estrellas.](#)
- 13-11. [Panel de Oclusión Ambiental.](#)
- 13-12. [Efecto de distintas cantidades de muestreos.](#)
- 13-13. [Oclusión Ambiental con Color Cielo. Zenith es azul, Horizon es anaranjado, y el tipo es *Blend* para que el cielo sea un anaranjado total en el Nadir.](#)
- 13-14. [Valores de Bias de Oclusión Ambiental.](#)
- 14-1. [Menú Insertar Clave.](#)
- 14-2. [La ventana IPO.](#)
- 14-3. [Opciones de extensión de IPO.](#)
- 14-4. [IPOs extendidos.](#)
- 14-5. [El modo clave de IPO \(IPOKey\).](#)
- 14-6. [La ventana IPO.](#)
- 14-7. [Curva Ipo-tiempo lineal](#)
- 14-8. [La ventana de acciones con los botones de trayectorias.](#)
- 14-9. [Botones de Tracking](#)
- 14-10. [Trayectoria compleja de animación](#)
- 14-11. [Algunas imagenes de la cámara siguiendo al avión.](#)
- 14-12. [Una imagen de la animación final.](#)
- 15-1. [Menú de Insertar Clave.](#)
- 15-2. [Clave de Referencia y Velocidad IPO.](#)
- 15-3. [Claves Absolutas.](#)
- 15-4. [La cabeza femenina que queremos animar.](#)
- 15-5. [Todos los vértices ocultos salvo los de la cara.](#)
- 15-6. [Ojo izquierdo cerrado.](#)
- 15-7. [Sonriendo.](#)
- 15-8. [Claves en la Ventana IPO.](#)
- 15-9. [La curva IPO de la clave 1.](#)
- 15-10. [Botones de copy/paste.](#)
- 15-11. [Todos los IPOs.](#)
- 15-12. [Sequence.](#)
- 15-13. [RVK en la Ventana de Acción.](#)
- 15-14. [Configuración de la Retícula](#)
- 15-15. [Estrechando](#)
- 15-16. [Deformación Final de la Retícula](#)
- 15-17. [Algunos fotogramas de la animación resultante.](#)
- 16-1. [Opciones de Auto key](#)

- 16-2. [VentanaIPO de Acción Clavada](#)
- 16-3. [Agregando un Armature](#)
- 16-4. [Botones de Edición para un Armature](#)
- 16-5. [Opciones de Visualización para Armatures](#)
- 16-6. [Grupos de Vértices](#)
- 16-7. [El botón Pintura de Pesos \(Weight Paint\).](#)
- 16-8. [La opción de menú Modo Pose.](#)
- 16-9. [El botón Modo Pose.](#)
- 16-10. [Ventana de Acción](#)
- 16-11. [Ventana de Acción con una Restricción](#)
- 16-12. [IPO de Acción](#)
- 16-13. [Ventana NLA](#)
- 16-14. [Acción Expandida en la Ventana NLA](#)
- 16-15. [Restricción Expandida en la Ventana NLA](#)
- 16-16. [Opciones de la Banda de Acción NLA](#)
- 16-17. [Panel de Restricciones.](#)
- 16-18. [Restricción Copiar Posición \(Copy Location\).](#)
- 16-19. [Restricción Copiar Rotación \(Copy Rotation\).](#)
- 16-20. [Restricción Seguimiento \(Track To\).](#)
- 16-21. [Seguimiento Trabado \(Lock Track\).](#)
- 16-22. [Seguir Camino \(Follow Path\).](#)
- 16-23. [IK Solver Constraint.](#)
- 16-24. [Action Constraint.](#)
- 16-25. [Null Constraint.](#)
- 16-26. [El modelo del Brazo](#)
- 16-27. [Diseñando el armature](#)
- 16-28. [El armature en Vista Lateral.](#)
- 16-29. [Ubicando el armature en la Vista Lateral.](#)
- 16-30. [Solución IK de la muñeca.](#)
- 16-31. [Configurando los dedos.](#)
- 16-32. [Configurando el dedo.](#)
- 16-33. [Añadiendo la solución IK del dedo.](#)
- 16-34. [Configurando los otros dedos.](#)
- 16-35. [Vista general de los nombres.](#)
- 16-36. [Emparentando el pulgar.](#)
- 16-37. [Emparentando los otros dedos.](#)
- 16-38. [Configurando el IK para la muñeca. Seleccionando el hueso.](#)
- 16-39. [Configurando el IK para la muñeca. Completando la restricción.](#)

- 16-40. [Nombres de grupos de vértices.](#)
- 16-41. [Grupo de vértices ArmHi.R.](#)
- 16-42. [Asignando vértices a un grupo.](#)
- 16-43. [Diferentes Poses.](#)
- 16-44. [Una configuración de pierna \(equivocada\).](#)
- 16-45. [Asignando la restricción IK.](#)
- 16-46. [El esqueleto en modo pose.](#)
- 16-47. [Agregando los dedos y un poco mas de animación IK.](#)
- 16-48. [Moviendo la pierna.](#)
- 16-49. [Rigging con un hueso nulo.](#)
- 16-50. [Rigging con un hueso nulo.](#)
- 16-51. [Agregando los dedos.](#)
- 16-52. [Rig de la pierna completa.](#)
- 16-53. [Ampliación del pie.](#)
- 16-54. [Probando la configuración.](#)
- 16-55. [Araña-robótica de cuatro patas.](#)
- 16-56. [El modelo del brazo](#)
- 16-57. [El modelo del brazo con los ejes pivotes.](#)
- 16-58. [El brazo y su armadura](#)
- 16-59. [El brazo en Modo Posado](#)
- 16-60. [Piston hidráulico.](#)
- 16-61. [Pistón hidráulico en el brazo.](#)
- 16-62. [Pistón hidráulico en modo posado.](#)
- 16-63. [Pistón hidráulico con seguimiento mutuo.](#)
- 16-64. [Configuración para el Tracking \(seguimiento\).](#)
- 16-65. [Modo Posado para el brazo con elementos hidráulicos.](#)
- 16-66. [Añadiendo un tubo flexible.](#)
- 16-67. [Añadiendo la armature al tubo.](#)
- 16-68. [El Empty para la solución IKA.](#)
- 16-69. [Restricción IKA.](#)
- 16-70. [El brazo del robot completo en modo posado.](#)
- 16-71. [Algunas poses típicas de ciclo de andar.](#)
- 16-72. [Usa los botones Copiar, Pegar y Pegar Pose Invertida y serás feliz!](#)
- 16-73. [Un lindo paseo](#)
- 16-74. [Estimando el STRIDE. Refina la rejilla si es necesario !](#)
- 16-75. [Having a rest in the walk](#)
- 16-76. [Hola amigos!](#)
- 17-1. [Botones de render.](#)

- 17-2. [Tipos de imágenes y dimensiones.](#)
- 17-3. [Dimensiones de imagen predeterminadas.](#)
- 17-4. [Botones de Renderizado por partes.](#)
- 17-5. [Botón panorama.](#)
- 17-6. [Configuración test panorámico.](#)
- 17-7. [Renders no panorámicos.](#)
- 17-8. [Renderizado panorámico.](#)
- 17-9. [Renderizado "ojo de pez".](#)
- 17-10. [Panorámica 360 grados con lentes 16.0.](#)
- 17-11. [Panorámica de 360 grados con lente de 38.5.](#)
- 17-12. [Botones OSA.](#)
- 17-13. [Render sin OSA \(izquierda\) con OSA=5 \(centro\) y OSA=8 \(derecha\).](#)
- 17-14. [Formatos de imagen y animación.](#)
- 17-15. [Configuración de los codec del AVI.](#)
- 17-16. [Botones para renderizar una animación.](#)
- 17-17. [Localización de la animación y las extensiones.](#)
- 17-18. [Botones de Motion Blur.](#)
- 17-19. [Cuadro 1 del cubo en movimiento, sin Desenfoque de Movimiento \(motion blur\).](#)
- 17-20. [Cuadro 2 del cubo en movimiento, sin Desenfoque de Movimiento \(motion blur\).](#)
- 17-21. [Cuadro 1 del cubo en movimiento, con Desenfoque de Movimiento \(motion blur\), 8 muestras, Bf=0,5.](#)
- 17-22. [Cuadro 1 del cubo en movimiento con Desenfoque de Movimiento \(motion blur\), 8 muestras, Bf=1,0.](#)
- 17-23. [Cuadro 1 del cubo en movimiento, con Desenfoque de Movimiento \(motion blur\), 8 muestras, Bf=3,0.](#)
- 17-24. [Escena de prueba de Profundidad de Campo.](#)
- 17-25. [Poniendo el Empty Foco.](#)
- 17-26. [NURBS circle tracking the focus Empty.](#)
- 17-27. [IPO Speed para el recorrido circular NURBS.](#)
- 17-28. [Camara con seguimiento sobre el Empty Focal.](#)
- 17-29. [Valores de Motion blur.](#)
- 17-30. [Render final con Motion blur.](#)
- 17-31. [Una escena con materiales de Caricatura.](#)
- 17-32. [Botones de borde de caricatura.](#)
- 17-33. [Valores del borde de caricatura.](#)
- 17-34. [Escena re-renderizada con bordes de caricatura activados.](#)
- 17-35. [El botón Renderizador Unificado.](#)
- 17-36. [Botones adicionales del Renderizador Unificado.](#)

- 17-37. [El submenú de postprocesamiento del Renderizador Unificado.](#)
- 17-38. [Configuración de Render por campos.](#)
- 17-39. [Resultado de renderizar por campos.](#)
- 18-1. [Ejemplo de radiosidad](#)
- 18-2. [Decorado para la prueba de Radiosidad.](#)
- 18-3. [Material con la Radiosidad habilitada.](#)
- 18-4. [Botones Radiosidad para renderizado de radiosidad.](#)
- 18-5. [Habilitando la Radiosidad en los botones de Render.](#)
- 18-6. [Render de Radiosidad para mallas rudas \(izquierda\) y mallas refinadas \(derecha\).](#)
- 18-7. [Botón Gourad](#)
- 18-8. [Botones Radiosivos para Subdivisión](#)
- 18-9. [Botones de Radiosidad](#)
- 18-10. [Post proceso de radiosidad](#)
- 18-11. [Cylon Raider renderizado con radiosidad.](#)
- 18-12. [Añadir un plano](#)
- 18-13. [Color del plano](#)
- 18-14. [Cúpula del cielo.](#)
- 18-15. [Cúpula del cielo.](#)
- 18-16. [Material de la cúpula.](#)
- 18-17. [Configurando la cúpula y el plano como de 'una cara'.](#)
- 18-18. [Seleccionando todas las mallas.](#)
- 18-19. [Preparando la solución de radiosidad.](#)
- 18-20. [Propiedades de radiosidad.](#)
- 18-21. [Solución de radiosidad.](#)
- 18-22. [Render de la solución de radiosidad.](#)
- 18-23. [La malla de la nave.](#)
- 18-24. [Separando las partes de la nave que serán texturizadas.](#)
- 18-25. [Material "Metálico".](#)
- 18-26. [Cuatro texturas, desde la esquina superior izquierda, en el sentido de las agujas del reloj:
RaiderBM, RaiderDI, Markings, Raider.](#)
- 18-27. [Configuración de texturas.](#)
- 18-28. [Previsualización completa de material.](#)
- 20-1. [Ventana de botones de animación](#)
- 20-2. [El emisor.](#)
- 20-3. [Botones de partículas.](#)
- 20-4. [Propiedades de partículas.](#)
- 20-5. [Propiedades del Halo](#)
- 20-6. [Estrellas disparándose](#)

- 20-7. [Colocando las partículas con DupliVert.](#)
- 20-8. [Renderizado de partículas con DupliVert.](#)
- 20-9. [Configuración de la hoguera.](#)
- 20-10. [Configuración de las partículas de fuego.](#)
- 20-11. [Material de Llamas.](#)
- 20-12. [Textura de las llamas.](#)
- 20-13. [Material IPO del fuego](#)
- 20-14. [Render Final.](#)
- 20-15. [La explosión](#)
- 20-16. [Material para la nube de la explosión.](#)
- 20-17. [Material para las chispas.](#)
- 20-18. [Textura de ambos.](#)
- 20-19. [Particle system for the cloud](#)
- 20-20. [Sistema de partículas para las chispas](#)
- 20-21. [Botones de multiplicación de partículas](#)
- 20-22. [Material de los fuegos artificiales 1](#)
- 20-23. [Añadiendo un segundo material al emisor.](#)
- 20-24. [Material 2](#)
- 20-25. [Propiedades de partícula](#)
- 20-26. [Propiedades de celosía](#)
- 20-27. [Efectos de deformación de celosía](#)
- 20-28. [Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.](#)
- 20-29. [Propiedades de partículas estáticas](#)
- 20-30. [Propiedades de material](#)
- 20-31. [Propiedades de banda de colores de textura](#)
- 20-32. [Propiedades de textura en los botones de material](#)
- 20-33. [Resultado final](#)
- 20-34. [Panel Interacción de Partículas \(Particle Interaction panel\).](#)
- 20-35. [Indicador del Campo de Fuerza.](#)
- 20-36. [Indicador del Campo de Vértice.](#)
- 20-37. [Panel de Efectos \(Effects panel\).](#)
- 20-38. [Cuadro 40 del Ejemplo.](#)
- 20-39. [Ejemplo de Configuración de Desviación \(Deflection\).](#)
- 20-40. [Resultado final del ejemplo.](#)
- 21-1. [Efecto Build](#)
- 21-2. [Panel de control de Onda](#)
- 21-3. [Origen de la Onda](#)

- 21-4. [Tipo de frente de ondas](#)
- 21-5. [Controles de frente de onda](#)
- 21-6. [Características del frente de onda](#)
- 21-7. [Controles de tiempos de onda](#)
- 22-1. [A simple scene to play with.](#)
- 22-2. [The parent mesh can be any primitive.](#)
- 22-3. [The man is parented to the circle.](#)
- 22-4. [The Animation Buttons](#)
- 22-5. [In every vertex of the circle a man is placed.](#)
- 22-6. [Changing the size of the circle in Edit Mode.](#)
- 22-7. [A second row of Magritte's men.](#)
- 22-8. [Show object's axis to get what you want.](#)
- 22-9. [Negative Y Axis is aligned to vertex normal \(pointing to the circle's center\)](#)
- 22-10. [Positive Y axis is aligned to normal](#)
- 22-11. [Positive X axis is aligned to normal](#)
- 22-12. [Positive Z axis is aligned to normal \(weird, huh?\)](#)
- 22-13. [Strange tentacle and SubSurfed version.](#)
- 22-14. [Local reference of the tentacle.](#)
- 22-15. [DupliVerts not rotated.](#)
- 22-16. [DupliVerts rotated.](#)
- 22-17. [Our model complete.](#)
- 22-18. [Contorno del eslabón](#)
- 22-19. [Sección transversal del eslabón](#)
- 22-20. [Curve's settings: Curve Path y Curve Follow.](#)
- 22-21. [Borrando el origen.](#)
- 22-22. [Siguiendo el eje correcto.](#)
- 22-23. [DupliFrames!](#)
- 22-24. [Valores para DupliFrames. Notar que "DupEnd: 35" terminará el eslabón antes del final de la curva.](#)
- 22-25. [Making Dupli's Real.](#)
- 22-26. [Skineando el eslabón.](#)
- 22-27. [Skinned link.](#)
- 22-28. [Using a curve path to model the chain.](#)
- 22-29. [Curve settings.](#)
- 22-30. [Ajustando los DupliFrames.](#)
- 22-31. [Rotando el eslabón.](#)
- 22-32. [Cadena con Dupliframe.](#)
- 22-33. [A Surface Circle.](#)

- 22-34. [Trapezoidal cross-section.](#)
- 22-35. [Trapezoidal cross section, rotated and translated.](#)
- 22-36. [Rotation IPO for the cross section.](#)
- 22-37. [Translation IPO for the cross section.](#)
- 22-38. [Size IPO for the cross section.](#)
- 22-39. [Using a curve path to model the chain.](#)
- 22-40. [Using a curve path to model the chain.](#)
- 22-41. [Un tenedor feo.](#)
- 22-42. [Una lattice de 2x2x2.](#)
- 22-43. [Usa una resolución adecuada pero no exageres.](#)
- 22-44. [Deformar una lattice es un placer !](#)
- 22-45. [Doblando cosas.](#)
- 22-46. [Un lindo tenedor.](#)
- 22-47. [Pobre tipo...](#)
- 22-48. [Doblando cosas.](#)
- 22-49. [Escala la rejilla para que calce la botella.](#)
- 22-50. [Edita la Lattice para que el pequeñín entre comodamente en la botella](#)
- 22-51. [¿ Claustrofóbico ?](#)
- 22-52. [Render Final. Creanme, se lo merecía!](#)
- 23-1. [Columnas sobre un plano.](#)
- 23-2. [El render de un plano.](#)
- 23-3. [El render con la niebla incorporada de Blender.](#)
- 23-4. [Configuración del plano.](#)
- 23-5. [El plano con frame duplicado \(Dupliframe\).](#)
- 23-6. [Reformando el frame duplicado del plano IPO.](#)
- 23-7. [Reformando el frame duplicado del plano IPO.](#)
- 23-8. [Propiedades basicas del material.](#)
- 23-9. [Limite de altura de la textura.](#)
- 23-10. [Basic Propiedades de material para textura de nubes.](#)
- 23-11. [Propiedades de la textura de nube.](#)
- 23-12. [Render final.](#)
- 24-1. [Ajustes de Softbody.](#)
- 24-2. [Ajustes de Bake.](#)
- 24-3. [Ejemplo de configuración de pesos.](#)
- 24-4. [Ejemplo de configuración de viento.](#)
- 24-5. [Ejemplo de IPO de fuerza del viento.](#)
- 25-1. [Resultado final.](#)
- 25-2. [Sacando el cubo de la vista de cámara.](#)

- 25-3. [Un render del cubo solido.](#)
- 25-4. [Y un render del cubo alamabrico.](#)
- 25-5. [Establezca un nombre de archivo de salida.](#)
- 25-6. [Añadiendo un corte de vídeo](#)
- 25-7. [Colocando el corte.](#)
- 25-8. [Mezclando dos cortes de vídeo](#)
- 25-9. [Botón previsualizar del editor de secuencia.](#)
- 25-10. [Añadiendo una ventana de previsualización.](#)
- 25-11. [Secuencia con 4 cortes de cubos alambricos añadidos juntos](#)
- 25-12. [Botón IPO de Secuencia.](#)
- 25-13. [Definiendo el brillo de unacapa con un IPO](#)
- 25-14. [Cuatro cubos alambricos combinados con efectos de fundido.](#)
- 25-15. [Nombrado el Meta Corte](#)
- 25-16. [Mascara de animación.](#)
- 25-17. [Substrayendo la mascara del vídeo.](#)
- 25-18. [Máscara eliminada.](#)
- 25-19. [Máscara multiplicada.](#)
- 25-20. [Añadiendo los dos efectos](#)
- 25-21. [Dos capas cambiadas en el tiempo.](#)
- 25-22. [Añadiendo un círculo de 20 vértices.](#)
- 25-23. [Revolucionando el círculo alrededor del cursor](#)
- 25-24. [Cámara dentro del túnel.](#)
- 25-25. [Visión de cámara del interior del túnel.](#)
- 25-26. [Rotación IPO del túnel sin entrada y salida.](#)
- 25-27. [Rotando la cámara para conseguir un efecto más dramático](#)
- 25-28. [Un túnel estupendo.](#)
- 25-29. [Aplicando el efecto de telón.](#)
- 25-30. [Añadiendo aleatoriedad con un IPO irregular](#)
- 25-31. [Logo saltarín](#)
- 25-32. [Propiedades del sistema de partículas.](#)
- 25-33. [Rectángulos con vértices duplicados](#)
- 25-34. [Configurando la niebla.](#)
- 25-35. [Configurando los parámetros de niebla](#)
- 25-36. [Rectángulos de partícula renderizados.](#)
- 25-37. [Use dos veces el logo de la animación](#)
- 25-38. [Animación de partículas combinado con animación de logo](#)
- 25-39. [Transición negro-blanco.](#)
- 25-40. [Imagen de transición vídeo blanco](#)

- 25-41. [Transición vídeo-negro](#)
- 25-42. [Combinando el túnel y el cubo alámbrico](#)
- 25-43. [Combinando el túnel, y los cubos alámbrico y sólido](#)
- 25-44. [Añadiendo la animación del logo con partículas](#)
- 25-45. [Añadiendo la animación del logo con zoom.](#)
- 25-46. [El último detalle](#)
- 25-47. [La secuencia final](#)
- 26-1. [Barra de herramientas de texto.](#)
- 26-2. [El GUI de nuestro ejemplo.](#)
- 26-3. [El resultado de nuestro script de ejemplo.](#)
- 28-1. [Selección del Path en el menú de preferencias de usuario.](#)
- 28-2. [Las opciones por defecto en el panel Render.](#)
- 28-3. [Selección de YafRay como raytracer y las dos solapas extra.](#)
- 28-4. [Solapa de YafRay por defecto con el fichero de exportación activado. \(botón XML\).](#)
- 28-5. [Exportación a fichero de YafRay desactivada.](#)
- 28-6. [El render en progreso con XML desactivado.](#)
- 28-7. [La escena.](#)
- 28-8. [La escena como se ve en el Viewport.](#)
- 28-9. [Rendered with Lamp at default settings.](#)
- 28-10. [The noise comes when sampling values is low.](#)
- 28-11. [The increase in sampling to 5 remove the noise.](#)
- 28-12. [The basic render from Area Light at default setting and value of 4.](#)
- 28-13. [Although the scene looks similar to Figure 28-11](#) but is brighter and have better shadows.
- 28-14. [The Spot lamp render with default settings.](#)
- 28-15. [Spot lamp with spherical light attenuation.](#)
- 28-16. [Spot lamp with Spherical attenuation enabled.](#)
- 28-17. [The Sun light at value 2. No wonder why this lamp is called Sun.](#)
- 28-18. [The Yafray GI tab.](#)
- 28-19. [The left part of image have the power of diffusion set at 2 while the right one at 4.](#)
- 28-20. [Using Full GI increases the time as well as realism.](#)
- 28-21. [The Yafray GI tab. Options for Full method are visible.](#)
- 28-22. [The Yafray tab.](#)
- 28-23. [Notice the noise on the wooden boards.](#)
- 28-24. [Increasing the sampling manually, and aperture size solves the problem.](#)
- 28-25. [HDRI render using Uffizi probe. It need more quality level.](#)
- 29-1. [Parámetro Texture Dir y botón de inicio.](#)
- 29-2. [Botones de cabecera de Yable.](#)
- 29-3. [Botones de configuración headerGeneral de Yable.](#)

- 29-4. [Configuración de Iluminación Global de Yable.](#)
- 29-5. [Ajustes Background.](#)
- 29-6. [Ajustes de Fog.](#)
- 29-7. [Ajuste de Depth of Field.](#)
- 29-8. [Ajuste del filtro Anti Noise.](#)
- 29-9. [Ajustes de Resolution.](#)
- 29-10. [Ajustes Rendering.](#)
- 29-11. [Selección de capas.](#)
- 29-12. [Botones de Material.](#)
- 29-13. [Moduladores de Materiales.](#)
- 29-14. [Panel GUI Light.](#)
- 29-15. [Opciones de Direct Light.](#)
- 29-16. [Configuración de cáusticas.](#)
- 29-17. [Escena de cáusticas.](#)
- 29-18. [Pintura del coche.](#)
- 29-19. [Elementos de cromo.](#)
- 29-20. [Parabrisas y otros cristales.](#)

List of Examples

- 26-1. [Cabecera del script](#)
- 26-2. [Dibujando el GUI](#)
- 26-3. [Manejando eventos](#)
- 26-4. [Cabecera del script](#)

[Next](#)

Introducción a Blender

I. Introducción a Blender

La primera parte de la documentación te guiará a través de la descarga, instalación y, si tu eliges descargar también las fuentes, construcción de Blender.

Blender exhibe un interface muy peculiar, altamente optimizado para la producción de gráficos en 3D. Al principio, puede parecer un poco complicado a un usuario sin experiencia, pero comprobarás su poder a la larga. Es muy recomendable leer [Chapter 3](#) detenidamente para familiarizarse tanto con el interface como las convenciones usadas en esta Documentación.

El último capítulo de esta parte, [Chapter 4](#), te permitirá entrever las capacidades de Blender. Por supuesto Blender puede hacer mucho más que esto, ya que esto es solo una introducción rápida al programa

Table of Contents

1. [Introducción](#)

- 1.1. [¿Qué es Blender?](#)
- 1.2. [La historia de Blender](#)
- 1.3. [Sobre el Software Libre y la licencia GPL](#)
- 1.4. [Consigue ayuda - la comunidad de usuarios de Blender](#)

2. [Instalación](#)

- 2.1. [Descarga y ejecución de Blender](#)
 - 2.1.1. [Windows](#)
 - 2.1.2. [OSX](#)
 - 2.1.3. [Linux](#)
 - 2.1.4. [FreeBSD](#)
 - 2.1.5. [Irix](#)
 - 2.1.6. [Solaris](#)
- 2.2. [Instalando Blender a partir del código fuente](#)
 - 2.2.1. [Obteniendo el código fuente](#)
 - 2.2.2. [Librerías necesarias](#)
 - 2.2.3. [Herramientas necesarias](#)
 - 2.2.4. [Instalando Blender](#)
 - 2.2.5. [Soporte técnico](#)

3. [Comprendiendo la interfaz](#)

3.1. [Conceptos de la Interface de Blender](#)

- 3.1.1. [El teclado y el ratón](#)
- 3.1.2. [El sistema de ventanas](#)
- 3.1.3. [Tipos de ventanas](#)
- 3.1.4. [Contextos, Paneles y Botones](#)
- 3.1.5. [Caja de Herramientas\(ToolBox\)](#)
- 3.1.6. [Pantallas\(Screens\)](#)
- 3.1.7. [Escenas](#)

3.2. [Navegando en el Espacio 3D](#)

- 3.2.1. [La dirección de la vista \(rotando\)](#)
- 3.2.2. [Trasladando y Escalando la Vista.](#)
- 3.2.3. [Proyección en Perspectiva y Ortográfica](#)
- 3.2.4. [Modo de Dibujado](#)
- 3.2.5. [Vista Local](#)
- 3.2.6. [El sistema de Capas](#)

3.3. [Funciones fundamentales](#)

- 3.3.1. [Cargando archivos](#)
- 3.3.2. [Guardando archivos](#)
- 3.3.3. [Renderizando](#)
- 3.3.4. [Preferencias de usuario y Temas](#)
- 3.3.5. [Configurando la escena por defecto](#)

4. [Su primera animación en 30 + 30 minutos](#)

- 4.1. [Calentando](#)
- 4.2. [Construyendo el cuerpo](#)
- 4.3. [Veamos como se ve Gus](#)
- 4.4. [Materiales y Texturas](#)
- 4.5. [Construyendo un esqueleto \(Rigging\)](#)
- 4.6. [Asignación del esqueleto \(Skinning\)](#)
- 4.7. [Posando](#)
- 4.8. [¡Gus anda!](#)

[Prev](#)

[Home](#)

[Next](#)

Volumen I de la Documentación
de Blender - Guía de Usuario

Introducción

Chapter 1. Introducción

1.1. ¿Qué es Blender?

Blender es un paquete integrado de herramientas, para la creación de un amplio rango de contenido 3D. Ofrece total funcionalidad para el modelado, renderizado, animación, post-producción, creación y reproducción de contenido interactivo con los singulares beneficios de ser multiplataforma y tener un tamaño inferior a 2.5MB.

Destinado para artistas y profesionales 3D, Blender puede ser usado para crear visualizaciones 3D, tanto estáticas como de video de alta calidad, mientras que la incorporación de un motor para 3D en tiempo real, permite la creación de contenido interactivo que puede ser reproducido independientemente.

Originalmente desarrollado por la compañía 'Not a Number' (NaN), Blender ahora es desarrollado como 'Software Libre', con el código fuente disponible bajo la licencia GNU GPL.

Características principales:

- Paquete de creación totalmente integrado, ofreciendo un amplio rango de herramientas esenciales para la creación de contenido 3D, incluyendo modelado, animación, renderizado, post-producción de video y creación de juegos.
- Ejecutable de pequeño tamaño, para una sencilla distribución;
- Multiplataforma, con un interfaz basado en OpenGL, listo para usar en todas las versiones de Windows, Linux, OSX, FreeBSD, Irix y Sun;
- Arquitectura 3D de alta calidad permitiendo un rapido y eficiente desarrollo;
- Canales de soporte gratuito via www.blender.org;
- Una comunidad de más de 250.000 usuarios;

Puedes descargarte la última versión de Blender de download.blender.org.

1.2. La historia de Blender

En 1988 Ton Roosendaal co-fundó el estudio de animación Holandés *NeoGeo*. NeoGeo rápidamente se convirtió en el estudio más grande de animación 3D en Holanda y en una de las más destacadas casas de animación en Europa. NeoGeo creó producciones, que fueron premiadas (European Corporate Video Awards de 1993 y 1995), para grandes clientes corporativos tales como la compañía multinacional de electrónica Philips. En NeoGeo, Ton fué el responsable tanto de la dirección artística como del desarrollo interno del software. Después de una cuidadosa deliberación Ton decidió que la actual herramienta 3D utilizada en el estudio de NeoGeo era demasiado vieja y voluminosa para mantener y actualizar y necesitaba ser reescrita desde el principio. En 1995 esta reescritura comenzó y estaba destinado a convertirse en el software de creación 3D que ahora conocemos como *Blender*. Mientras NeoGeo continuaba refinando y mejorando Blender, Ton se dió cuenta que Blender podría ser utilizado como una herramienta para otros artistas fuera del estudio NeoGeo.

En 1998, Ton decidió crear una nueva compañía llamada Not a Number (NaN) derivada de NeoGeo para fomentar el mercado y desarrollar Blender. En la base de NaN estaba el deseo de crear y distribuir gratuitamente una suite de creación 3D compacta y multiplataforma. En ese momento, esto fué un concepto revolucionario ya que la mayoría de los programas comerciales de modelado costaban miles de dólares. NaN esperaba conseguir una herramienta de modelado y animación de un nivel profesional al alcance del público en general. El modelo de negocio de NaN consistía en proporcionar productos comerciales y servicios alrededor de Blender. En 1999 NaN asistió a su primera conferencia en el Siggraph en un esfuerzo aún mayor para promocionar Blender. La primera convención del Siggraph para Blender en 1999 fué un auténtico éxito y provocó un enorme interés tanto de la prensa como de los asistentes a la convención. ¡Blender fué un gran éxito y se confirmó su tremendo potencial!

En alas del gran éxito del Siggraph, a principios del año 2000 NaN consiguió una financiación de 4.5 millones de euros procedente de unos inversores. Este gran aporte de dinero permitió a NaN expandir rápidamente sus operaciones. Pronto NaN alardeó de tener más de 50 empleados trabajando alrededor del mundo intentando mejorar y promocionar Blender. En el verano del 2000, Blender 2.0 fué publicado. Esta versión de Blender integraba un motor de juegos a la suite 3D. Al final del 2000, el número de usuarios registrados en el sitio web de NaN sobrepasó los 250,000.

Desafortunadamente, las ambiciones y oportunidades de NaN no coincidieron con las capacidades de la compañía ni con la realidad del mercado de la época. Este sobredimensionamiento de la empresa condujo a una reestructuración creando una compañía (NaN) mas pequeña y con nuevos fondos procedentes de los inversores. Seis meses mas tarde, el primer producto comercial de NaN, *Blender Publisher* fué lanzado. Este producto fué dirigido al emergente mercado de medios interactivos en 3D

basados en entornos web. Debido a las decepcionantes ventas y al continuo clima de dificultades económicas, los nuevos inversores decidieron dar por terminadas las actividades de NaN. Esto también incluía parar el desarrollo de Blender. Si bien existían claramente defectos en la actual versión de Blender, con una arquitectura interna del software compleja, características inacabadas y un GUI no muy común, la magnífica ayuda de la comunidad y los clientes que habían comprado el Blender Publisher en el pasado provocó que Ton no pudiera permitir que Blender desapareciera en el olvido. Como relanzar una nueva compañía con un equipo suficientemente grande de desarrolladores no era factible, en Marzo de 2002 Ton Roosendaal fundó la organización no lucrativa *Blender Foundation* (Fundación Blender)

El primer objetivo de la Fundación Blender fue encontrar una manera de continuar el desarrollo y la promoción de Blender como un proyecto de [Código Abierto](#) basado en la comunidad de usuarios. En Julio de 2002, Ton logró obtener de los inversores de NaN un "sí" para que la Fundación Blender llevara a cabo su plan para que Blender fuera código abierto. La campaña de "Liberar a Blender" tenía que obtener 100,000 EUR para que la Fundación pudiese comprar los derechos del código fuente de Blender y los de propiedad intelectual a los inversores de NaN y posteriormente liberar Blender a la comunidad de código abierto. Con un entusiasta grupo de voluntarios, entre los que se encontraban varios ex-empleados de NaN, fue lanzada la campaña de "Liberar a Blender". Para el deleite y la sorpresa de todo el mundo, la campaña alcanzó el objetivo de 100,000 EUR en tan solo 7 semanas. El Domingo 13 de Octubre de 2002, Blender fue liberado al mundo bajo los términos de la Licencia Pública General de GNU (GPL). El desarrollo de Blender continúa hasta nuestros días conducido por un equipo de valientes y dedicados voluntarios procedentes de diversas partes del mundo y liderados por el creador de Blender, Ton Roosendaal.

Historia de Blender y hoja de ruta

- 1.00 Enero de 1995 Desarrollo de Blender en el estudio de animación NeoGeo
- 1.23 Enero de 1998 Versión para SGI (IrisGL) publicada en la web
- 1.30 Abril de 1998 Versión para Linux y FreeBSD, se porta a OpenGL y a X
- 1.3x Junio de 1998 Creación de NaN
- 1.4x Septiembre de 1998 Versión para Sun y Linux Alpha publicada
- 1.50 Noviembre de 1998 Primer manual publicado
- 1.60 Abril de 1999 C-key (nuevas características necesitan ser desbloqueadas, \$95), la versión de Windows se libera
- 1.6x Junio de 1999 Versión para BeOS y PPC publicada
- 1.80 Junio de 2000 Fin de la C-key, Blender es totalmente gratuito de nuevo
- 2.00 Agosto de 2000 Motor de tiempo-real y player interactivo
- 2.10 Diciembre de 2000 Nuevo motor, físicas y Python
- 2.20 Agosto de 2001 Sistema de animación de personajes
- 2.21 Octubre de 2001 Blender Publisher lanzado
- 2.2x Diciembre de 2001 Versión para Mac OSX publicada

- *13 Octubre de 2002 Blender se convierte en código abierto, Primera Conferencia de Blender*
- 2.25 Octubre de 2002 Blender Publisher está de nuevo disponible gratuitamente
- Tuhopuu1 Octubre de 2002 Se crea la rama experimental de Blender, un lugar de pruebas para los programadores.
- 2.26 Febrero de 2003 La primera versión de Blender siendo código abierto
- 2.27 Mayo de 2003 La segunda versión de Blender siendo código abierto
- 2.28x Julio de 2003 La primera de las series 2.28x.
- 2.30 Octubre de 2003 En la segunda conferencia de Blender el interfaz 2.3x es presentado.
- 2.31 Diciembre de 2003 Actualización al interfaz de la versión 2.3x, ya que es estable.
- 2.32 Enero de 2004 Gran revisión de la capacidad del render interno.

[Prev](#)

Introducción

[Home](#)

[Up](#)

[Next](#)

Sobre el Software Libre y la
licencia GPL

1.3. Sobre el Software Libre y la licencia GPL

Cuando uno escucha hablar sobre el "software libre", la primera cosa que le viene a la mente es "gratuito". Aunque esto es verdad en la mayoría de los casos, el término "software libre" tal como es usado por la Fundación para el Software Libre (creadores del proyecto GNU y de la Licencia Pública General GNU) significa "libre, como en libertad" mas que "gratis, como en cerveza gratis" (N. del T. En inglés tienen el problema de que la palabra "free" significa gratis y libre, por eso al definir "free software" tienden a hacer distinción entre software gratuito y software libre). Software libre es, en este sentido, aquel software que puedes usar, copiar, modificar y distribuir sin ningún límite. Esto contrasta con la licencia de la mayoría de paquetes de software comerciales, en donde se te permite usar el software en una sola computadora pero no esta permitido hacer copias y menos aún ver el código fuente. El software libre permite una increíble libertad al usuario. Como añadidura a todo esto, debido a que el código fuente esta disponible a todo el mundo, los fallos pueden ser detectados y corregidos de una manera más eficiente y rápida.

Cuando un programa es licenciado bajo la licencia pública general GNU (la GPL):

- Tienes el derecho de usar el programa para cualquier propósito;
- tienes el derecho de modificar el programa, y tener acceso al código fuente;
- tienes el derecho de copiar y distribuir el programa;
- tienes el derecho de mejorar el programa, y publicar tus propias versiones.

A cambio de esos derechos, también tienes algunas responsabilidades, que están diseñadas para proteger tu libertad y la libertad de otros, por lo que si distribuyes un programa GPL:

- Debes proporcionar una copia de la GPL con el programa, así el usuario es consciente de los derechos que le otorga la licencia.
- Debes incluir el código fuente o hacer que esté disponible gratuitamente.
- Si modificas el código y distribuyes una versión modificada, dichas modificaciones deben estar licenciadas bajo la GPL y tienes que facilitar el código fuente modificado. (No puedes usar código GPL como parte de un programa propietario.)
- No puedes licenciar el programa fuera de los términos de la licencia GPL. (No puedes transformar un programa GPL en un producto propietario).

Para más información sobre la licencia GPL, visita [el sitio web del proyecto GNU](http://www.gnu.org). Para referencia, una copia de la licencia pública general de GNU se incluye en el Volumen II.

[Prev](#)

La historia de Blender

[Home](#)

[Up](#)

[Next](#)

Consigue ayuda - la comunidad
de usuarios de Blender

1.4. Consigue ayuda - la comunidad de usuarios de Blender

Al estar Blender disponible de forma gratuita desde un principio, incluso cuando el código era cerrado, ayudó mucho en su adopción. Una gran estable y activa comunidad de usuarios ha estado creciendo alrededor de Blender desde el año 1998.

La comunidad mostró lo mejor de si misma en el momento crucial de la liberación de Blender, convirtiendolo en código abierto bajo la licencia GNU GPL hacía finales del verano del año 2002.

La comunidad está ahora dividida en dos sitios, con muchas funciones superpuestas:

1. La comunidad de desarrollo, centrada alrededor de la Fundación Blender <http://www.blender.org/>. Aquí es donde estan los proyectos de desarrollo, las listas de funcionalidades y documentación, el repositorio CVS con las fuentes de Blender, todas las fuentes de documentación y foros de discusión. Los desarrolladores de Blender, los que hacen scripts de python, los documentadores, y cualquiera que trabaje en el desarrollo de Blender, generalmente, puede ser encontrado aquí
2. La comunidad de usuarios, centrada alrededor de un sitio independiente <http://www.elysiun.com/>. Aquí es donde los artistas, los creadores de juegos y los aficionados a Blender se reunen para mostrar sus creaciones, obteniendo críticas para mejorarlas, y donde pueden pedir ayuda para obtener una mejor comprensión de las funcionalidades de Blender. Algunos tutoriales de Blender y una base con el conocimiento básico se pueden encontrar también aquí.

Estas dos páginas web no son los únicos recursos de Blender. La comunidad alrededor del mundo ha creado una gran cantidad de sitios independientes, con diferentes idiomas o especializadas en diferentes áreas. Una constante actualización de los recursos para Blender se encuentra en cualquiera de los sitios mencionados anteriormente.

Para una crítica instantanea hay tres canales de chat disponibles en irc.freenode.net. Puedes unirte a la conversación con tu programa IRC favorito.

Los canales son **#blenderchat**, **#blenderqa** y **#gameblender**. El primero de ellos es accesible incluso sin un cliente IRC, usando un navegador con Java a través del sitio de elYsiun (<http://www.elysiun.com/>).

[Prev](#)

Sobre el Software Libre y la
licencia GPL

[Home](#)

[Up](#)

[Next](#)

Instalación

Chapter 2. Instalación

Referido a Blender v2.31

Blender está disponible en la web de la de la Fundación (<http://www.blender.org/>) como ejecutable y como código fuente. En la página principal busca la sección de descargas ('Downloads').

De todas formas, para una correcta comprensión de este libro, es altamente recomendable utilizar la versión incluida en el CDROM. Por lo tanto, cuando en el parrafo anterior mencionabamos "sección de descargas", también asumiremos que puede conseguirse desde el CDROM.

2.1. Descarga y ejecución de Blender

El ejecutable está disponible para 6 sistemas operativos:

- Windows
- Linux
- MacOSX
- FreeBSD
- Irix
- Solaris

Linux actualmente dispone de 4 opciones para las arquitecturas de Intel y PowerPC, con las librerías enlazadas estática o dinamicamente.

La diferencia entre enlazado dinámico y estático es importante. El estático tiene la librería OpenGL compilada en su interior. Esto hace que Blender corra en su sistema, sin utilizar la aceleración por hardware de su tarjeta gráfica. Utilice la versión estática para comprobar si Blender funciona cuando la versión dinámica falle. OpenGL se usa en Blender para todo el dibujado, incluyendo menús y botones. Esta dependencia hace que la instalación, en tu sistema, de la librería OpenGL sea necesaria. No todos los fabricantes de tarjetas gráficas 3D dan soporte a OpenGL, especialmente para las tarjetas más baratas destinadas al mercado de videojuegos.

Desde que los 'renders' se hacen en Blender con el motor de renderizado utilizando la memoria principal y el procesador de tu computadora, una tarjeta gráfica con aceleración por hardware no supone ninguna

diferencia en el tiempo de renderizado.

2.1.1. Windows

2.1.1.1. Instalación rápida

Descargue el fichero `blender-2.3#-windows.exe`, donde 2.3# es el número de la versión, de la sección de descargas del sitio web de Blender. Comience la instalación haciendo 'doble click' sobre el fichero descargado. En este proceso se le harán varias preguntas, para las cuales las respuestas que hay por defecto deberían bastar. Después que la instalación haya finalizado, puede comenzar a usar Blender directamente, o bien, puede usar la entrada que se habrá creado en el menu Inicio.

2.1.1.2. Instrucciones detalladas

Descargue el fichero `blender-2.3#-windows.exe` de la sección de descargas del sitio web de Blender. Elija descargarlo, seleccione un directorio y pulse "salvar" (Save). Entonces, con el explorador, dirijase al directorio donde haya guardado el fichero y haga "doble-click" para comenzar la instalación.

El primer diálogo le muestra la licencia de Blender. Si quiere que la instalación continúe tendrá que aceptarla. Después de aceptar la licencia, seleccione los componentes que desee instalar (solo hay uno, Blender) y las acciones adicionales que quiera llevar a cabo. Hay tres: Añadir un enlace directo al menú de Inicio, añadir un icono de Blender al escritorio y asociar los archivos `.blend` con el programa Blender. Por defecto, todas las acciones están seleccionadas. Si no quiere que se ejecute alguna de esas acciones simplemente deseleccionela. Cuando esté hecho, simplemente presione en *siguiente (Next)*.

Seleccione un lugar para instalar los archivos (el lugar que viene por defecto debería valer), y presione en *siguiente (Next)* para instalar Blender. Presione *cerrar (Close)* cuando la instalación haya terminado.

Por último, se le preguntará si quiere que el programa de instalación ejecute a continuación Blender. Una vez que esté instalado Blender puede ejecutarse desde el menú de Inicio (una entrada se ha creado por el programa de instalación con el nombre de "Blender Foundation") o bien haciendo "doble-click" en un archivo de Blender (* `.blend`).

2.1.2. OSX

2.1.2.1. Instalación

Descargue el fichero `blender-2.3#-darwin-6.6-powerpc.dmg` de la sección de descargas del sitio web de Blender. Descomprimalo haciendo "doble-click" en el archivo. Esto abrirá un directorio con varios ficheros.

Desde que Blender usa OpenGL para dibujar su interfaz, y MacOS X también lo utiliza para dibujar su escritorio, necesita verificar primero que dispone de suficiente memoria VRAM en su sistema. Por debajo de 8 MB de memoria VRAM Blender no funcionará. Por encima de 16 MB de memoria VRAM necesitará configurar su sistema para que tenga "miles de colores" (1000s of colors) en (System Preferences -> Displays).

Ahora puede usar Blender simplemente con hacer "doble-click" en el icono de Blender. O arrastrar el icono de Blender al panel principal (Dock) para crear un acceso directo. Blender, por defecto, se iniciará en una pequeña ventana. Utilice el boton "+" en la cabecera de la ventana para maximizarlo. Más consejos y avisos sobre la versión para MacOS X se encuentran en el archivo `OSX tips.rtf` en el directorio donde esté Blender instalado.

2.1.3. Linux

2.1.3.1. Instalación rápida

Descargue el fichero `blender-2.3#-linux-glibc#.#.#-ARCH.tar.gz` de la sección de descargas del sitio web de Blender. Aquí 2.3# es la versión de Blender, #.#.# es la versión de la librería glibc y *ARCH* es la arquitectura de la computadora, que puede ser *i386* o *powerpc*. Debería escoger la correspondiente a su sistema, recuerde elegir entre enlazado dinamico o estático.

Desempaque el fichero en la localización de su elección. Esto creará un directorio denominado `blender-2.3#-linux-glibc#.#.#-ARCH`, en el cual encontrará el binario de **blender**.

Para comenzar Blender unicamente abre una consola y ejecuta el comando `./blender`, por supuesto cuando el servidor gráfico 'X' esté funcionando.

2.1.3.2. Instrucciones detalladas

Descargue el fichero `blender-2.3#-linux-glibc#.#.#-ARCH.tar.gz` de la sección de descargas del sitio web de Blender. Elija descargarlo, seleccione un directorio y pulse "salvar" (Save). Entonces navegue hasta la localización que haya elegido para la instalación de blender (p.ej. `/usr/local/`) y desempaquete el fichero (con el comando `tar xzf /ruta/donde/está/blender-2.3#-linux-glibc#.#.#-ARCH.tar.gz`). Si usted desea, puede renombrar el directorio resultante del desempaquetado `blender-2.3#-linux-glibc#.#.#-ARCH` a algo más corto, como por ejemplo `blender`.

Ahora Blender está instalado y se puede ejecutar desde la linea de comandos (consola) simplemente poniendo `./ruta/donde/está/blender` y presionando la tecla "enter". Si usa KDE o GNOME, puede comenzar Blender usando el explorador de archivos que utilice habitualmente navegando hasta donde esté el ejecutable de Blender y haciendo "doble-click" en él.

Si usa Sawfish, puede que quiera añadir una línea como esta ("*Blender*" (*system "blender &"*)) a su archivo `.sawfish/rc` de configuración.

Añadir un icono para blender en KDE

1. Seleccione el "Editor de Menú" en el submenú System del menú de KDE.
2. Seleccione el submenú etiquetado como "Gráficos" en la lista.
3. Pinche en el botón "Nuevo dispositivo". Una caja de dialogo aparecerá para que elija un nombre. Cree el nombre y pinche en "OK". "Blender" o "Blender 2.3#" serían unas elecciones bastantes lógicas, pero elegir otro cualquiera no afectará al funcionamiento del programa.
4. Será devuelto al listado del menú, y el submenú "Gráficos" se expandirá, con su nueva entrada resaltada. En la sección de la derecha, asegúrese de que los siguientes campos son rellenados: "Nombre", "Comentario", "Comando", "Tipo" y "Ruta de trabajo".
 - El campo "Nombre" debería estar ya rellenado , pero puede modificarlo en cualquier momento.
 - Rellene el campo "Comentario". Aquí es donde define la etiqueta que aparecerá sobre el icono cuando pase con el puntero por encima.
 - Pinche en el icono de la carpeta al final del campo "Comando" para buscar el icono de blender. Seleccione el icono y pulse en "OK" para volver al "Editor de Menú".
 - El campo "Tipo" debería ser "Application".
 - El campo "Ruta de trabajo" debería ser el mismo que el del campo "Comando", con el nombre del programa suprimido. Por ejemplo, si en el campo "Comando" pone `/home/user/blender-publisher-#.##-linux-glibc#.##-ARCH/blender`, el campo "Ruta de trabajo" debería ser `/home/user/blender-publisher-#.##-linux-glibc#.##-ARCH/`.
5. Pinche en "Aplicar" y cierre el "Editor de Menú".

Para añadir un enlace a Blender en el panel de KDE, pulse el boton derecho del ratón sobre un espacio vacío del panel, y entonces pulse "Añadir", luego "Botón", después "Gráficos" y por último selecciona "Blender" (o el nombre que haya elegido en el paso 3). Alternativamente, puede navegar a través del submenú "Panel de configuración" del menú de KDE y hacer "Añadir", "Botón", "Gráficos" y "Blender".

Para añadir un icono para Blender en el escritorio, abra el navegador Konqueror (que se puede encontrar en el Panel, o en submenú "Sistema" del menú de KDE) y navegue hasta el icono de Blender que se encuentra donde descomprimió por primera vez el programa. Pinche y seleccione el icono y muevalo desde el navegador hasta un espacio vacío de su escritorio. Se le preguntará si quiere copiarlo, moverlo o enlazarlo en esa localización. Elija enlazarlo.

Añadir un icono para blender en GNOME

1. Selecciona "Editar menús" en el submenú Panel del menú de GNOME.
2. Selecciona el submenú "Gráficos", y pinche en el botón "Nuevo dispositivo".
3. En el panel derecho, rellene los campos "Nombre:", "Comentario:" y "Comando:". Rellene el campo "Nombre:" con el nombre del programa, por ejemplo "Blender". Puede darle el nombre que más le guste, solo es lo que aparecerá en el menu, pero no afectará a la funcionalidad del programa. Rellene el campo "Comentario:" con un comentario descriptivo. Esto es lo que se mostrará en las etiquetas emergentes. Rellene el campo "Comando:" con la ruta hasta el programa blender, por ejemplo, `/home/user/blender-publisher-#.##-linux-glibc#.##-ARCH/blender`
4. Pinche en el botón "No Icono" para elegir un icono. Puede haber o no un icono para Blender en la localización por defecto. Puede hacer uno, o buscar el que viene con KDE. El icono de KDe debería estar en `/opt/kde/share/icons/hicolor/48x48/apps/blender.png`. Si su directorio de instalación es diferente, puede buscarlo usando este comando en un terminal o una consola: **`find / -name "blender.png" -print`**
5. Pinche en el botón "Salvar" y cierre el "Editor de Menú".

Para añadir un icono al panel, pincha con el botón derecho del ratón en un espacio vacío del Panel, entonces selecciona "Programas", luego "Gráficos", y después "Blender". Alternativamente, podría pinchar en el menú GNOME, seleccionar "Panel", entonces "Añadir al panel", luego "Lanzar desde el menu", después "Gráficos", y por último "Blender".

Para añadir un icono para Blender en el escritorio, abra el navegador Nautilus (doble-click en el icono de la casa en la esquina superior izquierda de su escritorio, o pinche en el menú GNOME, entonces "Programas", luego "Aplicaciones", y "Nautilus"). Navegue hasta el directorio que contiene el icono del programa Blender. Píche con el botón derecho del ratón, y muevaló hacia el escritorio. Un menú aparecerá preguntándole si quiere copiar, mover, enlazar o cancelar. Seleccione enlazar.

2.1.4. FreeBSD

2.1.4.1. Instalación

Descargue el archivo `blender-2.3#-freebsd-#.##-i386.tar.gz` de la sección de descargas del sitio web de Blender. Aquí 2.3# es la versión de Blender, #.# es la versión de FreeBSD y i386 es la arquitectura de la computadora.

Para comenzar Blender únicamente abra una consola y ejecute el comando **`./blender`**, por supuesto cuando el servidor gráfico 'X' esté funcionando.

2.1.5. Irix

2.1.5.1. Instalación

Descargue el archivo `blender-2.3#-irix-6.5-mips.tar.gz` desde la sección de descargas de la web de Blender. Aquí 2.3# es la versión de Blender, 6.5 es la versión de Irix y *mips* es la arquitectura del ordenador.

Para comenzar Blender unicamente abra una consola y ejecute el comando **`./blender`**, por supuesto cuando el servidor gráfico 'X' esté funcionando. Blender se desarrolló originalmente para la plataforma IRIX, pero actualmente esta plataforma no está siendo activamente mantenida en todas sus versiones. Para algunas estaciones de trabajo se ha informado de la existencia de algunos problemas que afectan al rendimiento.

2.1.6. Solaris

2.1.6.1. Instalación

Descargue el archivo `blender-2.3#-solaris-2.8-sparc.tar.gz` desde la sección de descargas de la web de Blender. Aquí 2.3# es la versión de Blender, 2.8 es la versión de Solaris y *sparc* es la arquitectura de la computadora.

Actualmente no hay instrucciones disponibles para Solaris. Por favor utilice los foros del sitio web de Blender para obtener ayuda.

[Prev](#)

Consigue ayuda - la comunidad de usuarios de Blender

[Home](#)[Up](#)[Next](#)

Instalando Blender a partir del código fuente

2.2. Instalando Blender a partir del código fuente

Relevante a Blender v2.31

Este documento describe las herramientas necesarias para instalar Blender a partir del código fuente, procedente tanto del CVS como de un paquete de fuentes. La instalación a partir del CVS requiere el uso de más herramientas. Mientras que esto puede ser un poco más problemático que instalarlo a partir de un paquete de fuentes, puede ser necesario para algunas personas. Por ejemplo, cuando quieres instalar Blender para una plataforma no soportada o cuando quieres implementar algunas características nuevas.

Esta es la versión inicial de este documento. Lo que significa que está incompleta y que algunos procedimientos o conceptos pueden no ser correctos para tu sistema. Por favor recuérdalo cuando lo leas. También recuerda que Blender es un producto complejo y requiere un entorno adecuado para instalarlo.

2.2.1. Obteniendo el código fuente

Los siguientes párrafos describirán como y donde obtener el código fuente para instalar Blender.

2.2.1.1. Obtenga el último paquete estable de fuentes

Las fuentes están disponibles en el CDROM que acompaña este libro. También puede descargarlas desde la web, <http://www.blender3d.org/Download/?sub=Source>

2.2.1.2. Obtenga las últimas fuentes desde el CVS

CVS son las siglas de Sistema Concurrente de Versiones. Es una herramienta de configuración que mantiene varios ficheros del código fuente en el repositorio central. CVS permite a los desarrolladores actualizarse rápidamente al último estado del repositorio y subir sus cambios. La herramienta mantiene una lista de los cambios entre cada versión de un fichero. Para obtener el estado actual del repositorio, no necesita tener un "usuario" para acceder a las fuentes. Esta característica es opcional, pero en un desarrollo de código abierto es casi una obligación. Para subir sus cambios al repositorio necesita tener acceso de desarrollador. Como este documento solo describe como obtener el último estado de las fuentes, los procedimientos para subir sus cambios no serán descritos aquí.

Para obtener el último estado de las fuentes escribe:

```
export CVSROOT=:pserver:anonymous@cvs.blender.org:/cvs01
```

```
cvs login
```

```
password: Enter
```

```
cvs -z3 co blender
```

Por favor no utilice un nivel mayor de compresión para acceder al servidor de Blender

Si ya ha obtenido las fuentes del servidor, puede usar el comando **update** para actualizar las fuentes al estado actual del repositorio. Utilice el comando **cd** directorio_de_blender para situarse en el directorio raíz de Blender y a continuación escriba:

```
cvs -z3 update .
```

2.2.2. Librerías necesarias

Blender es un programa que usa una gran cantidad de paquetes externos para expandir su funcionalidad. Cada uno de esos paquetes tiene, como Blender, un sistema de versiones. Las versiones más recientes de tales paquetes probablemente tendrán más características y menos problemas. Como desarrollador es excitante trabajar con las últimas características disponibles para obtener lo mejor del programa. De todas formas, el número de desarrolladores es mucho menor que el número de usuarios, que no están interesados en la "última" característica sino en tener una aplicación que funcione. Desde que Blender es capaz de funcionar en múltiples plataformas, las librerías tienen que disponer de las mismas funcionalidades básicas para todas ellas.

La tabla de abajo muestra las librerías y la mínima versión necesaria de esas librerías. Con el tiempo puede que esas versiones mínimas sean incrementadas ya que la demanda de nuevas características es bastante alta.

Table 2-1. Versión mínima de las librerías

Librería	Versión
glibc	2.2.4
libjpeg	6b
libpng	1.0.14

libsdl	1.0
libz	1.1.4
mesa	3.4.2
openAL	N/A
openGL	1.1 (1.2 for engine)
python	2.2

No todas las librerías son necesarias en todos los sistemas. La siguiente tabla da una visión general de las actuales plataformas soportadas y las librerías requeridas en esas plataformas. Una 'X' significa que es necesaria, una '-' significa que no es necesaria y una 'O' significa que es opcional.

Table 2-2. Requerimientos de las librerías dependiendo del sistema utilizado

Librería	Linux	Windows	FreeBSD	IRIX	MacOS X
glibc	X	-	X	X	X
libjpeg	X	X	X	X	X
libpng	X	X	X	X	X
libsdl	O	O	O	O	O
libz	X	X	X	X	X
mesa	X	X	X	-	-
openAL	X	X	X	X	X
openGL	-	-	-	X	X
python	X	X	X	X	X

2.2.3. Herramientas necesarias

Teniendo las librerías necesarias instaladas y las fuentes de Blender descargadas en tu sistema significa que ya eres capaz de instalar Blender. El proceso de creación requiere que algunas herramientas se encuentren disponibles en tu sistema. En la tabla de abajo, se muestra una lista con las herramientas y su mínima versión. La tercera columna muestra si la herramienta es necesaria solamente para el CVS ('X'). Si la herramienta no es necesitada para la construcción a partir del paquete de fuentes se usará '-'.

Table 2-3. Mínima versión de las herramientas

Herramienta	Versión	CVS	Nota

autoconf	2.53	X	
automake	1.6.2	X	
cvs	1.11.1p1	X	
docbook	3.1	O	
doxygen	N/A	O	
gawk	3.1.0	X	
gcc	2.96	-	
gettext	0.11	-	
gmake	3.79.1	-	
m4	1.4	X	
sed	3.02	X	
sh	2.05.1	-	
Visual C++	6.0 SP5	-	Solo Windows

Tip Python

Python no está incluido en esta tabla, aunque es necesario para instalar Blender. La razón de esto es que python también es necesario como librería externa y por tanto tendría que estar ya instalado tal como se indicaba en la sección anterior.

2.2.4. Instalando Blender

Hay dos sistemas de construcción para usarlos con el compilador gcc u otros compiladores; Makefiles, los cuales perduran desde el periodo en el que Blender era desarrollado por NaN, y el automake/ autoconf mas conocido por "configure". Utilizar "configure" puede sobrescribir los Makefiles de NaN, asi que tienes que elegir entre uno y otro sistema.

Para Windows MSVC, Blender soporta el uso de ficheros de proyecto y espacios de trabajo.

Los ficheros que describen detalladamente el procedimiento de construccion están disponibles en el directorio raiz de Blender y son:

- INSTALL: información general, enlaces para descargar las librerías
- INSTALL.AUTO: utilización de autoconf y scripts de configuración
- INSTALL.MAKE: utilización de los makefiles
- INSTALL.MSVC: utilización de los ficheros de proyecto de Microsoft Visual C

2.2.5. Soporte técnico

- portal: <http://www.blender.org>
- visión general: http://www.blender.org/docs/get_involved.html
- lista de correo: <http://www.blender.org/mailman/listinfo/bf-committers/>
- reporte de fallos: http://projects.blender.org/tracker/?group_id=9
- IRC: [#blendercoders](irc://irc.freenode.net)

[Prev](#)

Instalación

[Home](#)

[Up](#)

[Next](#)

Comprendiendo la interfaz

Chapter 3. Comprendiendo la interfaz

By Martin Kleppmann

Si es nuevo en Blender, debería asimilar como trabajar con la interfaz de usuario antes de ponerse a modelar. Los conceptos detrás de la interfaz de Blender hacen que esta no sea muy estandar, sino que sea diferente de los otros paquetes 3D. Especialmente los usuarios de Windows necesitarán coger el truco a la manera en la que Blender maneja los controles, tales como botones o movimientos de ratón. Pero esta diferencia es de hecho la gran fuerza de blender: una vez comprenda la manera en la que funciona Blender, encontrará que puede hacer su trabajo extraordinariamente rápido y de manera muy productiva.

Por otro lado, la interfaz de Blender ha cambiado sustanciosamente desde la versión 2.28 a la versión 2.3, así que incluso los usuarios más experimentados podrán sacar partido a este capítulo.

3.1. Conceptos de la Interface de Blender

Aplicable a Blender v2.31

La interface de usuario es el mecanismo de interacción mutua entre el usuario y el programa. El usuario se comunica con el programa mediante el teclado y el mouse, el programa responde por medio de lo que muestra en pantalla.

3.1.1. El teclado y el ratón

La interface de Blender saca provecho de los ratones de tres botones y una amplia gama de atajos de teclado (en el Volumen II se encuentra una discusión detallada). Si el ratón sólo tiene dos botones, es posible emular el botón central (la [Section 3.3.4](#) describe como). Es posible usar un ratón con rueda, pero no es obligatorio, ya que también existen atajos de teclado que cumplen la misma función.

Este libro asume las siguientes convenciones para describir la entrada de usuario:

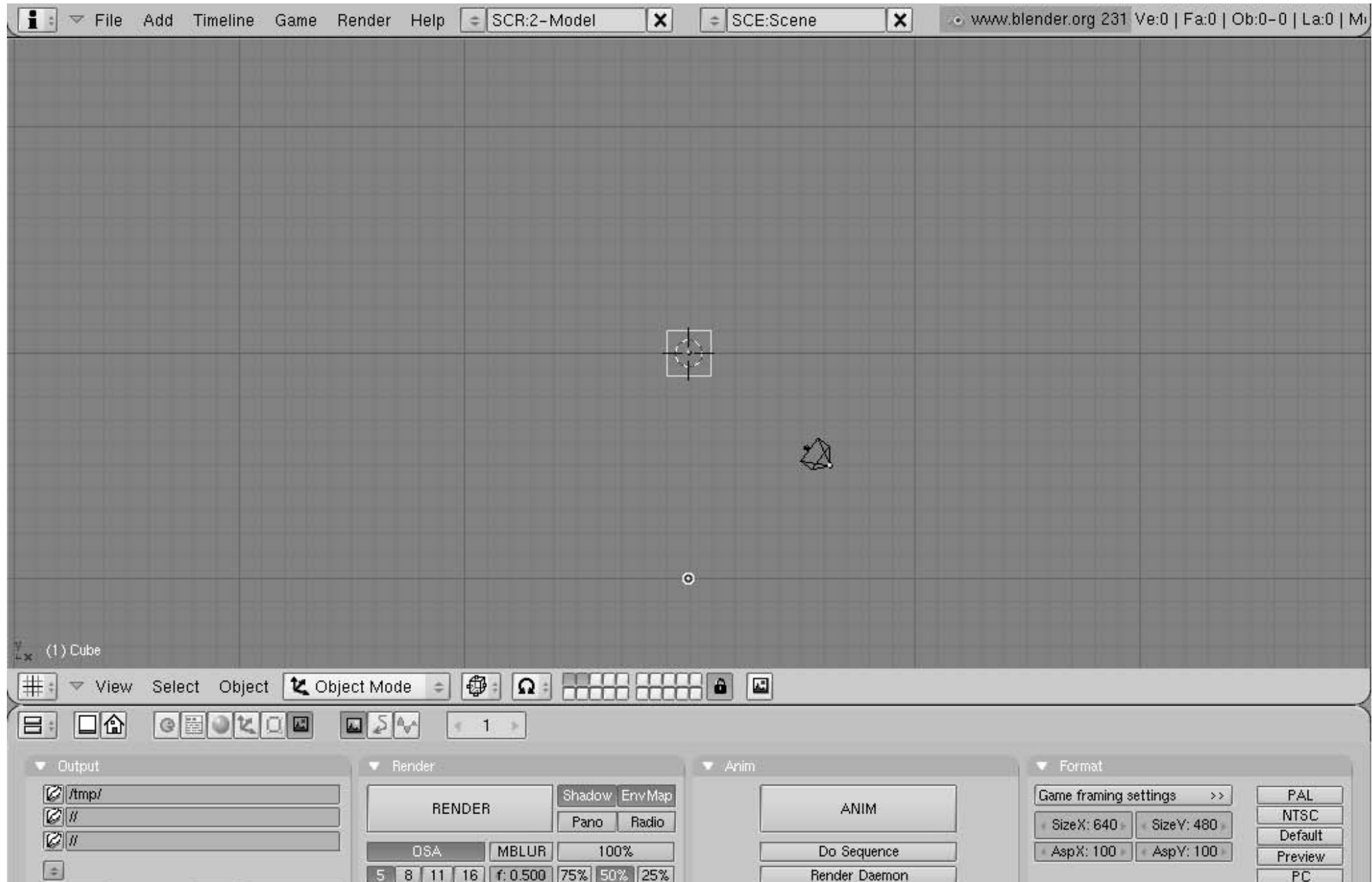
- Los botones del ratón se abrevian como **BIR** (botón izquierdo del ratón), **BCR** (botón central del ratón) y **BDR** (botón derecho del ratón).
- Si el ratón tiene una rueda, **BCR** se refiere a hacer click con la rueda como si ésta fuera un botón, mientras que **RR** significa girar la rueda.
- Las letras de los atajos se nombran añadiendo **KEY** a la letra, es decir **GKEY** se refiere a la letra *g* en el teclado. Las teclas pueden ser combinadas con los modificadores **SHIFT**, **CTRL** y/o **ALT**. Generalmente para las teclas modificadas el sufijo **KEY** se descarta, por ejemplo **CTRL-W** o **SHIFT-ALT-A**.
- **NUM0** a **NUM9**, **NUM+** y así sucesivamente, se refiere a las teclas ubicadas en el teclado numérico. Generalmente NumLock debería ser activada.
- En el caso de otras teclas, se les refiere usando sus nombres, tal como **ESC**, **TAB**, **F1** a **F12**.
- Otras teclas especiales para tener en cuenta son las teclas de dirección, **ARRIBA**, **ABAJO**, **IZQUIERDA** y **DERECHA**.

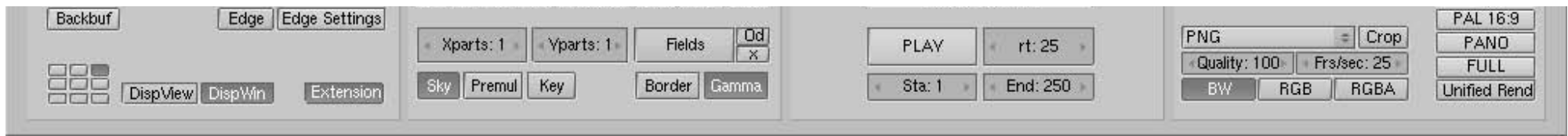
Dado que Blender hace un uso tan extensivo del ratón y el teclado, entre los usuarios se ha hecho popular una "regla de oro": mantener una mano en el ratón y la otra en el teclado! Si normalmente usa un teclado que es considerablemente diferente de la distribución de teclas inglesa, podrá ir pensando en cambiar a uno con ésta o la distribución americana para el trabajo con Blender. Las teclas de mayor uso están agrupadas de tal manera que pueden ser alcanzadas con la mano izquierda en posición estándar (dedo índice sobre **FKEY**) en la distribución de teclado inglesa. Se asume que se usa el ratón con la mano derecha.

3.1.2. El sistema de ventanas

Ahora es tiempo de iniciar Blender y empezar a explorar.

Figure 3-1. La escena por defecto de Blender.





La [Figure 3-1](#) muestra la vista de pantalla que debe obtenerse al haber iniciado Blender(excepto por las flechas y el texto añadido). Por defecto está separada en tres ventanas: el menu principal en la parte alta, la vista 3D grande y la Ventana de Botones abajo. La mayoría de las ventanas tiene un encabezado (la franja con un fondo gris claro que contiene botones con iconos - por esto se le llamará, también, *Barra de Herramientas*); si está presente, el encabezado puede ubicarse en la parte alta (como en la Ventana de Botones) o en la parte baja (como en la vista 3D) del área de una ventana.

Si se mueve el ratón sobre una ventana, es posible notar que su encabezado cambia a un gris más claro. Esto significa que ésta está "enfocada"; todos los atajos de teclado que se presionen afectarán ahora el contenido de esta ventana.

Es posible personalizar el sistema de ventanas de Blender para satisfacer las necesidades y gustos del usuario. Una ventana nueva puede ser creada fraccionando una existente. Esto se hace "enfocando" la ventana que se quiere fraccionar (poner el puntero del ratón dentro de esta), hacer click en el borde con **BCR** o **BDR** y seleccionar *Split Area* ([Figure 3-2](#)). Ahora es posible ubicar el nuevo borde haciendo click con **BIR**, o cancelar el fraccionamiento presionando la tecla **ESC**. La nueva ventana será un clon de la ventana que fue fraccionada, pero ahora puede ser convertida a una ventana de otro tipo, o hacer que muestre la escena desde otro punto de vista.

Tip Elementos de la Interface

Las etiquetas en los botones de la interface, los elementos de menú y, en general, cualquier texto mostrado en la pantalla se resalta en este libro *de esta manera*.

Figure 3-2. El menu de Split para crear ventanas nuevas.

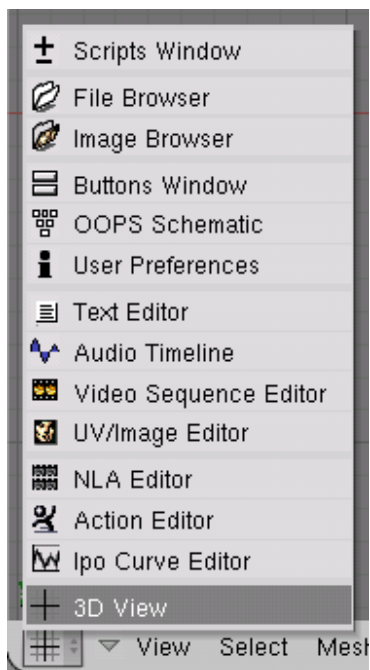


Un nuevo borde vertical es creado al escoger *Split Area* en un borde horizontal y viceversa. Cada ventana puede ser redimensionada arrastrando un borde con el **BIR**. Para reducir el número de ventanas, se hace click en un borde entre dos ventanas con **BCR** o **BDR** y se escoge *Join Areas*. La ventana resultante recibe las propiedades de la ventana enfocada previamente.

Para ajustar la posición de un encabezado se hace click con el **BDR** sobre el encabezado y se escoge *Top* o *Bottom*. También es posible ocultar el encabezado seleccionando *No Header*, pero esta opción sólo se recomienda si todos los atajos de teclado relativos se conocen. Un encabezado oculto se puede volver a mostrar haciendo click en el borde de la ventana con **BCR** o **BDR** y seleccionando *Add Header*.

3.1.3. Tipos de ventanas

El marco de cada ventana puede contener diferentes tipos y conjuntos de datos, dependiendo de lo que se está haciendo. Estos pueden incluir modelos 3D, animación, materiales de la superficie, scripts Python, y así sucesivamente. Se puede seleccionar el tipo de cada ventana haciendo click en el botón de más a la izquierda en su encabezado con el **BIR** ([Figure 3-3](#)).

Figure 3-3. El menú de selección del tipo de ventana.

Se explicarán las funciones y el uso de cada tipo de ventana a lo largo del libro. Por ahora sólo se prestará atención a los tres tipos de ventana que se encuentran en la escena por defecto de Blender:

Vista 3D

Provee una vista gráfica de la escena en la cual se está trabajando. Es posible ver la escena desde cualquier ángulo con una variedad de opciones; para información detallada, vease [Section 3.2](#). Tener varias vistas en la misma pantalla es útil si se desea observar el efecto de los cambios desde distintas perspectivas al mismo tiempo.

Buttons Window(Ventana de Botones)

Contiene la mayoría de las herramientas para editar objetos, superficies, texturas, luces y mucho más. Esta ventana se usa constantemente si los atajos de teclado no se saben de memoria. Por supuesto, es posible tener más de una ventana de estas, cada una con un conjunto de herramientas diferente.

Preferencias de Usuario (Menu Principal)

Usualmente esta ventana está oculta, así que sólo la parte del menú está visible - para detalles, ver la [Section 3.3.4](#). Raramente se usa, pues contiene ajustes de la configuración global.

En Blender v2.30 hay varias novedades. Primero que todo, los encabezados de ventana tienden a ser mucho más limpios, menos saturados por botones y los menús ahora están presentes en muchos encabezados.

La mayoría de los encabezados, inmediatamente después del botón de menú "Tipo de Ventana", exhibe un grupo de menús; esta es una de las nuevas características principales de la interface 2.30. Los menús ahora permiten acceder directamente a muchas de las características y comandos que anteriormente sólo eran accesibles por medio de atajos de teclado o botones misteriosos. Los menús pueden ser ocultados y mostrados por medio del botón triangular junto a ellos.

Los menús no son sólo dependientes de la ventana (cambian con el tipo de ventana) sino también dependientes del contexto (cambian de acuerdo con el objeto seleccionado) así que siempre son muy compactos, mostrando sólo las acciones que pueden ser realizadas.

Todas las entradas de menú muestran el atajo de teclado relativo, si lo hay. El flujo de trabajo de Blender se optimiza al máximo cuando se usan atajos de teclado. Así que el resto de este libro ofrecerá mayormente atajos de teclado, en lugar de elementos de menú. De todas formas los menús son muy preciados, pues ofrecen una visión general, hasta donde es posible, de todas las herramientas y comandos que Blender ofrece.

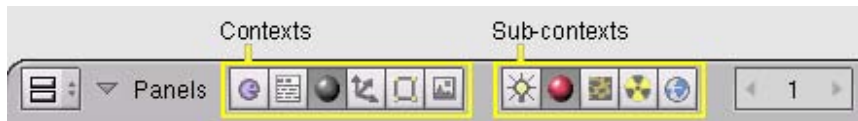
Una característica de las ventanas que a veces resulta práctica para hacer edición precisa es la de maximizar a pantalla completa. Si se usa el elemento de menú *View>Maximize Window* apropiado o el atajo de teclado **CTRL-ABAJO**, la ventana enfocada se extenderá para llenar la pantalla completa. Para regresar al modo de tamaño normal, se usa el elemento de menú *View>Tile Window* o **CTRL-ARRIBA**.

3.1.4. Contextos, Paneles y Botones

Los botones de Blender son mucho más emocionantes que los de la mayoría de otras interfaces de usuario y se volvieron aún mejores en la versión 2.30. Esto es mayormente debido al hecho de que son basados en vectores y dibujados en OpenGL, lo que los hace elegantes y permite hacer zoom en ellos.

Los botones se hallan agrupados mayormente en la Ventana de Botones. Desde Blender 2.3 la Ventana de Botones muestra *seis* contextos principales, los cuales se pueden seleccionar por medio de alguno de los botones en el primer grupo en el encabezado ([Figure 3-4](#)), a su vez cada uno de estos puede ser subdividido en un número variable de sub-contextos, a los cuales se puede acceder a través de los botones del segundo grupo en el encabezado ([Figure 3-4](#)):

Figure 3-4. Contextos y Sub-Contextos



- *Lógica* - atajo **F4**
- *Scripts* - sin atajo
- *Shading* - atajo **F5**
 - *Lámpara* - sin atajo
 - *Material* - sin atajo
 - *Textura* - atajo **F6**
 - *Radiosidad* - sin atajo
 - *Mundo* - atajo **F8**
- *Objeto* - atajo **F7**
- *Edición* - atajo **F9**
- *Escena* - atajo **F10**
 - *Renderizado* - sin atajo
 - *Animación/Reproducción* - sin atajo
 - *Sonido* - sin atajo

Usualmente, una vez el contexto ha sido seleccionado por el usuario, el subcontexto es determinado por Blender basándose en el objeto activo. Por ejemplo, en el contexto "Shading", si una lámpara está seleccionada, el subcontexto muestra los Botones de Lámpara, si una malla u otro objeto renderizable está seleccionado, entonces el subcontexto activo vendrá a ser el de Botones de Material y si una cámara está seleccionada el subcontexto activo es Mundo.

La novedad más notable en la interface es probablemente la presencia de *Paneles* para agrupar lógicamente los botones. Cada panel es del mismo tamaño. Pueden ser movidos haciendo click con el **BIR** en su encabezado y arrastrando. Los paneles pueden ser alineados haciendo click con el **BDR** en la ventana de botones y escogiendo la distribución deseada en el menú que aparece ([Figure 3-5](#)).

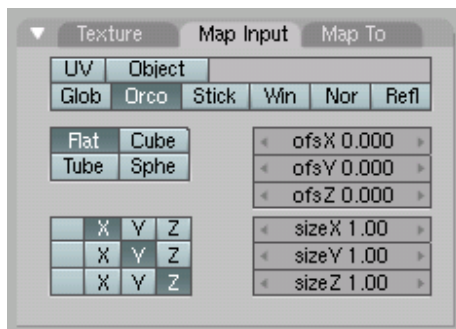
Figure 3-5. Menu de la Ventana de Botones.



La **RR** desplaza los paneles en su dirección de alineamiento, **CTRL-RR** y **CTRL-BCR** hacen zoom de acercamiento y alejamiento. Paneles individuales pueden ser colapsados/expandidos haciendo click con el **BIR** en el triángulo ubicado a la izquierda de su encabezado.

Los paneles particularmente complejos se organizan en *Fichas*. Haciendo click con el **BIR** en una ficha en el encabezado del panel cambian los botones mostrados ([Figure 3-6](#)). Las fichas se pueden "desligar" de un panel para formar paneles independientes haciendo click sostenido con el **BIR** en su encabezado y arrastrándolas fuera. De manera similar, paneles separados se pueden convertir en un solo panel con fichas soltando el encabezado de un panel sobre otro.

Figure 3-6. Panel con fichas.



Como un último elemento de interface, hay varios tipos de botones que se disponen en las fichas de los paneles:

Botones de Operación. Estos son botones que efectúan una operación cuando se les hace click (con el **BIR**, como todos los botones). Pueden ser identificados por el color pardusco que tienen en el esquema visual por defecto de Blender. ([Figure 3-7](#)).

Figure 3-7. Un botón de operación



Botón Alternante. Los botones alternantes vienen en varios tamaños y colores ([Figure 3-8](#)). Los colores verde, violeta y gris no modifican la función, sólo ayudan a facilitar la agrupación visual y reconocer el contenido de la interface más rápidamente. Hacer click en un botón de estos no genera acción alguna, sólo alterna un estado entre "on" y "off".

Algunos botones también tienen un tercer estado que se puede identificar por que el texto se vuelve amarillo (el botón Ref en la [Figure 3-8](#)). Usualmente el tercer estado significa "negativo" y el estado normal "on" significa "positivo".

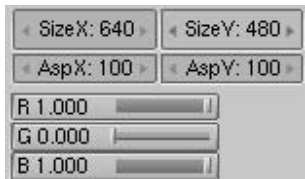
Figure 3-8. Botones alternantes



Botones de Radio. Los botones de radio son grupos particulares de botones alternantes mutuamente excluyentes. Tan sólo uno de estos botones puede estar "on" en un momento dado.

Botones Numéricos. Los botones numéricos ([Figure 3-10](#)) pueden ser identificados por sus títulos, que contienen ":" seguido por un número. Los botones numéricos son manipulados de varias formas: Para incrementar el valor, se hace click con el **BIR** en la parte derecha del botón, donde se observa un triángulo pequeño; para decrementarlo, se hace click en la parte izquierda del botón, donde se encuentra otro triángulo. Para cambiar el valor en un rango más amplio, se hace click sostenido con el **BIR** y se arrastra el ratón a la derecha o a la izquierda. Si mantiene oprimida la tecla **CTRL** mientras lo hace, el valor cambia con saltos grandes; si sostiene oprimida la tecla **SHIFT**, tendrá un control más sutil sobre los valores. La tecla **ENTER** presta la misma utilidad que el **BIR**.

Figure 3-9. Botones numéricos



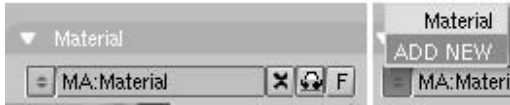
Es posible ingresar un valor usando el teclado si se sostiene la tecla **SHIFT** y se hace click con el **BIR**. Presionando la combinación **SHIFT-RETROCESO** se elimina el valor contenido en el botón; con **SHIFT-IZQUIERDA** se mueve el cursor al inicio; y **SHIFT-DERECHA** mueve el cursor al final. Presionando **ESC** se restaura el valor original.

Algunos botones de número contienen un deslizador y no sólo un número con triángulos laterales. El mismo método de operación se aplica para estos, excepto que el click con el **BIR** se debe efectuar en la parte izquierda o derecha del deslizador, en tanto que hacer click en la etiqueta o el número automáticamente iniciará el modo de entrada por teclado.

Botones de Menú. Los botones de menú se usan para escoger de una lista de elementos creada dinámicamente. Los botones de menú son usados principalmente para enlazar "DataBlocks" entre sí. (DataBlocks son estructuras como Mallas, Objetos, Materiales, Texturas y otros; al enlazar un material a un objeto, éste queda asignado). Un ejemplo de este tipo de bloque de botones se muestra en la [Figure 3-10](#). El primer botón (con los diminutos triángulos apuntando arriba y abajo) abre un menú que permite seleccionar el DataBlock al cual efectuar el enlace, esto se hace sosteniendo oprimido el **BIR** y liberándolo cuando el puntero se halle sobre el elemento deseado. El segundo botón muestra el tipo y nombre del DataBlock enlazado y permite editar su nombre después de hacer click con el **BIR**. El botón con la "X" elimina el enlace, el botón con el "carro" genera un nombre automáticamente para el DataBlock y el botón con la "F" especifica que el DataBlock deberá ser almacenado en el archivo aún si no está siendo usado (no tiene enlace alguno).

**Objetos sin enlace alguno**

Los datos sin enlace *sólo* se pierden *hasta* que se cierra Blender. Esta es una poderosa característica que permite deshacer errores. Si un objeto es eliminado el material que tiene asignado pasa a estar sin enlace, pero aún existe! Solo se debe re-enlazar a otro objeto o presionar el botón "F".

Figure 3-10. Botones de enlace de DataBlocks

3.1.5. Caja de Herramientas(ToolBox)

Al presionar la tecla **ESPACIO** en la Vista 3D, o sostener el **BIR** o el **BDR** con el ratón quieto durante más de medio segundo se abre la Caja de Herramientas. Esta contiene 6 contextos principales dispuestos en dos líneas, cada uno de los cuales abre menús y submenús.

Tres de estos contextos abren los mismos tres menús presentes en el encabezado de la vista 3D; los otros tres, *Add* permite añadir objetos nuevos a la escena mientras *Edit* y *Transform* muestran todas las operaciones que se pueden efectuar sobre el(los) objetos seleccionado(s). ([Figure 3-11](#)).

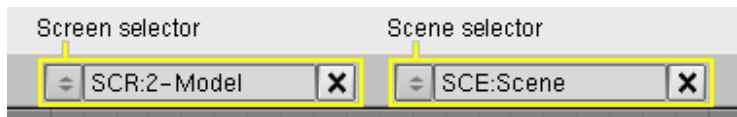
Figure 3-11. La Caja de Herramientas



3.1.6. Pantallas(Screens)

La flexibilidad de las ventanas de Blender permite crear entornos de trabajo personalizados para diferentes tareas, tales como modelado, animación y creación de scripts. A menudo resulta útil intercambiar rápidamente entre diferentes entornos dentro del mismo archivo. Esto es posible al crear varias Pantallas: Todos los cambios a las ventanas que se describen en [Section 3.1.2](#) y [Section 3.1.3](#) se almacenan en una pantalla, de modo que si las ventanas de una pantalla son modificadas, las otras no resultarán afectadas en lo absoluto. Pero la escena en la cual se está trabajando es la misma en todas las pantallas.

Blender incluye por defecto tres pantallas diferentes; están disponibles a través del Botón de Menú *SCR* en el encabezado de la Ventana de Preferencias de Usuario mostrado en la [Figure 3-12](#). Para cambiar a la pantalla siguiente, alfabéticamente, se debe presionar la combinación **CTRL-DERECHA**; para cambiar a la pantalla previa, alfabéticamente, se presiona la combinación **CTRL-IZQUIERDA**.

Figure 3-12. Selectores de Pantalla y Escena

3.1.7. Escenas

También es posible tener varias escenas dentro del mismo archivo de Blender. Las escenas pueden usar objetos de otras o ser completamente independientes entre sí. Con el botón de menú *SCE*, ubicado en el encabezado de la Ventana de Preferencias de Usuario, se puede crear o seleccionar una escena ([Figure 3-12](#)).

Cuando se crea una escena nueva, es posible escoger entre cuatro opciones para controlar su contenido:

- *Empty* Crea una escena vacía.
- *Link Objects* crea la escena nueva con el mismo contenido de la escena seleccionada actualmente. Los cambios en una escena afectarán igualmente a la otra.
- *Link ObData* crea la nueva escena basada en la escena seleccionada actualmente, con enlaces a las mismas mallas, materiales, etc. Esto significa que la posición de los objetos puede ser cambiada, junto con otras propiedades, pero modificaciones de las mallas, materiales, etc. afectarán igualmente la otra escena, a menos que se indique manualmente la creación de copias con usuario único (single-user copies).
- *Full Copy* crea una escena completamente independiente con copias del contenido de la escena seleccionada actualmente.

[Prev](#)[Instalando Blender a partir del código fuente](#)[Home](#)[Up](#)[Next](#)[Navegando en el Espacio 3D](#)

3.2. Navegando en el Espacio 3D

Relevante para Blender v2.31

Blender te permite trabajar en un espacio tridimensional, pero las pantallas de nuestros monitores son sólo bi-dimensionales. Para poder trabajar en tres dimensiones, debes ser capaz de cambiar tanto tu punto de vista como la dirección en la que ves la escena. Esto es posible en todas las vistas 3D.

Aunque vamos a describir la ventana de vista en 3D, la mayoría de las ventanas no-3D usan una serie de funciones equivalentes, por ejemplo es incluso posible arrastrar y escalar una Ventana de Botones y sus Paneles.

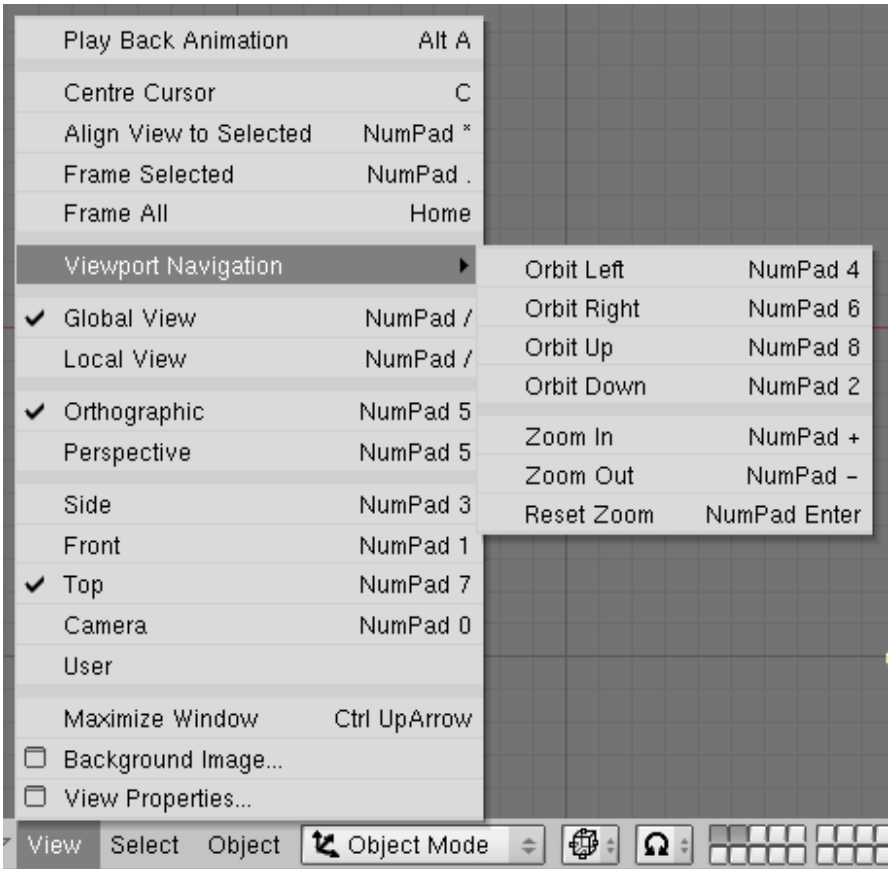
3.2.1. La dirección de la vista (rotando)

Blender ofrece tres direcciones de vista por defecto: Lateral, Frontal, and Superior. Como Blender usa un sistema de coordenadas "de mano derecha" con el eje Z apuntando hacia arriba, la vista "Lateral" corresponde a una mirada desde el eje X axis, en la dirección negativa; la vista "Frontal" es desde el eje Y; y la "Superior" desde el eje Z. Puedes seleccionar la dirección de la vista para una ventana 3D con las *Vista* entradas de Menú ([Figure 3-13](#)) o presionando los atajos de teclado **NUM3** para "Lateral", **NUM1** para "Frontal", y **NUM7** para "Superior".

Tip Atajos de teclado

Recuerda que la mayoría de los atajos de teclado afectan a la ventana activa, así que comprueba que el cursor del ratón está en el área donde quieres trabajar antes de usar los atajos de teclado!

Figure 3-13. El menú de vistas de una ventana 3D.



Aparte de estas tres direcciones por defecto, la vista puede ser girada con cualquier ángulo que necesites. Haz clic y arrastra **MMB** en el área de

la ventana: si comienzas en el medio de la ventana y te mueves hacia arriba y hacia abajo o hacia la derecha y la izquierda, la vista gira alrededor del centro de la ventana. Si comienzas en el borde y no te mueves hacia el centro, puedes girar alrededor de tu eje de vista. Juega un poco con esta función hasta que controles su manejo.

Para cambiar el ángulo de vista en pasos sucesivos, usa **NUM8** y **NUM2**, que corresponden al arrastre vertical. **MMB** o usa **NUM4** y **NUM6**, que corresponden al arrastre horizontal. **MMB**

3.2.2. Trasladando y Escalando la Vista.

Para trasladar la vista, pulsa **SHIFT** y arrastra **MMB** en la Ventana 3D. Para hacerlo en pasos sucesivos, usa las combinaciones de teclas **CTRL-NUM8**, **CTRL-NUM2**, **CTRL-NUM4** y **CTRL-NUM6** como con las rotaciones.

Puedes escalar la vista (acercarte y alejarte) pulsando **CTRL** y arrastrando **MMB**. Los atajos de teclado son **NUM+** y **NUM-**. Los sub-menús **View>>Viewport Navigation** contienen también estas funciones.

Tip La Rueda del Ratón

Si tienes un ratón con rueda, puedes realizar todas las acciones que harías con **NUM+** y **NUM-** girando la rueda (**MW**). La dirección de la rotación selecciona la acción.

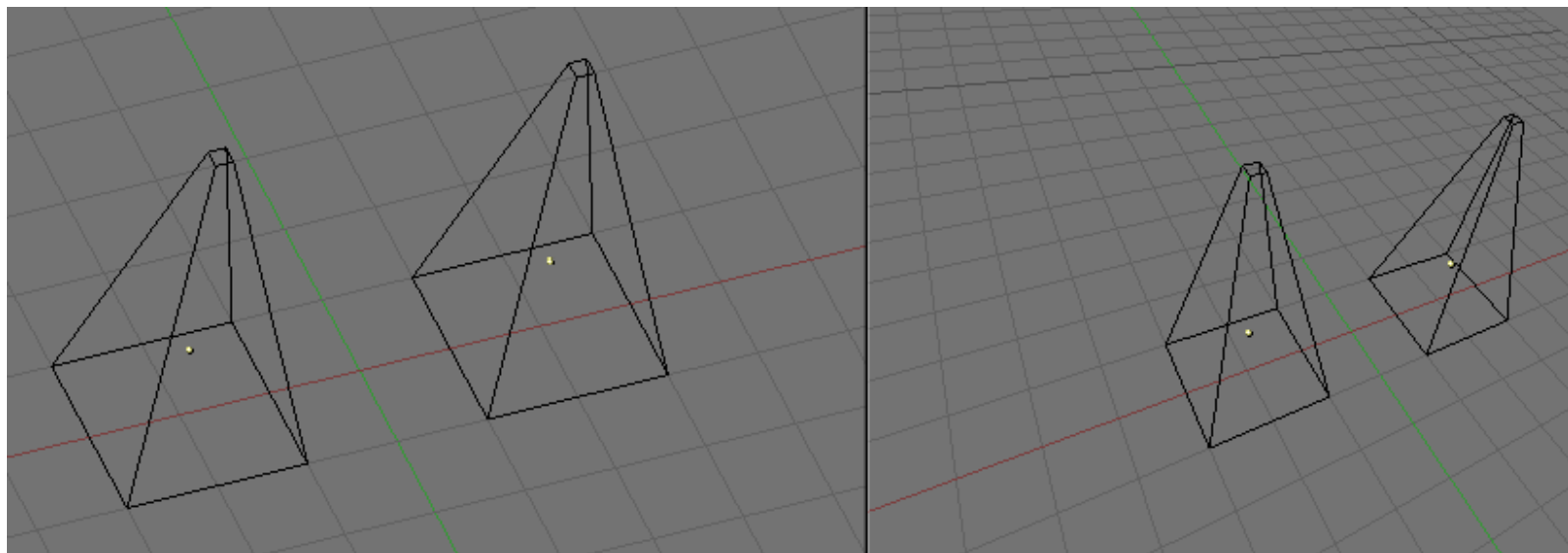
Tip Si te pierdes...

Si te pierdes en el espacio 3D, lo que no es raro, dos atajos de teclado te serán útiles: **HOME** cambia la vista para que puedas ver todos los objetos (**Vista>>Encajar todos** entrada de menú,) mientras **NUM.** escala la vista a los objetos actualmente seleccionados (**Vista>>Encajar seleccionados** entrada de Menú.)

3.2.3. Proyección en Perspectiva y Ortográfica

Cada Ventana 3D soporta dos tipos diferentes de proyección. Los mostramos en [Figure 3-14](#) : ortográfica (izquierda) y perspectiva (derecha).

Figure 3-14. Ortográfica (izquierda) y perspectiva (derecha).



Nuestro ojo está acostumbrado a ver en perspectiva porque los objetos distantes parecen más pequeños. La proyección Ortográfica, a menudo, parece incorrecta al principio porque los objetos permanecen del mismo tamaño independientemente de su distancia: es como ver la escena desde un punto infinitamente distante. Al contrario, la vista ortográfica es muy útil (es el modo por defecto en Blender y en la mayoría de aplicaciones 3D), porque ofrece un aspecto más "técnico" de la escena, facilitando el dibujo y los cálculos de proporciones.

Tip Perspectiva and Ortográfica

La vista en Perspectiva se construye geométicamente de esta manera: tienes una escena en 3D y tu eres un observador situado en un punto O. La escena con perspectiva 2D se construye situando un plano, una hoja de papel donde la escena 2D se dibuja en frente del punto O, perpendicular a la dirección de la vista. Para cada punto P en la escena 3D se dibuja una línea, pasando entre O y P. El punto de intersección S entre esta línea y el plano es la proyección en perspectiva de ese punto. Proyectando todos los puntos P de la escena obtendremos una vista en perspectiva.

En una proyección ortográfica, también llamadas "ortonormales", por otra parte, tienes una dirección de vista pero no un punto de vista O. La línea que se dibuja atraviesa el punto P de forma paralela a la dirección de la vista. Las intersecciones S entre la línea y el plano conforman la proyección ortográfica. Y proyectando todos los puntos P de la escena obtenemos la vista ortográfica.

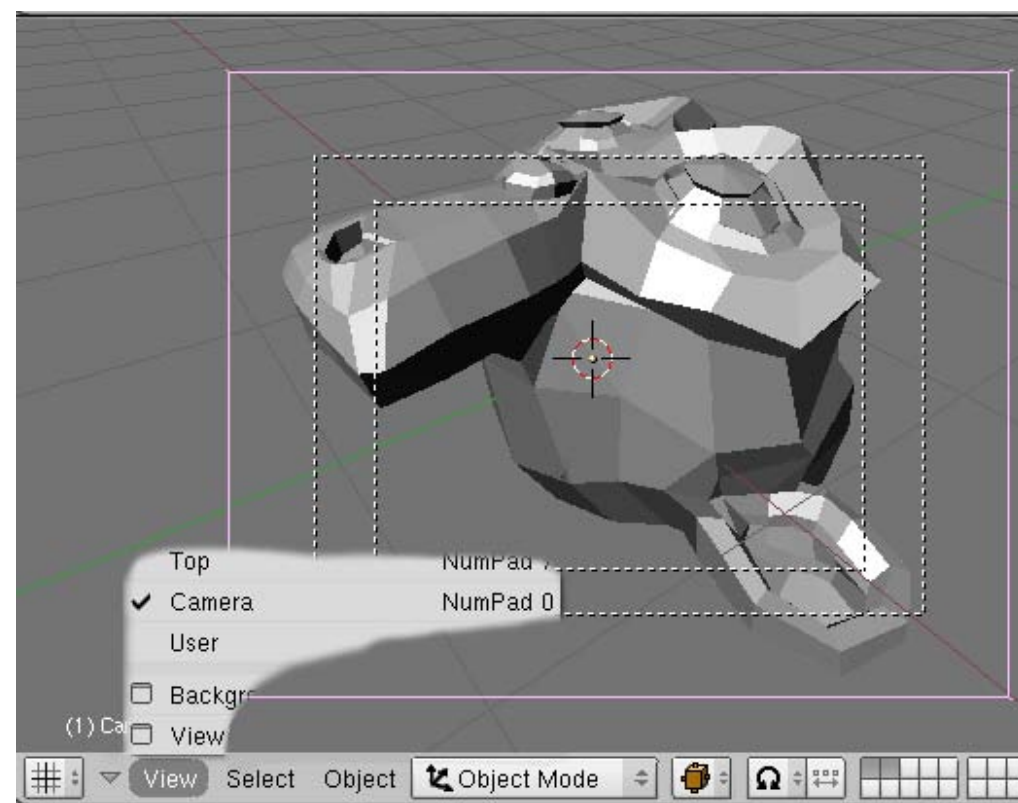
Para cambiar la proyección de una ventana 3D Viewport, elige *Vista>>Ortográfica* o *Vista>>Perspectiva* entradas de Menú ([Figure 3-13](#)). El atajo de teclado **NUM5** cambia entre los dos modos.

Tip Proyección de Cámara

Recuerda que cambiando la proyección de una vista 3D no afecta a la forma en que una escena será renderizada. El Renderizado es en perspectiva por defecto. Si necesitas crear un renderizado ortográfico, selecciona la cámara y presiona *Orto* en los Botones de Edición (**F9**) en el panel *Cámara*.

La entrada de Menú *Vista>>Cámara* pone la vista 3D en modo Cámara (Atajo: **NUM0**). Entonces la escena se muestra como será renderizada luego (vea [Figure 3-15](#)): la imagen renderizada contendrá todo lo que abarque la línea de puntos exterior. Es posible acercarse y alejarse en esta vista, pero para cambiar el punto de vista tendremos que mover o girar la Cámara.

Figure 3-15. Demostración de vista de Cámara.



3.2.4. Modo de Dibujado

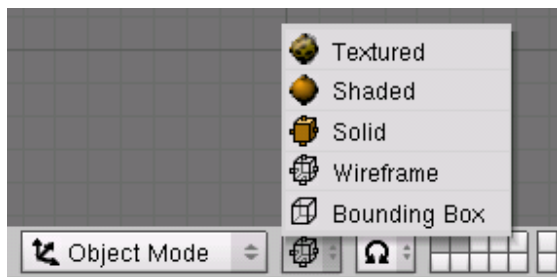
Dependiendo de la velocidad de tu ordenador, de la complejidad de tu escena y del tipo de trabajo que estés haciendo, puedes alternar entre diferentes modos de dibujado:

- *Texturizado* - Intenta dibujarlo todo tan completamente como sea posible, aunque no es aún una alternativa al renderizado. Nota que si no hay ninguna luz en tu escena todo saldrá negro.
- *Sombreado* - Dibuja superficies sólidas incluyendo el cálculo de iluminación. Como en el modo Texturado, no verás nada sin luces.

- **Sólido** - Las Superficies se dibujan como Sólidos, pero también funciona sin luces.
- **Enrejado** - Los objetos sólo consisten en líneas que hacen reconocibles las formas. Este es el modo de dibujado por defecto.
- **Caja de Límites** - Los objetos no son dibujados en absoluto; en vez de eso, este modo enseña sólo las cajas rectangulares que corresponden al tamaño y a la forma de cada objeto.

El modo de dibujado puede ser elegido con el botón de menú apropiado en la cabecera ([Figure 3-16](#)) o con atajos de teclado: **ZKEY** cambia entre Enrejado y Sólido, **SHIFT-Z** cambia entre Enrejado y Sombreado.

Figure 3-16. Un botón de modo de dibujado de una vista 3D.



3.2.5. Vista Local

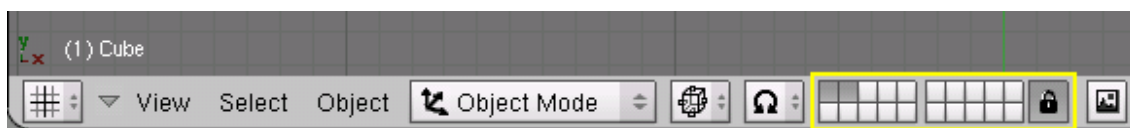
En vista local, sólo los objetos seleccionados son mostrados, lo que puede facilitar la edición en escenas complejas. Para entrar en vista local, primero selecciona los objetos que quieres (ver [Section 5.1](#)) y después usa la entrada de menú; *Vista>>Vista Local* usa la entrada de menú *Vista>>Vista Global* para volver a la vista global. ([Figure 3-13](#)). El atajo es **NUM/**, que cambia entre vista Local y Global.

3.2.6. El sistema de Capas

Las escenas en 3D a menudo se hacen mucho más confusas a medida que aumenta su complejidad. Para mantenerlas bajo control, los objetos pueden ser agrupados en "capas", para que sólo las capas que elijas sean mostradas en cualquier momento. Las capas de 3D difieren de las capas que puedes conocer de las aplicaciones gráficas en 2D: no tienen influencia en el orden de dibujado y están ahí (a excepción de algunas funciones especiales) únicamente para ofrecer al modelador un mejor control de las vistas.

Blender utiliza 20 capas; tú eliges las que se muestran con los pequeños botones sin título de la cabecera ([Figure 3-17](#)). Para seleccionar sólo una capa, haz clic en el botón apropiado con **LMB**; para seleccionar más de una, presiona **Mayúsculas** mientras haces clic.

Figure 3-17. Los botones de Capas de una vista 3D.



Para seleccionar capas con el teclado, presiona desde la tecla **1KEY** hasta la tecla **0KEY** (en el área principal del teclado) para las capas de 1 a 10 (la fila superior de botones), y **ALT-1** hasta **ALT-0** para las capas 11 a 20 (la fila inferior). La tecla **Mayúsculas** para selecciones múltiples también funciona con estos atajos.

Por defecto, el botón de bloquear a la derecha del bloque de capas está activado; esto significa que los cambios a la visibilidad de las capas afecta a todas las ventanas 3D. Para seleccionar ciertas capas sólo en una ventana, desactiva primero el bloqueo.

Para mover los objetos seleccionados a una capa diferente, presiona **MKEY**, selecciona la capa que quieres en el diálogo emergente, y finalmente haz clic en OK.

3.3. Funciones fundamentales

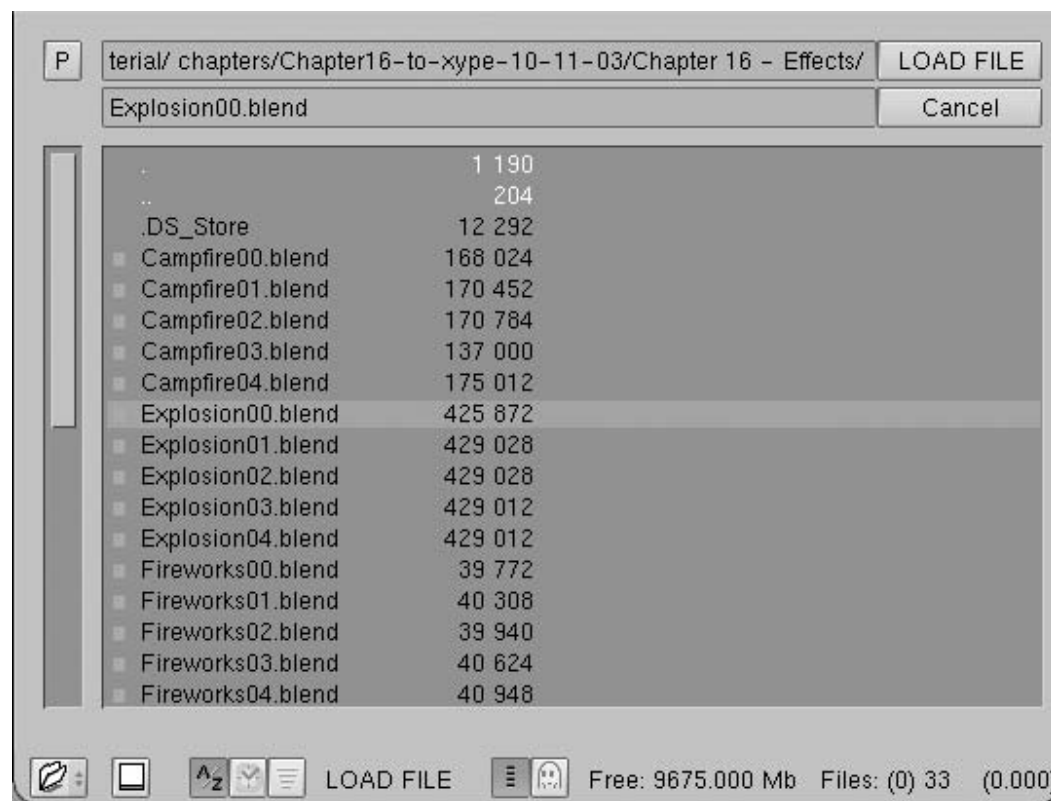
Relevante para Blender v2.31

3.3.1. Cargando archivos

Blender usa el formato de archivo *.blend* para salvar casi todo: Objetos, escenas, texturas, e incluso todas tus opciones de la ventana de preferencias del usuario.

Para cargar un archivo Blender del disco, presiona **F1**. La ventana activa se transformará temporalmente en la Ventana de selección de archivos como se muestra en [Figure 3-18](#). La barra de la izquierda puede ser arrastrada con **LMB** para ver el resto de la lista. Para cargar un archivo, selecciónalo con **LMB** y presiona **ENTER**, o simplemente haz clic en él con **MMB**.

Figure 3-18. Ventana de Selección de Archivos - cargando.



La caja de texto superior muestra la ruta del directorio actual, y la inferior contiene el nombre de archivo seleccionado. El botón P (**PKEY**) te sitúa en el directorio anterior y el botón

con la coma mantiene un listado de rutas usadas recientemente. En Sistemas operativos Windows, este último también contiene una lista de todos los discos (C:, D:, etc).

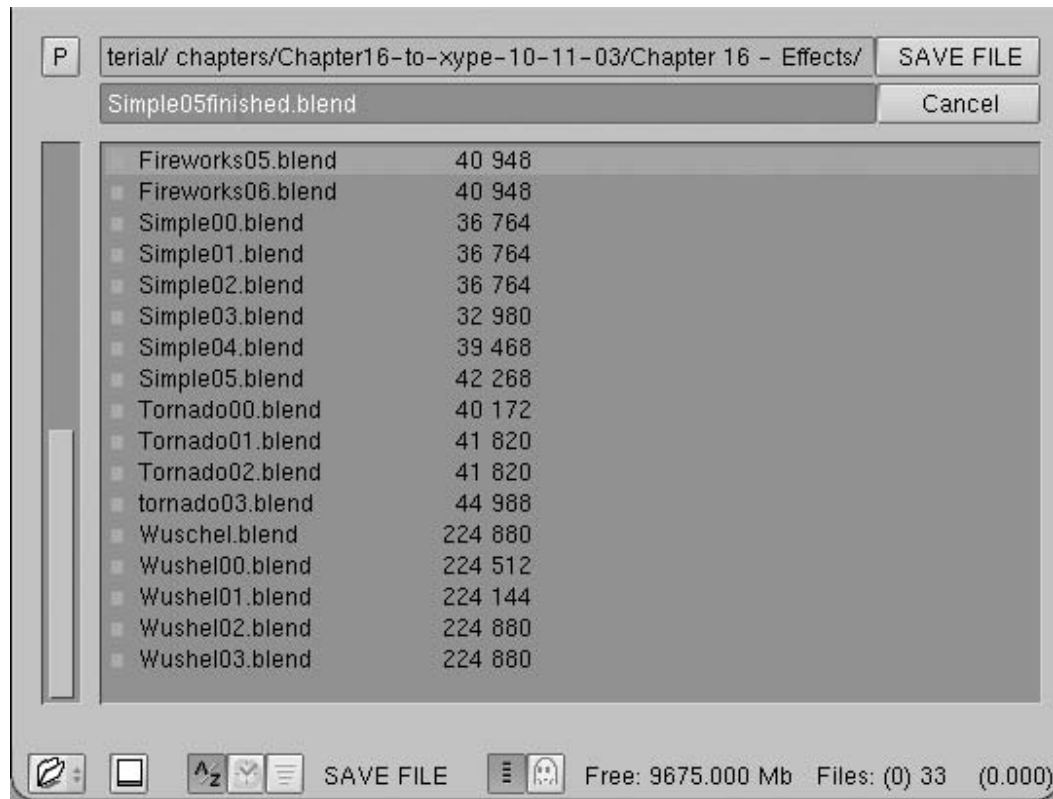
Note

Blender espera que tú sepas lo que estás haciendo! Cuando cargas un archivo, no se te pregunta si quieres guardar los cambios de la escena que tenías abierta: completar el diálogo de carga de un archivo se considera suficiente confirmación de que no lo has hecho por accidente. Asegúrate de salvar tus archivos.

3.3.2. Guardando archivos

Guardar archivos es como cargarlos: Cuando presionas **F2**, la ventana activa cambia temporalmente a una ventana de selección de archivos, como se muestra en [Figure 3-19](#). Haz clic en la ventana de edición inferior para escribir un nombre de archivo. Si no lo acabas con ".blend," la extensión es añadida automáticamente. Después presiona **ENTER** para guardar el archivo. Si ya existe un archivo con el mismo nombre, tendrás que confirmar que quieres guardar el archivo en la pregunta de sobre-escribir.

Figure 3-19. Ventana de Selección de Archivos - guardando.



El diálogo de salvar tiene una pequeña cualidad añadida para ayudarte a crear múltiples versiones de tu trabajo: Presionando **NUM+** o **NUM-** incrementará o disminuirá un número contenido en el nombre del archivo. Para simplemente salvar sobre el archivo actualmente cargado evitando el diálogo de salvar, presiona **CTRL-W** en vez de **F2** y sólo tendrás que confirmar la pregunta.

3.3.3. Renderizando

Esta sección te ofrece sólo un repaso rápido de lo que necesitarás para renderizar tu escena. Encontrarás una descripción detallada de todas las opciones en [Chapter 17](#).

Las Opciones de Render están en el contexto *Escena* y en el sub-contexto *Botones de Renderizado* ([Figure 3-20](#)) al que se accede haciendo clic en , o presionando **F10**.

Figure 3-20. Opciones de Renderizado en los Botones de Render.



Por ahora sólo nos interesa el panel *Formato*. El tamaño (número de pixeles horizontales y verticales) y el formato de archivo de la imagen a crear se eligen aquí. Puedes fijar el tamaño usando los botones *SizeX* y *SizeY*. Haciendo clic en la caja de selección de debajo (en [Figure 3-20](#), está seleccionado "Targa") abre un menú con todos los formatos de salida disponibles para imágenes y animaciones. Para imágenes fijas podemos elegir Jpeg, por ejemplo.

Ahora que las opciones están completas, la escena puede ser renderizada presionando el botón *RENDER* en el panel *Render* o presionando **F12**. Dependiendo de la complejidad de la escena, esto puede tardar normalmente entre unos pocos segundos y varios minutos, y el progreso es mostrado en una ventana separada. Si la escena contiene una animación, sólo se renderiza el fotograma actual (Para renderizar la animación completa, ver [Section 17.5](#)).

Si no ves nada en la vista renderizada, asegúrate de que tu escena está construida convenientemente. ¿Tiene luz? Está la cámara situada correctamente, y apunta en la buena dirección? ¿Están visibles todas las capas que quieres renderizar?

Note Una imagen renderizada no se salva automáticamente en el disco. Si estás satisfecho con el renderizado, debes salvarlo presionando **F3** y usando el diálogo de salvar tal como describimos en [Section 3.3.2](#). La imagen es salvada en el formato que hayas seleccionado previamente en los botones de Display.

Tip Extensiones de archivo

Blender *no* añade la extensión del formato automáticamente a los archivos de imagen! Tienes que escribir la extensión explícitamente, si la necesitas.

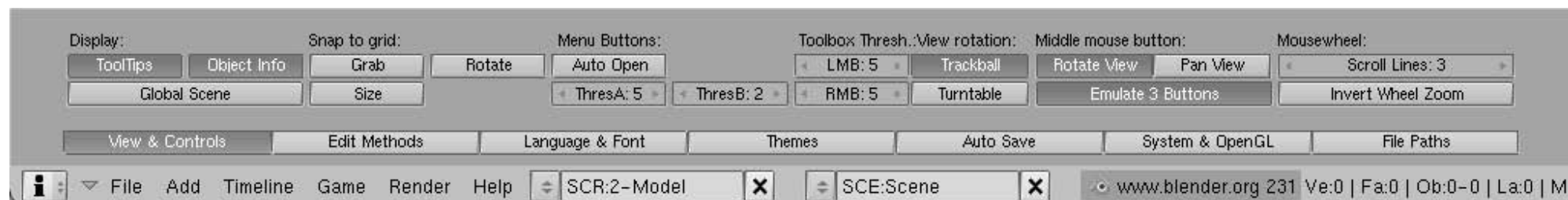
3.3.4. Preferencias de usuario y Temas

Blender tiene unas pocas opciones que no son guardadas con cada archivo, sino que se aplican a todos los archivos de un usuario. Estas preferencias incluyen detalles de la interfaz del usuario, y propiedades del sistema como el ratón, las fuentes y los lenguajes.

Como las preferencias del usuario son raramente necesarias, están escondidas bajo el menú principal. Para hacerlas visibles, baja el borde de la ventana del menú (normalmente el

borde superior en la pantalla). Las opciones están agrupadas en siete categorías que pueden ser seleccionadas con los botones violeta mostrados en [Figure 3-21](#).

Figure 3-21. Ventana de Preferencias del Usuario.



Ya que la mayoría de los botones se explican por sí mismos o muestran una útil ayuda si mantienes el cursor parado sobre ellos, no vamos a describirlos en detalle aquí. En lugar de eso, te ofrecemos un repaso de las categorías de Preferencias:

Vista & Controles

Las opciones concernientes a cómo debe reaccionar la interfaz a las acciones del usuario, como qué método de rotación debe usarse en las vistas 3D. Aquí puedes activar también la emulación del ratón de 3 botones si el tuyo tiene sólo dos. **MMB** puede ser emulado con **ALT-LMB**.

Métodos de Edición

Te permite especificar los detalles del funcionamiento de ciertos comandos de edición como *duplicar*.

Lenguaje & Fuentes

Selecciona una fuente TrueType alternativa para mostrar en la interfaz, y elige entre los lenguajes disponibles para la interfaz.

Temas

Desde la versión 2.30 Blender permite la utilización de Temas para definir colores personalizados para la interfaz. Puedes crear y manejar temas desde aquí.

Guardar Automáticamente

Puedes configurar Guardar Automáticamente para tener una copia de emergencia en caso de que algo vaya mal. Estos archivos se llamarán Nombre.blend1, Nombre.blend2, etc.

Sistema & OpenGL

Debes consultar esta sección si experimentas problemas con los gráficos o con la salida de audio, o si no dispones de un teclado numérico y quieres imitarlo (para portátiles). Además, aquí puedes también configurar el esquema de luces para los modos de dibujado Solido y Sombreado.

Directorios de Archivos

Escoge el directorio por defecto para varios diálogos de carga de archivos.

3.3.5. Configurando la escena por defecto

No te gusta la configuración de ventanas por defecto de Blender, o quieres unas opciones específicas de render para cada nuevo proyecto que empieces, o quieres guardar tu Tema? No hay problema. Puedes usar cualquier archivo de escena como opciones por defecto cuando arranca Blender. Convierte la escena en la que estas trabajando en la opción por

defecto presionando **CTRL-U**. La escena será copiada en un archivo denominado .B.blend en tu directorio principal.

Puedes borrar el proyecto actual y volver a la escena por defecto en cualquier momento presionando **CTRL-X**. Pero recuerda salvar primero tus cambios a la escena previa!

[Prev](#)

Navegando en el Espacio 3D

[Home](#)

[Up](#)

[Next](#)

Su primera animación en 30 + 30 minutos

Chapter 4. Su primera animación en 30 + 30 minutos

Este capítulo le guiará paso a paso a través de la animación de un pequeño personaje, "el hombre galleta". Describiremos completamente todas las acciones, pero asumiremos que se ha leído enteramente el [Chapter 3](#), y que entiende las convenciones que se van a usar en el libro.

En los primeros 30 minutos de este tutorial contruiremos un hombre de galleta *estático*. Entonces, en los siguientes 30 minutos, le proporcionaremos un esqueleto y le animaremos para que camine (walk cycle).

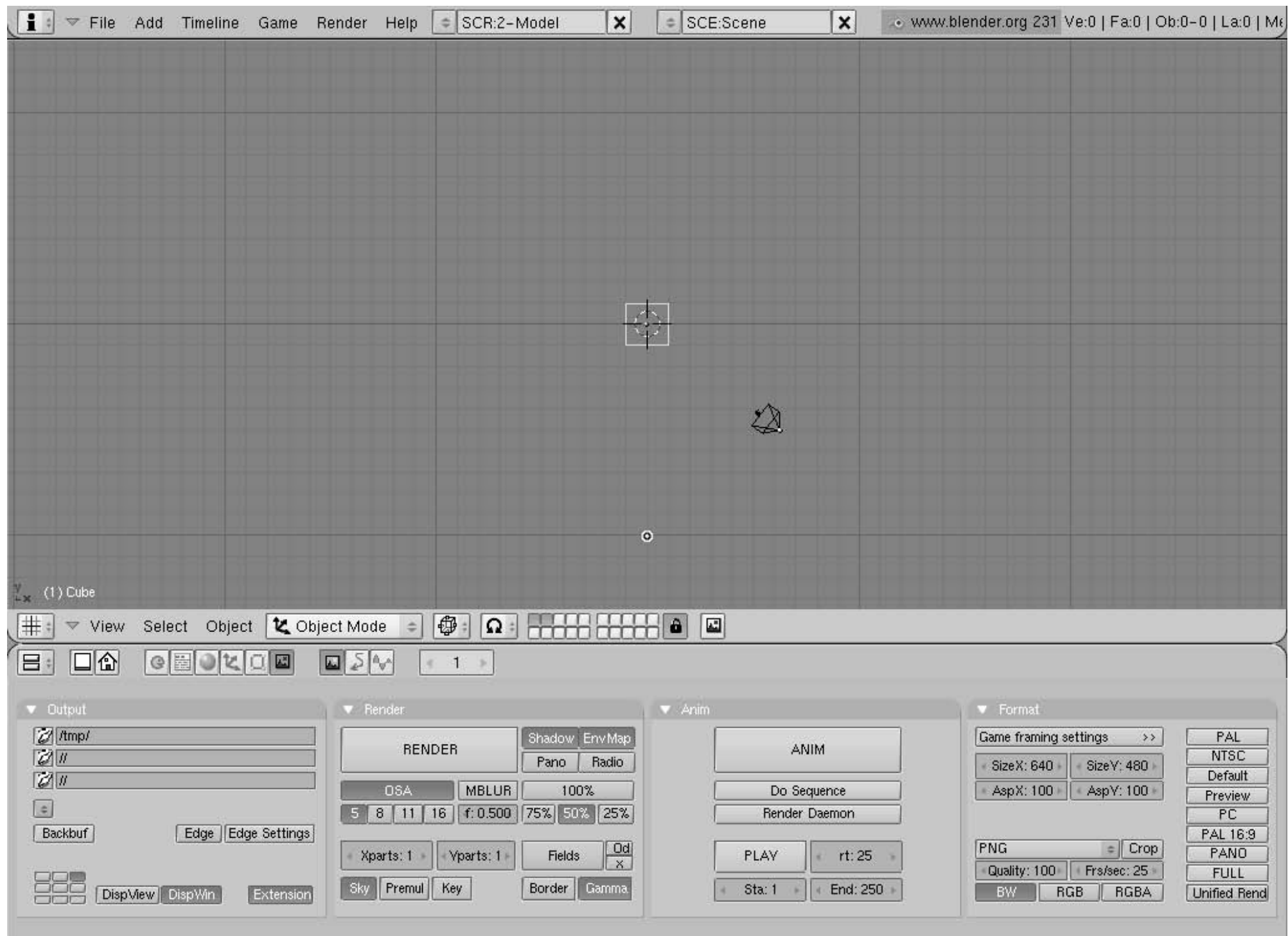
4.1. Calentando

Relativo a Blender v2.31

Comencemos.

1. Arranque Blender haciendo doble-click en su icono o ejecutelo desde la línea de comandos. Blender se abrirá mostrándole, desde la vista superior (top view), la configuración por defecto: una cámara y un plano. El plano está de color rosa, lo que significa que está seleccionado ([Figure 4-1](#)). Elimine el plano con la **Tecla X** y confirme su eliminación pinchando en *Erase Selected* en el diálogo que aparecerá.

Figure 4-1. Blender recién iniciado.



Ahora seleccione la cámara con el **RMB** y presione la **Tecla M**. Una pequeña caja de herramientas, como esta [Figure 4-2](#), aparecerá bajo su puntero, con el primer botón ya

seleccionado. En la fila de arriba seleccione el botón más a la derecha y luego pinche en el botón *OK*. Esto moverá su cámara a la capa (layer) 10.

Blender le proporciona 20 capas para ayudarle a organizar su trabajo. Puede ver que capas son visibles actualmente a través del grupo de veinte botones en la barra de herramientas de la ventana 3D (3D windows toolbar) ([Figure 4-3](#)). Puede elegir que capa quiere que sea visible con el **LMB** y seleccionar que varias capas sean visibles a la vez con **SHIFT-LMB**

Figure 4-2. Caja de herramientas para el control de capas.



Figure 4-3. Controles de visibilidad de las capas.



[Prev](#)

Funciones fundamentales

[Home](#)

[Up](#)

[Next](#)

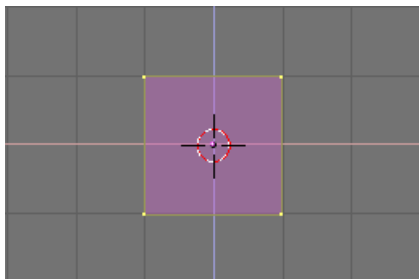
Construyendo el cuerpo

4.2. Construyendo el cuerpo

Relativo a Blender v2.31

Cambie a la vista frontal (front view) con **NUM1** y añada un cubo presionando en la **Barra Espaciadora (SPACE)** y seleccione el menú *Add*, el submenú *Mesh*, y luego *Cube*. (En el libro usaremos **Barra Espaciadora**>>ADD>>Mesh>>Cube como abreviatura para esta clase de acciones). Un cubo aparecerá ([Figure 4-4](#)). Este cubo está en *Modo Edición (EditMode)*, un modo en el cual puedes mover los vértices que componen la malla (mesh). Por defecto, todos los vertices están seleccionados (amarillos). Los vértices sin seleccionar son rosas.

Figure 4-4. Nuestro cubo en Modo Edición, todos los vértices seleccionados.



Llamaremos a nuestro hombre galleta "Gus". Nuestro primer cometido es construir el cuerpo de Gus modificando nuestro cubo en Modo Edición. Para ver las herramientas que utilizaremos para este proposito, pulse el botón que muestra un cuadrado con vértices amarillos en la cabecera de la ventana ([Figure 4-5](#)), o presione **F9**.

Figure 4-5. El botón de la ventana de Edición.

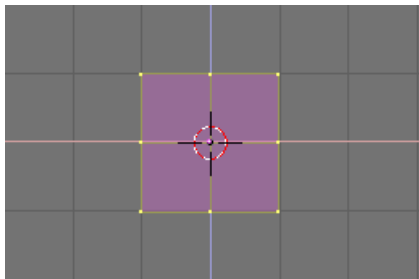


Ahora localice el botón *Subdivisión (Subdivide)* en el panel *Herramientas de Malla (Mesh Tools)* y presionele una vez ([Figure 4-6](#)). Esto partirá cada lado del cubo en dos, creando nuevos vértices y caras ([Figure 4-7](#)).

Figure 4-6. Ventana de Edición para una malla.



Figure 4-7. El cubo, subdividido una vez.



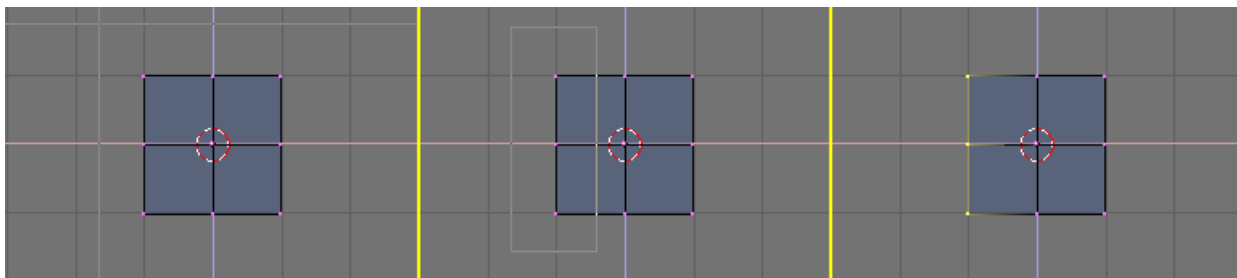
Con el cursor sobre la ventana 3D (3D Window), presione la **Tecla A** para deselectionar todos los elementos. Los vértices se volverán rosas. Ahora presione la **Tecla B**; el cursor se transformará en un par de líneas grises ortogonales. Mueva su ratón sobre la parte superior izquierda del cubo y a continuación presione y mantenga el **LMB**, entonces arrastre el ratón hacia abajo y hacia la derecha para seleccionar todos los vértices que se encuentren a la izquierda. Ahora deje de pulsar el **LMB**. Esta secuencia, la cual le permite seleccionar un grupo de vértices con una caja, está descrita en [Figure 4-8](#).

Tip Seleccionar con una caja (Box Select)

En algunas ocasiones puede tener vértices ocultos detrás de otros vértices, como nos ocurre aquí. Nuestro cubo subdividido tiene 26 vértices, aún cuando solo puede ver nueve porque los otros están ocultos.

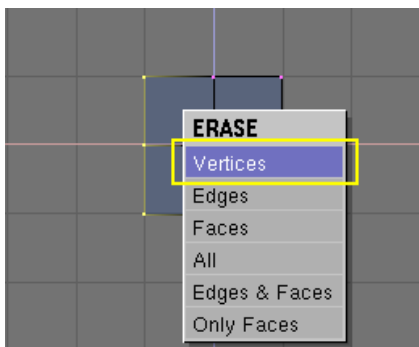
Una pulsación del **RMB** selecciona solo uno de esos vértices superpuestos, mientras que seleccionar con una caja selecciona todos. Por tanto, en este caso, incluso si solo ve tres vértices de color amarillo tiene actualmente seleccionados nueve vértices.

Figure 4-8. La secuencia de seleccionar con una caja un grupo de vértices.



Ahora presiona la **Tecla X** y, en el menú emergente, selecciona *Vertices* para eliminar los vértices seleccionados ([Figure 4-9](#)).

Figure 4-9. El menú emergente para borrar (Tecla X) en acción.



Tip Deshacer (Undo)

Desde la versión 2.3 blender tiene una nueva característica para deshacer pasos en mallas. Presionando la **Tecla U** en el modo Edición hace que Blender deshaga el último paso que hayamos hecho en la malla, y manteniendo presionada la **Tecla U** se retrocederan tantos pasos como nos permita el buffer de almacenamiento, mientras la combinación **MAYÚSCULAS-U** rehace los cambios. **ALT-U** abre un menú con una lista de posibles pasos a deshacer y así encontrar fácilmente el punto al que quiera retroceder.

Deshacer pasos en mallas solo trabaja en modo Edición y unicamente para una malla al mismo tiempo. Los datos que permiten deshacer no se pierden cuando sales del modo Edición hasta que empiece a editar una malla *different*.

Otra manera de volver al último paso es presionando la **Tecla ESC** en mitad de una acción. Esto cancela la acción y le devuelve al estado previo.

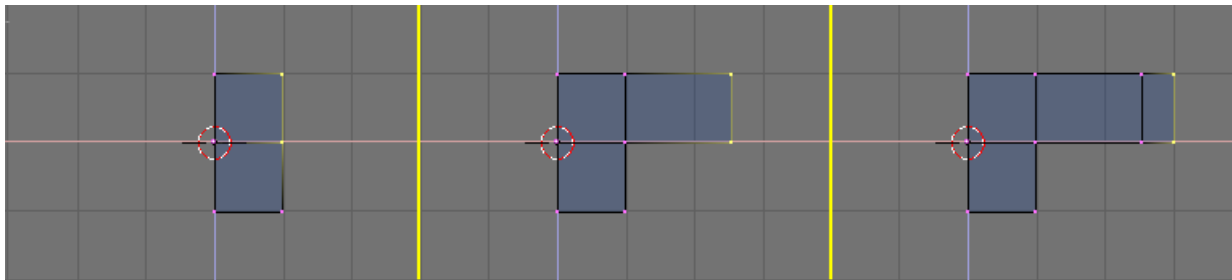
Ahora, usando la secuencia que acaba de aprender, seleccione con una caja los dos vértices de la esquina superior derecha ([Figure 4-10](#), ft). Presione la **Tecla E** y pinche en la entrada de menú *Extrude* para extruirlos. Esto creará nuevos vértices y nuevas caras que podrá mover y que seguirán al puntero del ratón. Muevalas a la derecha.

Para restringir el movimiento horizontal o verticalmente, pulse **MMB** mientras se este moviendo. Puede cambiar a un movimiento sin restricciones pulsando **MMB** de nuevo. Alternativamente puede usar la **Tecla X** para restringir el movimiento al eje x, la **Tecla Y** para restringirlo al eje y, etc.

Vamos a crear brazos y piernas a Gus. Mueva esos vértices nuevos un cuadrado y medio a la derecha, entonces pulse **LMB** para fijar su posición.

Extruya de nuevo con la **Tecla E** entonces mueva los vértices nuevos otro medio cuadrado a la derecha. [Figure 4-10](#) muestra esta secuencia.

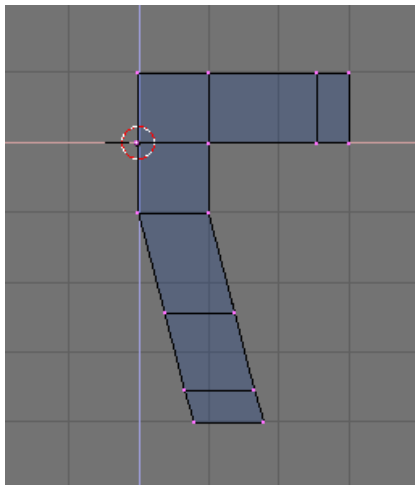
Figure 4-10. Extruyendo los brazos en dos pasos.



Gus debería tener actualmente un brazo izquierdo (ey! Gus nos está mirando). Contruiremos la pierna izquierda de la misma manera, extruyendo los vértices inferiores. Intente hacerlo tal y como se muestra en [Figure 4-11](#).

Note Usaremos la herramienta de extruir 3 veces para crear la pierna. ¡No se preocupe por los codos, pero necesitaremos una rodilla más tarde!

Figure 4-11. Medio cuerpo.



Tip **Vértices coincidentes**

Si está extruyendo y, en el proceso de mover los vértices cambia de opinión y pulsa **ESC** para deshacerlo, los vértices extruidos seguirán estando allí, ¡en su posición original!. Puede moverlos, escalarlos o rotarlos, pero probablemente no quiera extruirlos de nuevo. Para deshacer totalmente la extrusión busque el botón *Remove Doubles*, resaltado en [Figure 4-12](#). Esto eliminará los vértices coincidentes.

Figure 4-12. La ventana de botones de edición.



Note El CD contiene un fichero .blend con este ejemplo, guardado en varias fases del modelado. El primer fichero, `Quickstart00.blend` contiene lo que debería haber conseguido hasta ahora.

Los pasos subsiguientes son numerados progresivamente, `Quickstart01.blend`, `Quickstart02.blend` y así, mientras `Quickstart.blend` contiene el resultado final. Este estándar se aplica a todos los otros ejemplos en este libro.

Ahora crearemos la otra mitad de Gus:

1. Seleccione todos los vértices (**Tecla A**) y elige la entrada *3D Cursor* en el menú *Rotation/Scaling Pivot (eje o punto de rotación/escalado)* de la cabecera de la ventana 3D ([Figure 4-13](#)).
2. Presione **SHIFT-D** para duplicar todos los vértices, aristas y caras seleccionadas. Al duplicar nos encontraremos en modo de movimiento, presione **ESC** para salir de este modo sin mover los vértices.
3. Presione la **Tecla M** para abrir el menú *Mirror Axis (eje de reflexión)*. Elija *Global X (eje X global)*. El resultado es el mostrado en [Figure 4-14](#).

Figure 4-13. Configure el centro de referencia a el cursor.

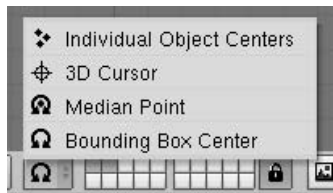
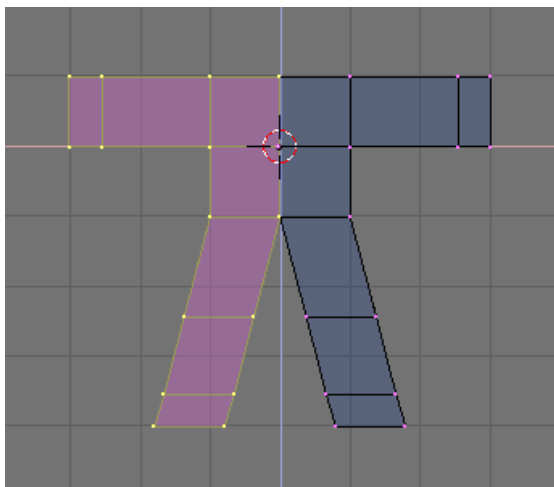


Figure 4-14. Voltee la copia de la mitad del cuerpo para obtener un cuerpo completo.



4. Deseleccione todo y entonces vuelva seleccionarlo presionando la **Tecla A** dos veces, entonces elimine los vértices coincidentes presionando el botón *Remove doubles* ([Figure 4-12](#)). Una ventana aparecerá, notificándole que ocho vértices han sido eliminados.

Tip Centro de referencia

En Blender, escalar, rotar y otras modificaciones de la malla ocurren con respecto a la posición del cursor, al centro del objeto o al baricentro de los elementos seleccionados, dependiendo de que esté seleccionado en el menú *Rotation/Scaling Pivot* [Figure 4-13](#). La cruz selecciona el cursor como referencia.

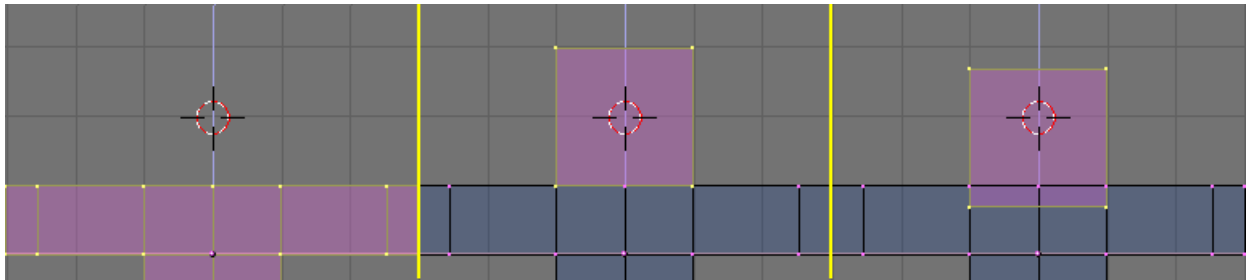
Tip Moviendo el cursor

Para situar el cursor en un punto específico de la rejilla, coloquelo próximo a donde quiere que esté y presione **SHIFT-S** para que aparezca el Menú Snap. La entrada *Curs->Grid* sitúa el cursor exactamente en un punto de la rejilla. La entrada *Curs->Sel* sitúa el cursor en el objeto seleccionado. Las otras entradas mueven los objetos, no el cursor.

Gus Necesita una cabeza:

1. Mueva exactamente el cursor a un cuadrado de la rejilla por encima del cuerpo de Gus ([Figure 4-15](#), left). Añada un nuevo cubo aquí (**SPACE**>>ADD>>Mesh>>Cube).
2. Presione la **Tecla G** para cambiar al Modo Movimiento y mover los vértices creados hacia abajo restringiendo el movimiento con **MMB**, más o menos una tercera parte de una unidad de rejilla ([Figure 4-15](#), right).

Figure 4-15. La secuencia de añadir una cabeza.



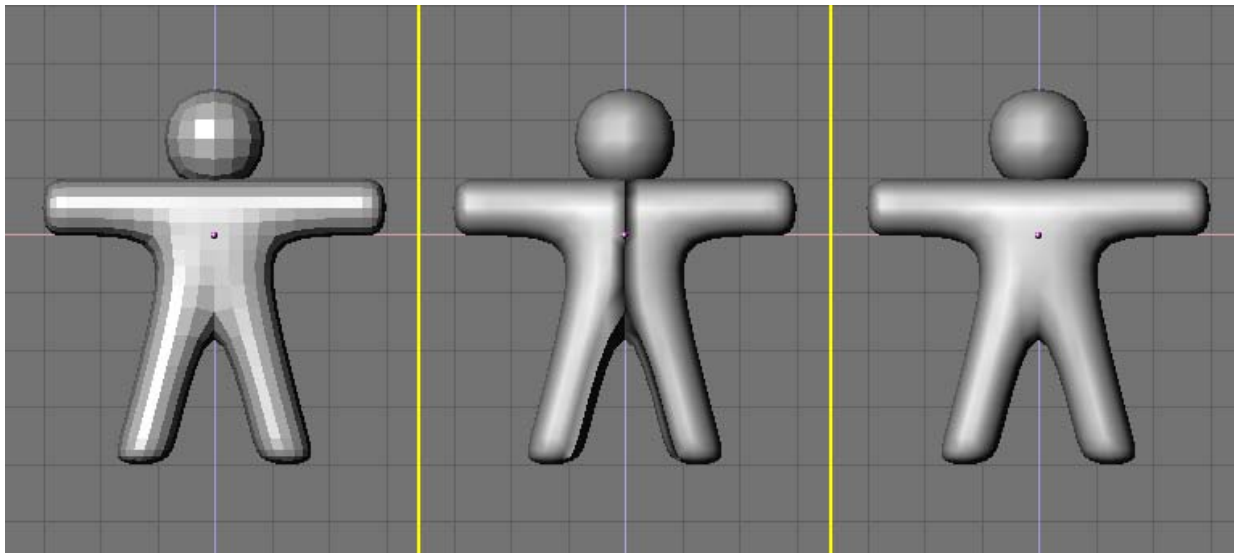
3. Esto produce una figura muy tosca. Para hacerla mas suave, localice el botón *SubSurf* (*Subdivisión de superficies*) [Figure 4-16](#) en el panel de *Mesh* y actívelo. Asegurese de poner a 2, ambos botones inferiores.
- Note** Subdivisión de superficies es una herramienta de modelado avanzado, que dinamicamente crea una malla mucho más densa a partir de los vértices de la malla inicial, haciendo de esta forma una malla mas suave y uniforme que sigue la forma original de la malla que es mucho más tosca. La forma del objeto está aún controlada por la localización de los vértices de la malla original, pero la forma que se renderiza es la malla subdividida, que es mucho más suave.

4. Salga del modo Edición (**TAB**) y cambie la visión del objeto del Modo Alambre (Wireframe mode) al Modo Sólido (Solid Mode) con la **Tecla Z** para echarle un vistazo a Gus [Figure 4-17](#) left.

Figure 4-16. La ventana de botones de Edición.



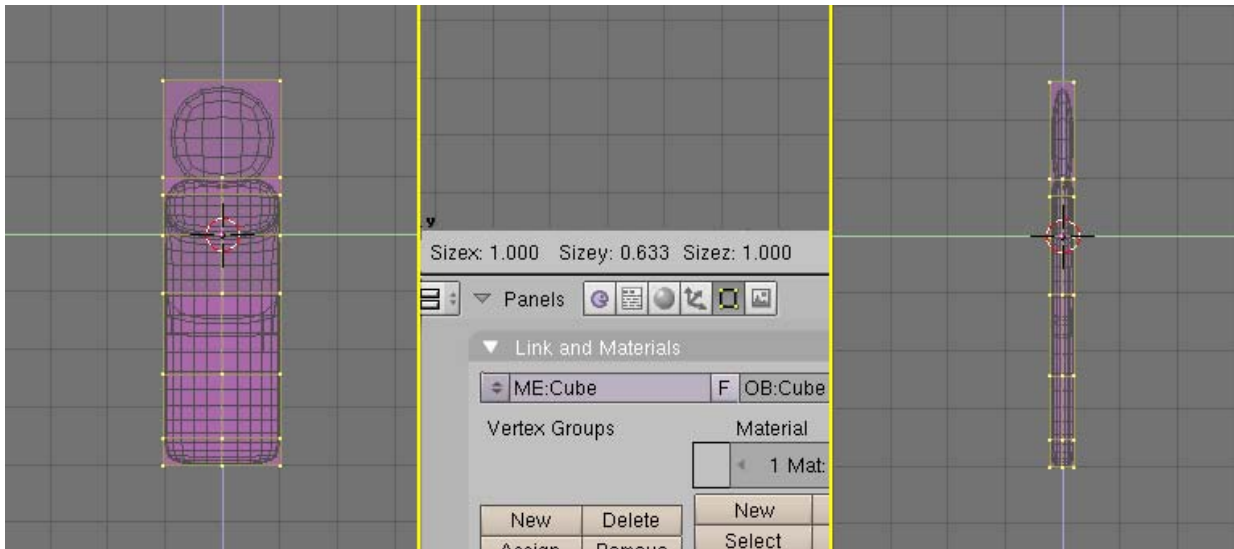
Figure 4-17. Haciendo a Gus más suave.



5. Para hacer que Gus parezca más liso, presione el botón *SetSmooth* [Figure 4-16](#). Ahora Gus aparecerá liso pero con divertidas líneas negras en medio ([Figure 4-17](#), middle). Esas líneas aparecen porque la malla subdividida está calculada usando las direcciones normales de la malla tosca, las cuales pueden no ser muy consistentes, esto es, que algunas normales de las caras pueden apuntar hacia dentro, algunas hacia fuera, si extrusiones o volteados han sido hechos. Para resetear las normales, regrese al modo Edición (**TAB**), seleccione todos los vértices (**AKEY**), y presione **CTRL-N**. Pulse con **LMB** en *Recalc normals outside* (*Recalcular las normales hacia fuera*) en la ventana que aparecerá. Ahora Gus debería aparecer correctamente, tal como se muestra [Figure 4-17](#).

Presione **MMB** y arrastre el ratón alrededor de Gus para verle desde todos los ángulos. Oh, ¡es demasiado grueso!. Para corregir esto, cambia a la vista lateral **NUM3**. Ahora, entra en el Modo Edición (si no te encuentras en él), y regresa al modo Alambre (**Tecla Z**), y selecciona todos los vértices con la **Tecla A** ([Figure 4-18](#)).

Figure 4-18. Adelgazando a Gus utilizando el escalado restringido.



Vamos a hacer a Gus más delgado:

1. Presione la **Tecla S** y comience a mover el ratón horizontalmente. Pulse **MMB** para restringir el escalado solo a un eje o presione la **Tecla Y** para obtener el mismo resultado. Si ahora movemos el ratón veremos como

Gus comienza a hacerse más delgado pero manteniendo la misma altura.

2. Los tres numeros en la barra de herramientas de la ventana 3D muestran el factor de escalado. Una vez que el escalado está restringido, solo uno de esos numeros variará. Presione y mantenga **CTRL**. El factor de escalado ahora variará de 0.1 en 0.1. Reduzca a Gus con un factor de 0.2, introduciendo su dimensión pulsando **LMB**.
3. Regrese a la vista frontal y al Modo Sólido (**Tecla Z**), entonces rote su vista con **MMB**. ¡Gus está mucho mejor ahora!

[Prev](#)

Su primera animación en 30 + 30 minutos

[Home](#)

[Up](#)

[Next](#)

Veamos como se ve Gus

4.3. Veamos como se ve Gus

Relativo a Blender v2.31

Estamos listos para ver nuestro primer renderizado, pero antes de eso, tenemos algún trabajo que hacer.

1. **Shift-LMB** en la esquina superior derecha para accionar el pequeño botón de los botones de visibilidad de las capas en la barra de herramientas de la ventana 3D ([Figure 4-19](#)) para conseguir que tanto la capa 1 (la capa de Gus) como la capa 10 (la capa de la cámara) sean visibles.

Figure 4-19. Haciendo visibles las capas 1 y 10.



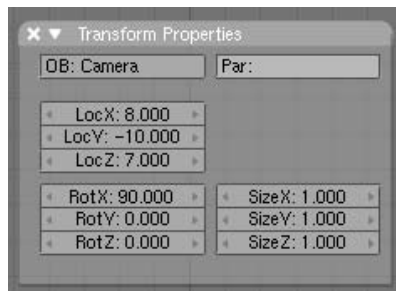
Note Recuerde que la *última* capa seleccionada es la capa activa, por lo que todas los añadidos subsecuentes se harán automaticamente en la capa 10.

2. Seleccione la cámara (**RMB**) y muevalá a la siguiente localización (x=7, y=-10, z=7). Puede hacer esto presionando la **Tecla G** y arrastrando la cámara mientras mantiene presionado **CTRL** para moverla en pasos de 1 unidad en 1 unidad de rejilla.

Tip Introduciendo localizaciones y rotaciones más precisas

Si prefiere introducir numericamente los valores de la localización de un objeto puede hacerlo presionando la **Tecla N** y modificando los "NumButtons" en el Panel que aparece ([Figure 4-20](#)). Recuerde presionar **OK** para confirmar su entrada.

Figure 4-20. El panel para la entrada numérica de la posición/rotación del objeto.



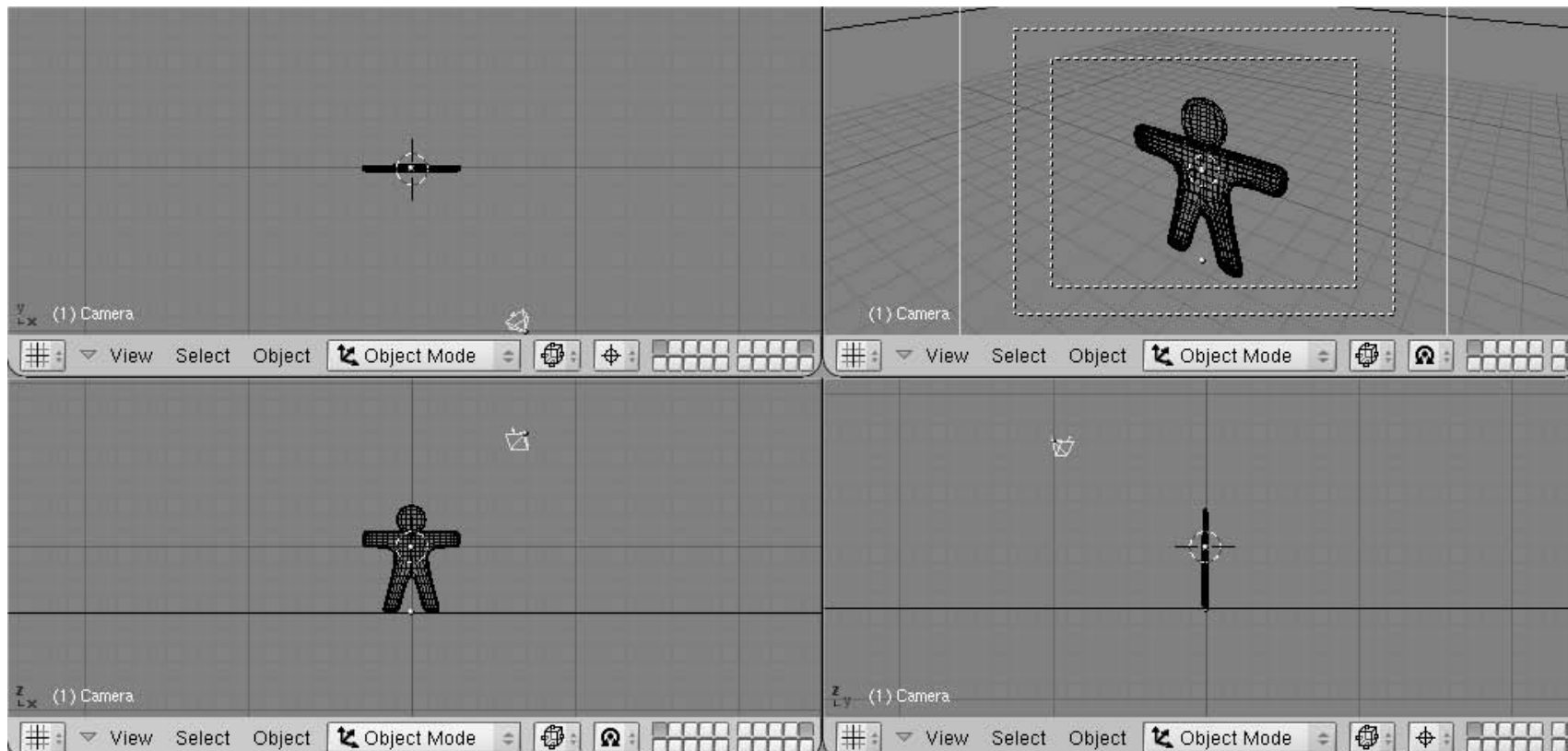
Para conseguir que la cámara apunte a Gus, mantenga su cámara seleccionada y entonces seleccione a Gus con **SHIFT-RMB**. La cámara debería estar de color rosa y Gus de color rosa claro. Ahora presione **CTRL-T** y seleccione la entrada *Old Track* en la ventana emergente. Esto forzará a la cámara a seguir a Gus y siempre le seguirá. Esto significa que puede mover la cámara donde quiera y puede estar seguro que Gus siempre estará en el centro de la vista de la cámara.

Tip Seguimiento (Tracking)

Si el objeto a seguir ha sufrido ya una rotación, como ocurre muy a menudo, el resultado de pulsar **CTRL-T** puede no ser el esperado. Si le ocurre esto, seleccione el objeto que realiza el seguimiento (en nuestro caso la cámara), y presione **ALT-R** para borrar cualquier rotación del objeto. Una vez que haga esto la cámara realmente seguirá a Gus.

[Figure 4-21](#) muestra las vistas superior, frontal, lateral y de cámara de Gus. Para obtener una vista de cámara presione la **Tecla NUM0**.

Figure 4-21. La posición de la cámara con respecto a Gus.



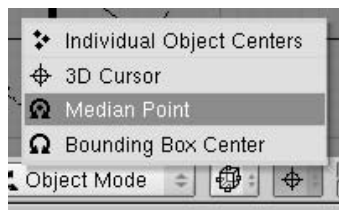
Ahora necesitamos crear el terreno para que Gus se mantenga de pie.

1. En la vista superior (**NUM7**), y *fuera* del modo Edición, añade un plano (**SPACE>>ADD>>Mesh>>Plane**).

Note Es importante salir del modo Edición, porque si no el objeto añadido formará parte del objeto que actualmente se encuentra en modo Edición, como hicimos con la cabeza de Gus cuando la añadimos. Si el cursor está donde se muestra [Figure 4-21](#), el plano se añadirá en mitad del cuerpo de Gus.

2. Cambia a modo Objeto y a la vista frontal (**NUM1**) y mueva (**Tecla G**) el plano hasta los pies de Gus, usando **CTRL** para mantenerlo alineado con Gus.
3. Cambiamos el centro de referencia de cursor (donde lo configuramos al principio) a objeto presionando el botón resaltado [Figure 4-22](#).
4. Ve a la vista de cámara (**NUM0**) y, con el plano aún seleccionado, presiona la **Tecla S** para comenzar a escalarlo.

Figure 4-22. Seleccione el centro de referencia al centro del objeto.

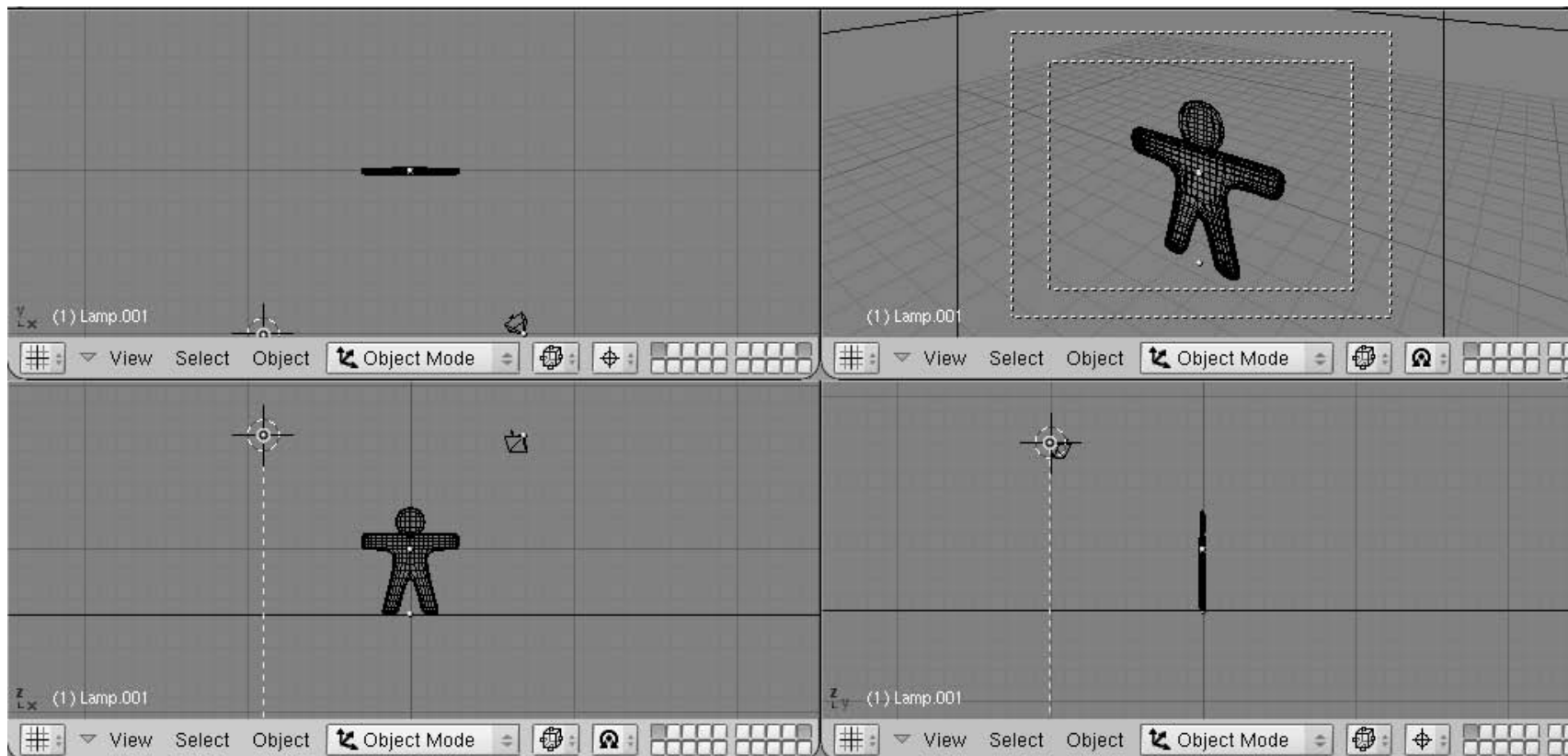


5. Alargue el plano hasta que sus lados se extiendan más allá del area de visión de la cámara , que está indicado por el rectángulo de rayas blancas en la vista de cámara.

Ahora, ¡algo de luz!

1. En la vista superior (NUM7), añada una lámpara de luz (Lamp) (SPACE>>ADD>>Lamp) en frente de Gus, pero en el otro lado de la cámara; por ejemplo en (x=-9, y=-10, z=7) ([Figure 4-23](#)).

Figure 4-23. Insertando un lámpara.



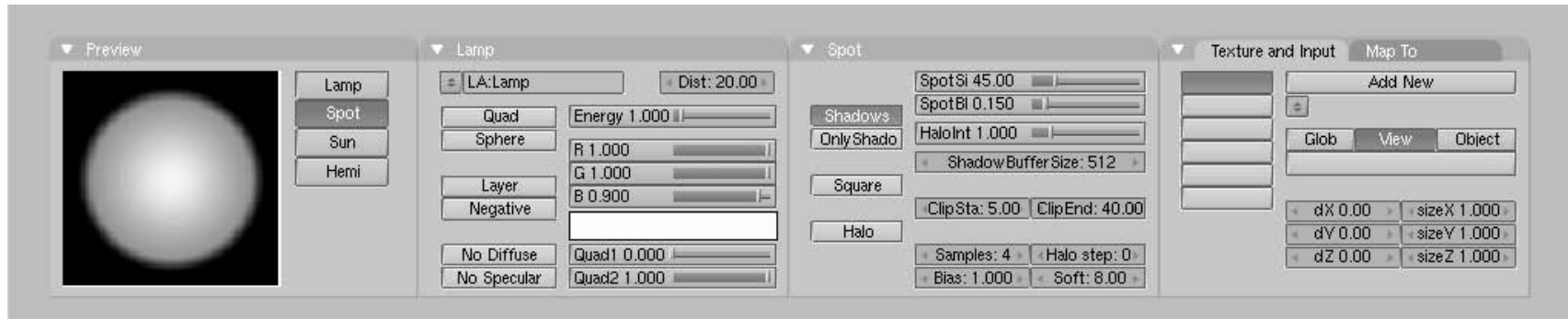
2. Cambie a los botones de lámpara (Lamp Buttons) en el contexto de materiales a través del botón con una lámpara en la barra de herramientas de la ventana de botones ([Figure 4-24](#)) o presione simplemente F5.

Figure 4-24. El botón para el panel de lámparas.



3. En la ventana de botones, en el panel *Preview*, presione el botón *Spot* para convertir la lámpara en una Spotlight (luz tipo foco) ([Figure 4-25](#)) de color amarillo claro (R=1, G=1, B=0.9). Ajuste *ClipSta*: a 5, *Samples*: a 4, y *Soft*: a 8.

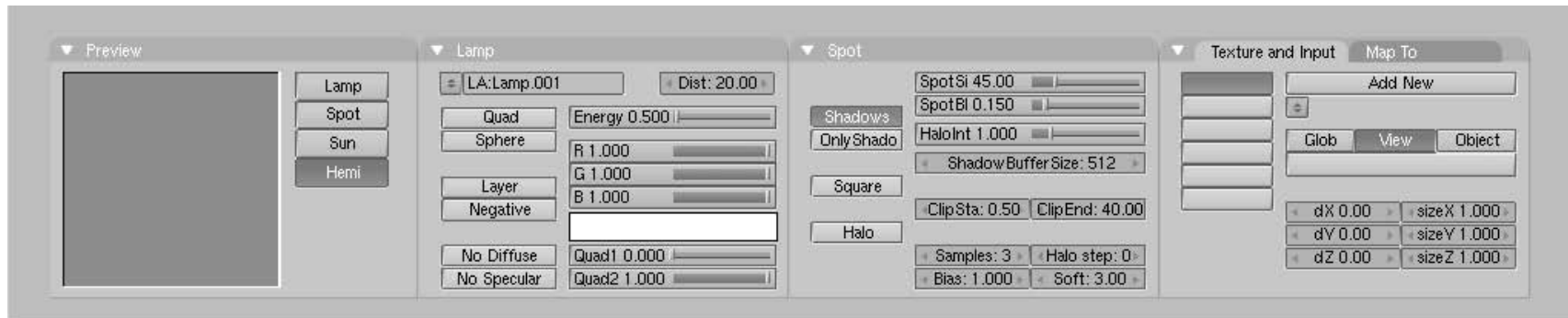
Figure 4-25. Configuración de una luz Spot.



4. Haga que la luz Spot siga a Gus, tal como hizo con la cámara, seleccionando la luz, luego presionando **SHIFT**, luego Gus, y por último **Ctrl-T**. Si ha añadido la luz Spot en la vista superior no debería necesitar borrar su rotación a través de **Alt-R**.

5. Añada una segunda lámpara en la misma localización que la luz Spot, y de nuevo en la vista superior , con (**SPACE**>>**ADD**>>**Lamp**). Haga esta lámpara, una lámpara *Hemi* con una energía de valor 0.6 ([Figure 4-26](#)).

Figure 4-26. La configuración de la lámpara Hemi



Tip ¿Dos lámparas?

Usar dos o más lámparas ayuda a producir una iluminación más suave y realística porque en realidad la luz natural nunca viene de un solo punto. Aprenderá más sobre esto en [Chapter 12](#).

Estamos casi listos para renderizar. Como primer paso, vaya al panel Scene (Escena) presionando el boton que muestra una imagen en la barra de herramientas de la ventana de botones ([Figure 4-27](#)) o presione **F10**.

Figure 4-27. El panel de botones de renderizado.



En los botones de render, en el panel *Format*, ponga el tamaño de la imagen a 640x480. En el panel *Render* pulsa el botón *Shadows* (sombras) que esta arriba en el centro, y el botón *OSA* en el centro a la izquierda

([Figure 4-28](#)). Esos últimos botones habilitarán las sombras y el oversampling (alisado) el cual prevendrá los bordes de sierra.

Figure 4-28. La ventana de botones de renderizado



Ahora presione el botón *RENDER* o pulse **F12**. El resultado, tal como se muestra en [Figure 4-29](#), es actualmente bastante lamentable. Aún necesitamos poner materiales, y montones de detalles, tales como ojos, y cosas por el estilo.

Figure 4-29. Su primer renderizado. ¡Felicidades!



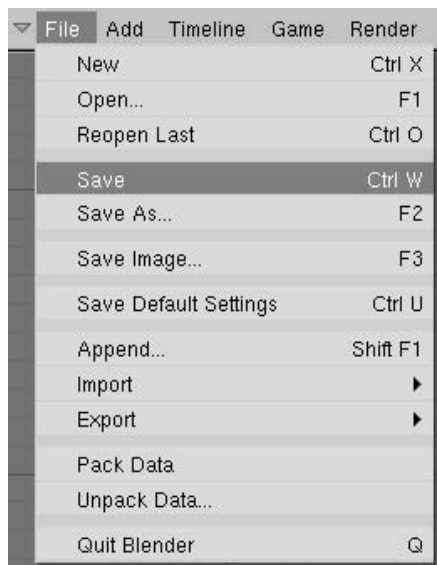


Guardando

Si no lo ha hecho ya, ahora sería un buen momento para guardar su trabajo, a través del menú *File>>Save* como se muestra en [Figure 4-30](#), o bien pulsando **CTRL-W**. Blender le avisará si intenta sobrescribir un fichero existente.

Blender hace copias automáticas en el directorio temporal del sistema. Por defecto, esto ocurre cada 4 minutos y el nombre del fichero es un número. Cargar esos ficheros es otro método de deshacer fallos.

Figure 4-30. El menú para guardar.



[Prev](#)

Construyendo el cuerpo

[Home](#)

[Up](#)

[Next](#)

Materiales y Texturas

4.4. Materiales y Texturas

Relativo a Blender v2.31

Es hora de dar a Gus algún buen material que simule una galleta:

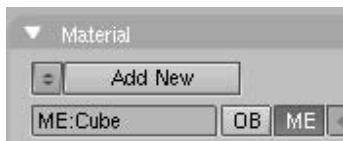
1. Seleccione a Gus. Entonces, en la cabecera de la ventana de botones, seleccione la ventana de materiales presionando el botón de la bola roja ([Figure 4-31](#)) o usando la tecla **F5**.

Figure 4-31. El botón de la ventana de botones de materiales.



2. La ventana de botones estará casi vacía porque Gus no tiene aún materiales asignados. Para añadir un material, pincha en el Panel *Material* (donde están los dos triángulos, uno hacia arriba y otro hacia abajo) y selecciona *ADD NEW* (Añadir nuevo) ([Figure 4-32](#)).

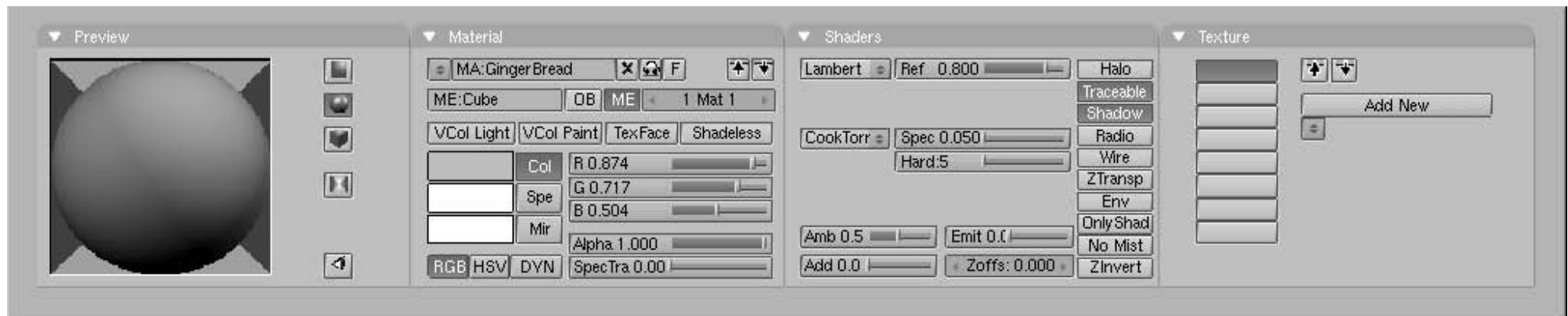
Figure 4-32. El botón en el menú de los materiales.



3. La ventana de botones se llenará con Paneles y botones y un hueco con el nombre del material, que por defecto será "Material", que aparecerá próximo al botón cuadrado de color blanco. Cambie esto a algo más significativo, como GingerBread (galleta de gengibre).

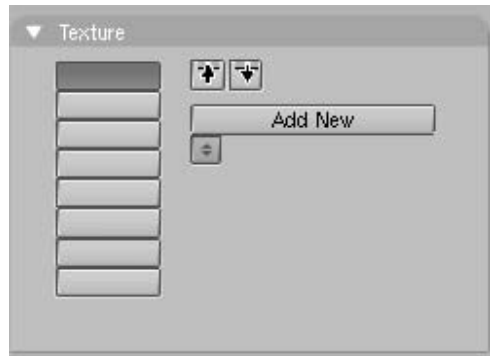
4. Modifique los valores como aparece en [Figure 4-33](#) para obtener un primer material áspero.

Figure 4-33. La ventana de botones del material y un primer material de gengibre.



5. Presione el botón de menú en el area del panel de texturas ([Figure 4-34](#)) y seleccione *Add new*. Con esto añadiremos una textura en el primer canal. Llamela "GingerTex."

Figure 4-34. Panel texturas en la ventana de materiales



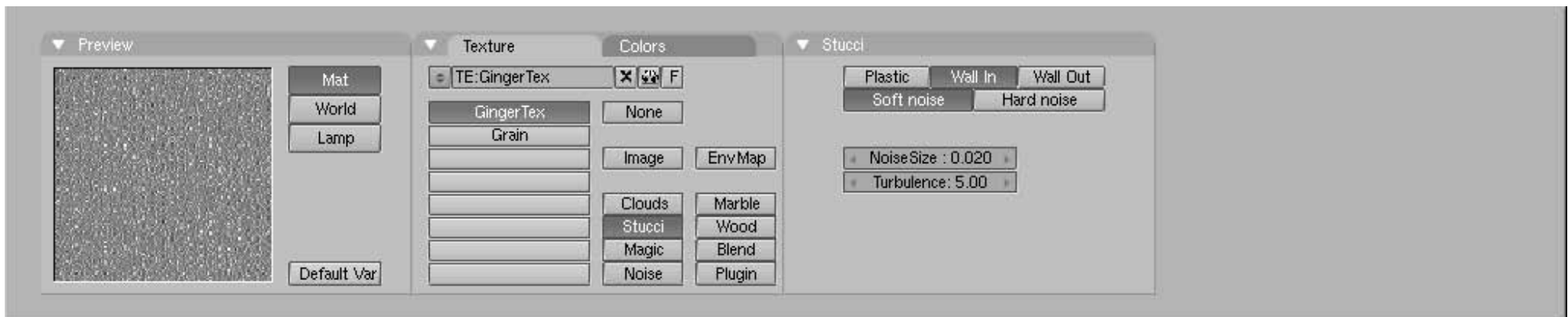
6. Seleccione la ventana de texturas pinchando en el botón [Figure 4-35](#) o presionando **F6**.

Figure 4-35. El botón de la ventana de texturas.



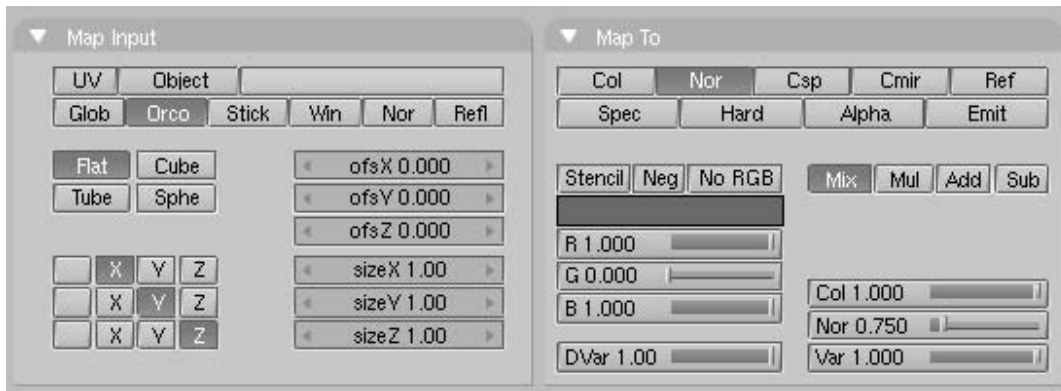
7. De las columnas de botones que aparecen en el panel *Texture (Textura)* seleccione *Stucci* y configure todos los parámetros como en [Figure 4-36](#).

Figure 4-36. La ventana de los botones de textura con una textura stucci.



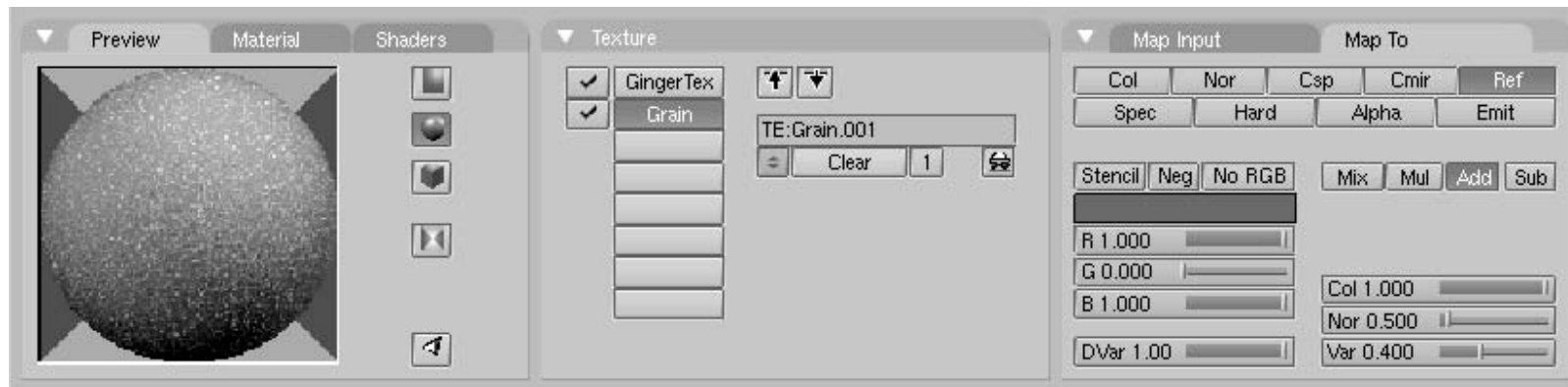
8. Vuelva a la ventana de materiales (**F5**) y configure las pestañas *Map Input* y *Map To* del panel de *Texturas* como en [Figure 4-37](#). Deseleccione el botón *Col* y seleccione el botón *Nor*, entonces mueva el deslizador *Nor* a 0.75. Esos cambios harán que nuestra textura stucci actue como un "bumpmap" (mapa de rugosidad) y haga que Gus parezca más de galleta.

Figure 4-37. Configuración para la textura Stucci en la ventana de botones del material.



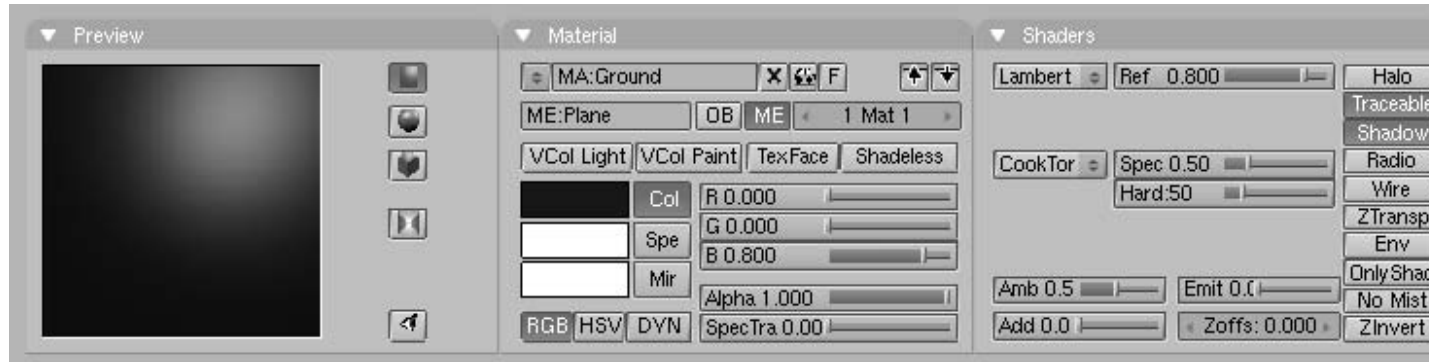
9. Ahora añada una segunda textura, llámela "Grain", y seleccione solo la propiedad *Ref* y un valor de 0.4 en el deslizador *Var* ([Figure 4-38](#)). La textura es una textura de *Noise* (Ruido).

Figure 4-38. Configuración para una textura adicional de ruido en el canal 2.



Dele al suelo un material apropiado, como el azul oscuro que se muestra en [Figure 4-39](#).

Figure 4-39. Un material muy simple para el suelo.



Para darle algunos toques finales, le añadiremos ojos y otros detalles.

1. Primero haremos que la única capa visible sea la 1 pinchando con **LMB** en el botón de la capa 1 ([Figure 4-40](#)). Esto ocultará las luces, la cámara y el suelo.

Figure 4-40. Los botones de visibilidad de las capas en la barra de herramientas.



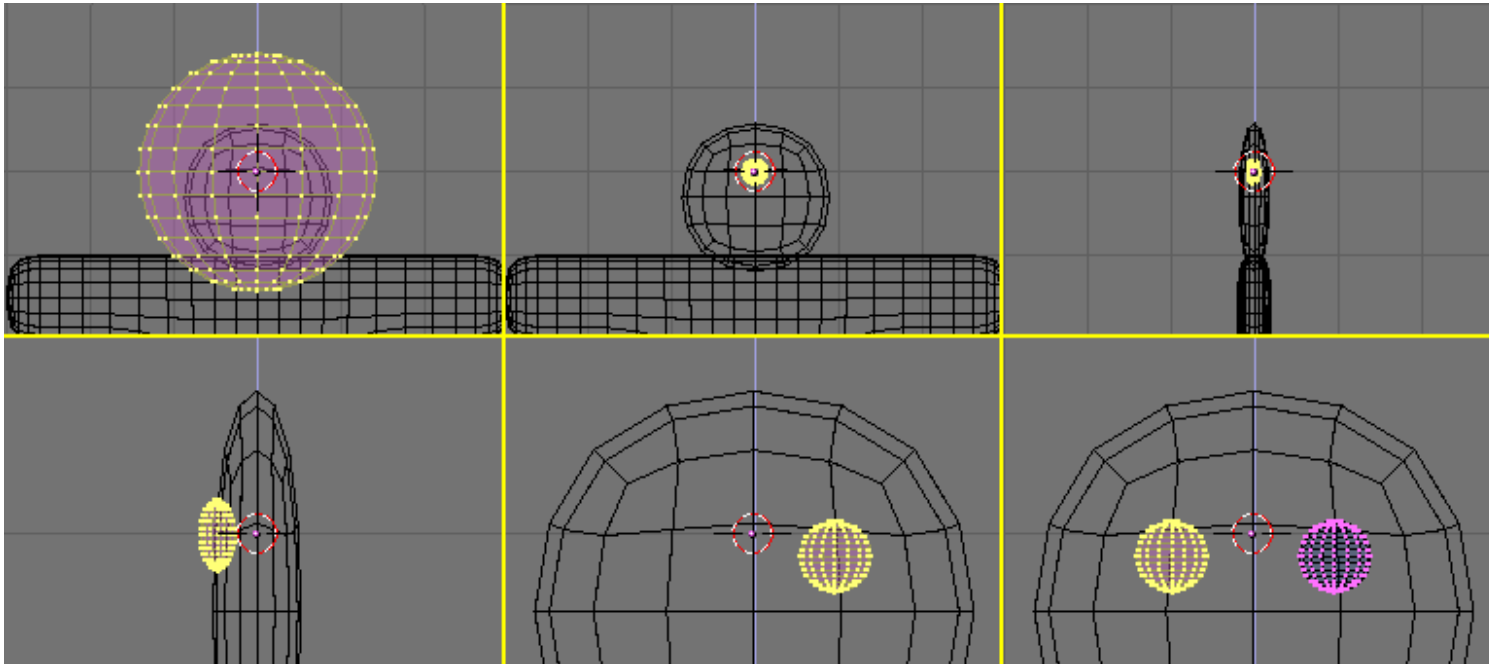
2. Situe el cursor en el centro de la cabeza de Gus. ¡Recuerde que está en 3D así que asegúrese de comprobar al menos dos vistas para comprobarlo!.

3. Añada una esfera (**SPACE**>>**ADD**>>**Mesh**>>**UVsphere**). Se le preguntará por el número de *Segments* (*Segmentos*) (meridianos) y *Rings* (*Anillos*) (paralelos) en los que se dividirá la esfera. El valor por defecto de 32 es más de lo que necesitamos aquí, así que usaremos un valor de 16 para ambos. La esfera está en la primera imagen arriba a la izquierda en la secuencia [Figure 4-41](#).

4. Disminuya la esfera escalandola (**SKEY**) con un factor de 0.1 en todas las dimensiones, entonces cambie a la vista lateral (**NUM3**) y escálela solo en la dirección horizontal (**Tecla**

Y) un valor de 0.5 (vea las siguientes dos imagenes en [Figure 4-41](#)).

Figure 4-41. Secuencia de creación de los ojos.



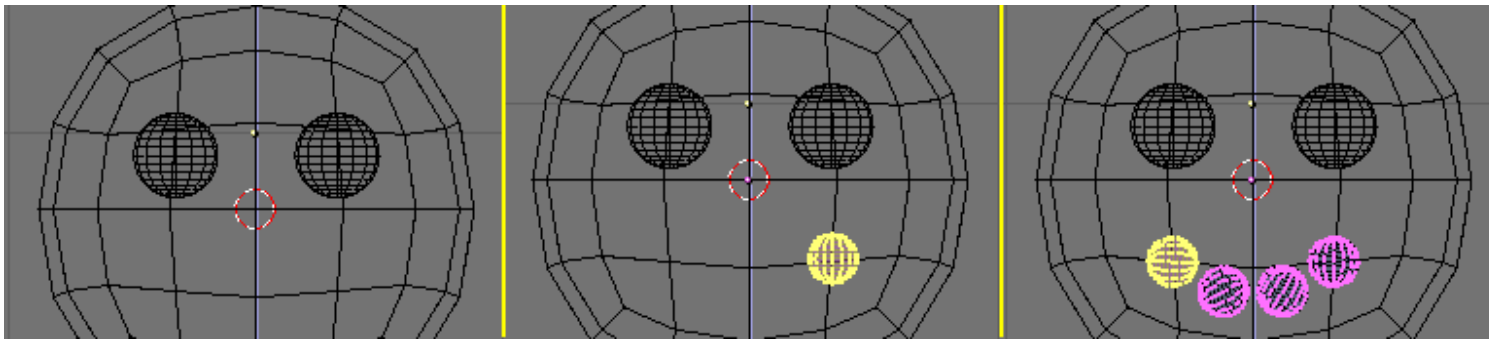
5. Aplique un poco el Zoom si es necesario con **NUM+**, **MW**, o **CTRL-MMB**, y mueva la esfera (**GKEY**) a la izquierda para que esté en la mitad de la cabeza (como se muestra en la primera imagen de la segunda fila de [Figure 4-41](#)).

6. Vuelva a la vista frontal (**NUM1**) y mueva la esfera a la derecha. Situela donde Gus debería tener un ojo.

7. Voltee un duplicado alrededor del cursor siguiendo la secuencia que aprendió cuando volteó el cuerpo de Gus. (Seleccione el cursor 3D, en modo Edición seleccione todo **AKEY**, y haga **SHIFT-D**, **ESC**, **MKEY**, *Global X*). Ahora Gus tiene dos ojos.

8. Salga del modo Edición (**TAB**), y sitúe el cursor tan cerca como pueda del centro de la cara de Gus. Añada una nueva esfera y escálela y muevala exactamente como antes, pero hágala más pequeña y sitúela más abajo y a la derecha del cursor, centrada sobre el vértices de la malla subdividida [Figure 4-42](#)).

Figure 4-42. Creando una boca con herramientas de spinning (revolución).



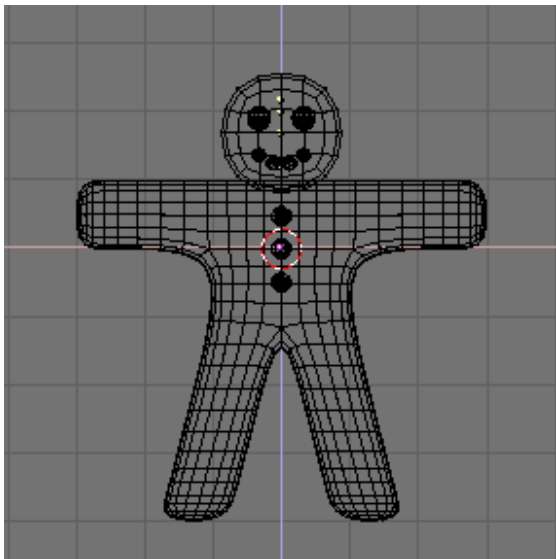
9. Ahora, en los botones de Edición (**F9**), localice el grupo de botones en la parte superior del Panel *Mesh Tools* (*Herramientas de malla*) ([Figure 4-43](#)). Configure *Degr*: a 90, *Steps*: a 3, y verifique que esta pulsado el botón *Clockwise*:. Entonces, con todos los vértices aún seleccionados, presione *SpinDup*. Esto creará tres duplicados de los vértices seleccionados en un arco de 90 grados, centrados alrededor del cursor. El resultado debería ser la boca de Gus, tal como se muestra en la última imagen de secuencia mostrada en [Figure 4-42](#).

Figure 4-43. Los botones de la herramienta Spin en la ventana de botones de Edición.



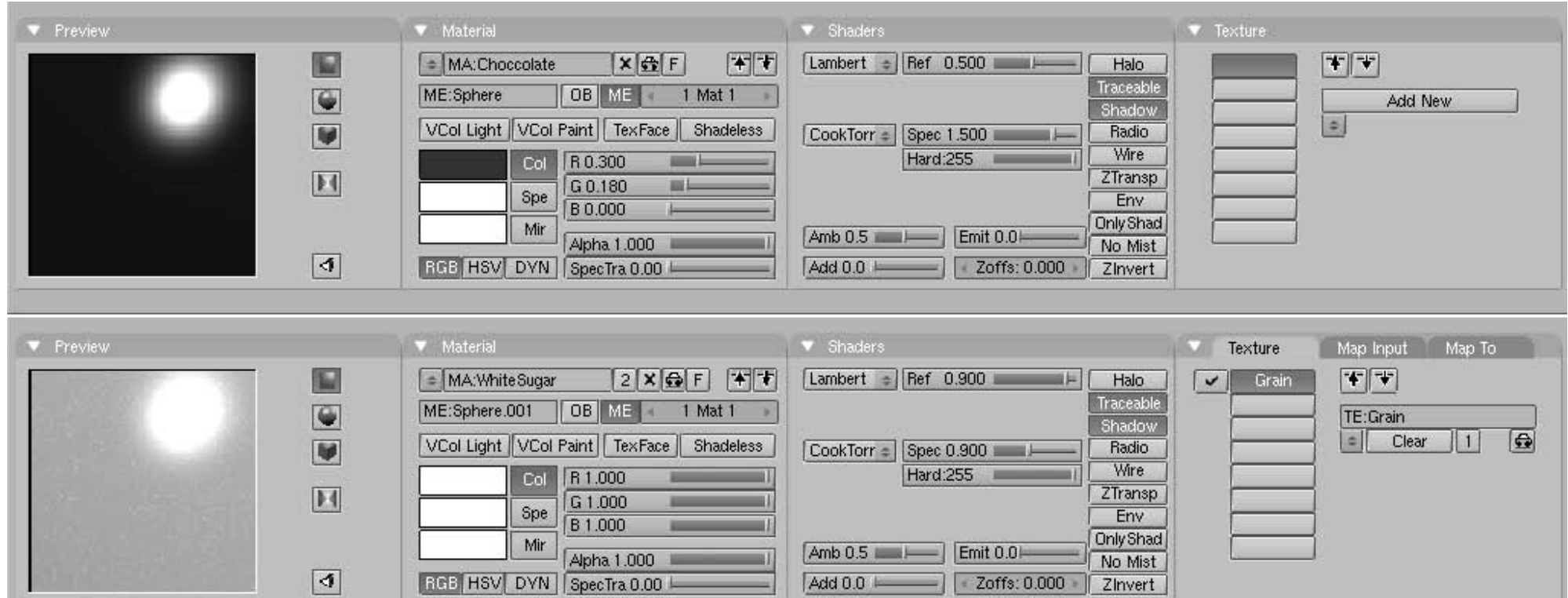
Ahora que ha aprendido el truco, añada tres más de esas elipsoides para crear los botones de Gus. Una vez que haya hecho un botón, puede simplemente salir del modo Edición, presionar **SHIFT-D** para crear un duplicado, y mover el duplicado a su lugar, tal y como se muestra en [Figure 4-44](#).

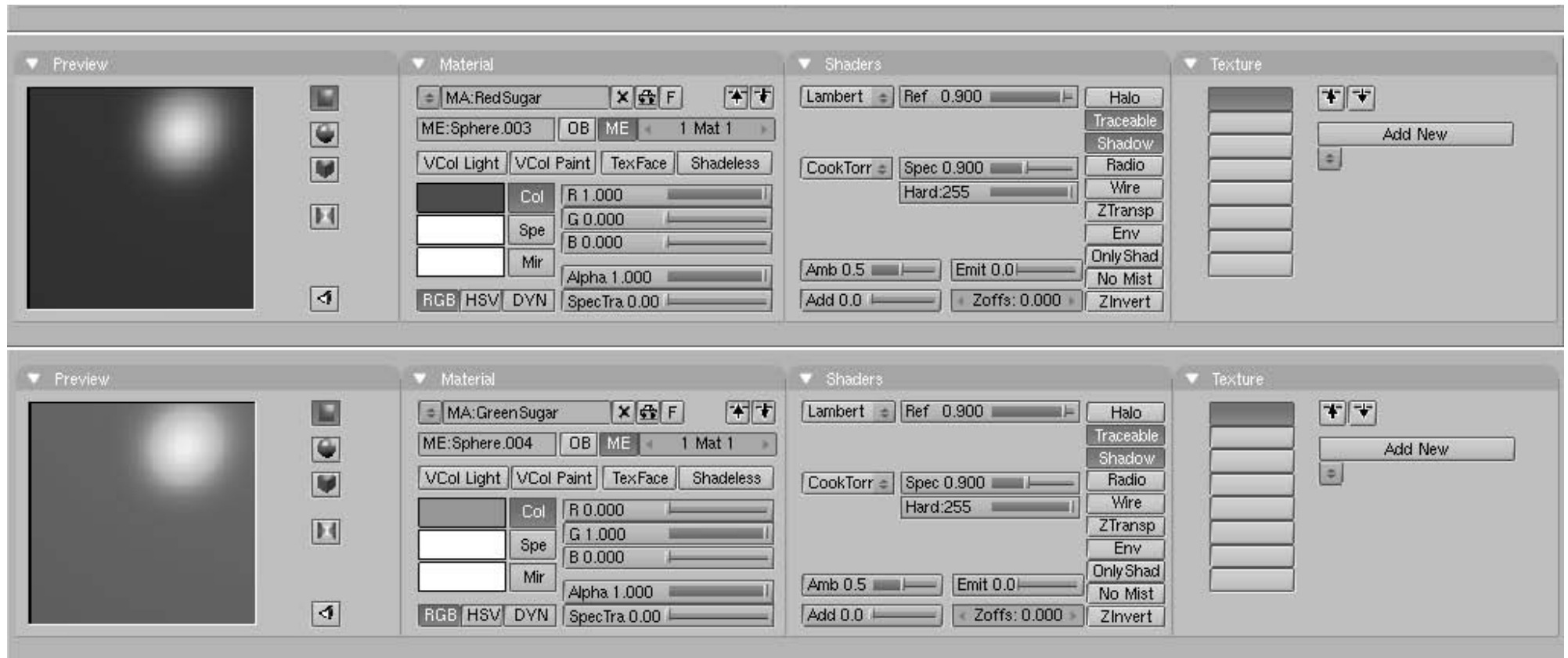
Figure 4-44. ¡Gus completado!



Dé a los ojos un material parecido al chocolate, como el que se muestra en la parte superior [Figure 4-45](#). Dé a la boca un material blanco parecido al azúcar, como el segundo que se muestra en [Figure 4-45](#), y dé a los botones un material rojo, blanco y verde parecido de nuevo al azúcar. Estos materiales son mostrados de arriba a abajo en [Figure 4-45](#).

Figure 4-45. Algunos materiales de caramelo.

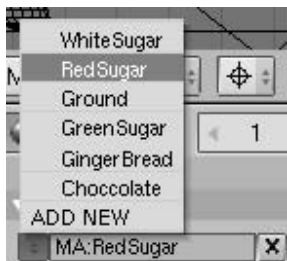




Tip Objetos compartiendo un material

Para dar a un objeto el mismo material que otro objeto, seleccione ese material de la lista que aparecerá cuando presione el botón de menú en el Panel *Material*.

Figure 4-46. Seleccionando un material existente en el menú de los materiales.



Una vez que haya terminado de asignar los materiales, haga la capa 10 visible de nuevo (ya debe saber como), para que esas luces y la cámara también aparezcan, y haga un nuevo renderizado (F12). El resultado debería asemejarse más o menos a esto [Figure 4-47](#).

Figure 4-47. Renderizado estático de Gus una vez terminado.

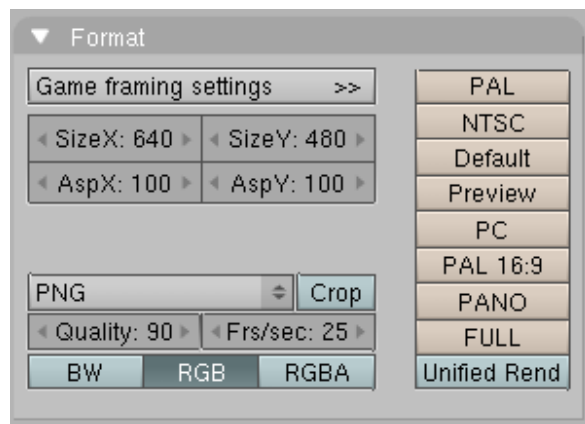


Salve su imagen, si así lo desea, presionando **F3**. Introduzca el nombre de su imagen en la ventana de archivos y guardelo.

Tip Tipos de imagen y extensiones

Debe elegir el formato de la imagen (JPEG, PNG, y demás) configurandolo en los botones de renderizado *antes* de presionar **F3** ([Figure 4-27](#)) y usando para ello el menú ([Figure 4-48](#)) en el Panel *Format*. Blender *no* añade una extension al nombre del fichero; debe introducir uno si lo desea.

Figure 4-48. Menú de selección del tipo de fichero en la ventana de los botones de renderizado.



[Prev](#)

Veamos como se ve Gus

[Home](#)[Up](#)[Next](#)

Construyendo un esqueleto (Rigging)

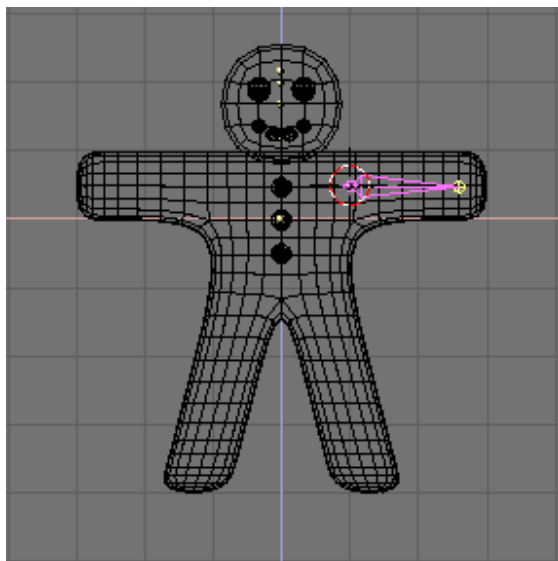
4.5. Construyendo un esqueleto (Rigging)

Relativo a Blender v2.31

Si hubiesemos querido hacer una imagen estática, nuestro trabajo hasta este punto sería suficiente, pero ¿queremos que Gus se mueva!. El siguiente paso será darle a Gus un esqueleto, o armadura, con la cual se podrá mover. Esto se llama el arte del rigging. Gus tendrá un esqueleto muy simple: cuatro extremidades (dos brazos y dos piernas) y una pocas uniones (sin codos, solo con rodillas), pero sin pies o manos. Para añadir el esqueleto:

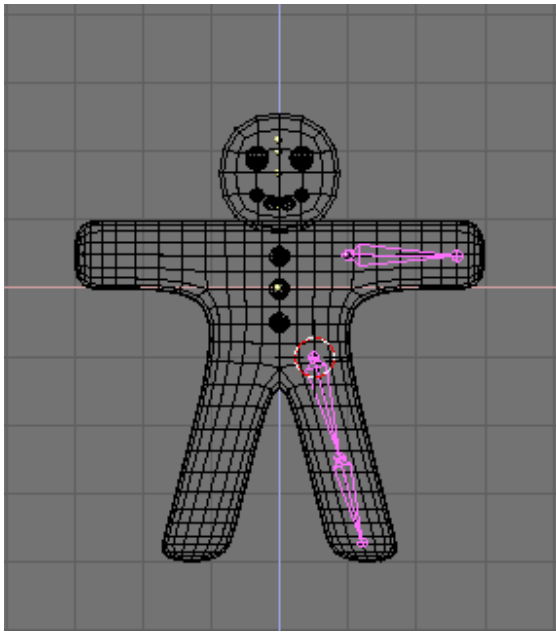
1. Situe el cursor donde debería estar el hombro, presione **SPACE (Espacio)>>Add>>Armature**. Un objeto romboidal aparecerá, un hueso del sistema del esqueleto, que va desde el cursor 3D al puntero del ratón. Situe el fin del hueso en la mano de Gus ([Figure 4-49](#)) con **LMB**. Esto fijará el hueso y creará uno nuevo que partirá del punto final del anterior, produciendo una cadena de huesos. De momento no necesitamos más huesos, así que presione **ESC** para salir.

Figure 4-49. Añadiendo el primer hueso del brazo.



2. Permanecio en modo Edición, entonces mueva el cursor hasta la unión de la pierna con el tronco y añada un nuevo hueso (**SPACE>>ADD>>Armature**) hasta la rodilla. Presione **LMB** y un nuevo hueso debería aparecer automaticamente ahí. Lleve este hueso hasta el pie ([Figure 4-50](#)).

Figure 4-50. Añadiendo el segundo y el tercer hueso, una cadena de huesos en la pierna.

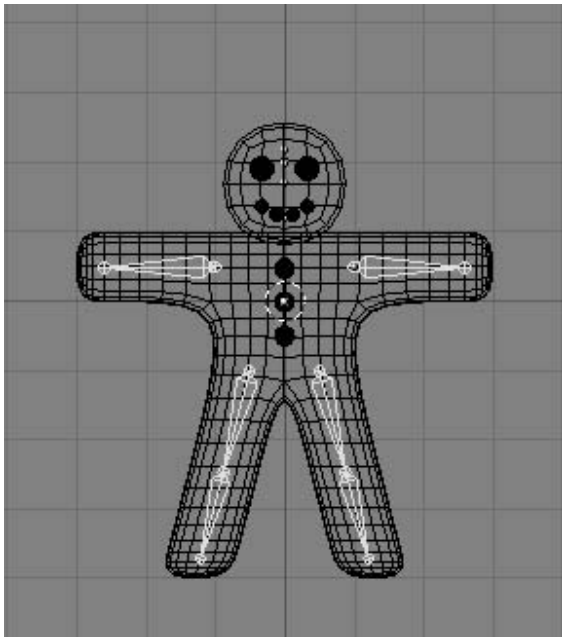


Tip Posición de los huesos

Los huesos que está añadiendo deformarán la malla del cuerpo de Gus. Para producir un resultado correcto, intente situar las uniones de los huesos como se muestra en las ilustraciones.

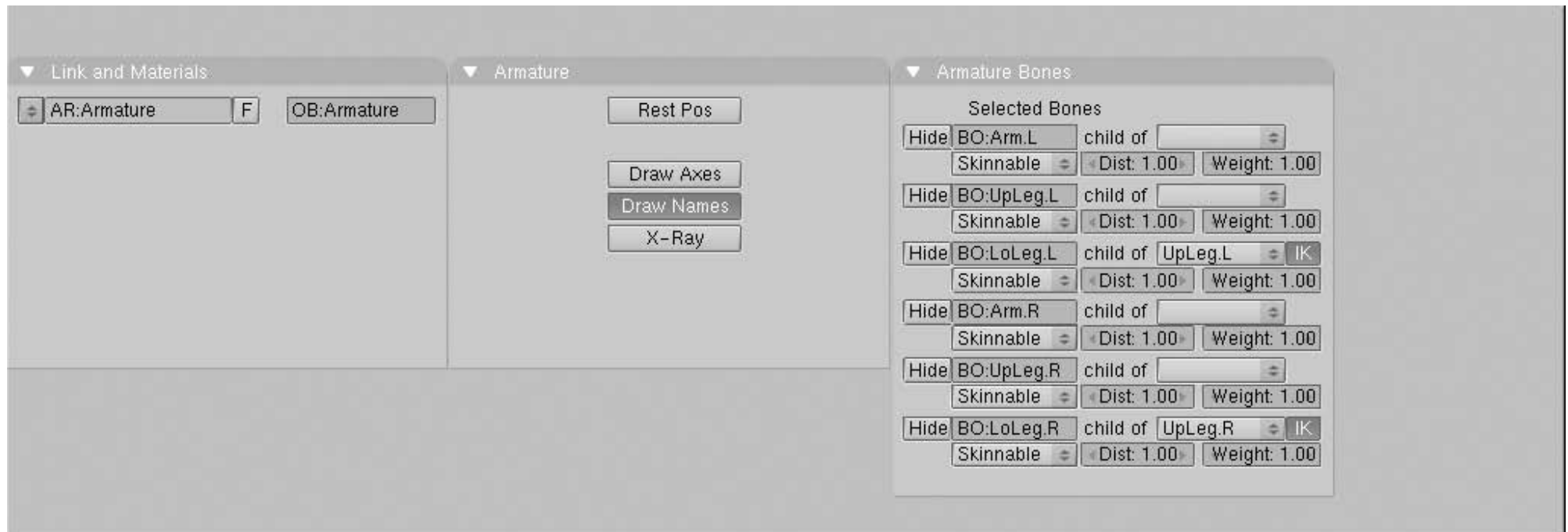
3. Ahora sitúe el cursor en el centro y seleccione todos los huesos con la **Tecla A**. Duplíquelos con **Shift-D** salga de modo de movimiento con la **Tecla ESC** entonces voltee los con la **Tecla M** relativamente al cursor y al eje X Global como hizo con las mallas ([Figure 4-51](#)).

Figure 4-51. El esqueleto completo después de duplicarlo y voltearlo.



Una vez que haya seleccionado todos los huesos (**Tecla A**), la ventana de botones de edición debería mostrar un panel *Armature Bones* (huesos del esqueleto) el cual debería mostrar los botones de la armadura ([Figure 4-52](#)).

Figure 4-52. la ventana de botones de edición para una armadura.



Presione el botón *Draw Names* (*Dibujar nombres*) para ver los nombres de los huesos, entonces **SHIFT-LMB** en los nombres en la ventana de botones de edición ([Figure 4-52](#)) para cambiarlos a algo más apropiado tal como Arm.R, Arm.L, UpLeg.R, LoLeg.R, UpLeg.L, y LoLeg.L. Salga del modo Edición con (**TAB**).



Renombrando los huesos

Es muy Importante que llame a sus huesos con un sufijo '.L' or '.R' para distinguir los que pertenecen a la izquierda (left) y los que pertenecen a la derecha (right), así el editor de acciones (Action editor) será capaz de voltear automáticamente sus poses.

[Prev](#)[Materiales y Texturas](#)[Home](#)[Up](#)[Next](#)[Asignación del esqueleto \(Skinning\)](#)

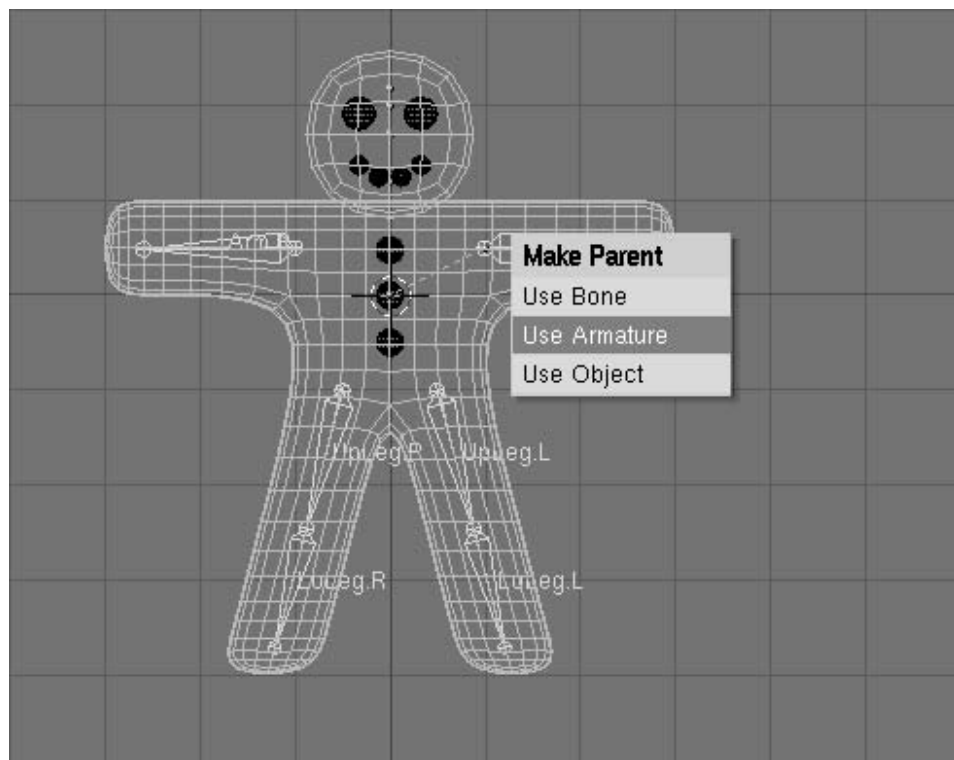
4.6. Asignación del esqueleto (Skinning)

Relativo a Blender v2.31

Ahora debemos conseguir que una deformación en el esqueleto provoque una deformación proporcional en el cuerpo. Haremos esto con la técnica Skinning, la cual asigna los vértices a los huesos de tal forma que los primeros están sujetos al movimiento de los últimos.

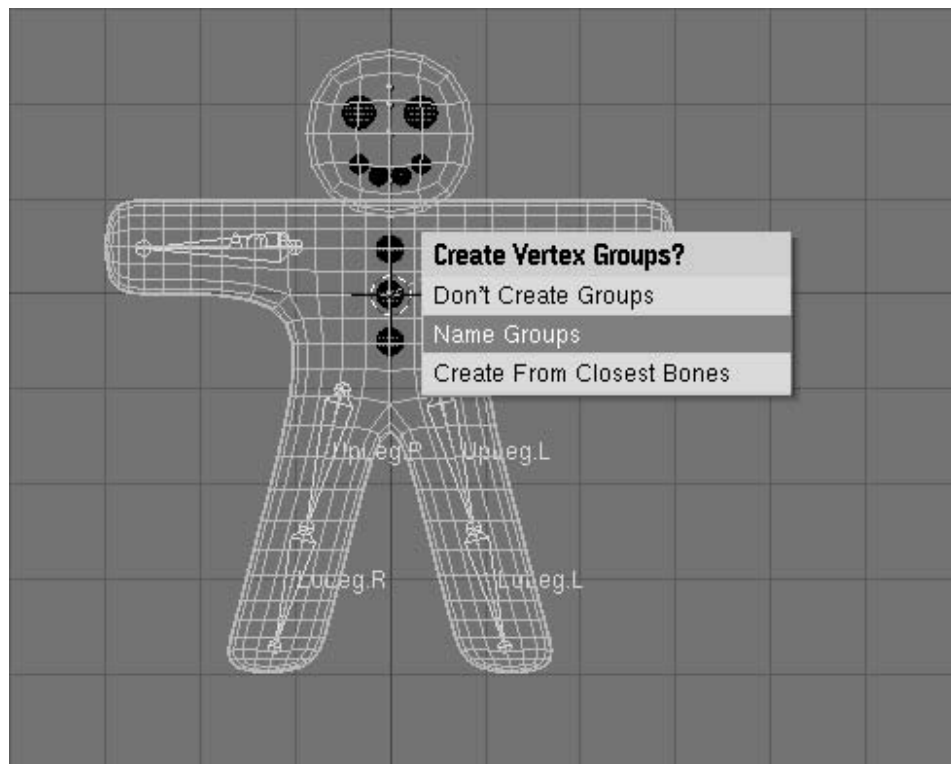
1. Selecciona el cuerpo de Gus, entonces presionando la tecla **SHIFT** selecciona el esqueleto de tal forma que el cuerpo esta morado y el esqueleto está de color rosa claro.
2. Presiona **CTRL-P** para encajar el cuerpo al esqueleto. Un dialogo emergente aparecera ([Figure 4-53](#)). Selecciona la entrada *Use Armature* (*Usar esqueleto*).

Figure 4-53. La ventana emerge cuando encajamos un Objeto a una esqueleto.



3. Un nuevo menú aparecerá, preguntandole si quiere que Blender no haga nada, cree grupos de vértices vacios o grupos de vértices poblados. ([Figure 4-54](#)).

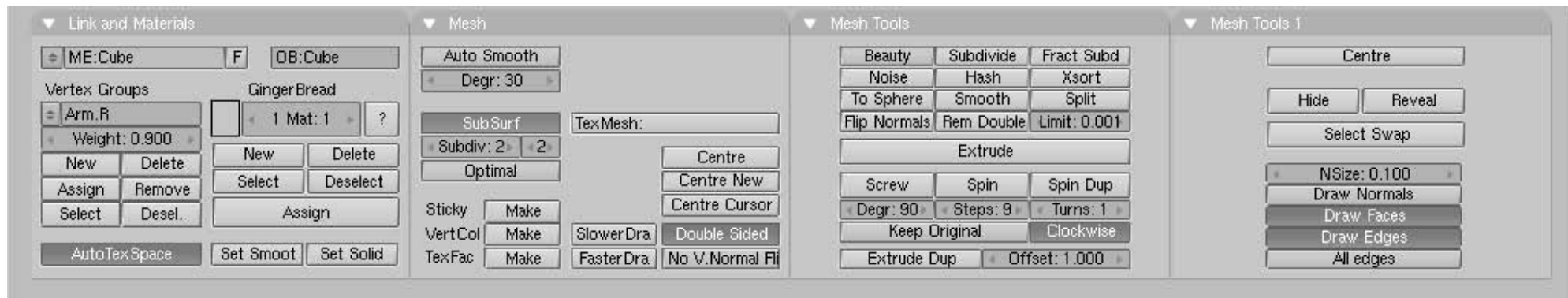
Figure 4-54. Opciones de Skinning automático.



4. Usaremos la opción de skinning automático. Vaya y seleccione *Create From Closest Bones* (Crear a partir de los huesos más cercanos).

Ahora seleccione únicamente el cuerpo de Gus y entre en modo Edición (**TAB**). Notará la presencia en la ventana de botones de edición (Edit buttons) (**F9**) y en el panel *Herramientas de malla 1* (*Mesh Tools 1*), de un menú y botones para grupos de vértices (Vertex Group) ([Figure 4-55](#)).

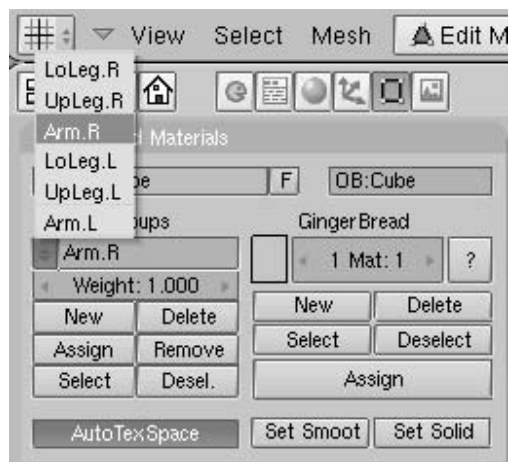
Figure 4-55. Los botones del grupo de vértices en la ventana de botones de edición de una malla.



Presionando en el menú de botones un menú con todos los grupos de vértices disponibles emergerá (seis en nuestro caso, pero con un verdadero personaje, con manos y pies completamente equipados de huesos, ¡puedes tener docenas de ellos!) [Figure 4-56](#). Los botones *Select* (Seleccionar) y *Deselect* (Deseleccionar) le mostrarán que vértices pertenecen

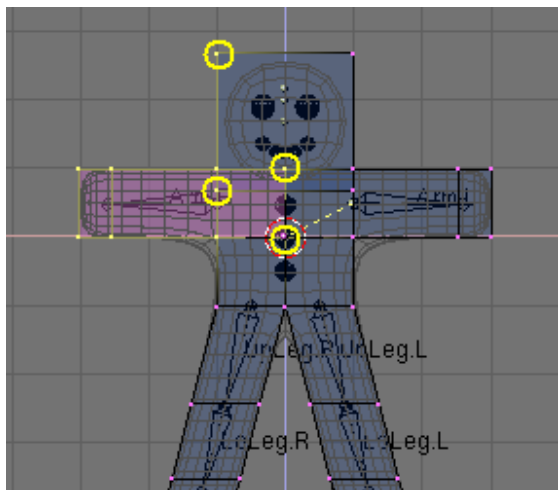
a cada grupo.

Figure 4-56. El menú con los grupos de vértices automáticamente creados en el proceso de skinning.



Seleccione el grupo del brazo derecho (Arm.R) y, con todos los vértices deseleccionados (**Tecla A**, si es necesario) presione *Select*. Debería ver algo como [Figure 4-57](#).

Figure 4-57. Gus en Modo Edición con todos los vértices del grupo Arm.R seleccionados.

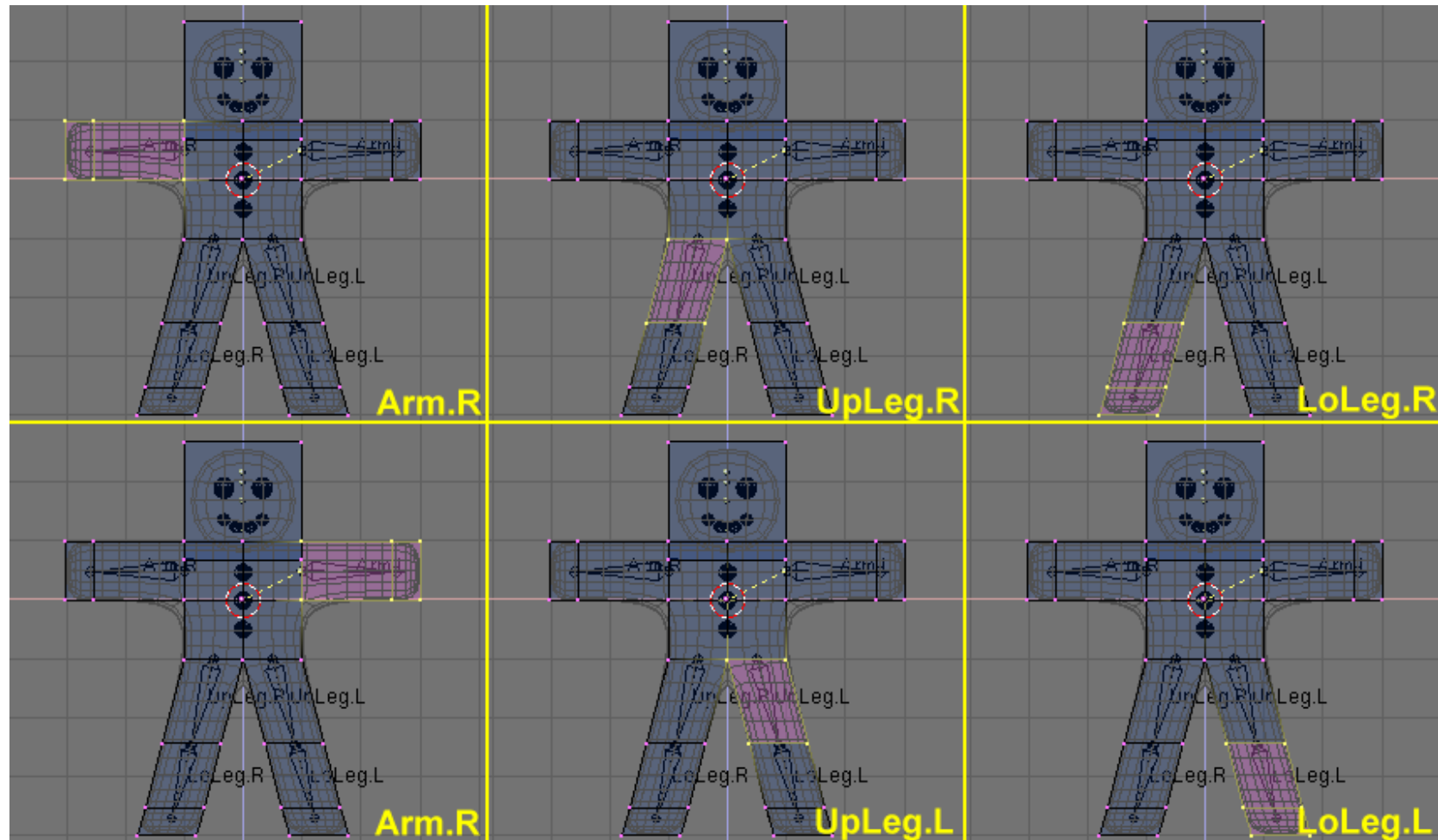


Los vértices marcados con círculos amarillos en [Figure 4-57](#) pertenecen al grupo de deformación, pero no deberían pertenecer. El proceso de autoskinning encontró que estaban muy cerca del hueso y los añadió al grupo de deformación. No les queremos en este grupo porque, dado que algunos son de la cabeza y otros son del pecho, añadirlos al grupo de deformación, deformaría esas partes del cuerpo. Para eliminarlos del grupo, deselecciona todos los otros vértices, aquellos que deberían *permanecer* en el grupo usando la caja de selección (**Tecla B**), pero usa **MMB**, no **LMB**, para definir la caja, así todos los vértices que estén en la caja se deseleccionarán.

Una vez que solo los vértices 'indeseables' estén seleccionados, presione el botón *Remove* ([Figure 4-55](#)) para eliminarlos del grupo Arm.R.

Deseleccione todos (**Tecla A**) entonces compruebe otro grupo. Compruebelos todos y asegurese de que se parecen a estos [Figure 4-58](#).

Figure 4-58. Los seis grupos de vértices.



Tip Grupos de vértices

Sea cuidadoso cuando asigne o elimine vértices de los grupos de vértices. Si observa deformaciones inesperadas, puede haber olvidado algunos vértices, o puede haber situado demasiados vértices en el grupo. Puede modificar sus grupos de vértices en cualquier momento.

Tip Otros detalles

Nuestras deformaciones solo afectaran al cuerpo de Gus, no a sus ojos, boca o botones, los cuales son objetos independientes. Mientras que esto no es un problema a considerar en esta simple animación, debe ser tenido en cuenta en proyectos más complejos, por ejemplo emparentando o uniendo las distintas partes del cuerpo para crear una malla única (describiremos en detalle esas opciones en los próximos capítulos).

[Prev](#)

Construyendo un esqueleto (Rigging)

[Home](#)

[Up](#)

[Next](#)

Posando

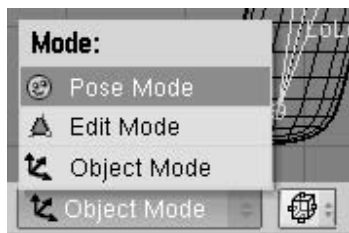
4.7. Posando

Relativo to Blender v2.31

Una vez que haya dado creado el esqueleto y se lo haya asignado a Gus puede comenzar a jugar con él como si fuera un muñeco, moviendo sus huesos y viendo los resultados.

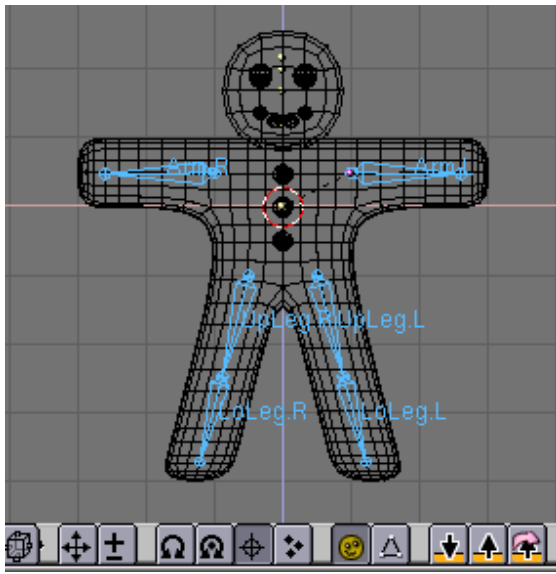
1. Selecciona unicamente la armadura, entonces selecciona *Modo Posado* (*Pose Mode*) en el menú de los modos ("Mode" Menu) ([Figure 4-59](#)). Esta opción solo aparecerá si una armadura está seleccionada.

Figure 4-59. El botón para cambiar al modo posado en la barra de herramientas de la ventana 3D.



2. La armadura se volverá azul. Está ahora en Modo Posado. Si selecciona un hueso se volverá azul claro, no rosa, y si lo mueve(**TECLA G**), o lo rota (**TECLA R**), ¡el cuerpo se deformará!

Figure 4-60. ¡Ahora está en modo posado!



Tip Posición inicial

Blender recuerda la posición original de los huesos. Puede volver a poner el esqueleto (armature) en dicha posición simplemente presionando el botón *RestPos* en el panel de botones de edición de esqueletos (Armature Edit Buttons) ([Figure 4-52](#)).

Tip Cinemática directa (FK) e inversa (IK)

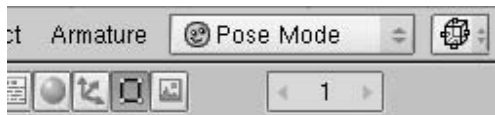
Mientras manejamos los huesos en modo posado notarás que se comportan como cuerpos rígidos, cuerpos inextensibles con juntas esféricas al final. Solo puedes mover el primer hueso de la cadena y todos los otros le seguirán. Todos los huesos subsecuentes en la cadena no podrán moverse, solo pueden ser rotados, así que si es rotado respecto al hueso anterior de la cadena todos los huesos subsecuentes de la cadena seguirán su rotación.

Este procedimiento, llamado *Cinemática directa (Forward Kinematics)* es fácil de seguir, pero precisar la localización hasta del último hueso de la cadena es difícil. Podemos utilizar otro método, *Cinemática Inversa (Inverse Kinematics)* donde solo tenemos que definir la posición del *último* hueso de la cadena, y el resto asume una posición, automáticamente calculada por Blender, para mantener la cadena sin huecos. De este modo posicionar las manos y los pies es mucho más sencillo.

Haremos que Gus ande definiendo cuatro poses diferentes relativas a cuatro diferentes estados de una zancada. Blender hará el trabajo de crear una animación fluida.

1. Primero, verifique que está en el frame (imagen) 1 de la línea temporal. El número de frames aparece en un botón a la derecha de la barra de herramientas de la ventana de botones ([Figure 4-61](#)). Si no está puesto a 1, pongalo ahora.

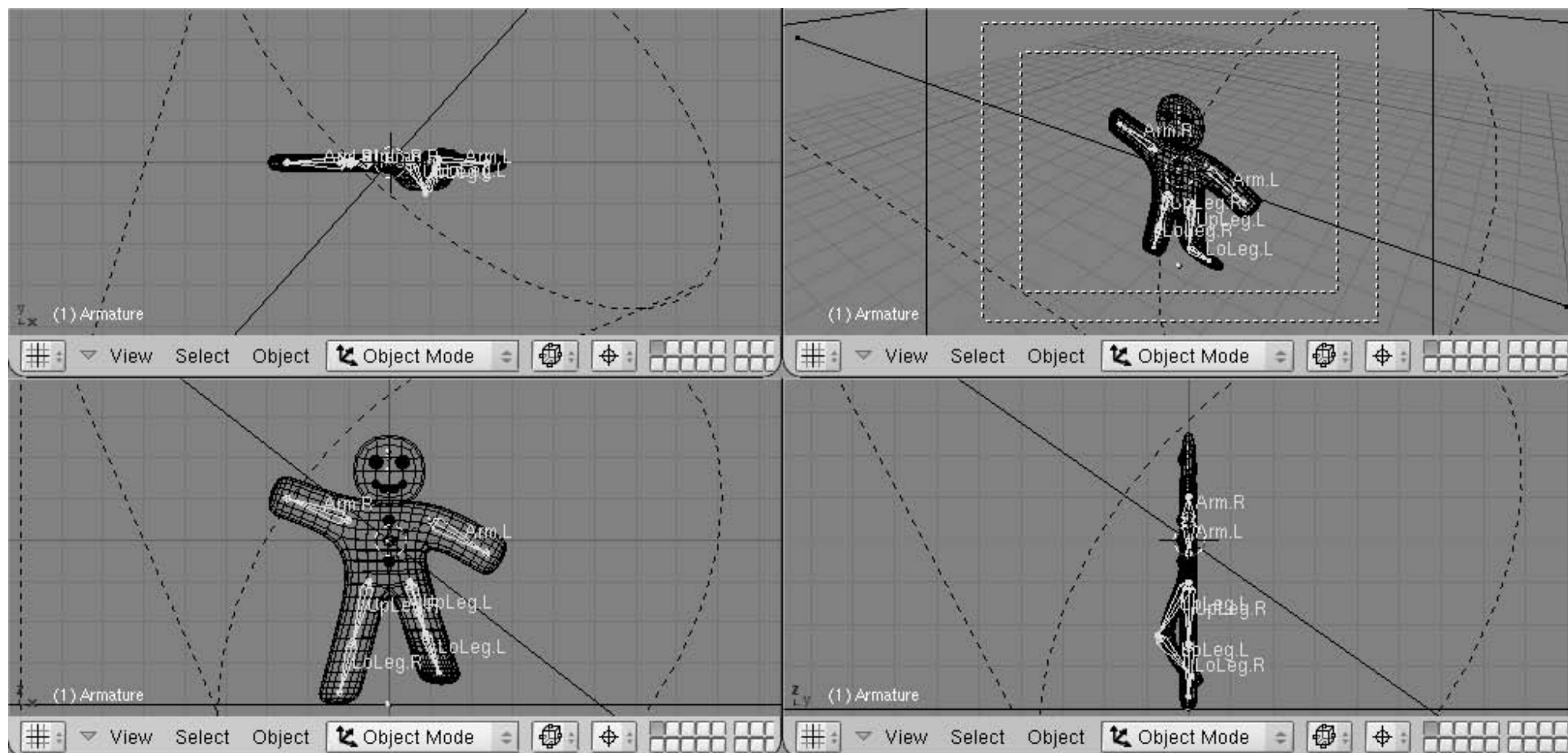
Figure 4-61. El frame actual en la barra de herramientas de la ventana de botones.



2. Ahora, rotando solo un hueso al mismo tiempo (**Tecla R**), levantaremos UpLeg.L y doblaremos LoLeg.L hacia atrás mientras levantamos un poco Arm.R y bajamos otro poco

Arm.L, tal como se muestra en [Figure 4-62](#).

Figure 4-62. Nuestra primera pose.



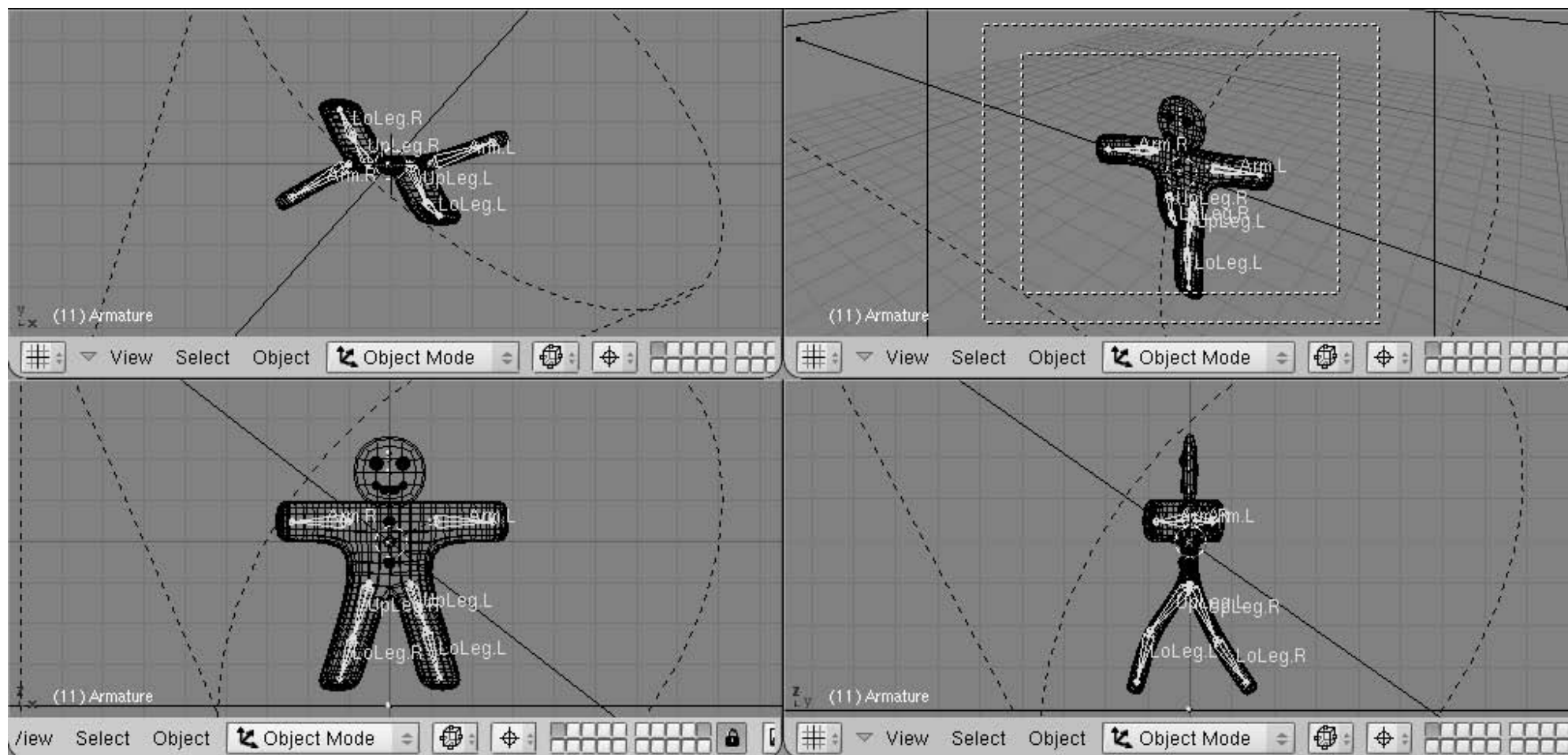
3. Seleccione todos los huesos con la **Tecla A**. Con el puntero del ratón en la ventana 3D, presione la **Tecla I**. Un menú emergerá [Figure 4-63](#). Seleccione *LocRot* en este menú. Esto obtendrá la posición y la orientación de todos los huesos y los guardará en una pose en el frame 1. Esta pose representa a Gus en la mitad de la zancada, mientras esta moviendo su pierna izquierda hacia delante y sobre el suelo.

Figure 4-63. Asignando la pose al frame.



4. Ahora muevase al frame 11 introduciendolo numericamente o presionando la **Tecla Arriba**. Entonces mueva a Gus a una posición diferente , como [Figure 4-64](#), con su pierna izquierda adelantada y la derecha retrasada, ambas ligeramente dobladas. ¡Gus está caminando en el sitio!

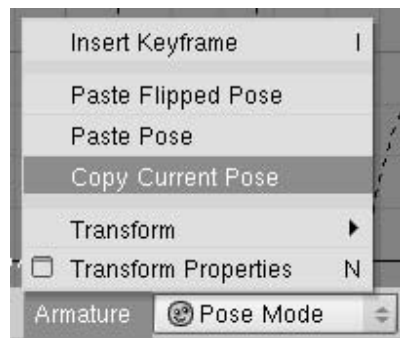
Figure 4-64. Nuestra segunda pose.



5. Seleccione todos los huesos de nuevo y presione la **Tecla I** para guardar la pose en el frame 11.

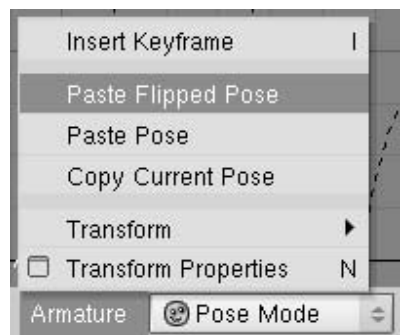
6. Ahora necesitamos una tercera pose en el frame 21, con la pierna derecha levantada, porque estamos en el medio de la otra mitad de la zancada. Esta pose es el reflejo de la pose que hemos definido en el frame 1. Otra forma sería, volver al frame 1 y en el menú *Armature* en la cabecera de la ventana 3D seleccione la entrada *Copy Pose (Copiar Pose)* ([Figure 4-65](#)). Ha copiado la pose actual al buffer.

Figure 4-65. Copiando la pose al buffer.



7. Vaya al frame 21 y pegue la pose con la opción *Paste Flipped Pose (Pegar pose volteada)* en el menú *Armature* ([Figure 4-66](#)). Este botón pegará la pose seleccionada, intercambiando las posiciones de los huesos con el sufijo .L con aquellos huesos que tienen el sufijo .R, efectivamente ¡dándolos la vuelta!

Figure 4-66. Pegar la copia como una nueva, y volteada, pose.



La pose está ahí pero ¡aún no ha sido guardada!. Debe presionar la **Tecla I** con todos los huesos seleccionados.

8. Ahora aplique el mismo procedimiento para copiar la pose del frame 11 al frame 31, también volteandola.

9. Para completar el ciclo, necesitamos copiar la pose del frame 1 *sin* voltear al frame 41. Copiela de la manera usual, y peguelo utilizando la entrada *Paste Pose (Pegar pose)*. Finalice la secuencia guardando la pose con la **Tecla I**.



Comprobando la animación

Para previsualizar la animación, seleccione el frame actual a 1 y presione **Alt-A** en la ventana 3D

[Prev](#)

Asignación del esqueleto (Skinning)

[Home](#)

[Up](#)

[Next](#)

¡Gus anda!

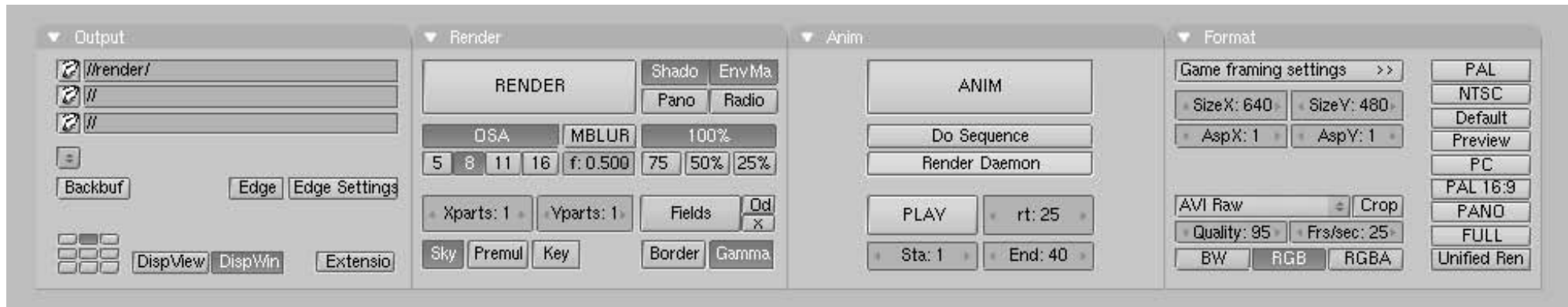
4.8. ¡Gus anda!

Relativo a Blender v2.31

El primer paso es el nucleo del proceso de caminar, y una vez que lo tenga definido hay técnicas para hacer que un personaje ande a través de un camino complejo. Pero, para el proposito de este capítulo de iniciación, este primer paso es suficiente.

1. Cambie a la ventana de Renderizado con (**F10**) y configure el comienzo (start) y el final (end) de la animación a 1 y a 40 respectivamente ([Figure 4-67](#)). Debido a que la imagen (frame) 41 es identica a la imagen 1, solo necesitamos renderizar las imagenes que van de la 1 a la 40 para producir el ciclo completo.

Figure 4-67. Configurando los botones de renderizado para una animación.



2. Seleccione AVI Raw como el tipo de fichero que vamos a utilizar en el panel *Formato (Format)* ([Figure 4-67](#)). Mientras que esto no es una buena elección generalmente, principalmente por problemas de tamaño de archivo (como se explicará más tarde), tiene la ventaja de ser rápido y funcionar en cualquier computadora, asi que cumple nuestras necesidades. (Puede también seleccionar el formato AVI Jpeg para obtener un fichero de un tamaño más reducido, pero utilizando la compresión por pérdidas Jpeg y obteniendo una película que algún reproductor externo puede no ser capaz de reproducir).

3. Finalmente, presione el botón *Animar (ANIM)* en el panel *Animación (Anim)*. Recuerde que *todas* las capas que quiera usar en la animación ¡deben ser seleccionadas!. En nuestro caso, las capas seleccionadas serán la 1 y la 10.

Tip Deteniendo un render

Si ha cometido un error, como olvidar seleccionar la capa 10, puede parar el proceso de renderizado con la **Tecla ESC**.

Nuestra escena es muy simple, y Blender probablemente renderizará las 40 imagenes en unos pocos segundos. Mire como van apareciendo en pantalla.



Imágenes estáticas

Por supuesto puede siempre renderizar cada una de las imágenes de la animación como una imagen estática, simplemente seleccionando la imagen que desee renderizar y pulsando el botón *Renderizar (RENDER)*.

Una vez que el renderizado se haya completado debería obtener un fichero llamado *0001_0040.avi* en un subdirectorio *render* de su actual directorio - el mismo que contiene el fichero *.blend*. Puede reproducir este fichero directamente en Blender presionando el botón *Reproducir (Play)* que está debajo del botón *Animar (ANIM)* ([Figure 4-67](#)). La animación se reproducirá una y otra vez. Para detenerla presione **ESC**.

Solo hemos producido un ciclo muy básico para poder andar . Hay muchas más cosas en Blender, ¡pronto las descubrirá!.

[Prev](#)

Posando

[Home](#)

[Up](#)

[Next](#)

Modelado, Materiales e Iluminación

II. Modelado, Materiales e Iluminación

Como has podido ver en el capítulo de "introducción rápida" en [Part I](#) la creación de una escena en 3D necesita al menos tres cosas: Modelos, Materiales e Iluminación. En este capítulo analizaremos en profundidad cada una de ellas.

Una vez que hayas terminado con esta parte puedes elegir el [Part IV](#) si quieres sacar partido de los motores de renderizado de imagenes que tiene Blender, o también puedes optar por leer el [Part III](#) para aprender todo sobre la animación en Blender. [Part V](#) por otro lado te proporcionará mas información sobre las herramientas de modelado avanzado.

Table of Contents

5. [Modo Objecto](#)

- 5.1. [Selección de objetos](#)
- 5.2. [Movimiento \(traslación\) de objetos](#)
- 5.3. [Rotación de objetos](#)
- 5.4. [Cambio de tamaño/imagen especular de objetos](#)
- 5.5. [Panel de propiedades de transformación](#)
- 5.6. [Duplicación](#)
- 5.7. [Emparentar \(Agrupar\)](#)
- 5.8. [Seguimiento \(Tracking\)](#)
- 5.9. [Otras Acciones](#)
- 5.10. [Operaciones booleanas](#)

6. [Modelado de Malla básico](#)

- 6.1. [Objetos Básicos](#)
- 6.2. [Modo Edición](#)
 - 6.2.1. [Estructuras: Vértices, Aristas y Caras](#)
 - 6.2.2. [Edición Básica](#)
 - 6.2.3. [Deshacer para Mallas](#)
- 6.3. [Suavizando \(Smoothing\)](#)
- 6.4. [Extrudado](#)
 - 6.4.1. [La Hoja](#)
 - 6.4.2. [La Empuñadura](#)
 - 6.4.3. [La Guarnición](#)

6.5. [Giro \(Spin\) y Giro duplicado \(SpinDup\)](#)

6.5.1. [Spin](#)

6.5.2. [SpinDup](#)

6.6. [Rosca \(Screw\)](#)

6.7. [Herramienta Curvar \(Warp\)](#)

6.8. [Garfios de Objeto \(Object Hooks\)](#)

6.8.1. [Agregando garfios](#)

6.8.2. [Usando garfios](#)

6.8.3. [Opciones de ModoEdición](#)

6.8.4. [Panel de Garfios](#)

7. [Modelado de Malla avanzado](#)

7.1. [Subdivisión de Superficies Catmull-Clark](#)

7.2. [Pliegues pesados para superficies de subdivision](#)

7.3. [Herramientas de Arista](#)

7.3.1. [Herramienta Cuchillo \(Knife Tool\)](#)

7.3.2. [Bucle de Caras \(Face Loop\)](#)

7.4. [Herramientas de solevado \(Bevel\)](#)

7.5. [Modelado Simétrico](#)

7.6. [Herramienta de Edición Proporcional \(PET\)](#)

7.7. [Ruido \(Noise\)](#)

7.8. [Herramienta Diezmador \(Decimator\)](#)

8. [Los Meta Objects](#)

9. [Curvas y Superficies](#)

9.1. [Curvas](#)

9.1.1. [Curvas Bèziers](#)

9.1.2. [Curvas NURBS](#)

9.1.3. [Un ejemplo funcional](#)

9.2. [Superficies](#)

9.3. [Texto](#)

9.3.1. [Caracteres especiales](#)

9.4. [Extruir a lo largo de una trayectoria](#)

9.5. [Curva Taper](#)

9.6. [Curva Deformar](#)

9.6.1. [Introducción](#)

9.6.2. [La Interfaz](#)

9.6.3. [Ejemplo](#)

9.7. [Skinning](#)

10. [Materiales](#)

- 10.1. [Difusión](#)
- 10.2. [Reflexión especular](#)
- 10.3. [Materiales en la práctica](#)
 - 10.3.1. [Colores del Material](#)
 - 10.3.2. [Los Sombreadores \(The Shaders\)](#)
 - 10.3.3. [Manipulando Materiales \(Tweaking Materials\)](#)
- 10.4. [Sombreadores de Rampa \(Ramp Shaders\)](#)
 - 10.4.1. [Introducción](#)
 - 10.4.2. [Interfaz](#)
 - 10.4.3. [Un ejemplo](#)
- 10.5. [Reflexiones con el Trazador de rayos](#)
- 10.6. [Transparencias con el trazador de rayos](#)
- 10.7. [Materiales Múltiples](#)
- 10.8. [Materiales Especiales](#)
 - 10.8.1. [Materiales Halo](#)
 - 10.8.2. [Lens Flares](#)

11. [Texturas](#)

- 11.1. [Texturas](#)
 - 11.1.1. [Texturas desde el Punto de Vista del Material](#)
 - 11.1.2. [Texturas \(Textures themselves\)](#)
 - 11.1.3. [ImageTexture](#)
- 11.2. [Mapas de Ambiente \(Environment Maps\)](#)
- 11.3. [Mapas de Desplazamiento](#)
 - 11.3.1. [Mapas de Desplazamiento en Objetos](#)
 - 11.3.2. [Interfaz](#)
 - 11.3.3. [Utilización de un Mapa de desplazamiento](#)
 - 11.3.4. [Ejemplo](#)
- 11.4. [Cristal Solido y Hueco](#)
 - 11.4.1. [Cristal Solido](#)
 - 11.4.2. [Cristal Hueco](#)
- 11.5. [EditorUV y Selección de Caras](#)
 - 11.5.1. [Introducción](#)
 - 11.5.2. [El Editor UV](#)
 - 11.5.3. [Herramientas de Desplegado \(Unwrapping\)](#)
 - 11.5.4. [Editando las coordenadas UV](#)
 - 11.5.5. [Despliegue LSCM](#)
 - 11.5.6. [Pintado de Texturas \(Texture Paint\)](#)
 - 11.5.7. [Renderizado y coordenadas UV](#)

11.6. [Desenrollando a Suzanne](#)

11.6.1. [Organizandose con "Costuras"](#)

11.6.2. [Desenrollando la malla](#)

11.6.3. ["Cosiendo el Mapa"](#)

11.6.4. [Otra vez por favor ;-\)](#)

11.6.5. [Finalizando](#)

11.6.6. [¿ Y ahora qué ?](#)

11.7. [Plugins de Textura](#)

12. [Iluminación](#)

12.1. [Introducción](#)

12.2. [Tipos de luces](#)

12.2.1. [Luz Solar \(Sun Light\)](#)

12.2.2. [Luz Hemi](#)

12.2.3. [Luz lámpara](#)

12.2.4. [Luz focal](#)

12.3. [Sombras con el trazador de rayos](#)

12.4. [Sombras con el buffer](#)

12.5. [Luz Volumétrica](#)

12.6. [Ajustando la iluminación](#)

12.6.1. [Tres puntos de luz](#)

12.6.2. [Tres puntos de luz - Exteriores](#)

12.6.3. [Luz tipo Area](#)

12.6.4. [Iluminación Global \(y Sombreado Global\)](#)

13. [El Mundo y el Universo](#)

13.1. [El Fondo del Mundo](#)

13.2. [Exposición](#)

13.3. [Niebla \(Mist\)](#)

13.4. [Estrellas](#)

13.5. [Oclusión Ambiental](#)

[Prev](#)

[Home](#)

[Next](#)

¡Gus anda!

Modo Objecto

Chapter 5. Modo Objecto

por *Martin Kleppmann*

La geometría de una escena de Blender se construye a partir de uno o más objetos: luces, curvas, superficies, cámaras, mallas y los objetos básicos descritos en [Section 6.1](#). Cada objeto puede ser movido, rotado y cambiado de tamaño en *ObjectMode*. Para cambios más detallados en la geometría, se puede trabajar sobre la malla de un Objeto en *EditMode* (véase [Section 6.2](#)).

Una vez se ha añadido un objeto básico mediante el menú **SPACE>>Add**, Blender cambia a modo *EditMode* por defecto si el objeto es una Malla (Mesh), una Curva (Curve) o una Superficie (Surface). Se puede cambiar a *ObjectMode* presionando **TAB**. La estructura de alambre del objeto, si la hay, debería aparecer en rosa, lo que significa que el objeto está ahora seleccionado y activo.

5.1. Selección de objetos

Concerniente a Blender v2.31

Para seleccionar un objeto, pulse sobre él en **RMB**. Para seleccionar múltiples objetos, mantenga pulsado **SHIFT** y pulse en **RMB**. Generalmente, el último objeto para seleccionar es el que está *activo*: Aparece en rosa pálido, mientras que los objetos seleccionados no activos aparecen en color morado. La definición de objeto activo es importante por varias razones, incluyendo la "paternidad" de los mismos.

Para deselectar el objeto activo, haga click sobre él de nuevo con **RMB**, si hay varios objetos seleccionados pulse la tecla **SHIFT** para mantener los otros como están. Pulse **AKEY** para seleccionar todos los objetos de la escena (si no hay ninguno seleccionado en ese momento) o para deselectarlos todos (si uno o más están seleccionados).

BKEY activa *Border select*. Use la opción Border select para seleccionar un grupo de objetos dibujando un rectángulo mientras mantiene pulsada **LMB**. Se seleccionarán todos los objetos dentro del rectángulo o que estén en contacto con él.

Note Border select añade objetos a la selección previa, por lo tanto para seleccionar unicamente el contenido del rectángulo, deseccione todo primero con **AKEY**. Use **MMB** mientras dibuja el borde para deselectar todos los objetos dentro del rectángulo.

[Prev](#)

Modelado, Materiales e
Iluminación

[Home](#)

[Up](#)

[Next](#)

Movimiento (traslación) de
objetos

5.2. Movimiento (traslación) de objetos

Concerniente a Blender v2.31

Para mover grupos de objetos, pulse **GKEY** para activar el *modo Grab* (modo Agarrar) para todos los objetos seleccionados. Dichos objetos se verán como estructuras de malla de alambre que pueden moverse con el ratón (sin pulsar ningún botón del mismo). Para confirmar la posición nueva, pulse **LMB** o **ENTER**; para salir del modo Grab pulse **RMB** or **ESC**. La cabecera de la ventana 3D muestra la distancia que se está moviendo el objeto.

Para restringir el movimiento a un solo eje del sistema global de coordenadas, entre en el modo Grab, mueva el objeto en el eje deseado, y pulse **MMB**. Para desactivar esta restricción pulse de nuevo **MMB**. Como nueva característica en la versión 2.3 se puede restringir el movimiento a un eje dado pulsando **XKEY**, **YKEY** o **ZKEY**. Cada tecla restringe el movimiento al eje correspondiente *global*, tal como hace **MMB**. Una segunda pulsación de la *misma* tecla restringe el movimiento al eje correspondiente *local* del objeto. Una tercera pulsación de la misma tecla quita las restricciones. Se dibujan unas líneas para que se pueda visualizar mejor la restricción.

Una vez que el modo Grab está activado se puede introducir la traslación del objeto numéricamente simplemente tecleando el número. Esto permitirá introducir la primera coordenada mostrada en la cabecera de la ventana 3D. Se puede cambiar la coordenada con **TAB**, usando **NKEY** para salir/volver a entrar al modo de introducción numérica, **ENTER** para finalizar y **ESC** para salir. **BACKSPACE** dejará los valores originales. Nótese que debe usarse la tecla **.KEY** y no la **NUM.** para los decimales.

Si mantiene pulsada la tecla **CTRL** mientras mueve el objeto activará el modo *snap* (movimiento por bloque), y el objeto se moverá en un número completo de unidades de Blender (cuadros de la rejilla). El modo Snap termina cuando se suelta la tecla **CTRL**, por lo que debe asegurarse de confirmar la posición antes de hacerlo.

La ubicación de objetos seleccionados puede dejarse al valor por defecto pulsando **ALT-G**.

Note Si le es difícil colocar un objeto de forma muy precisa, mantenga pulsado **SHIFT** mientras realiza el movimiento. De esta manera un movimiento grande de ratón se convertirá en un movimiento pequeño del objeto, lo que permite un ajuste más fino.



Sistema gestual de Blender

Se puede entrar en el modo Grab dibujando una línea recta mientras se tiene pulsado **LMB**.

[Prev](#)

Modo Objecto

[Home](#)

[Up](#)

[Next](#)

Rotación de objetos

5.3. Rotación de objetos

Concerniente a Blender v2.31

Para rotar objetos, active el modo Rotación (Rotate) pulsando **RKEY**. Al igual que en el modo Grab, puede cambiarse la rotación moviendo el ratón, confirmar con **LMB** o **ENTER**, cancelar con **RMB** o **ESC**.

La rotación en el espacio 3D se produce alrededor de un eje, y hay varias formas de definir dicho eje. Blender define un eje mediante una dirección y un punto por el que pasa. Por ejemplo, por defecto la dirección de un eje es ortogonal a su pantalla.

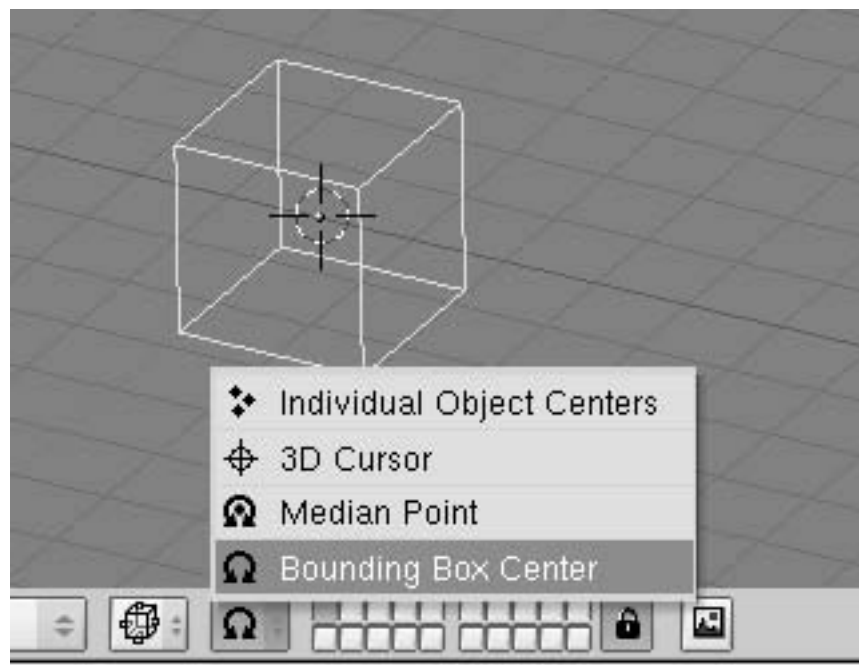
Si está viendo la escena desde delante desde un lado o desde arriba el eje de rotación será paralelo a una de las coordenadas globales del sistema de ejes. Si está viendo la escena desde un ángulo el eje de rotación estará en ángulo también, lo que puede originar una rotación muy extraña de su objeto. En ese caso necesitará mantener el eje de rotación paralelo a los ejes del sistema de coordenadas. Active y desactive este modo pulsando **MMB** durante el modo Rotación y vea el indicador de ángulo en la cabecera de la ventana.

De igual modo, una vez está en modo Rotación, puede pulsar **XKEY**, **YKEY** o **ZKEY** para restringir la rotación a uno de los ejes *globales de referencia*. Pulsando **XKEY-XKEY** (dos veces **XKEY**) se restringe la rotación alrededor del eje x del *objeto*. Esto mismo es válido para la doble pulsación de **YKEY** y **ZKEY**. Como en el modo Grab, una tercera pulsación elimina las restricciones.

Es posible introducir números para una rotación exacta, al igual que con las traslaciones.

Seleccione el punto por el que pasa el eje de rotación con el menú pertinente en la cabecera de la ventana 3D, como se describe más abajo. ([Figure 5-1](#)).

Figure 5-1. Los botones de selección de punto de rotación



- *Centro del cuadro limitador (Bounding Box)* - el eje pasa por el centro del cuadro limitador de la selección. (Si sólo hay un objeto seleccionado, el punto usado es el punto central del objeto, que puede no ser necesariamente el centro geométrico. En [Figure 5-1](#) está en el medio del borde de la derecha, marcado por un punto granate. Para más información sobre este punto véase [Section 6.2](#).)
- *Punto de la Mediana* - el eje pasa por el punto de la mediana de la selección. Esta diferencia sólo es importante en el modo Edición, y el punto 'Mediana' es el baricentro de todos los vértices.
- *Cursor 3D* - el eje pasa por el cursor 3D. El cursor puede colocarse en cualquier sitio que se desee antes de rotar. Puede usarse esta opción para realizar fácilmente ciertas traslaciones al mismo tiempo que se rota el objeto.
- *Centros de objetos individuales* - cada objeto seleccionado recibe su propio eje de rotación, todos paralelos entre sí y que pasan por el punto central de cada objeto, respectivamente. Si selecciona sólo un objeto, conseguirá el mismo efecto que con el primer botón.

Si acaba de empezar a usar la rotación, no se preocupe demasiado por los detalles precedentes. Practique con las herramientas de Blender y se irá haciendo una idea de cómo trabajar con ellas.

Manteniendo pulsada **CTRL** se pasa al modo Snap. En el modo Snap las rotaciones están restringidas a 5 pasos. Manteniendo pulsada **SHIFT** se puede hacer un ajuste fino. La rotación de los objetos seleccionados puede dejarse al valor por defecto pulsando **ALT-R**.

Tip Sistema Gestual de Blender

También puede entrar en el modo Rotación dibujando una línea circular mientras mantiene pulsada **LMB**.

Movimiento (traslación) de
objetos

[Up](#)

Cambio de tamaño/imagen
especular de objetos

5.4. Cambio de tamaño/imagen especular de objetos

Concerniente a Blender v2.31

Para cambiar el tamaño de los objetos pulse **SKEY**. Al igual que en el modo Agarrar y Rotación cambie el tamaño moviendo el ratón, confirme con **LMB** o **ENTER**, y cancele con **RMB** o **ESC**.

El cambio de tamaño en 3D requiere un punto central. Este punto se define con los mismos botones que para el punto de soporte del eje en la rotación ([Figure 5-1](#)). Si aumenta el tamaño del objeto, todos los puntos del mismo se alejarán del punto central seleccionado; si lo disminuye, los puntos se moverán hacia él.

Por defecto, los objetos seleccionados cambian de tamaño uniformemente en todas direcciones. Para cambiar las proporciones (hacer el objeto más largo, ancho, etc.) puede bloquear el proceso de escalado a uno de los ejes globales de coordenadas al igual que haría para mover objetos. Para hacer esto entre el modo de cambio de tamaño, mueva el ratón un poco en la dirección del eje sobre el que quiere ejercer la acción y pulse **MMB**. Para volver al cambio de tamaño uniforme pulse **MMB** de nuevo. Podrá ver los factores de escala en la cabecera de la ventana 3D.

De nuevo todas las consideraciones referentes a restricciones con respecto a un eje dado hechas en el modo Agarrar se mantienen, así como las de entrada numérica de datos.

De nuevo **CTRL** activa el modo "Por Bloque" o Snap, con un escalado discreto a 0,1 pasos. Pulse **SHIFT** para un ajuste fino. El escalado de objetos seleccionados puede ponerse al valor por defecto pulsando **ALT-S**.

La imagen especular se hace con una aplicación diferente de la herramienta de escalado. Dicha imagen no es más que escalar con un factor negativo en una dirección. Para hacer una imagen especular en los ejes X o Y pulse **SKEY** para ir al modo escalado, y luego **NKEY** para pasar a introducción numérica de datos. Seleccione las coordenadas deseadas e indique '-1' como factor de escala.



Sistema Gestual de Blender

También puede pasar al modo escalado dibujando una línea en forma de V mientras mantiene pulsado **LMB**.

[Prev](#)

Rotación de objetos

[Home](#)

[Up](#)

[Next](#)

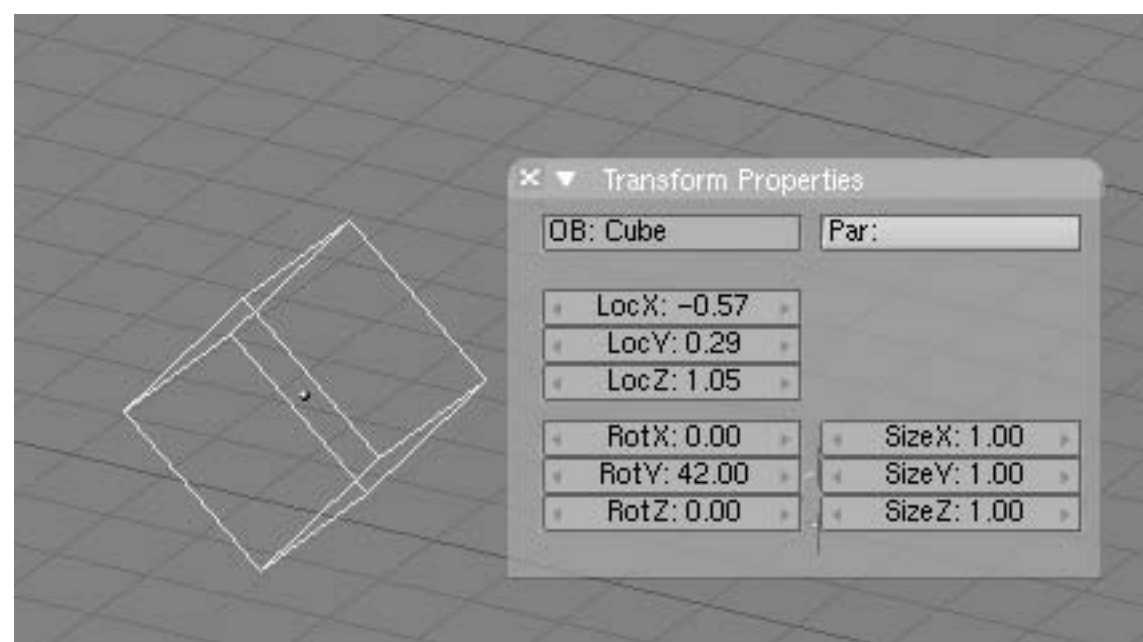
Panel de propiedades de
transformación

5.5. Panel de propiedades de transformación

Concerniente a Blender v2.31

Supongamos que quisiera que se mostrasen la posición/rotación/escalado de su objeto en números. O que quisiera introducir los valores de dichos parámetros de su objeto directamente. Para hacerlo, seleccione el objeto que quiere editar y pulse **NKEY**. El panel *Transform Properties* (Propiedades de transformación) ([Figure 5-2](#)) se muestra en pantalla. Pulse **SHIFT-LMB** y un número para introducir un valor, luego pulse OK para confirmar los cambios o mueva el ratón fuera de la ventana para cancelar.

Figure 5-2. El cuadro de diálogo numérico



El panel también muestra el nombre del objeto en el botón *OB:*. Puede editarlo desde este lugar.

5.6. Duplicación

Concerniente a Blender v2.31

Para duplicar un objeto pulse **SHIFT-D** y se creará una copia idéntica de los objetos seleccionados. La copia se crea en la misma posición, en modo Agarrar.

Será un objeto nuevo a excepción de que comparte Material, Textura e IPO con el original. Estos atributos se asocian a las dos copias, y el cambio de un material de uno de los objetos también afecta al otro. (Puede hacer materiales distintos para cada uno, como se describe en el capítulo de Materiales).

Puede crear un *Duplicado Asociado* en vez de un duplicado real pulsando **ALT-D**. Esto creará un objeto nuevo con *todos* sus datos asociados al objeto original. Si se ha modificado uno de los objetos asociados en el Modo Edición todas las copias también serán modificadas.

5.7. Emparentar (Agrupar)

Concerniente a Blender v2.31

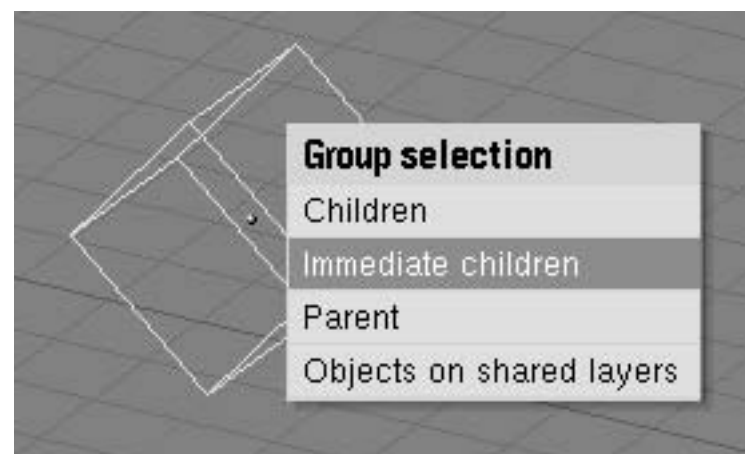
Para crear un grupo de objetos primero debe convertir a uno de ellos en *padre* de los otros. Para hacer esto, seleccione al menos dos objetos, pulse **CTRL-P**, y confirme la pregunta *¿Hacer padre?*. El objeto *activo* se convertirá en padre de todos los demás. El centro de todos los hijos estará ahora asociado al centro del padre por una línea de puntos. En este momento, el arrastre, la rotación y el escalado del padre tendrán el mismo efecto en los hijos.

El Emparentado es una herramienta muy importante con muchas aplicaciones avanzadas, como se verá en capítulos posteriores.

Pulse **SHIFT-G** con un objeto activo para ver el menú de Selección de Grupo ([Figure 5-3](#)). Este menú contiene:

- *Children* - Selecciona todos los hijos del objeto activo y los hijos de los hijos hasta la última generación.
- *Immediate Children* - Seleccionar todos los hijos del objeto activo pero no los hijos posteriores.
- *Parent* - Selecciona el padre del objeto activo.
- *Objects on shared layers* - En realidad esto no tiene nada que ver con padres e hijos. Selecciona todos los objetos de la(s) misma(s) capa(s) del objeto activo.

Figure 5-3. Selección de grupo

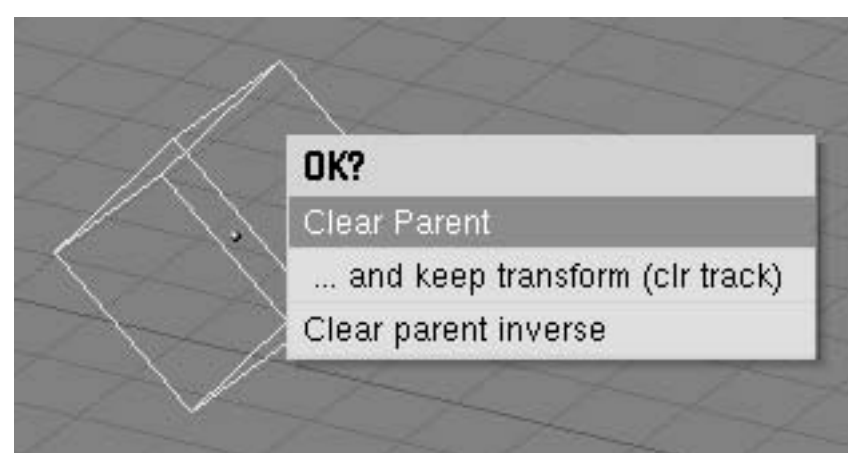


Mueva el hijo al padre borrando su origen (selecciónelo y pulse **ALT-O**)

Borre una relación parental con **ALT-P**. Puede ([Figure 5-4](#)):

- *Clear parent* - Libera los hijos, que vuelven a su ubicación, rotación y tamaño *originales*.
- *Clear parent...and keep transform* - Libera los hijos y *mantiene* la ubicación, rotación y tamaño que les dio el padre.
- *Clear parent inverse* - Coloca a los hijos con respecto al padre como si se hubieran colocado en la referencia Global. Esto borra definitivamente la transformación del padre sobre los hijos.

Figure 5-4. Liberación de hijos



[Prev](#)

Duplicación

[Home](#)

[Up](#)

[Next](#)

Seguimiento (Tracking)

5.8. Seguimiento (Tracking)

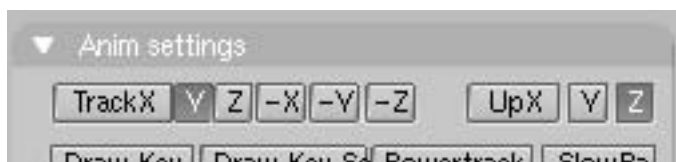
Concerniente a Blender v2.31

Para rotar un objeto de manera que se oriente hacia otro y mantener esta orientación si uno de los objetos se mueve seleccione al menos dos objetos y pulse **CTRL-T**. Aparecerá un cuadro de diálogo preguntándole si quiere usar una *restricción* de Seguimiento o el sistema de pista antiguo (Pre-2.30). La restricción de Pista se analizará en [Section 16.8](#) y es el método recomendable.

Se explicará aquí brevemente el sistema antiguo de seguimiento, de manera que se asume que tiene seleccionado *Old Track* en el cuadro de diálogo. Por defecto el objeto inactivo sigue al objeto activo de manera que los puntos de su eje local Y apuntan al objeto al que se hace el seguimiento. Sin embargo, esto puede no ocurrir si el objeto "seguidor" tiene ya una rotación. Se puede hacer un seguimiento con corrección cancelando la rotación (**ALT-R**) de este objeto.

La orientación del objeto "seguidor" también se puede fijar con el eje Z hacia arriba. Para cambiar esto, seleccione dicho objeto, cambie la ventana de Botones a Contexto de Objeto (, o **F7**) y seleccione el eje de seguimiento en la primera fila de seis botones, así como el eje que mirará hacia arriba en la segunda, que está en el panel *Anim Setting*. ([Figure 5-5](#)).

Figure 5-5. Configuración del eje de seguimiento



Para eliminar una restricción de seguimiento, seleccione el objeto "seguidor" y pulse **ALT-T**. Al igual que en la eliminación de las restricciones entre padres e hijos, debe elegir si quiere conservar o perder la rotación impuesta por el seguimiento.

5.9. Otras Acciones

Concerniente a Blender v2.31

Borrado

Pulse **XKEY** o **DEL** para borrar los objetos seleccionados. El uso de **XKEY** es más práctico para la mayoría de la gente, porque se puede pulsar fácilmente con la mano izquierda sobre el teclado.

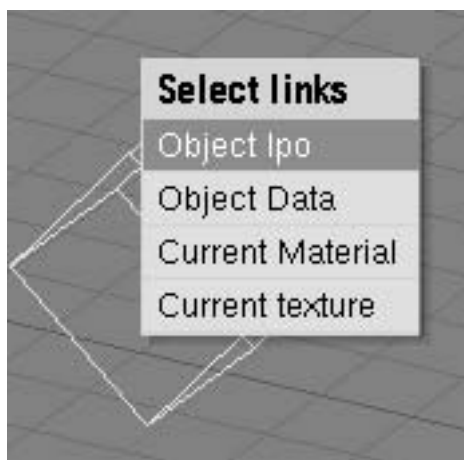
Unión

Pulse **CTRL-J** para unir todos los objetos seleccionados en un único objeto. (Los objetos debe ser del mismo tipo). El punto central del objeto resultante se obtiene del objeto *activo* anterior.

Seleccionar enlaces

Pulse **SHIFT-L** para seleccionar todos los objetos que compartan un enlace con el objeto activo. Puede seleccionar objetos que compartan enlace IPO, de datos, de material, o de textura ([Figure 5-6](#)).

Figure 5-6. Selección de enlaces.



5.10. Operaciones booleanas

Relativo a Blender v2.31

Las operaciones booleanas son acciones concretas que pueden usarse únicamente con objetos de tipo malla. Mientras que funcionan para todos los objetos malla, están especialmente indicadas para usarse con objetos cerrados sólidos con una región interna y externa bien definidas. Así, es muy importante definir consistentemente las normales en cada objeto, es decir, todas las normales de cada objeto deben apuntar hacia afuera. Vea [Chapter 6](#) para más información sobre normales y sobre por qué pueden producirse normales apuntando parcialmente hacia fuera y parcialmente hacia dentro.

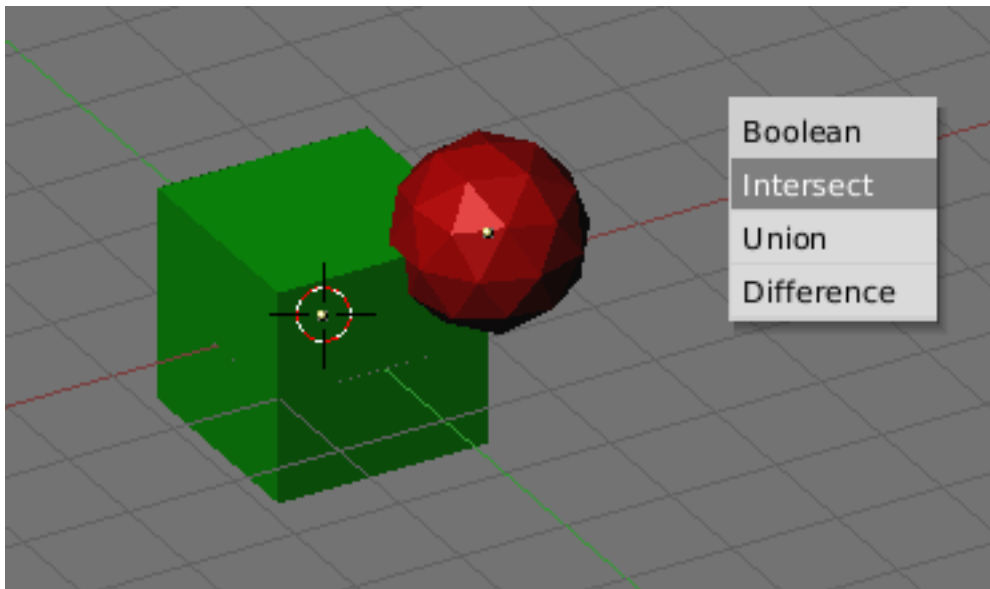
En el caso de objetos abiertos, el interior es definido matemáticamente extendiendo las caras del borde del objeto hasta el infinito. Como tal, puede encontrar que se produzcan resultados inesperados en estos objetos.

Una operación booleana nunca afecta a los operandos originales; el resultado es siempre un nuevo objeto de Blender.

Las operaciones booleanas se invocan seleccionando *exactamente* dos mallas y presionando **WKEY**. Hay tres tipos de operaciones booleanas para escoger en el menú emergente: *Intersect* (intersección), *Union* (unión) y *Difference* (diferencia).

Las operaciones booleanas también toman en consideración materiales y texturas UV, produciendo objetos con índices de materiales u objetos multimapeados UV.

Figure 5-7. Opciones para operaciones booleanas

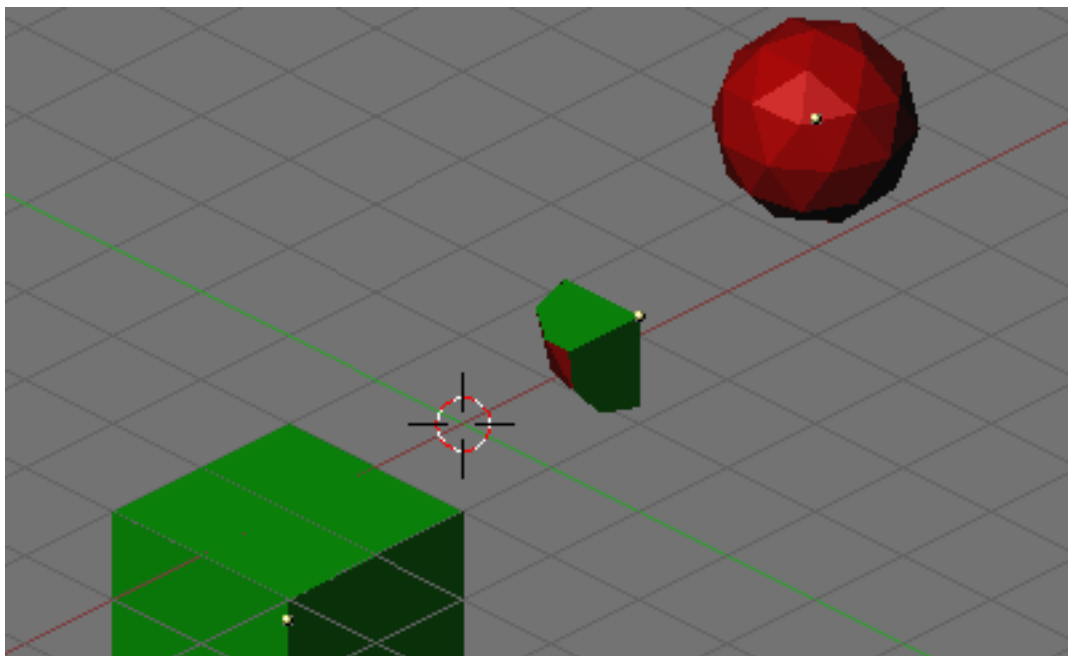


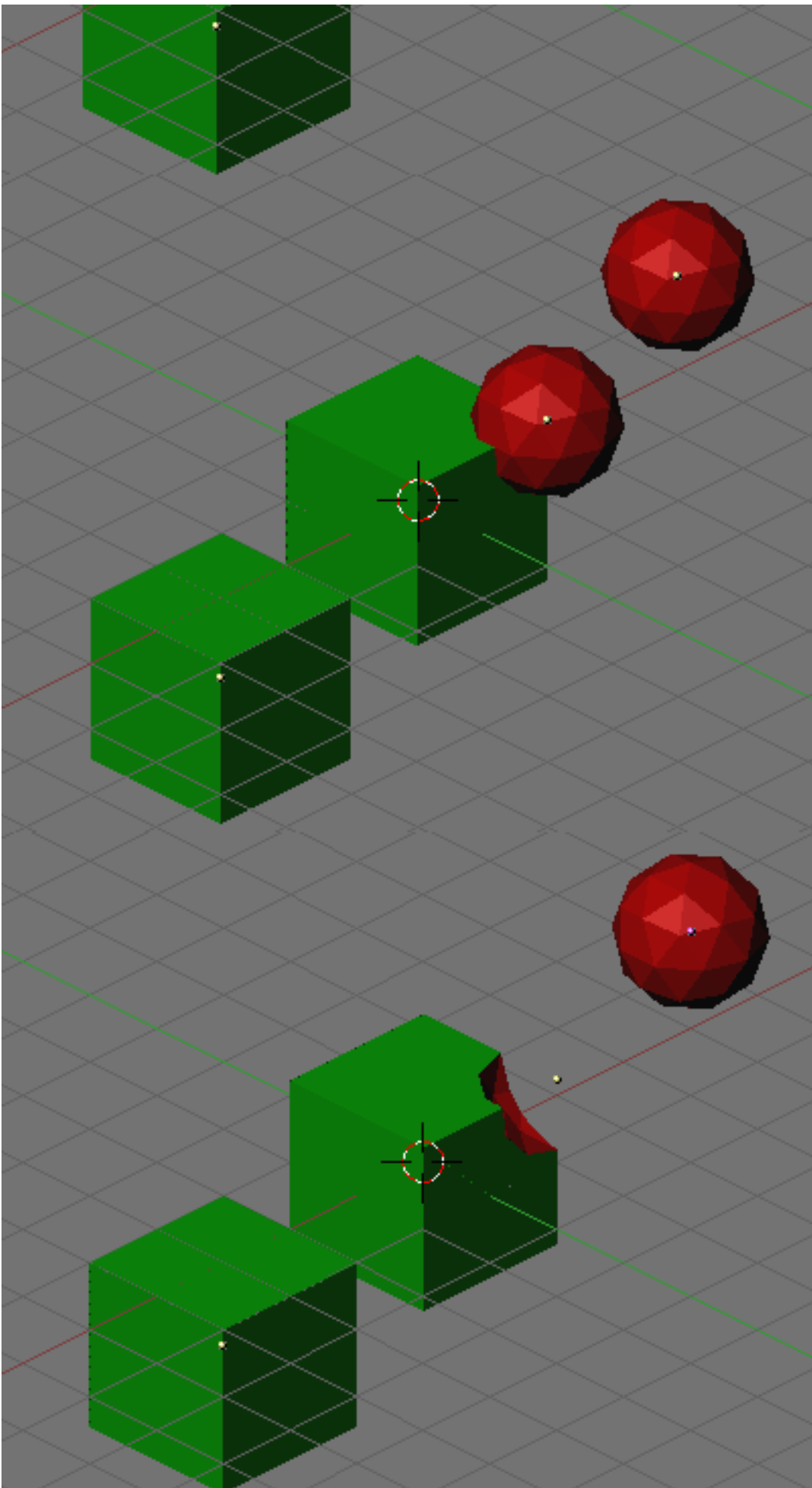
Considere el objeto [Figure 5-7](#).

- La operación *Intersect* crea un nuevo objeto cuya superficie encierra el volumen común a *ambos* objetos originales.
- La operación *Union* crea un nuevo objeto cuya superficie encierra el volumen de *ambos* objetos originales.
- La operación *Difference* es la única en la que el orden de selección es importante. El objeto activo (púrpura claro en vista de alambre) es restado del objeto seleccionado. Es decir, la superficie del objeto resultante encierra un volumen que corresponde al volumen del objeto seleccionado *e inactivo*, pero *no* al objeto seleccionado y *activo*.

[Figure 5-8](#) muestra el resultado de las tres operaciones.

Figure 5-8. Objetos resultantes para: intersección, unión, diferencia (de arriba a abajo).





El número de polígonos generados puede ser muy grande comparado con las mallas originales, especialmente al usar objetos cóncavos complejos. Además, los polígonos generados pueden ser generalmente de calidad pobre: muy largos y delgados y en ocasiones muy pequeños. Intente usar el Decimator de mallas (botones EditButtons **F9**) para reparar este problema.

Los vértices en la malla resultante que caen en la frontera de los dos objetos originales, frecuentemente no encajan, y los vértices del borde son duplicados. Esto es bueno en algunas circunstancias puesto que significa que puede seleccionar partes de la malla original seleccionando un vértice en el resultado y presionando el botón Select Linked (Seleccionar enlazados, **LKEY**). Ello es útil si desea asignar materiales y similares al resultado.

Note En ocasiones la operación booleana puede fallar con un mensaje diciendo "An internal error occurred -- sorry" ("Ocurrió un problema interno"). Si ello ocurre, intente mover o rotar los objetos ligeramente.

[Prev](#)

Otras Acciones

[Home](#)[Up](#)[Next](#)

Modelado de Malla básico

Chapter 6. Modelado de Malla básico

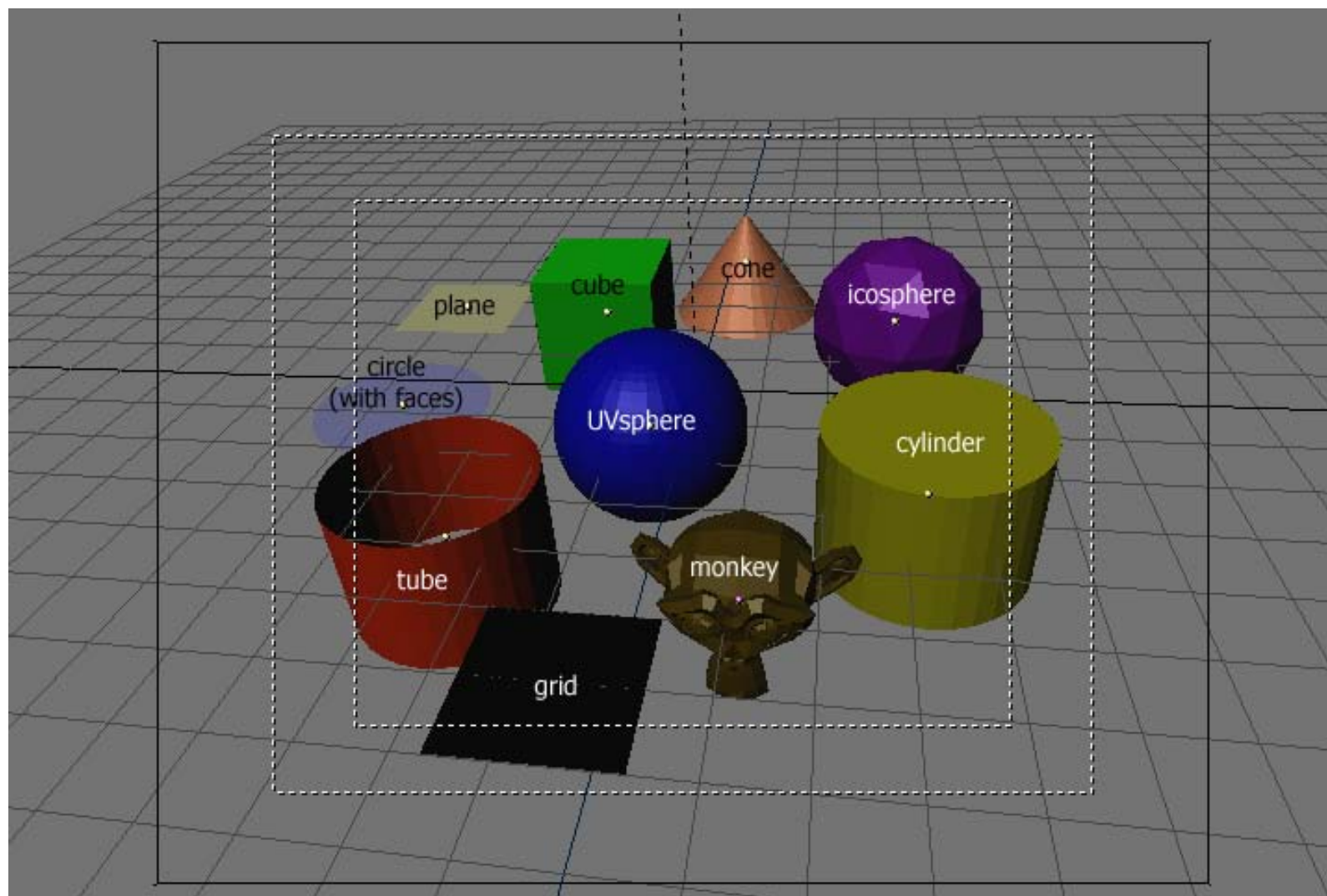
El objeto principal de una escena en 3D es normalmente una *Malla* (Mesh). En este capítulo primero vamos a enumerar los objetos básicos de malla, o *primitivas*, para despues seguir con una larga serie de secciones describiendo en detalle las acciones que se pueden realizar sobre objetos de malla.

6.1. Objetos Básicos

Relevante a Blender v2.31

Para crear un Objeto Básico pulse **SPACE** y seleccione "ADD>>Mesh", o acceda al menú 'add' pulsando **SHIFT-A** o simplemente mantenga presionado **LMB** sobre la ventana 3D por más de medio segundo. Seleccione del menú el objeto básico que le gustaría crear. En las líneas que siguen describiremos cada objeto básico o *primitive* que se puede crear en Blender. [Figure 6-1](#) también muestra la variedad de objetos básicos que pueden crearse.

Figure 6-1. Objetos Basicos o Primitivas



Plano (Plane)

Un plano estándar que contiene cuatro vértices, cuatro aristas y una cara. Es como una hoja de papel sobre una mesa; no es realmente un objeto tridimensional ya que es totalmente plano y no tiene espesor. Entre los objetos que pueden crearse a partir de planos se incluyen pisos, espejos y manteles.

Cube (Cubo)

Un cubo estándar que contiene ocho vértices, doce aristas, y seis caras, es un objeto tridimensional real. Entre los objetos que pueden crearse a partir de cubos podemos citar dados, cajas y cofres.

Circle (Circunferencia)

Una circunferencia estándar compuesta de n vértices. El número de vértices puede indicarse en la ventana que aparece cuando se crea la circunferencia. Mientras más vértices posea, más suave será el contorno. Ejemplos de objetos creados a partir de circunferencias son discos, platos o cualquier tipo de objeto plano y redondo.

UVSphere (EsferaUV)

Una esferaUV estándar está hecha de n segmentos y m anillos. El nivel de detalle puede indicarse en la ventana emergente que aparece cuando se crea la esferaUV. Aumentando el número de segmentos y anillos, la superficie de la esferaUV se hace más suave. Los segmentos son como meridianos terrestres, van de un polo al otro, en cambio los anillos son como paralelos terrestres. Ejemplos de objetos que pueden crearse a partir de esferasUV son pelotas, cabezas o las perlas de un collar.

Note

Si crea una esferaUV de seis segmentos y seis anillos, obtendrá algo que en la VistaSuperior es un hexágono (seis segmentos), con cinco anillos más dos puntos en los polos. O sea, un anillo menos que los esperados, o dos más si cuenta los polos como anillos de radio 0.

Icosphere (Icoesfera)

Una icoesfera está hecha de triángulos. El número de subdivisiones puede especificarse en la ventana que aparece cuando se crea la Icoesfera; aumentando el número de subdivisiones se hace más pulida la superficie de la icoesfera. Una Icoesfera de nivel 1 es un icosaedro, un sólido con 20 caras triangulares equiláteras. Cualquier nivel de subdivisión mayor subdividirá cada cara triangular en cuatro triángulos, resultando en una apariencia más esférica cada vez. Las icoesferas se utilizan normalmente para obtener una disposición de los vértices más isotrópica y económica que en una esferaUV.

Cylinder (Cilindro)

Un cilindro estándar hecho de a partir de una sección circular de n vértices. El número de vértices de sección transversal circular puede indicarse en la ventana emergente que aparece cuando se crea el objeto; mientras mayor sea el número de vértices, más suave será la sección circular transversal. Algunos objetos que pueden crearse a partir de cilindros incluyen barras y manijas.

Tube (Tubo)

Un tubo estándar hecho de n vértices. El número de vértices en la sección transversal circular hueca puede especificarse en la ventana emergente que aparece cuando se crea el objeto; a mayor número de vértices más suave se vuelve la sección circular hueca. Entre los objetos que pueden crearse a partir de tubos, podemos citar caños o vasos. (La diferencia básica entre un

cilindro y un tubo es que el primero posee sus extremos cerrados.)

Cone (Cono)

Un cono estándar hecho de n vertices. El número de vertices en la base circular puede especificarse en la ventana emergente que aparece cuando se crea el objeto; a mayor número de vertices más suave se vuelve la base circular. Entre los objetos que pueden crearse a partir de conos, podemos citar púas o sombreros de punta.

Grid (Rejilla o cuadrícula)

Una cuadrícula estándar hecha de n por m vertices. La resolución del eje X y el eje Y The resolution of the x-axis and y-axis puede especificarse en la ventana emergente que aparece cuando se crea el objeto; cuanto más alta sea la resolución, mayor será el número de vertices creados. Entre los objetos que pueden crearse a partir de cuadrículas se incluyen paisajes (con la herramienta de edición proporcional o PET) y otras superficies orgánicas.

Monkey (Mono)

Es un regalo de la vieja NaN a la comunidad y es vista como una broma de los programadores o "huevo de Pascuas". Crea una cabeza de mono (de hecho, de "mona") una vez que se pulsa sobre el botón *Oooh Oooh Oooh*. El nombre de la mona es *Suzanne* y es la mascota de Blender.

[Prev](#)

Operaciones booleanas

[Home](#)[Up](#)[Next](#)

Modo Edición

6.2. Modo Edición

Relativo a Blender v2.31

Cuando se trabaja con objetos geométricos en Blender, se lo puede hacer en dos modos: [Modo Objeto](#) y Modo Edición. Básicamente, como se vio en la sección anterior, las operaciones en Modo Objeto afectan a la los objetos completos, y la soperaciones en ModoEdición solo afectan a su geometría pero no a sus propiedades globales como la Posición o la Rotación.

En Blender puedes pasar de un modo al otro mediante **TAB** . El Modo Edición funciona solamente con un objeto a la vez: el objeto activo. Fuera de Modo Edición los objetos se dibujan en color rosado en la Ventana 3D (en modo alambre) cuando son seleccionados de lo contrario aparecen en color negro. El objeto Activo en Modo Edición se dibuja en color negro, pero cada vértice es resaltado en color rosado ([Figure 6-2](#)). Los vértices seleccionados son dibujados en color amarillo ([Figure 6-3](#)) y, si se encuentran presionados los botones apropiados(*Draw Faces* and *Draw Edges*) en el Panel *Mesh Tools 1* del Contexto Edición (**F9**) también se resaltarán las caras y las aristas.

Figure 6-2. Dos piramides, una en ModoEdición (izquierda) y una en Modo Objeto (derecha).

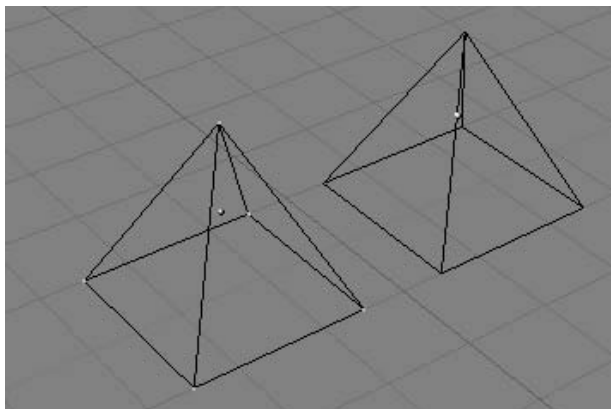
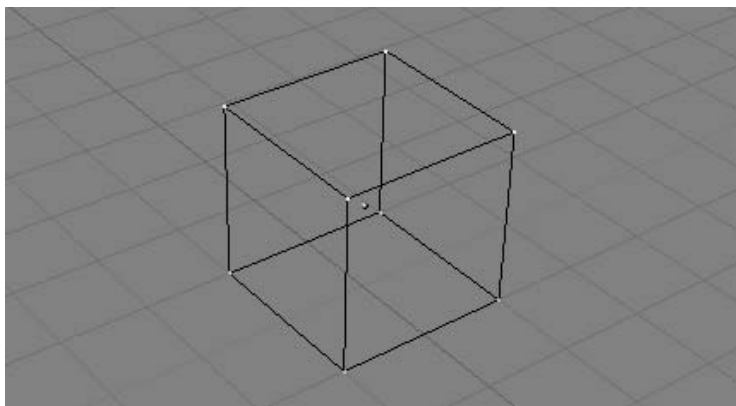


Figure 6-3. Cubo con los vértices seleccionados en amarillo.



6.2.1. Estructuras: Vértices, Aristas y Caras

En las mallas básicas, todo se construye a partir de tres estructuras: *Vértices*, *Aristas* y *Caras*. (Aquí no estamos incluyendo a Curvas, NURBS, etc.) Pero no hay razón para desanimarse: esta simplicidad nos provee un cúmulo de posibilidades que serán la base para todos nuestros modelos.

Vértices

Un vértice es elementalmente un punto o una posición en el espacio 3D. Normalmente es invisible al renderizar y en Modo Objeto. (No confundir el punto central de un objeto con un vértice, es similar pero es más grande y además no puedes seleccionarlo.)

Para crear un nuevo vértice, cambia a Modo Edición, mantén presionada **CTRL**, y pulsa con el **LMB**. Por supuesto, como la pantalla de la computadora es bidimensional, Blender no puede determinar las tres coordenadas del vértice a partir de un click de ratón, por lo que el nuevo vértice se ubica sobre el plano de profundidad del cursor 3D "dentro" de la pantalla. Cualquier vértice seleccionado previamente será conectado automáticamente al nuevo por medio de una arista.

Aristas

Una arista siempre conecta dos vértices con una línea recta. Las aristas son los "alambres" que vez cuando un malla se ve en la vista de alambre. Normalmente son invisibles en la imagen renderizada. Se usan para construir caras. Crea una arista seleccionando dos vértices y pulsando **FKEY**.

Caras

Una Cara es la estructura de más alto nivel en una malla. Las caras se usan para construir la superficie del objeto propiamente dicha. Las caras son lo que ves cuando renderizas una malla. Se la define cómo el área entre tres o cuatro vértices, con una arista en cada lado. Los triángulos siempre funcionan bien porque siempre son planos y fáciles de calcular.

Ten cuidado al crear caras de cuatro lados, porque internamente son separados en dos triángulos. Las caras de cuatro lados solo funcionan bien si la cara es prácticamente plana (todos sus puntos descansan en un plano imaginario) y convexa (en ninguna esquina el ángulo es mayor o igual a 180 grados). Este es el caso con las caras de un cubo por ejemplo. (Esta es la razón por la que no pueden verse ninguna diagonal en modo alambre, porque dividiría cada cara cuadrada en dos triángulos. Aunque tu puedes crear un cubo con caras triangulares, solo se vería más confuso en Modo Edición.)

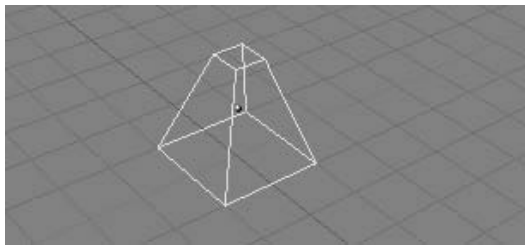
Sin embargo, el área entre tres o cuatro vértices, delineada por aristas, no tiene que ser una cara necesariamente. Si esta área no tiene una cara, será simplemente transparente o inexistente en la imagen renderizada. Para crear una cara, selecciona tres o cuatro vértices y pulsa **FKEY**.

6.2.2. Edición Básica

La mayoría de las operaciones simples de Modo Objeto (como seleccionar, mover, girar y escalar) funcionan de manera idéntica sobre vértices a como lo hacen sobre objetos. Por lo tanto puedes aprender muy rápido cómo utilizar las operaciones básicas de Modo Edición. La única diferencia notable es una nueva opción para escalar **ALT-S** que escala los vértices seleccionados en la dirección de las normales (achica-engorda). Por ejemplo, la pirámide truncada en [Figure 6-4](#), fue creada con los siguientes pasos:

1. Agrega un cubo en una escena vacía. Entra en Modo Edición.
2. Asegurate que todos los vértices estén deseleccionados (rosa). Usa la selección por área (**BKEY**) para seleccionar los cuatro vértices de la parte superior.
3. Verifica que el centro de escalado esté seteado como *cualquier cosa excepto* el cursor 3D (ver [Figure 5-1](#)), luego cambia a Modo Escalar (**SKEY**), reduce el tamaño y confirma con **LMB**.
4. Sal de Modo Edición pulsando **TAB**.

Figure 6-4. Pirámide truncada



Una característica adicional para Modo Edición es la herramienta de Simetría. Si tienes seleccionados algunos vértices y pulsas **MKEY** aparecerá un menú con nueve opciones. Puedes seleccionar una para rebatir el vértice seleccionado simétricamente respecto de cualquiera de los ejes X, Y o Z de las referencias Globales, local o según la vista.

En ModoEdición también se encuentra disponible el modo SelecciónCircular. Se lo invoca pulsando dos veces la tecla **BKEY** en lugar de una sola vez como harías para la Selección por Áreas. Aparecerá alrededor del cursor un círculo gris y con un click **LMB** se seleccionarán todos los vértices dentro de este círculo. Con **NUM+** y **NUM-** o la **MW**, si la hubiera, se agranda o reduce el círculo.

Todas las operaciones en ModoEdición son ejecutadas sobre los vértices; las caras y aristas se adaptan automáticamente ya que dependen de la posición de los vértices. Para seleccionar una arista debes seleccionar los dos puntos de los extremos o bien ubicar el ratón sobre la arista y pulsar **CTRL-ALT-MMB**. Para seleccionar una cara, se debe seleccionar cada esquina.

Las operaciones en Modo Edición son muchas y la mayoría se encuentran resumidas en la ventana de Botones de Contexto Edición, que se accede a través del botón de la cabecera () o por medio de **F9** ([Figure 6-5](#)). Observa el grupo de botones en el Panel *Mesh Tools 1*:

Figure 6-5. Botones de Edición



- *NSize*: - Determina la longitud, en Unidades de Blender, de las normales a las caras, si son dibujadas.
- *Draw Normals* - Activa/desactiva la representación de Normales. Si se encuentra activada, las normales de las caras se dibujarán como segmentos de color cian.
- *Draw Faces* - Si está activada, las caras se dibujan con un azul traslúcido, o rosado traslúcido si están seleccionadas. Si está en OFF, las caras son invisibles.
- *Draw Edges* - Las aristas se dibujan en negro, pero si este botón está activado, las aristas seleccionadas se dibujarán en amarillo. Las aristas que unan un nodo seleccionado y uno no seleccionado, tendrán un gradiente amarillo-negro.
- *All Edges* - En Modo Objeto se muestran solamente las aristas indispensables para definir la forma de un objeto. Con este botón puedes forzar a Blender a mostrar todas las aristas.

Note Por supuesto, todos estos colores son personalizables en el Editor de Temas.

Con **WKEY** puedes invocar el menú "Specials" ([Figure 6-6](#)). Con este menú puedes acceder rápidamente a muchas funciones de uso habitual en el modelado de polígonos.

Tip

Puedes acceder a los ítems en un menú emergente utilizando la tecla numérica correspondiente. Por ejemplo, pulsando **WKEY** y luego **1KEY** subdividirás las aristas seleccionadas sin tener que tocar para nada el ratón.

Figure 6-6. Menú Specials



- *Subdivide* - Cada arista seleccionada se divide en dos, en el punto medio se crean vértices nuevos, y las caras se subdividen también si es necesario.
- *Subdivide Fractal* - Igual que la anterior, pero los vértices nuevos son desplazados aleatoriamente dentro de un rango definido por el usuario.
- *Subdivide Smooth* - Como la anterior, pero los nuevos vértices son desplazados hacia el baricentro (centro de masa) de los vértices conectados.
- *Merge* - Funde los vértices seleccionados en un único vértice ubicado en el baricentro o en la posición del cursor.
- *Remove Doubles* - Funde todos los vértices seleccionados cuya distancia relativa sea menor que un umbral especificado (0.001 por omisión).
- *Hide* - Oculta los vértices seleccionados.
- *Reveal* - Muestra los vértices ocultos.
- *Select Swap* - Todos los vértices seleccionados se deselectionan y viceversa.
- *Flip Normals* - Cambia la dirección de las normales de las caras seleccionadas.
- *Smooth* - Suaviza una malla moviendo los vértices hacia el baricentro de los vértices enlazados.
- *Mirror* - Idéntica a **MKEY** descrita anteriormente.

Muchas de estas acciones tienen asignado un botón propio en el Panel *Mesh Tools* de la Ventana de Botones de Edición ([Figure 6-5](#)). El umbral para *Remove Doubles* se puede especificar allí también.

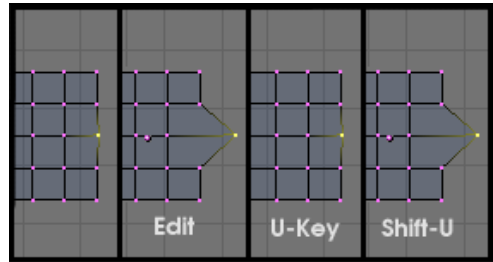
6.2.3. Deshacer para Mallas

A partir de Blender 2.3 disponemos de un verdadero Deshacer. Solo funciona para Mallas y únicamente en ModoEdición.

El comando Deshacer para mallas trabaja en segundo plano almacenando en memoria copias de tu malla a medida que es modificada. Pulsando **UKEY** en ModoEdición volvemos a la malla salvada deshaciendo la última operación de edición ([Figure 6-7](#)).

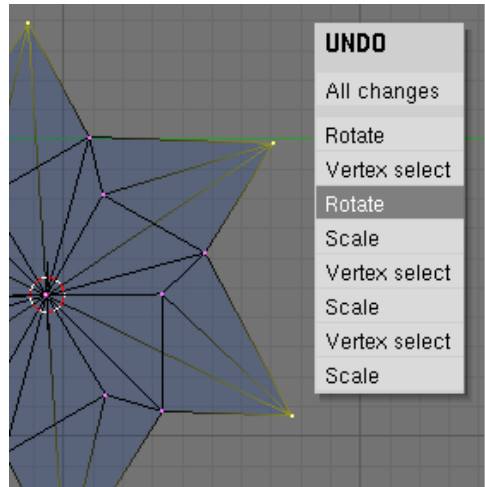
Las operaciones de Deshacer solo se almacenan para una malla a la vez. Puedes salir e ingresar a la misma malla sin perder información de Deshacer, pero una vez que editas otra malla, se perderá la información de Deshacer para la primera.

Figure 6-7. Deshacer y Rehacer



Pulsando **SHIFT-U** re-hace la última operación deshacer ([Figure 6-7](#)). Pulsando **ALT-U** invoca la menú Deshacer ([Figure 6-8](#)). Allí se listan todos los pasos de Deshacer por nombre para que puedas encontrar rápidamente el camino hasta un punto conocido aceptable de tu trabajo. El menú **ALT-U** contiene también la opción *All Changes*. Esta opción es más poderosa que simplemente pulsar varias veces **UKEY** y recuperará la malla a como estaba al comienzo de la sesión de edición, incluso si has agotado todos los pasos de Deshacer.

Figure 6-8. Undo Menu



El comando Deshacer de edición puede consumir mucha memoria. Una malla de 64.000 caras y vértices puede usar más de 3MB de RAM por cada paso. Si estás trabajando en una máquina con poca memoria RAM puedes ajustar el número máximo de pasos de Deshacer en la Ventana de Preferencias de Usuario, bajo *Edit Methods*. El rango permitido va de 1 a 64. El valor por omisión es 32.

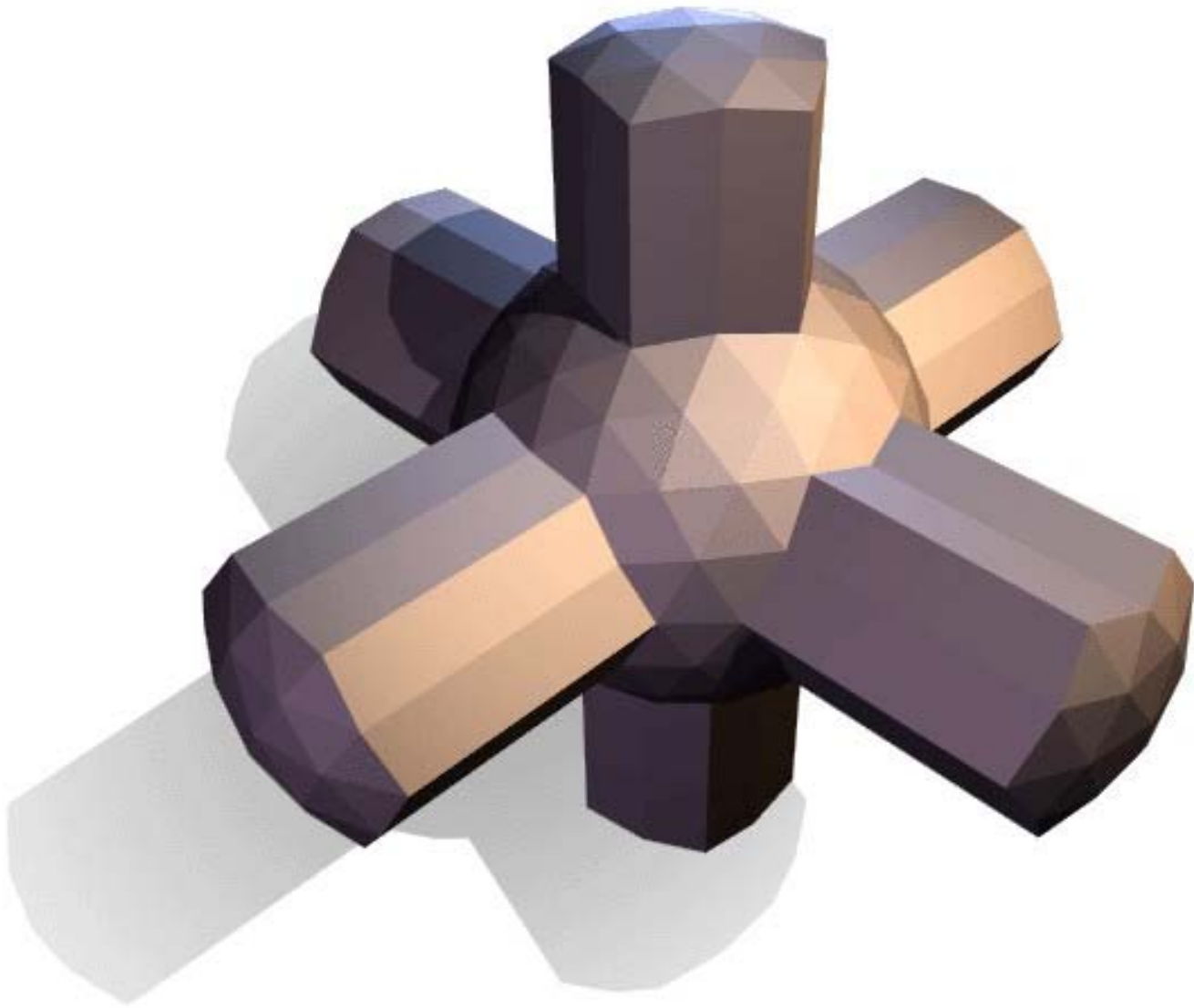
6.3. Suavizando (Smoothing)

Relevante a Blender v2.31

Como se vió en las secciones anteriores, los polígonos son una parte fundamental de Blender. La mayoría de los objetos en Blender son representados a través de polígonos, y objetos realmente curvos a menudo son aproximados mediante mallas poligonales.

Al renderizar imágenes, quizás observe que estos polígonos aparecen como una serie de pequeñas caras planas. ([Figure 6-9](#)). A veces este es el efecto deseado, pero muchas veces queremos que nuestros objetos se vean lisos y suaves. Esta sección describe cómo suavizar un objeto, y como aplicar el filtro de AutoSuavizado (AutoSmooth) para combinar rápidamente polígonos suavizados y facetados en el mismo objeto.

Figure 6-9. Objeto de prueba simple sin suavizar



Existen dos maneras de activar las características de suavizado de caras de Blender. La más fácil es suavizar o facetar un objeto entero seleccionando un objeto malla, en ModoObjeto, cambiar a los botones de Contexto de Edición (**F9**), y hacer click sobre el botón *Set Smooth* en el Panel *Link and Materials* ([Figure 6-10](#)). El botón no se queda presionado, pero fuerza a Blender a asignar el atributo de "suavizado" a cada cara de la malla. Ahora al renderizar la imagen con **F12** debería obtenerse la imagen que se ve en [Figure 6-11](#). Observe que el contorno del objeto aún se encuentra notoriamente facetado. Activar el suavizado no modifica realmente la geometría del objeto; solo cambia la forma en que se calcula el sombreado de las superficies creando la ilusión de una superficie suave y pulida.

Haga click sobre el botón *Set Solid* en el mismo Panel para recuperar el sombreado al que se ve en la [Figure 6-9](#).

Figure 6-10. Botones Set Smooth y Set Solid de la ventana de botones de edición

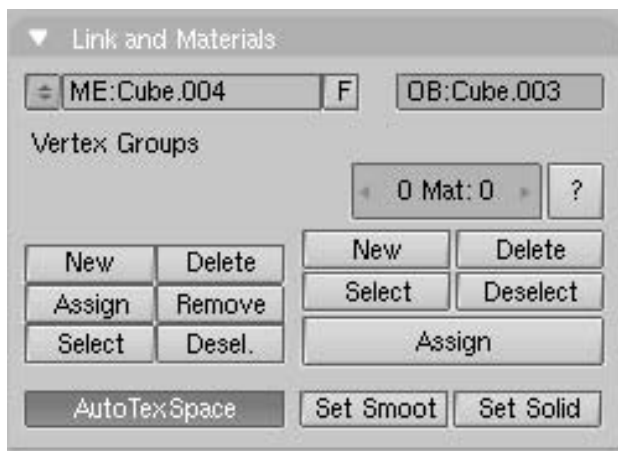
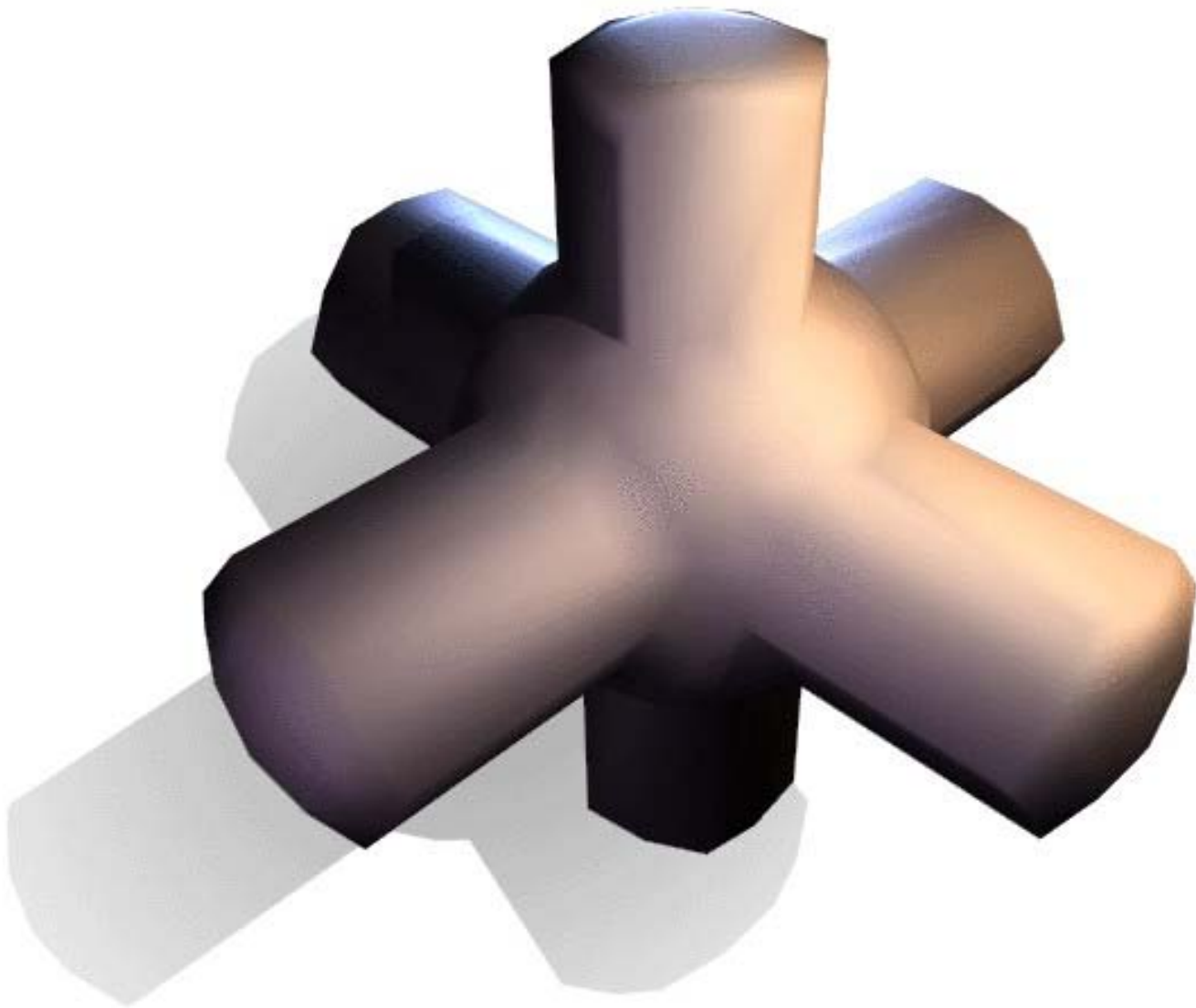


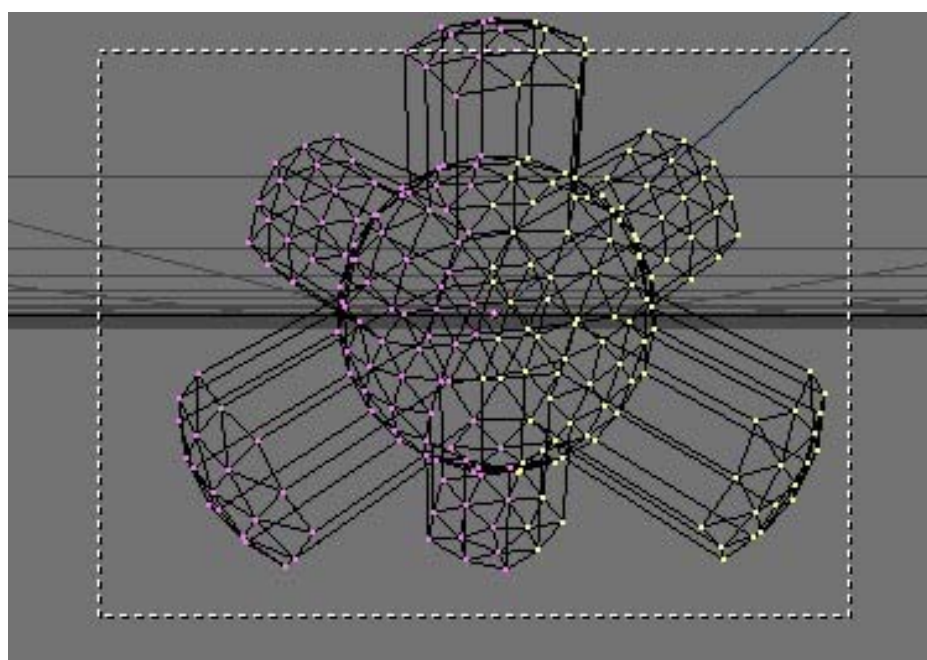
Figure 6-11. El mismo objeto pero completamente suavizado con 'Set Smooth'



Opcionalmente puede elegir qué caras suavizar ingresando en ModoEdición para el objeto con **TAB**, seleccionando las caras y haciendo click sobre el botón Set Smooth ([Figure 6-12](#)). Cuando la malla está en ModoEdición, solo las caras seleccionadas recibirán el atributo "suavizado". Puede facetar caras (eliminando el

atributo "suavizar") de la misma manera: seleccionando caras y haciendo click sobre el botón Set Solid.

Figure 6-12. Objeto en ModoEdición con algunas caras seleccionadas.



Puede ser complicado crear algunas combinaciones de caras pulidas y facetadas empleando solo las técnicas descritas. Aunque hay remedios (tales como separar grupos de caras seleccionándolos y pulsando **YKEY**), existe una manera más sencilla de combinar caras suaves y sólidas: utilizando AutoSmooth (AutoSuavizado).

Presione el botón *AutoSmooth* en el Panel *Mesh* de los botones de edición ([Figure 6-13](#)) para decirle a Blender qué caras deberán suavizarse basado en el ángulo entre caras ([Figure 6-14](#)). Los ángulos en un modelo que sean más agudos que el ángulo especificado en el botón numérico *Degr* no serán suavizados. Valores mayores produzcan caras suavizadas mientras que valores menores harán que se vea igual que una malla que fue facetada completamente.

Solo las caras que han sido puestas como suaves, serán afectadas por el AutoSuavizado. Una malla, o cualesquiera caras que hayan sido puestas como Sólidas, no cambiarán su sombreado cuando se active el AutoSuavizado. Esto permite un control adicional sobre que caras serán suavizadas y cuales no, soslayando las decisiones tomadas por el algoritmo de AutoSuavizado.

Figure 6-13. Grupo de botones AutoSmooth en la ventana de Botones de Edición.

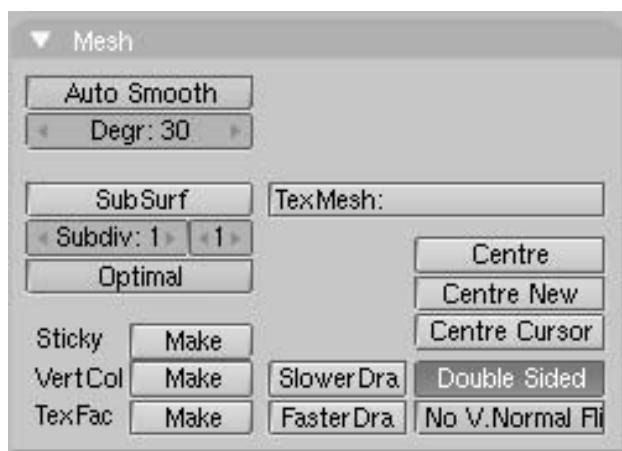
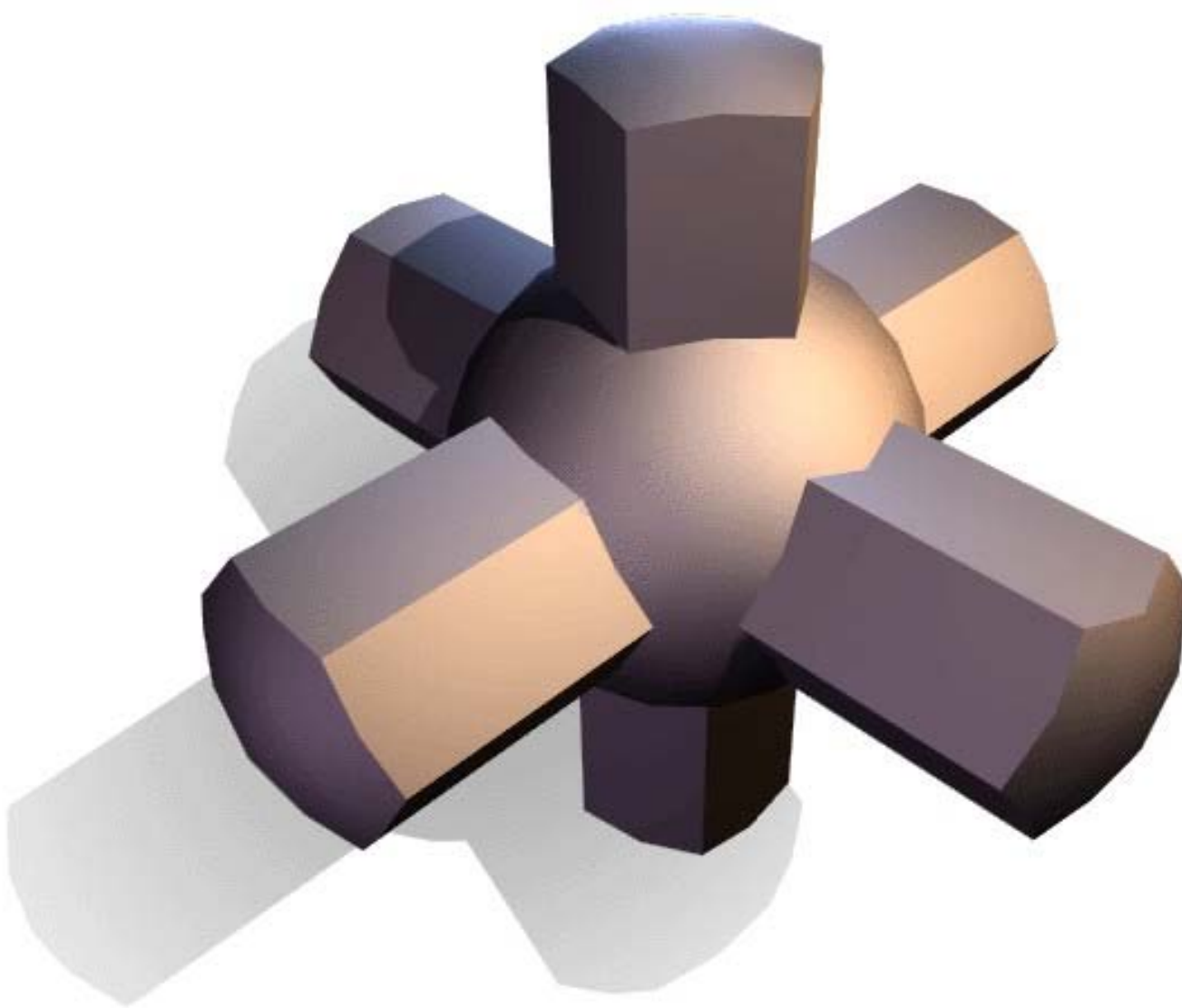


Figure 6-14. La misma prueba con AutoSuavizado activado



6.4. Extrudado

Relativo a Blender v2.31

El comando "Extrudar"(**EKEY**) es una herramienta de importancia capital cuando se trabaja con Mallas. Este comando permite crear cubos a partir de rectángulos y cilindros a partir de círculos, así como crear muy fácilmente las ramas de un árbol. Aunque el proceso es bastante intuitivo, los principios detrás la Extrusión son bastante complejos como se describe líneas abajo.

- Primero, el algoritmo determina el contorno exterior a extrudar; es decir, cuales de las aristas seleccionadas serán convertidas en caras. Por omisión, el algoritmo considera a las aristas que pertenecen a 2 o más de las caras seleccionadas como internas, y por ende no partes del contorno.
- Las aristas del contorno se convierten entonces en caras.
- Si las aristas en el contorno pertenecen solo a una cara de la malla, entonces todas las caras seleccionadas son duplicadas y conectadas a las caras recién creadas. Por ejemplo, los rectángulos se convertirán en cubos en esta etapa.
- En otros casos las caras seleccionadas son conectadas a las nuevas caras pero no duplicadas. esto evita que aparezcan caras indeseadas aprisionadas "dentro" de la malla resultante. Esta distinción es extremadamente importante ya que nos asegura la creación de volúmenes consistentemente coherentes todas las veces que se use Extrudar.
- Las aristas que no pertenezcan a ninguna cara seleccionada, es decir que formen un contorno "abierto", son duplicadas y se crea una nueva cara entre la nueva arista y la original.
- Los vértices aislados seleccionados que no pertenezcan a ninguna arista seleccionada son duplicados y una nueva arista se crean entre el vértice duplicado y el original.

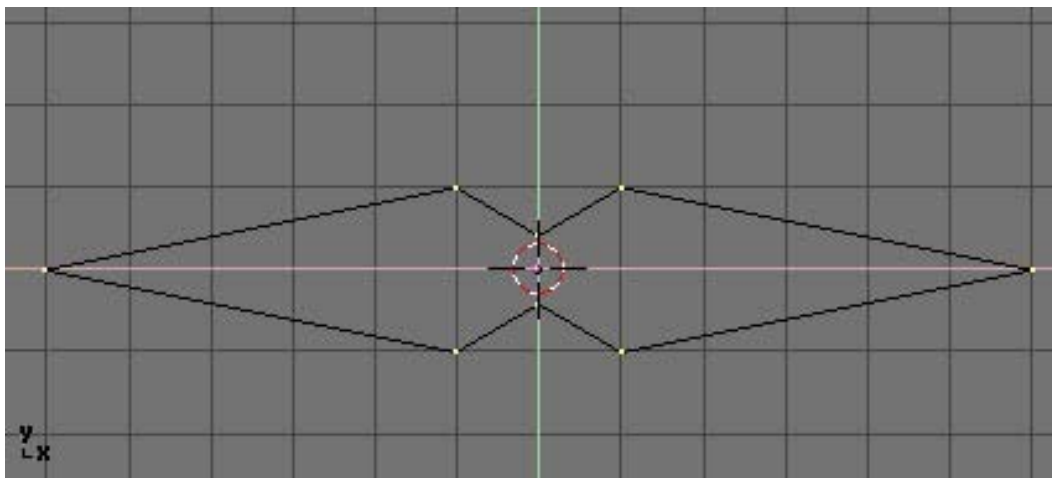
El modo Mover se activa automáticamente una vez finalizado el algoritmo Extrudar, de modo que las caras, aristas y vértices recién creados puedan moverse utilizando el ratón.

Extrudar es una de las herramientas de modelado más usadas en Blender. Es simple, directa y fácil de usar, pero aun así muy poderosa. La siguiente lección breve muestra cómo modelar una espada usando Extrudar.

6.4.1. La Hoja

1. Inicia Blender y borra el plano por defecto. En la vista superior añade un círculo malla con ocho vértices. Mueve los vértices para ajustarlos a la configuración mostrada en [Figure 6-15](#).

Figure 6-15. Circulo deformado para generar la sección transversal de la hoja.



2. Selecciona todos los vértices y redúcelos con **SKEY** para que la figura entre en dos unidades de rejilla. Cambia a la vista frontal con **NUM1**.

3. La figura que hemos creado es la base de la hoja. Usando Extrudar, crearemos la hoja en unos pocos pasos sencillos. Con todos los vértices seleccionados pulsa **EKEY**, o pulsa en el botón *Extrude* en el Panel *Mesh Tools* del Contexto de Edición (**F9** - [Figure 6-16](#)). Un diálogo emergente nos preguntará *Ok? Extrude* ([Figure 6-17](#)).

Pulsa sobre el texto o pulsa **ENTER** para confirmar, de lo contrario puedes mover el ratón fuera o presionar **ESC** para salir. Si ahora mueves el ratón verás que Blender ha duplicado los vértices, conectados a los originales con aristas y caras y que has entrado en modo Mover.

Figure 6-16. Botón Extrude en la ventana de Botones de Edición

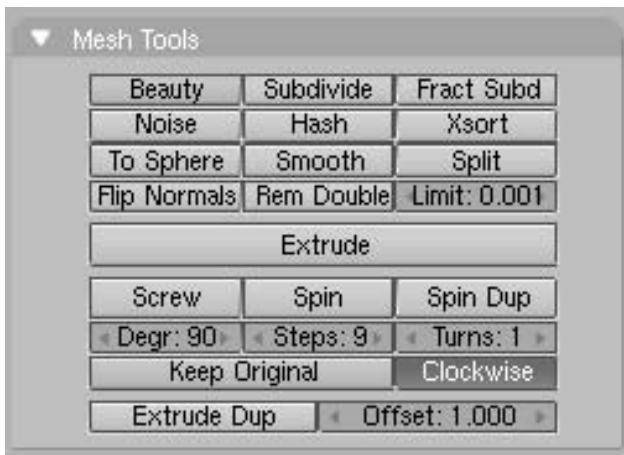
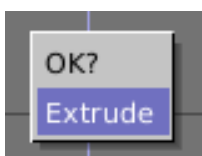
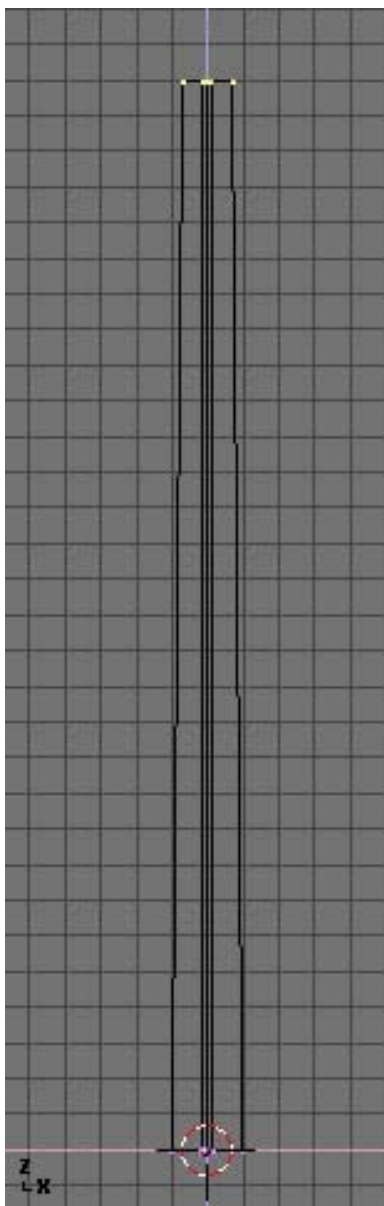


Figure 6-17. Caja de confirmación de Extrude.



4. Mueve los vértices nuevos hasta 30 unidades hacia arriba, restringiendo el movimiento con **CTRL**, luego pulsa **LMB** para confirmar su nueva posición y achicalos un poco con **SKEY** ([Figure 6-18](#)).

Figure 6-18. La Hoja

5. Presiona **EKEY** otra vez para extrudar la punta de la hoja, luego mueve los vértices 5 unidades hacia arriba. Para hacer que la hoja termine en un vértice, escala los vértices superiores hasta 0.000 (mantén pulsada **CTRL** para esto) y pulsa **WKEY**>*Remove Doubles* ([Figure 6-19](#)) o pulsa en el botón *Rem Doubles* en los Botones de Edición (**F9**). Blender informará que has eliminado siete u ocho vértices y quedará uno solo. ¡ La hoja está terminada ! ([Figure 6-20](#))

Figure 6-19. Menú Edición de Malla

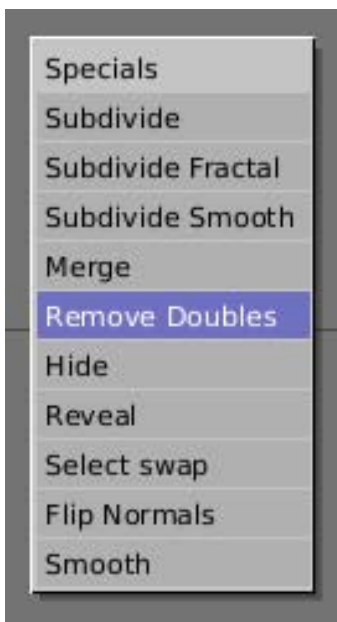
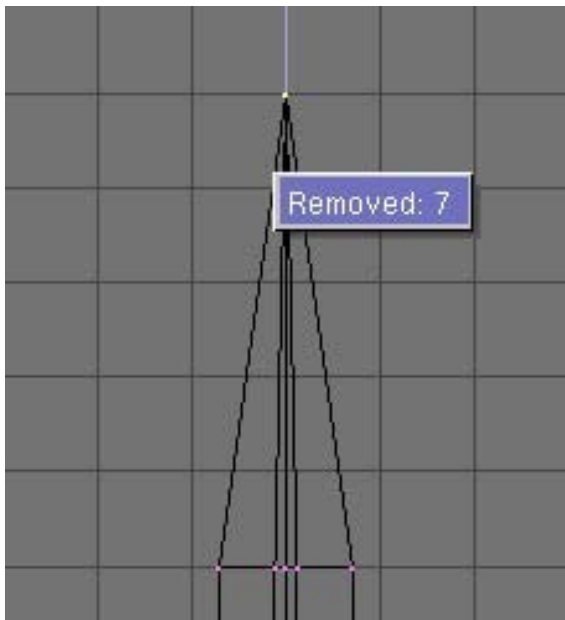


Figure 6-20. La hoja completa



6.4.2. La Empuñadura

6. Sal de Modo Edición y mueve la hoja a un lado. Agrega una UVSphere con 16 segmentos y anillos, y deselecciona todos los vértices con **AKEY**.

7. Selecciona los tres anillos de vértices superiores con **BKEY** y bórralos con **XKEY** >> *Vértices* ([Figure 6-21](#)).

Figure 6-21. La UV sphere para la empuñadura: los vértices se eliminan

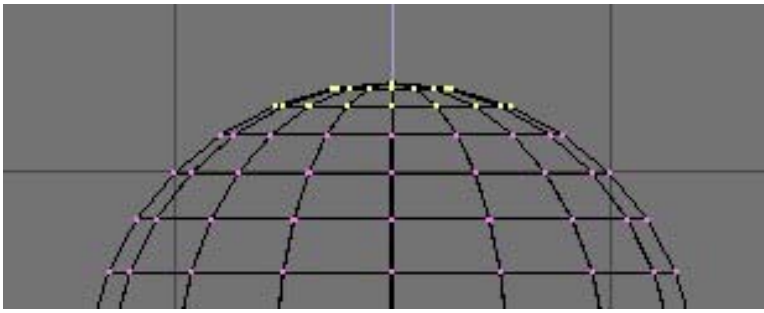
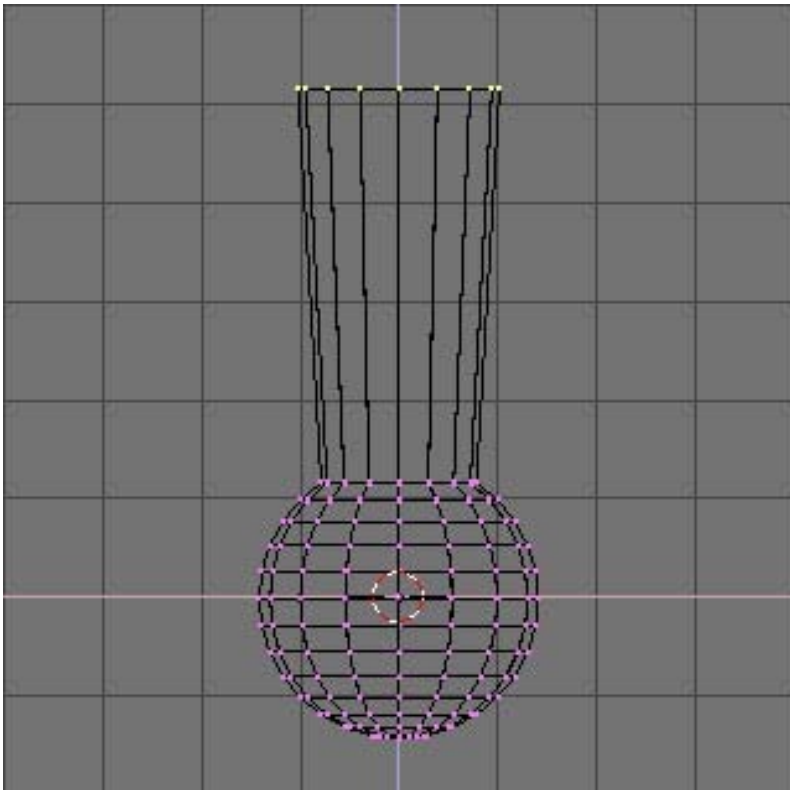


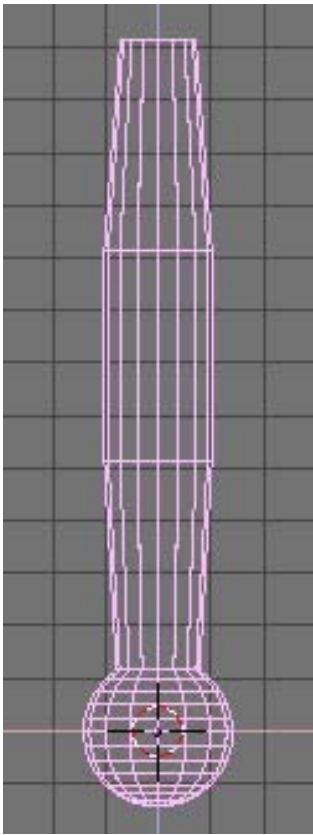
Figure 6-22. Primera extrusión de la empuñadura



9. Selecciona el anillo superior y extrudalo. Mueve hacia arriba cuatro unidades y agrandalo un poco ([Figure 6-22](#)), luego extruda dos veces y mueve cada vez otras cuatro unidades hacia arriba. Achica el ultimo anillo un poco ([Figure 6-23](#)).

10. Sal de ModoEdición y escala la empuñadura completa para darle proporciones acordes a la hoja. Ubicalo debajo de la hoja.

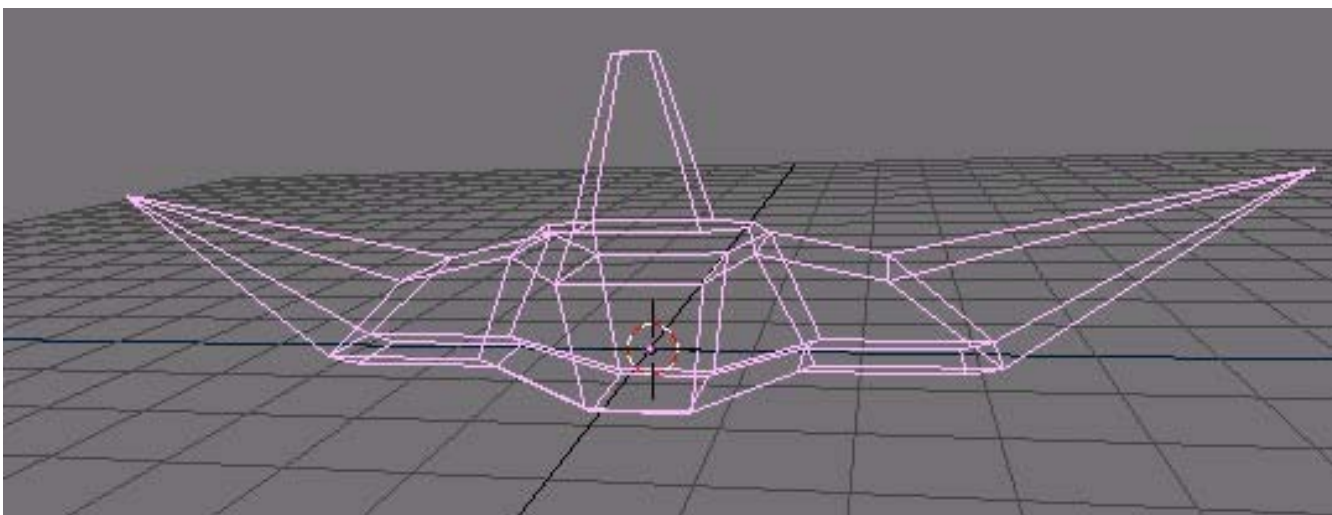
Figure 6-23. Empuñadura completa



6.4.3. La Guarnición

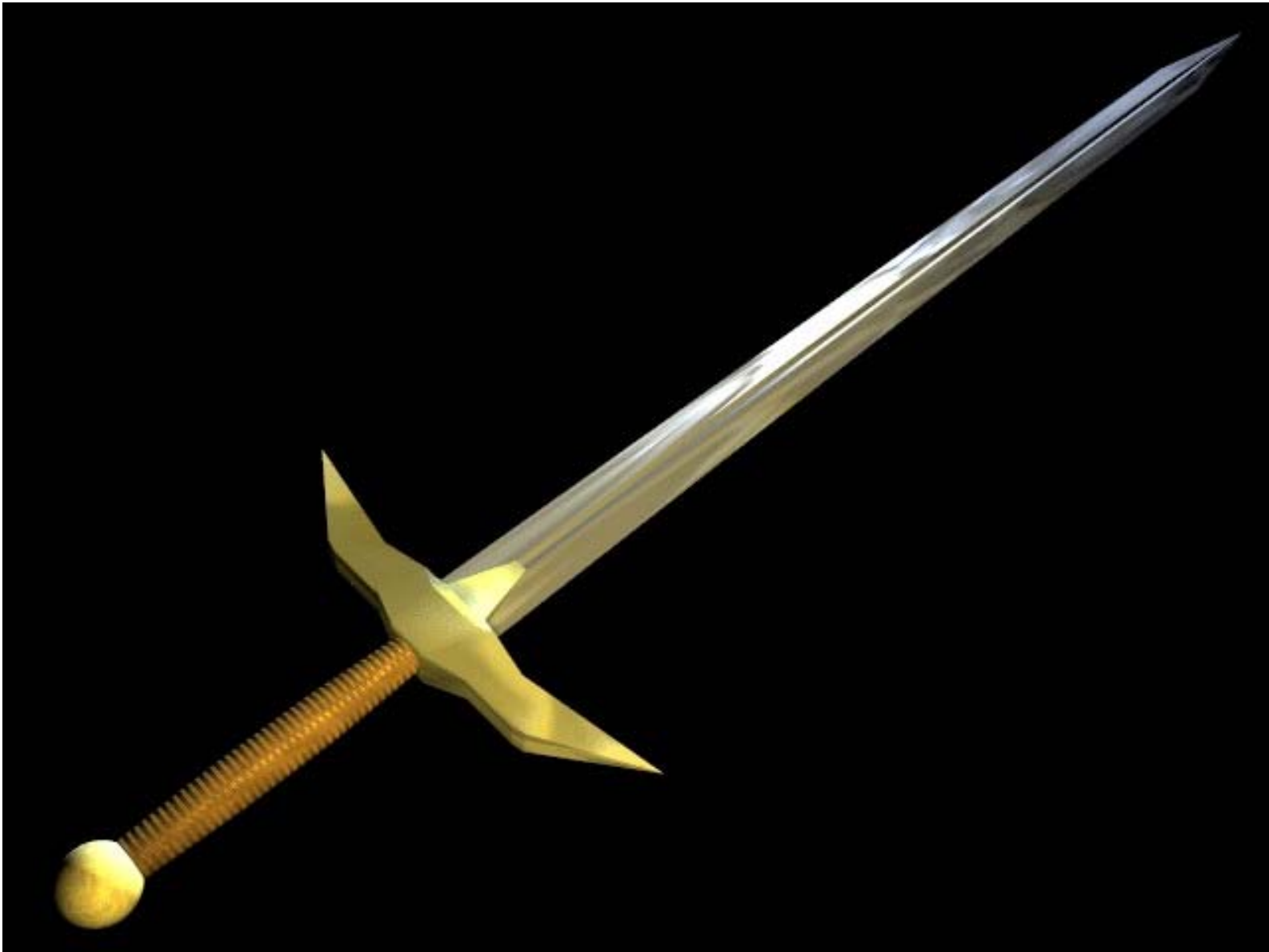
Para este momento ya te deberías haber acostumbrado a la secuencia 'extrudar>mover>escalar', así que intenta modelar una linda guarnición siguiéndola. Comienza con un cubo y extrude distintos lados un par de veces, escalando cuando haga falta. Deberías ser capaz de obtener algo similar a [Figure 6-24](#).

Figure 6-24. Guarnición Completa



Después de texturizar, la espada se ve como [Figure 6-25](#)

Figure 6-25. Espada terminada, con texturas y materiales



Como puedes ver, Extrudar es una herramienta muy potente que te permite modelar objetos relativamente complejos rápidamente (la espada completa fue modelada en menos de media hora). Sacando provecho de extrudar>mover>escalar hará tu experiencia como modelador en Blender mucho más agradable.

[Prev](#)[Suavizando \(Smoothing\)](#)[Home](#)[Up](#)[Next](#)[Giro \(Spin\) y Giro duplicado \(SpinDup\)](#)

6.5. Giro (Spin) y Giro duplicado (SpinDup)

Relevante a Blender v2.31

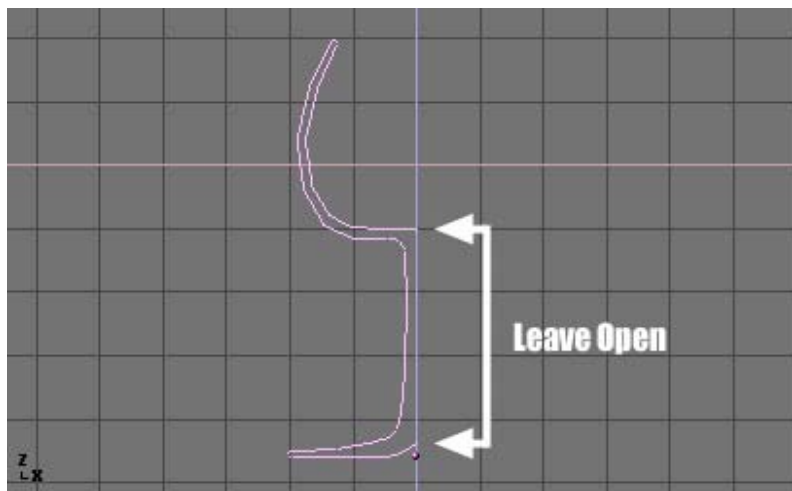
Spin y spin dup son otras dos poderosas herramientas de modelado que permiten crear fácilmente sólidos de revolución o estructuras axialmente periódicas..

6.5.1. Spin

Use la herramienta Spin para crear el tipo de objetos que generaría por medio de un lathe. (Por esta razón, esta herramienta es llamada a menudo una herramienta "lathe" o "barrido" en los textos n.)

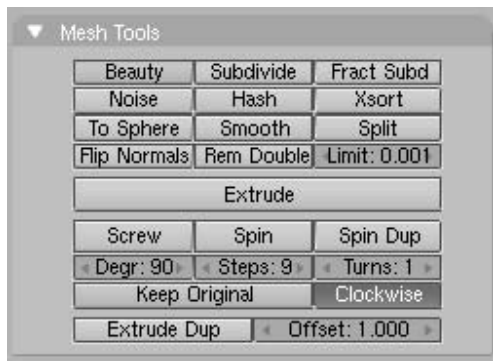
Primero, cree una malla representando el perfil de su objeto. Si está modelando un objeto hueco, es una buena idea darle un espesor al contorno. La [Figure 6-26](#) muestra el perfil para un vaso de vino que modelaremos como demostración.

Figure 6-26. Perfil del vaso



En ModoEdición, con todos los vertices seleccionados, acceda al Contexto Edición (**F9**). El botón *Degr* en el Panel *Mesh Tools* indica el número de grados que barrerá el objeto (en este caso queremos hacer un barrido de 360° completo). El botón *Steps* especifica cuantos perfiles se generarán en el barrido ([Figure 6-27](#)).

Figure 6-27. Botones de Spin

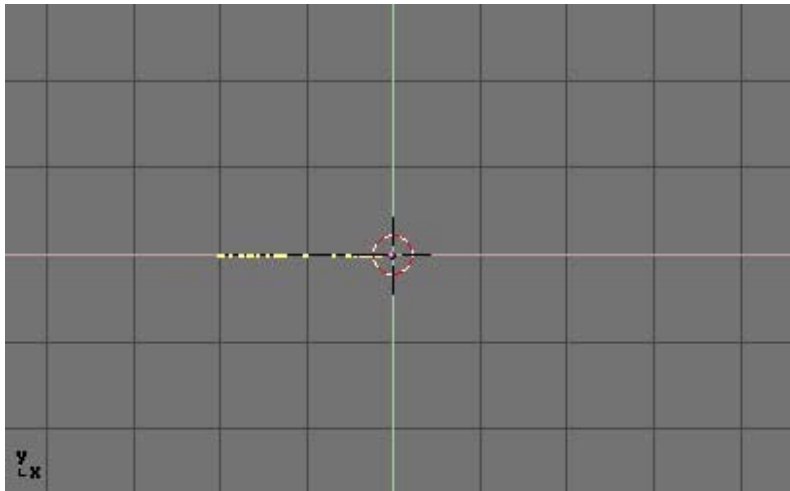


Al igual que Spin Duplicate (que se discute en la próxima sección), los efectos de Spin dependen de la posición del cursor y de qué ventana (vista) es la activa. Rotaremos el objeto alrededor del cursor en la Vista Superior. Cambie a la Vista Superior con **NUM7**.

1. Ubique el cursor en el centro del perfil seleccionando algún vértice central y ajustando el cursor esa posición con **SHIFT-S>>Curs->Sel**.

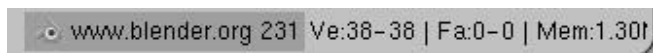
[Figure 6-28](#) muestra el perfil del vaso de vino en la Vista Superior con el cursor ubicado correctamente.

Figure 6-28. Perfil del vaso, vista superior en Modo Edición, justo antes de realizar el barrido.



Antes de continuar, observe el número de vértices del perfil. Encontrará estos datos en la barra de información en la parte superior de la interfaz de Blender ([Figure 6-29](#)).

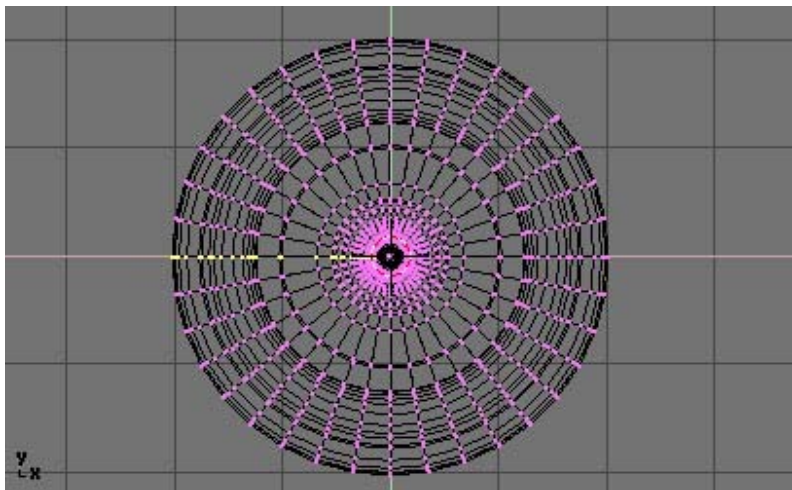
Figure 6-29. Datos de malla - número de caras y vértices.



2. Haga click en el botón "Spin". Si tiene más de una ventana abierta, el cursor cambiará a un aflecha con un signo de interrogación, y deberá hacer click sobre la ventana que tenga la Vista Superior antes de seguir. Si tiene solo una ventana abierta, el barrido se realizará inmediatamente.

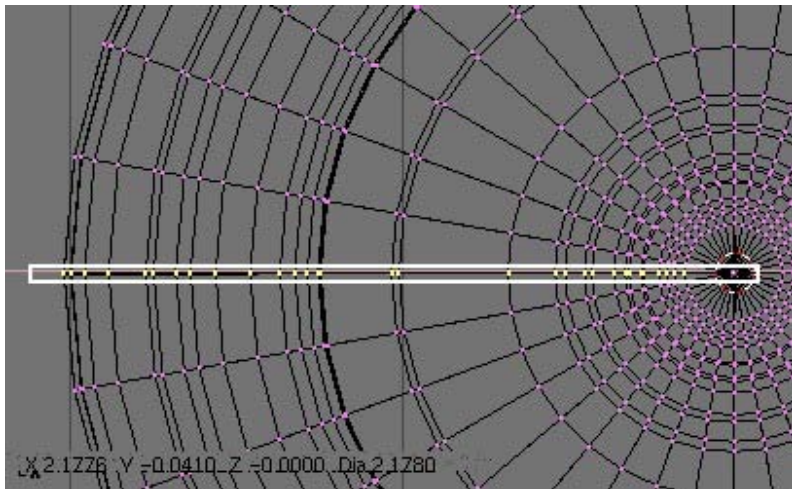
[Figure 6-30](#) muestra el resultado de un barrido exitoso.

Figure 6-30. Perfil barrido



3. La operación de spin deja vertices duplicados en el perfil. Puede seleccionar todos los vertices en la zona de la costura con la selección por área (**BKEY**) ([Figure 6-31](#)) y ejecutar un comando Remove Doubles (Borra Dobles).

Figure 6-31. Selección de vertices en la costura

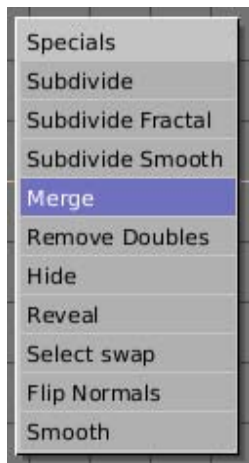


Observe la cuenta de vertices seleccionados antes y después de la operación Remove Doubles ([Figure 6-32](#)). Si todo va bien, la cuenta final de vertices (38 en este ejemplo) debería coincidir con el numero del perfil original observado en [Figure 6-29](#). De no ser así, algunos vertices fueron olvidados y deberá fundirlos manualmente. O aún peor, se han fundido demasiados vertices.

Figure 6-32. Cuenta de vertices después de eliminar dobles.**Tip Fundiendo dos vertices en uno**

Para fundir (soldar) dos vertices juntos, seleccione ambos vertices manteniendo pulsada **SHIFT** y haciendo click con **RMB** sobre ellos. Presione **SKEY** para empezar a escalar y mantenga presionada **CONTROL** mientras escala, para reducir los vertices a 0 unidades en los ejes X, Y y Z. Haga click con **LMB** para finalizar la operación de escalado y luego presione el botón *Remove Doubles* en la ventana de Botones de Edición.

Alternativamente, puede presionar **WKEY** y seleccionar *Merge* del menú emergente ([Figure 6-33](#)). Luego, en el siguiente menú, elija si el nuevo vertice se ubicará en el centro de los vertices seleccionados o en la posición del cursor. En nuestro caso, es mejor la primer opción.

Figure 6-33. El menú Merge

Todo lo que resta ahora es recalcular las normales, seleccionando todos los vertices y pulsando **CTRL-N**>>*Recalc Normals Outside*. En este momento puede abandonar ModoEdición y plaicar materiales o suavizar, poner algunas luces, un acámara y hacer un render. La [Figure 6-34](#) muestra nuestro vaso de vino terminado.

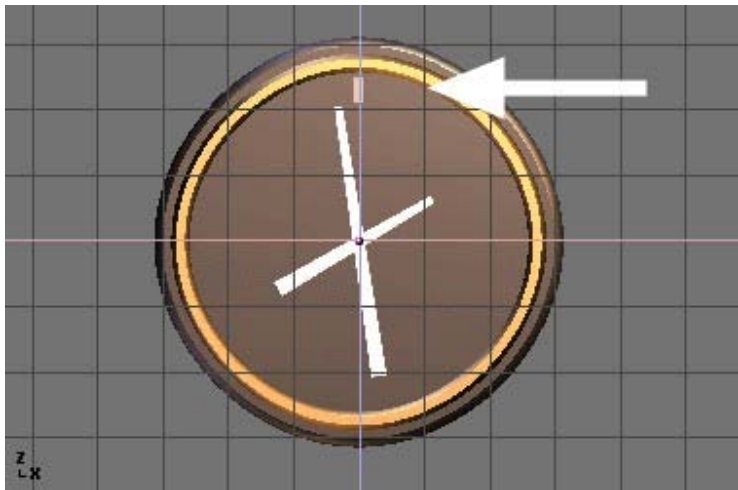
Figure 6-34. Render final de los vasos.



6.5.2. SpinDup

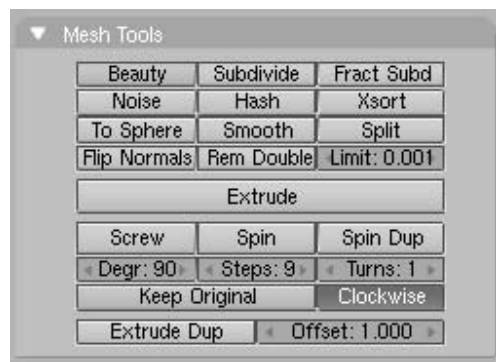
La herramienta Spin Dup es una excelente manera de obtener rápidamente una serie de copias de un objeto sobre una circunferencia. Por ejemplo, si hubiese modelado un reloj, y quisiera agregarle las marcas de las horas.

Figure 6-35. Marca de la hora indicada por la flecha



Modele solo una marca, en la posición de las 12 del reloj ([Figure 6-35](#)). Seleccione la marca y cambie a Contexto Edición con **F9**. Ponga 360 como el número de grados en el botón numérico *Degr*: en el Panel *Mesh Tools*. Queremos hacer 12 copias de nuestro objeto, así que ponga 12 en el parámetro *Steps* ([Figure 6-36](#)).

Figure 6-36. Botones Spin Dup



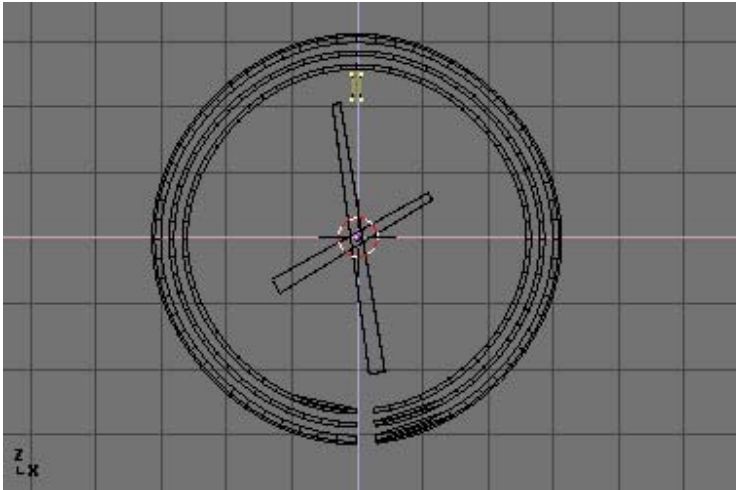
- Cambie la vista a una en la que desee rotar el objeto utilizando el teclado numérico. Tenga en cuenta que el resultado del comando *Spin Dupe* depende de la vista que esté usando al momento de presionar el botón.
- Ubique el cursor en el centro de rotación. Los objetos serán rotados alrededor de este punto.
- Seleccione el objeto que desea duplicar y entre en ModoEdición con **TAB**.
- En ModoEdición, seleccione los vertices que quiera duplicar (recuerde que puede seleccionar todos los vertices con **AKEY** o todos los vertices conectados con el vertice bajo el puntero del ratón con **LKEY**) Ver [Figure 6-37](#).



Ubicación del Cursor

Para ubicar el cursor en la posición precisa de un objeto o vertice, seleccione el objeto o vertice, y presione **SHIFT+S>>CURS>>SEL**.

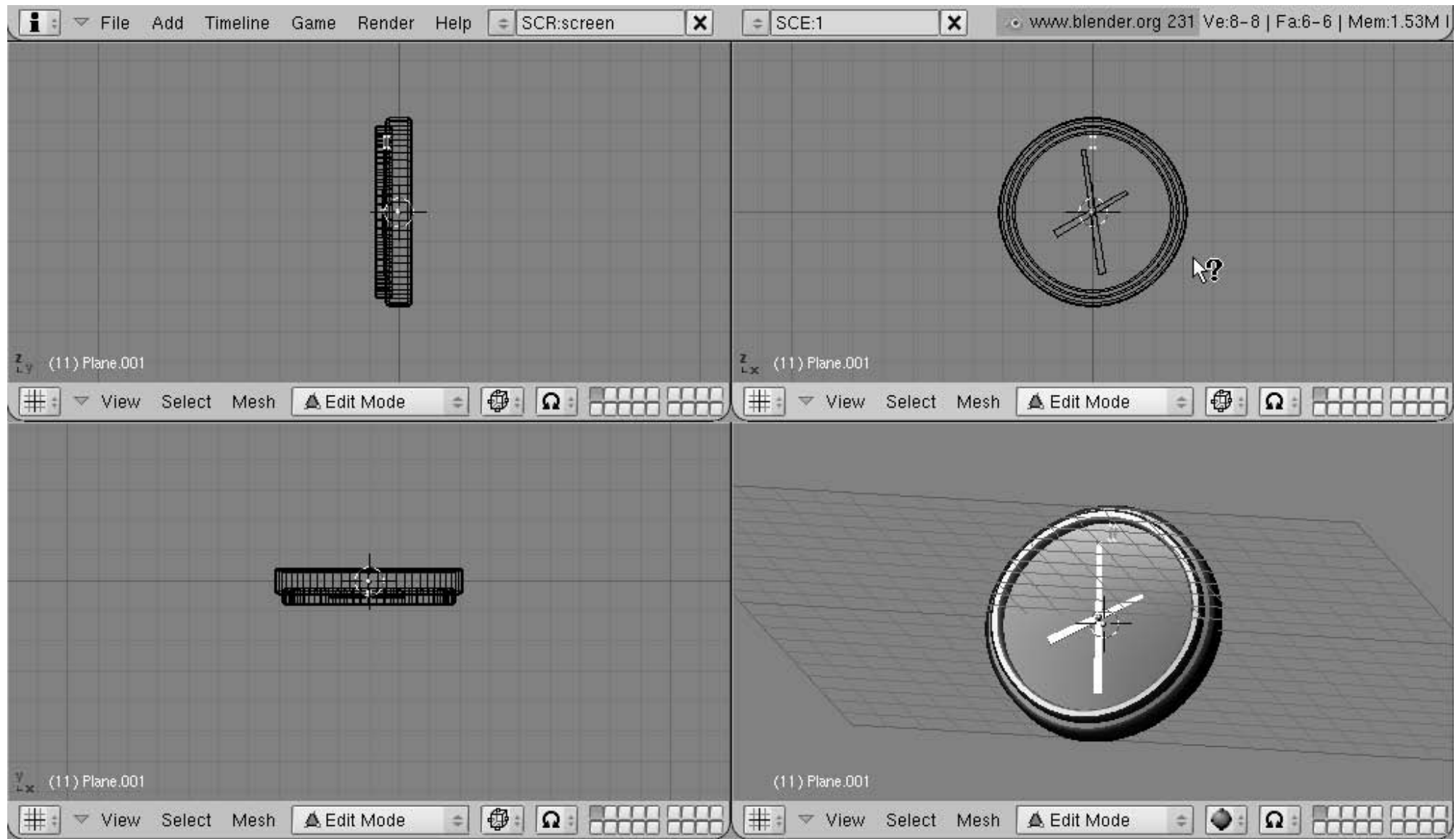
Figure 6-37. Malla seleccionada y lista para ser duplicada



- Pulse el botón Spin Dup. Si tiene más de una Ventana3D abierta, notará que el puntero del ratón cambia a una flecha con un signo de interrogación. Haga click en la ventana en la que quiera realizar su rotación. En este caso, queremos emplear la vista frontal ([Figure 6-38](#)).

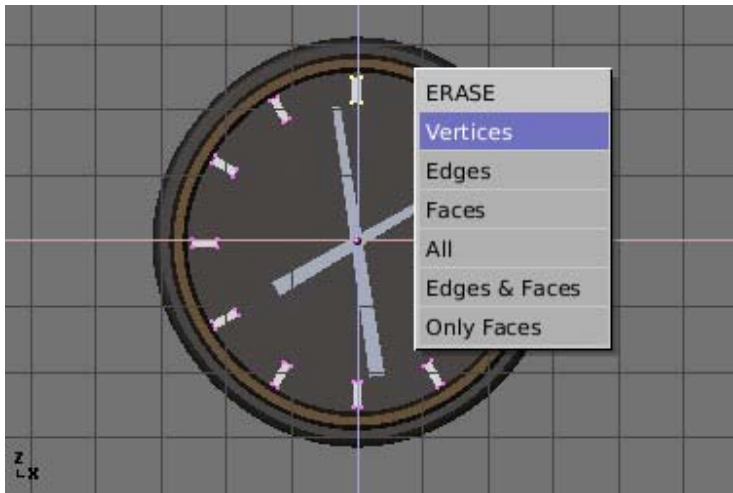
Si la vista que desea no es visible, puede descartar la flecha/signo de interrogación con **ESC** hasta cambiar a una ventana con la vista apropiada con el teclado numérico

Figure 6-38. Selección de la Vista para Spin Dup.



Cuando se duplica con revolución un objeto barriendo 360 grados, un duplicado del objeto aparece en la misma posición que el objeto original, generando geometría doble. Notará que después de hacer click sobre el botón *Spin Dup*, la geometría original permanece seleccionada. Para borrarla simplemente presione **XKEY**>>VERTICES. El objeto original será eliminado, pero la versión duplicada debajo del mismo, quedará intacta ([Figure 6-39](#)).

Figure 6-39. Eliminación de objetos duplicados



Tip Evitando duplicados

Si gusta de un poco de matemática, no necesitará preocuparse de los duplicados pues podrá evitarlos desde el comienzo. Solo haga 11 duplicados, no 12, y no sobre un giro completo de 360° , sino en un barrido de 330° (o sea $360 \cdot 11/12$). De esta manera ningún duplicado aparecerá sobre el objeto original

En general, para hacer n duplicados en 360 grados sin superposiciones, gire un objeto menos sobre $360 \cdot (n-1)/n$ grados.

[Figure 6-40](#) muestra el render final del reloj.

Figure 6-40. Render Final del Reloj.



[Prev](#)

Extrudado

[Home](#)

[Up](#)

[Next](#)

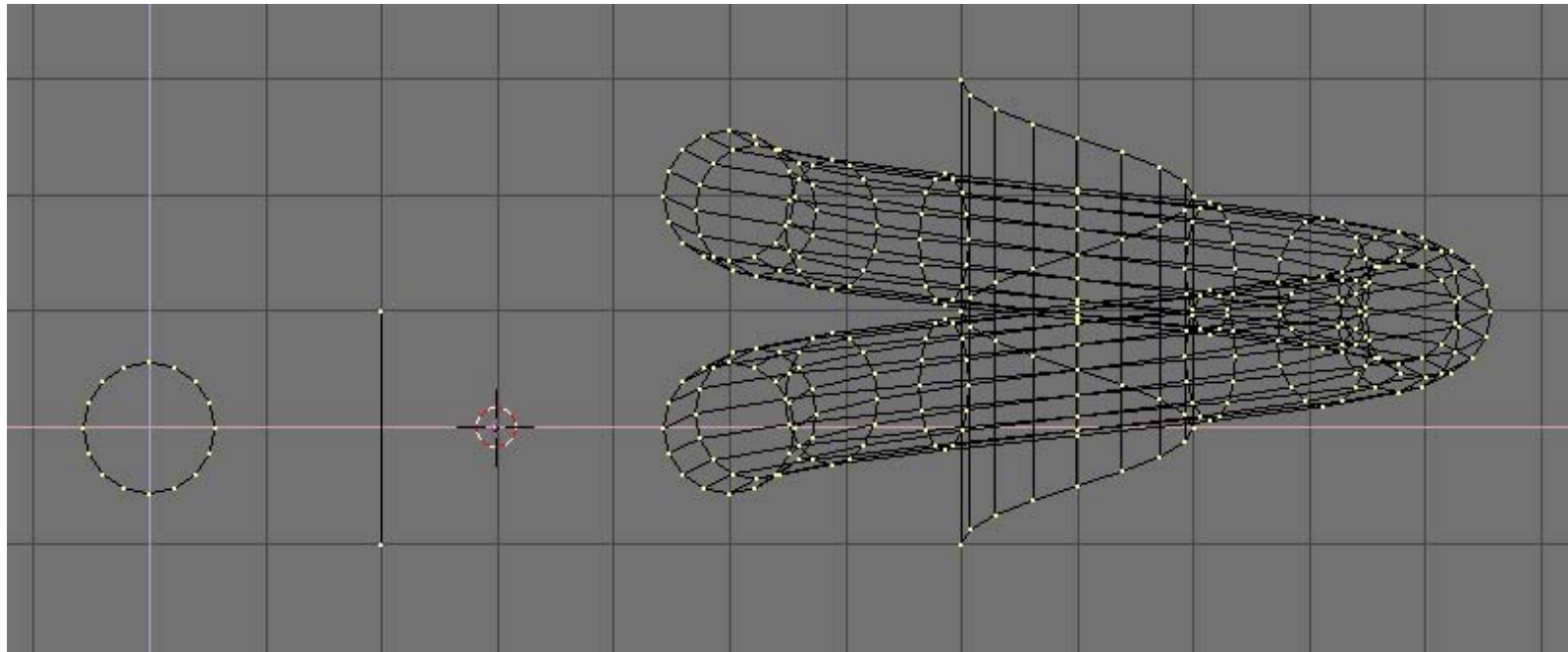
Rosca (Screw)

6.6. Rosca (Screw)

Relevante a Blender v2.31

La herramienta "Screw" o "Rosca", combina un "Spin" o "Giro" repetitivo con una traslación, para generar un objeto similar a una rosca o espiral (una helicoide). Emplea esta herramienta para crear roscas, resortes o estructuras en forma de concha.

Figure 6-41. Como hacer un resorte: antes (izqueirda) y después (derecha) de aplicar el comando Screw.



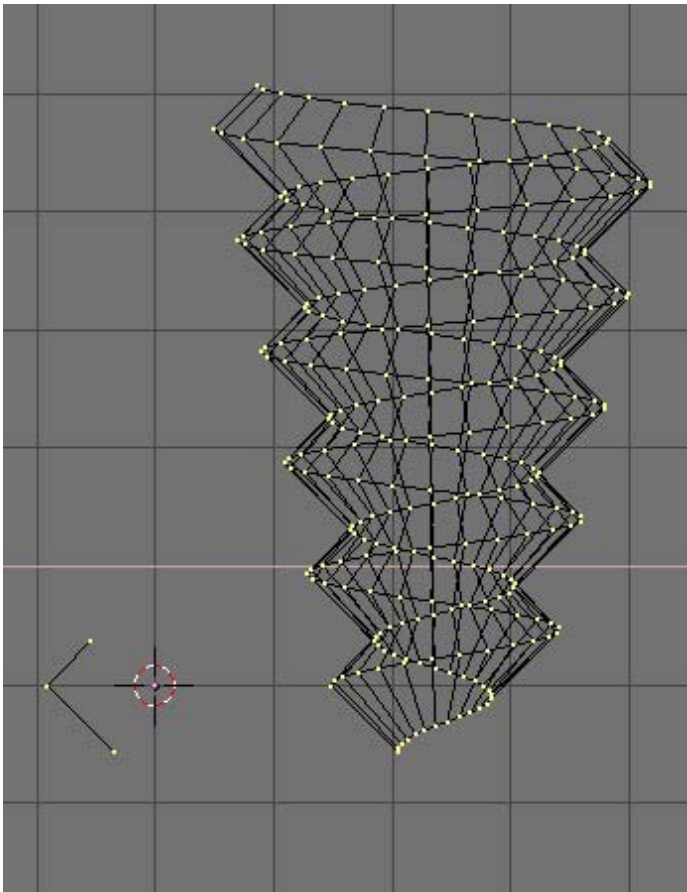
El modo de uso de la función "Screw" es estricto:

- Pon la Ventana3DSet en *Vista Frontal (NUM1)*.
- Pon el cursor3D en la posición en la que debe pasar el eje de rotación. Este eje debe ser vertical.
- Asegurate de disponer de una *poligonal abierta*. Esto puede ser tanto una arista única, como se ve en la figura, un semi círculo o cualquier otra. Solo necesitas asegurarte que haya dos extremos "libres"; dos vertices pertenecientes a una única arista enlazados luego a otro vertice. La función "Screw" localiza estos dos puntos y los usa para calcular el vector traslación que se suma al "Spin" en cada rotación ([Figure 6-41](#)). Si estos dos vertices están en el mismo lugar se crea un Spin normal. De otro modo, ¡pasan cosas interesantes!
- Selecciona todos los vertices que participarán en el "Screw".
- Asigna un valor a los Botones Numericos *Steps: (Pasos:)* y *Turns: (Vueltas:)* en el Panel *Mesh Tools* Panel. *Steps*: determina cuantas veces se repite el perfil en cada giro de 360°, mientras que *Turns*: establece el número de rotaciones de 360° completas que tendrán lugar.
- Pulsa *Screw*!

Si hubiera multiples ventanas3D, el puntero del ratón cambia a un signo de interrogación. Clickea sobre la Ventana3D en la que deberá ejecutarse el comando "Screw".

Si los dos extremos "libres" están alineados verticalmente el resultado es el visto más arriba. Si no, el vector traslación se mantiene vertical, igual a la componente vertical de vector que une estos dos vertices "libres", mientras que la componente horizontal genera un agrandamiento (o una reducción) de la rosca como se ve en [Figure 6-42](#).

Figure 6-42. Rosca creciente (derecha) obtenida con el perfil de la izquierda.



[Prev](#)

Giro (Spin) y Giro duplicado (SpinDup)

[Home](#)
[Up](#)

[Next](#)

Herramienta Curvar (Warp)

6.7. Herramienta Curvar (Warp)

Relevante a Blender v2.31

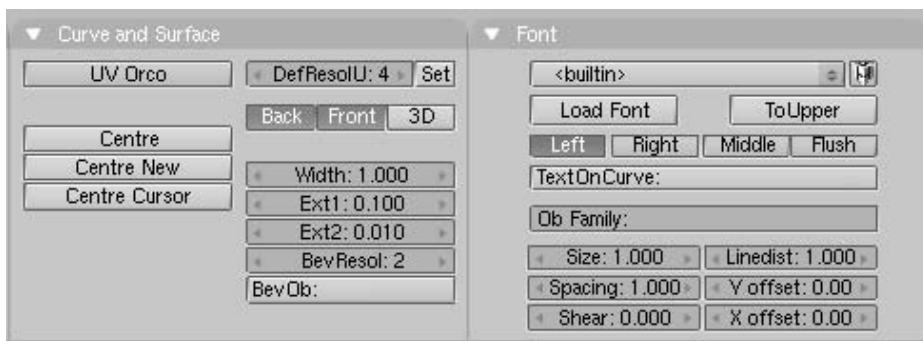
La herramienta Curvar (Warp) es una herramienta poco conocida en Blender, en parte debido a que no se la encuentra en la Ventana de Botones de Edición, y en parte porque solo es útil en casos muy específicos. No es algo que el usuario promedio de Blender vaya a utilizar todos los días.

Un bloque de texto curvado alrededor de un anillo es útil cuando se crean logotipos voladores, pero sería muy difícil de modelar sin el uso de la herramienta Curvar. Para nuestro ejemplo, curvaremos la frase "Amazingly Warped Text" alrededor de una esfera.

1. Primero cree la esfera.

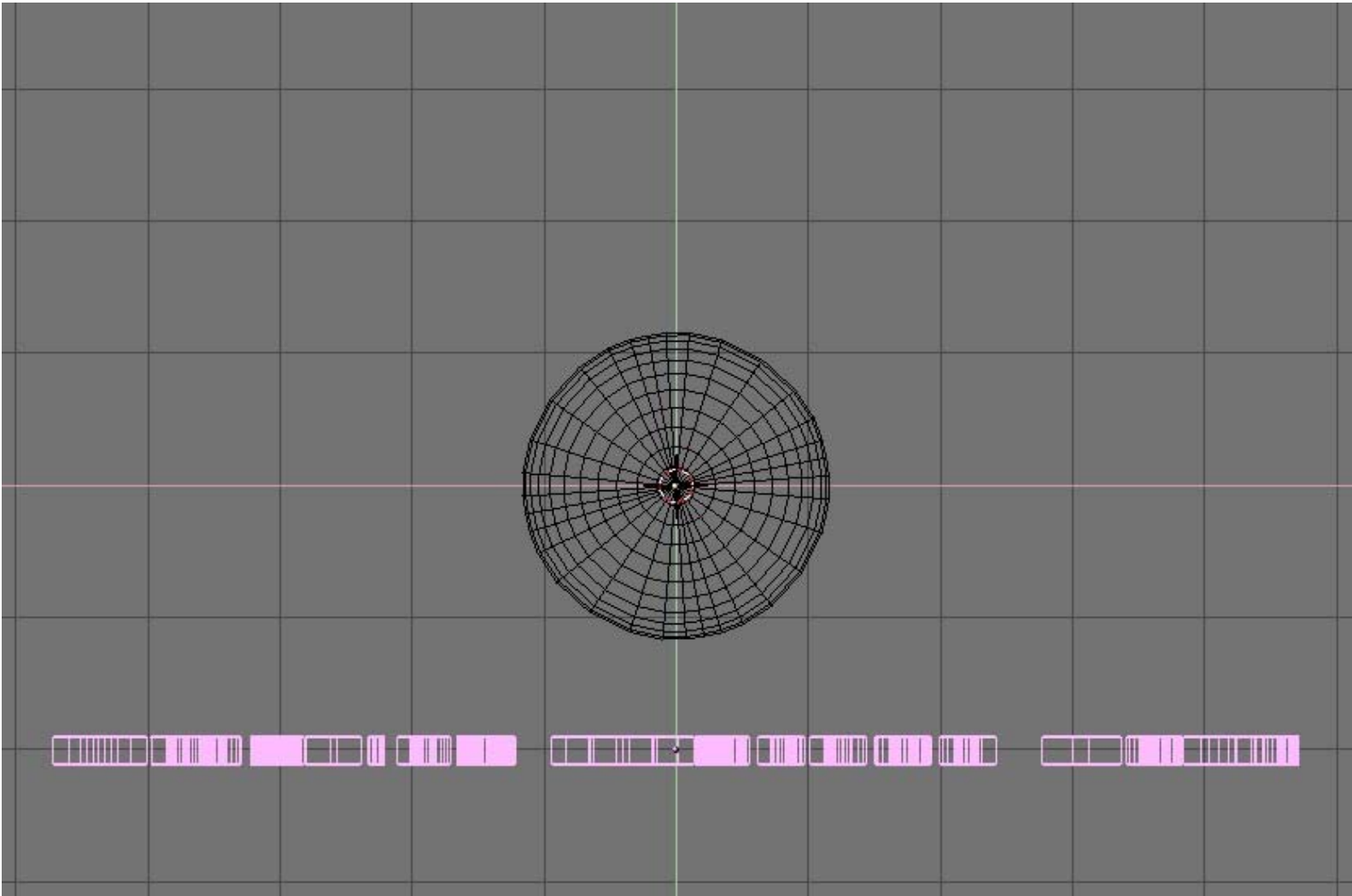
2. Luego cree el texto en VistaFrontal, en el Panel *Curve and Surface* del Contexto de Edición, ponga *Ext1* a 0.1 - haciendo el texto 3D, y ponga *Ext2* a 0.01, añadiendo un bonito chanfle al borde. Ponga *BevResol* a 1 o 2 para tener un chanfle suave y baje la resolución para que la cuenta de vertices no sea demasiado alta cuando luego subdividamos el objeto ([Figure 6-43](#) - y ver [Section 9.3](#)). Convierta el objeto a curvas, y luego a malla, (**ALT-C** dos veces) porque la herramienta Curvar no funciona sobre texto ni curvas. Subdivida la malla dos veces, de modo que la geometría cambie de forma limpiamente, sin defectos o artefactos.

Figure 6-43. Parámetros del Texto



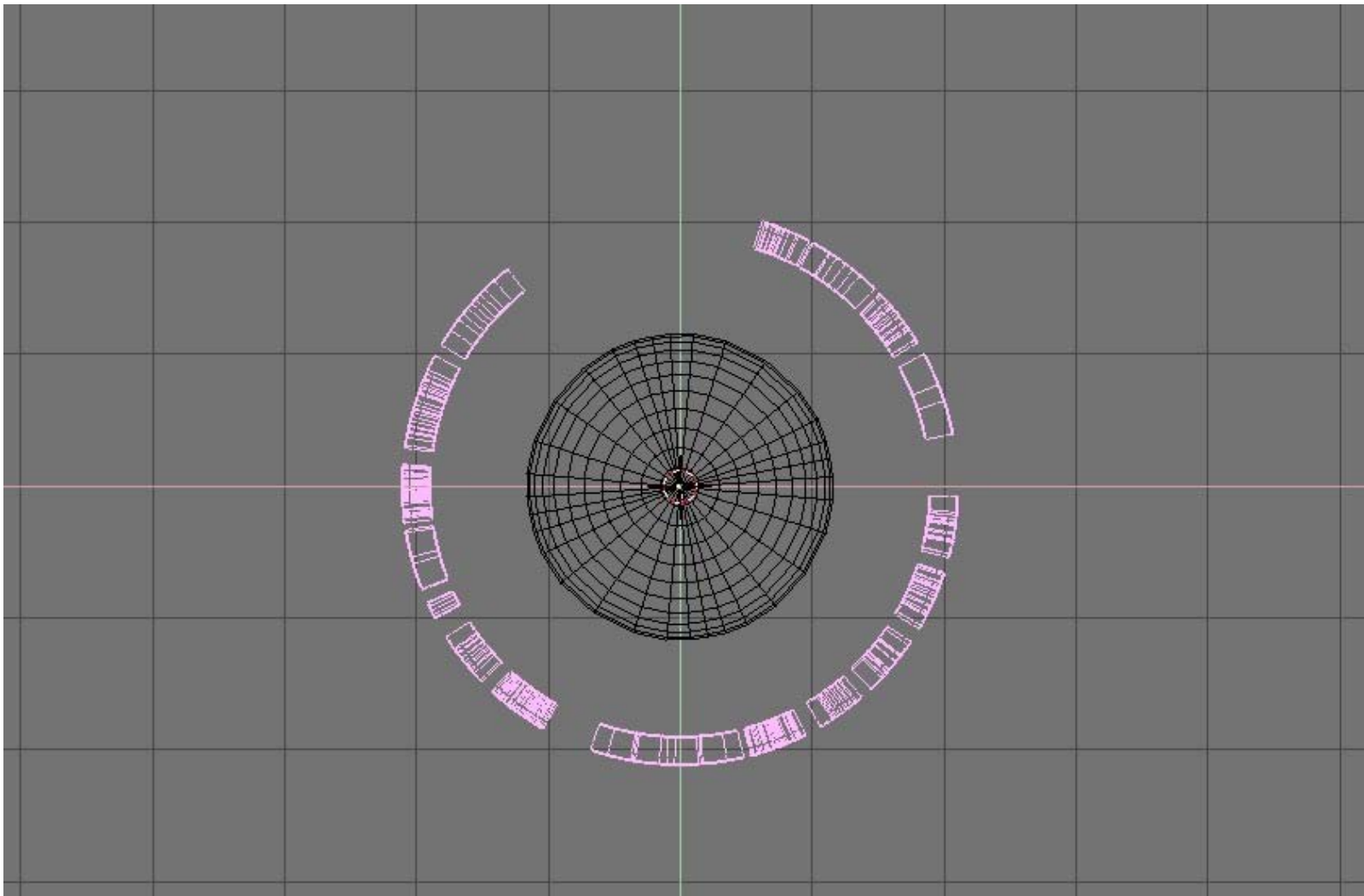
Cambie a Vista Superior y aleje la malla del cursor 3D. Esta distancia define el radio de curvatura a utilizar por Curvar. (Ver [Figure 6-44](#).)

Figure 6-44. Top view of text and sphere



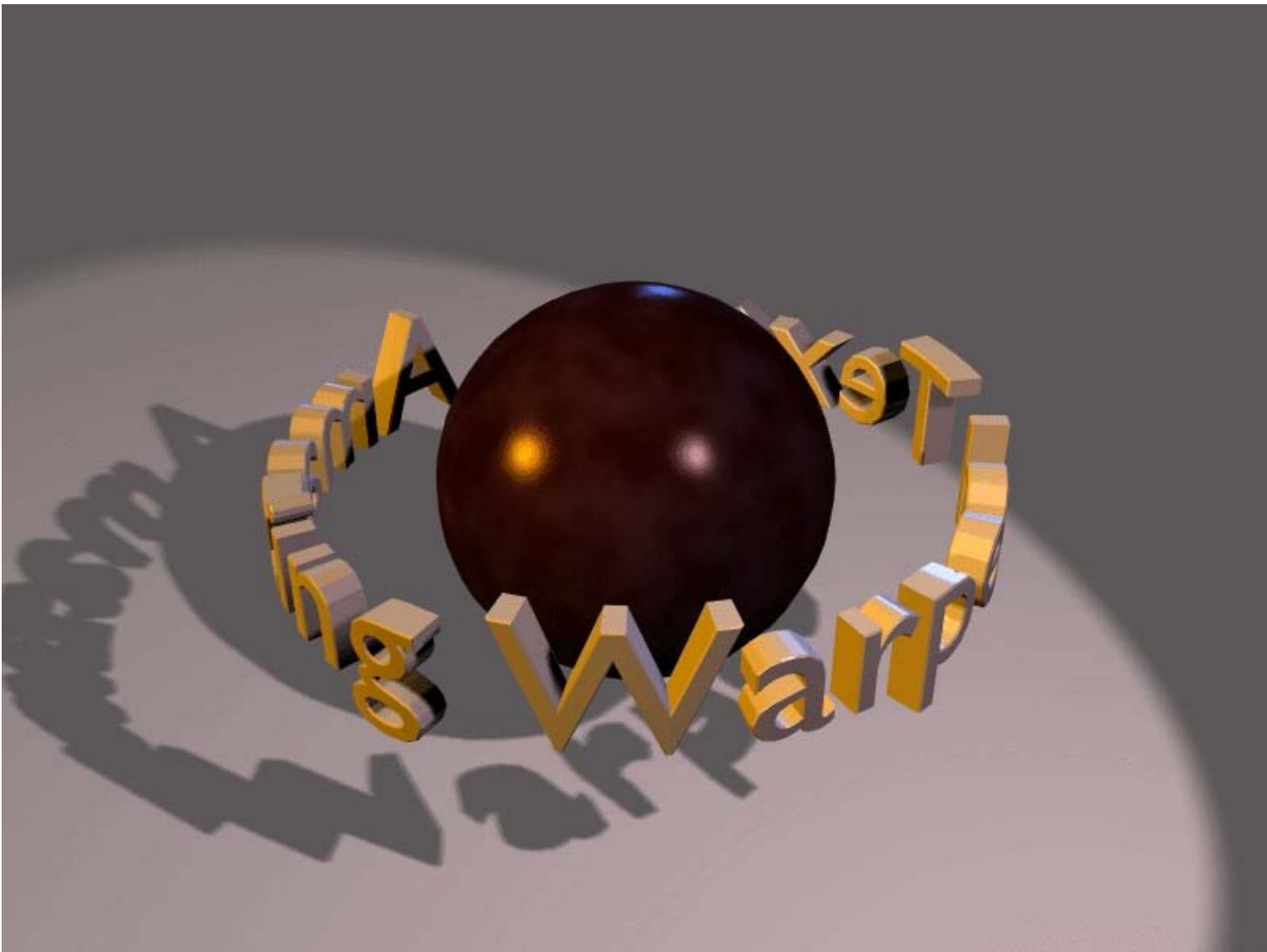
Ingrese en ModoEdición para la malla (**TAB**) y pulse **AKEY** para seleccionar todos los vertices. Presione **SHIFT-W** para activar la herramienta Curvar. Mueva el ratón arriba o abajo interactivamente para definir la cantidad de curvado. ([Figure 6-45](#)). Manteniendo pulsada **CTRL** haremos que el curvado se haga en intervalos de cinco grados.

Figure 6-45. Texto curvado



Ahora puede cambiar a vista de cámara, agregar materiales, luces y renderizar ([Figure 6-46](#)).

Figure 6-46. Render Final



6.8. Garfios de Objeto (Object Hooks)

por Kenneth Styrberg

Relativo a Blender v2.35

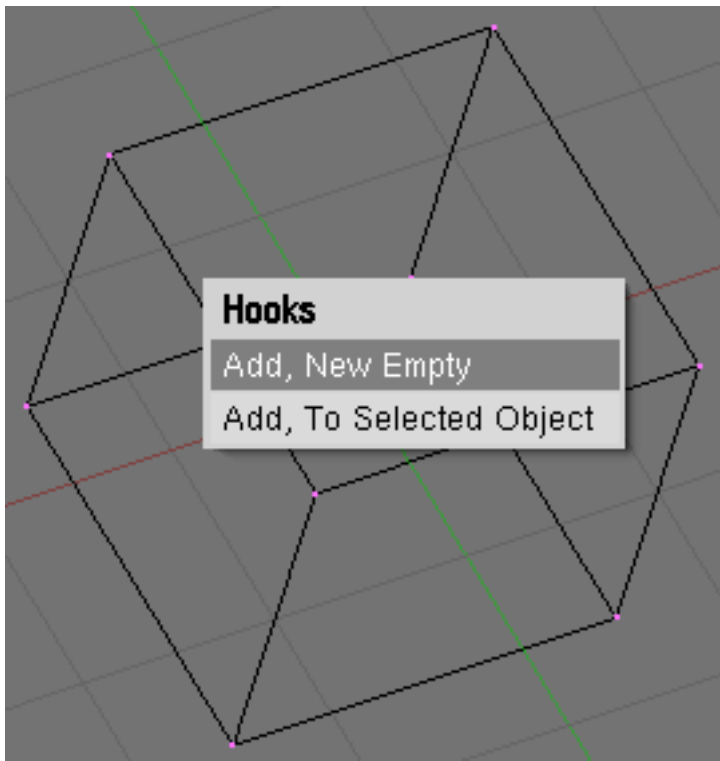
Los Garfios (Hooks) proveen acceso a nivel de objeto, a la geomtría subyacente de mallas, curvas, superficies o retículas. Un Garfio, es una característica de objeto, y es como el padre de un objeto, pero para vertices. En un objeto puede crear tantos garfios como quiera, y asignar para cada uno distintos vertices de influencia. También es posible superponer garfios, en cuyo caso se dispone de un factor de ponderación de peso para cada uno, que determina el nivel de influencia de cada garfio sobre los vertices compartidos.

Note Cuando remodela completamente algún objeto, muy probablemente también deba reasignar los garfios exitentes.

6.8.1. Agregando garfios

Como los garfios están relacionados con vertices o puntos de control, la mayoría de las opciones de edición se encuentran disponibles en el ModoEdición para mallas, curvas, superficies y retículas. Seleccione un número cualquiera de vertices y presione **CTRL-H** para acceder al menú Garfios.

Figure 6-47. Menú de garfios



Add, New Empty Crea un nuevo garfio y un nuevo objeto Empty, que será el padre de la selección y será ubicado en el centro de la misma.

Add, To Selected Object Cuando hay seleccionado otro objeto (puede hacerlo en ModoEdición con **CTRL-RMB**) se crea el nuevo grafio y se lo emparenta a ese objeto.

6.8.2. Usando garfios

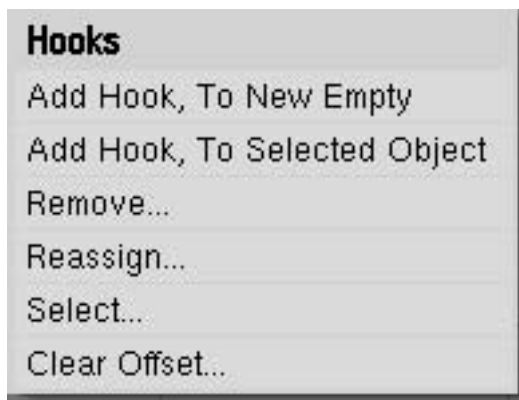
Dentro de ModoEdición los garfios se deshabilitan para permitir un modelado cómodo. Los garfios pueden usarse únicamente estando en Modo Objeto. Todas las opciones y transformaciones a nivel objeto son posibles, incluyendo el uso de jerarquías, restricciones, y animaciones de recorrido y de curvas IPO.

También puede hacer que el padre garfio sea un hijo del objeto original si no desea que las transformaciones del objeto deformen los garfios.

6.8.3. Opciones de ModoEdición

Una vez que en un objeto hay garfios disponibles, el menú **CTRL-H** nos ofrecerá opciones adicionales:

Figure 6-48. Menú extendido de garfios



Remove... (Eliminar...) Nos ofrecerá un nuevo menú con una lista de garfios para eliminar.

Reassign... (Reasignar...) Usela si desea reasignar nuevos vertices al garfio.

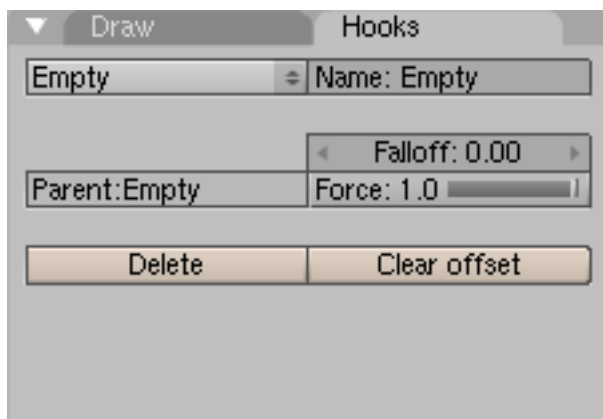
Select...(Seleccionar...) Para seleccionar los vertices de un garfio determinado.

Clear Offset...(Borrar Desplazamiento...) Neutralizará las trnasformaciones actuales del padre garfio.

6.8.4. Panel de Garfios

Puede encontrar botones para Garfios en la pestaña *Hooks* del contexto de objeto (**F7**). Aquí se le puede dar un nombre nuevo al garfio, el nombre por omisión es el nombre del padre; se le puede asignar un nuevo padre escribiendo el nombre del nuevo padre, o se le puede modificar el factor de ponderación de peso *Force*.

Figure 6-49. Panel de Garfios



Force (Fuerza) Como varios garfios pueden actuar sobre los mismos vertices, con este control puede ponderar la influencia de cada garfio. Las reglas de ponderación son:

- Si la suma total de todas las fuerzas es menor que 1.0, el resto, $1.0 - \text{Fuerzas}$, será el factor que la

posición original tiene como fuerza.

- Si la suma total de todas las "fuerzas" es mayor que 1.0, solo se utilizan las transformaciones producidas por garfios, promediando sus pesos.

Falloff Si es distinto de cero, el falloff es la distancia a la que la influencia de un garfio vale cero. Actualmente emplea un ainterpolación suave, parecida a la *Herramienta de Edición Proporcional*. (Ver [Section 7.6](#))

Delete Borra el garfio del objeto.

Clear offset Neutraliza la trnasformación actual de un garfio.

[Prev](#)

Herramienta Curvar (Warp)

[Home](#)

[Up](#)

[Next](#)

Modelado de Malla avanzado

Chapter 7. Modelado de Malla avanzado

Blender proporciona varias características de modelado avanzado de malla, mayormente orientadas a un manejo más sencillo de mallas complejas o a un manejo más económico del modelado de superficies suaves con un bajo número de vértices.

7.1. Subdivisión de Superficies Catmull-Clark

Relativo a Blender v2.31

Con cualquier malla regular como base, Blender puede calcular una superficie suavizada en el momento, mientras se modela o al hacer el render, usando Catmull-Clark Subdivision Surfaces o, abreviando, *SubSurf*. SubSurf es un algoritmo matemático para calcular una subdivisión "suavizada" de la malla. Esto permite un modelado de Malla de alta resolución sin la necesidad de guardar y mantener enormes cantidades de datos. Esta técnica también permite darle una apariencia más "orgánica" a los modelos.

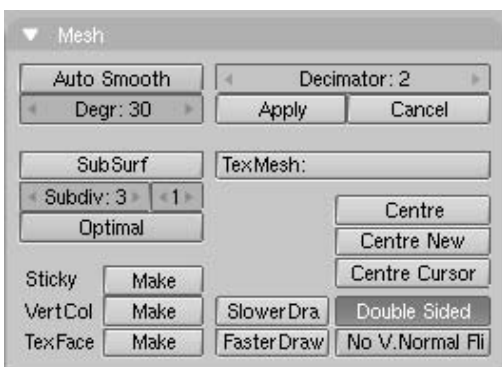
En realidad, una malla con SubSurf y una superficie NURBS tienen muchos puntos en común, puesto que ambos se basan en una "malla" de bajos polígonos para definir una superficie de "alta resolución". Pero también hay notables diferencias:

- NURBS permite un control más preciso de la superficie, ya que se pueden definir "pesos" independientes para cada punto de control de la malla de control. En una malla con SubSurf no se pueden controlar los "pesos".
- SubSurf posee una forma de modelado más flexible. Como una SubSurf es una operación matemática funcionando sobre una malla, se pueden usar todas las técnicas de modelado descritas en este capítulo sobre la malla. Existen más técnicas, que son mucho más flexibles, que las disponibles para NURBS control polygons.

SubSurf es una opción de la malla, que se activa en el Editing Context *Mesh* Panel (**F9** - [Figure 7-1](#)). Los Num Buttons debajo de este definen, a la izquierda, la resolución (o nivel) de subdivisión para propósitos de visualización 3D; el de la derecha, la resolución al momento del render. También se puede usar **SHIFT-O** estando en ModoObjeto. Esto activa o desactiva SubSurf. El nivel de SubSurf también puede ser controlado mediante **CTRL-1** a **CTRL-4**, pero esto sólo afecta el nivel de subdivisiones en la visualización.

Ya que los cálculos de SubSurf son realizados en tiempo real, mientras se modela y al momento del render, y como utilizan el CPU de manera intensiva, es, en general, una buena costumbre mantener el nivel de SubSurf bajo (distinto de cero) mientras se modela, y alto al momento del render.

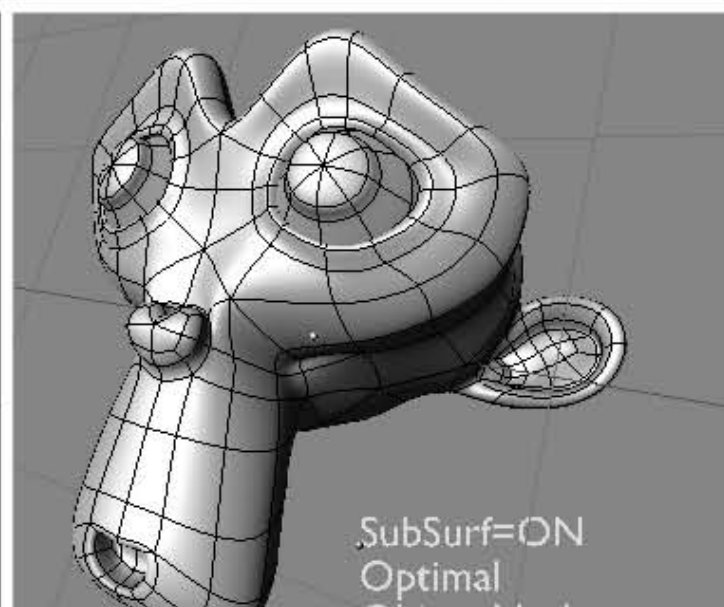
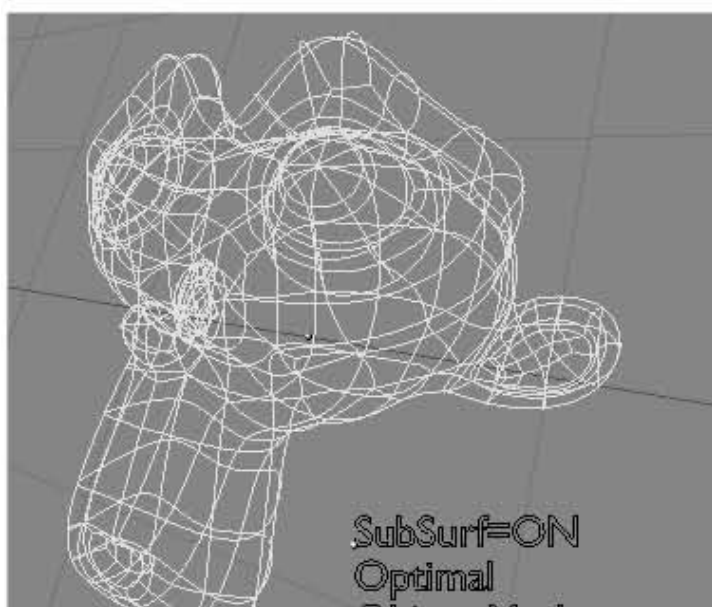
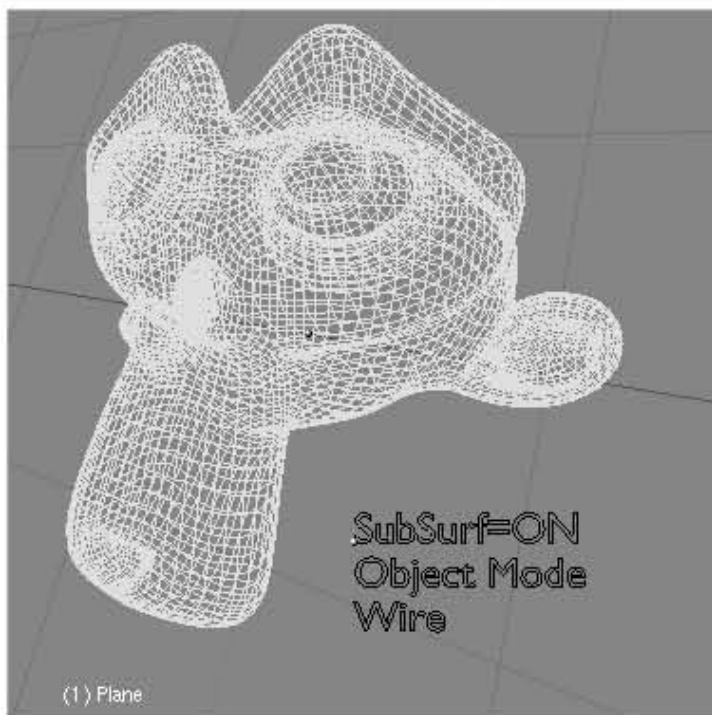
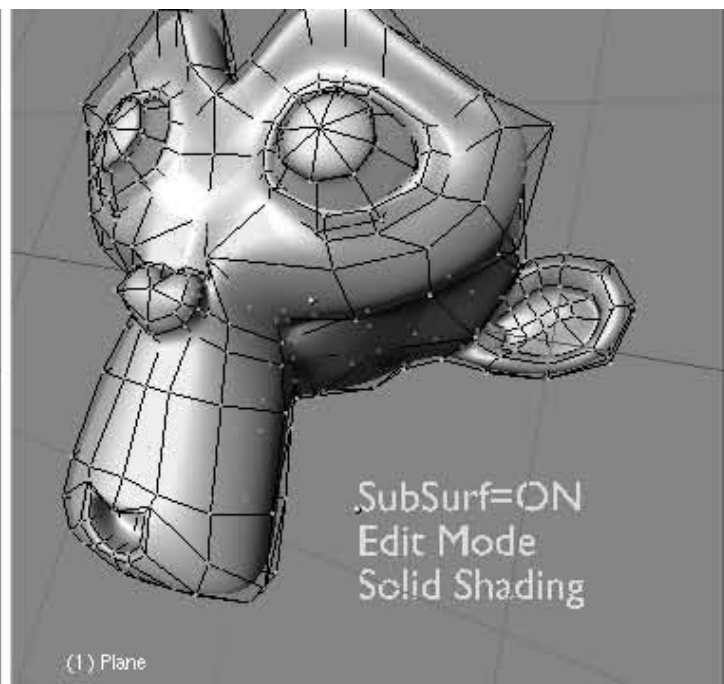
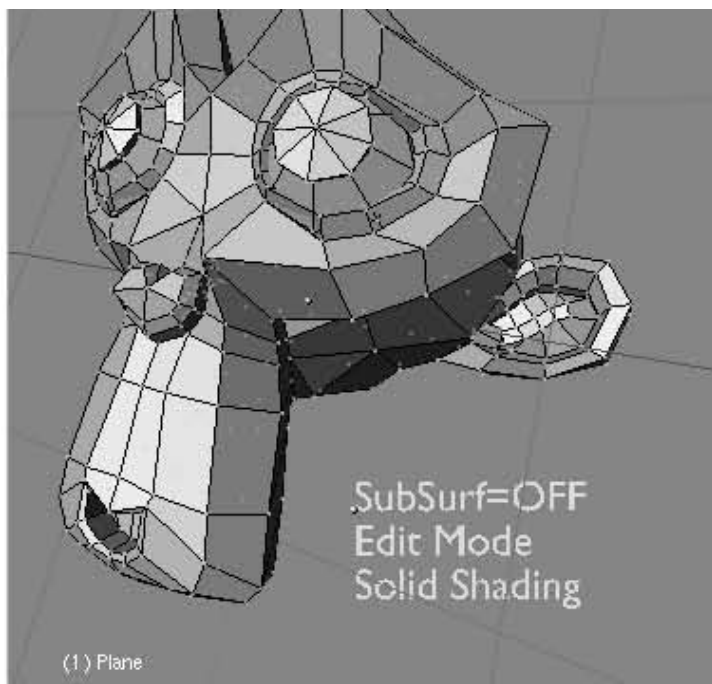
Figure 7-1. Botones SubSurf

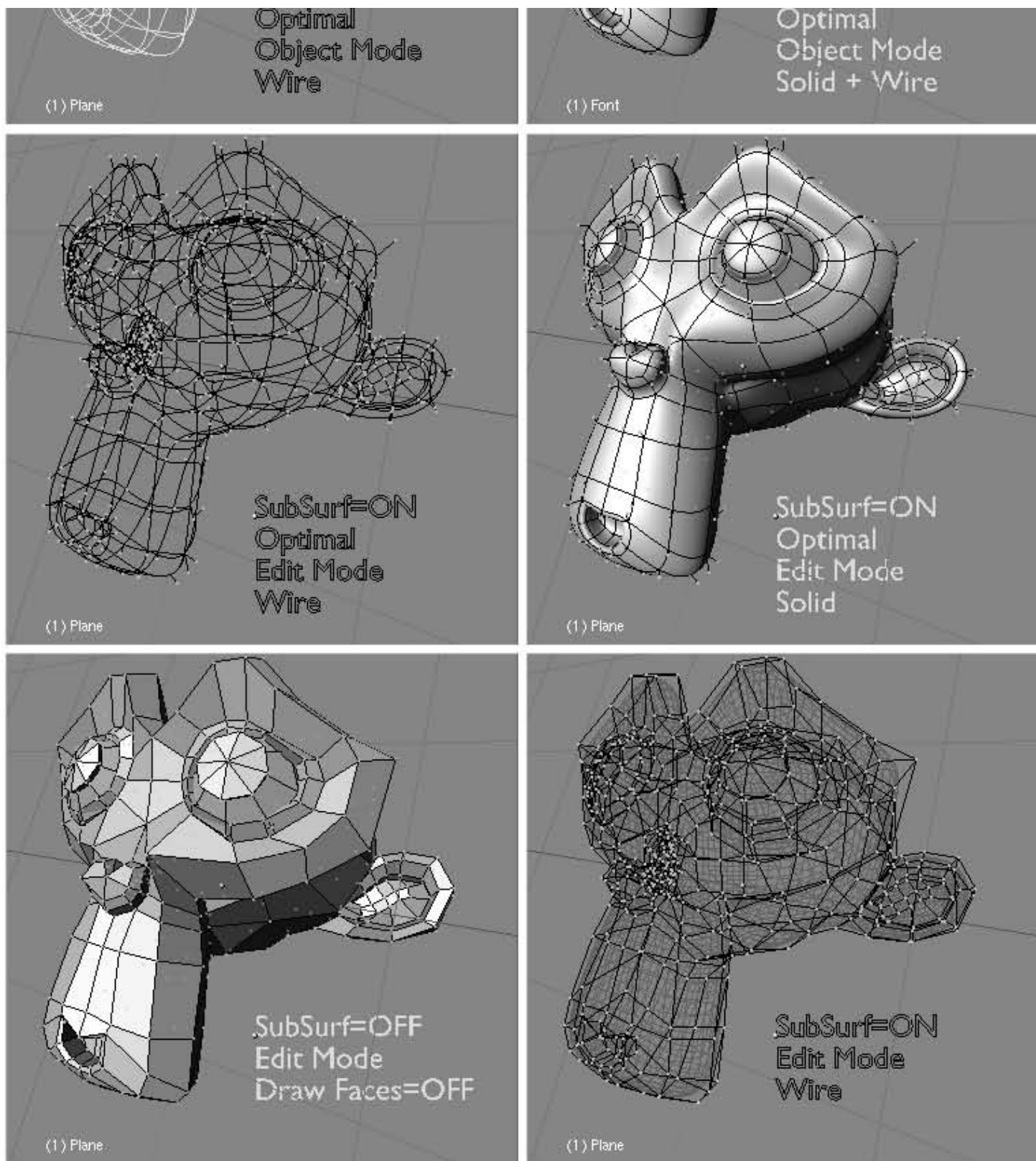


A partir de la versión 2.3, Blender tiene un nuevo botón relacionado a SubSurf: *Optimal*. Este botón cambia la forma en que las mallas con SubSurf son dibujadas y puede ser de gran ayuda al momento de modelar. [Figure 7-2](#) muestra una serie de imágenes de diferentes combinaciones en Suzanne's Mesh.

Figure 7-2. SubSurfed Suzanne.



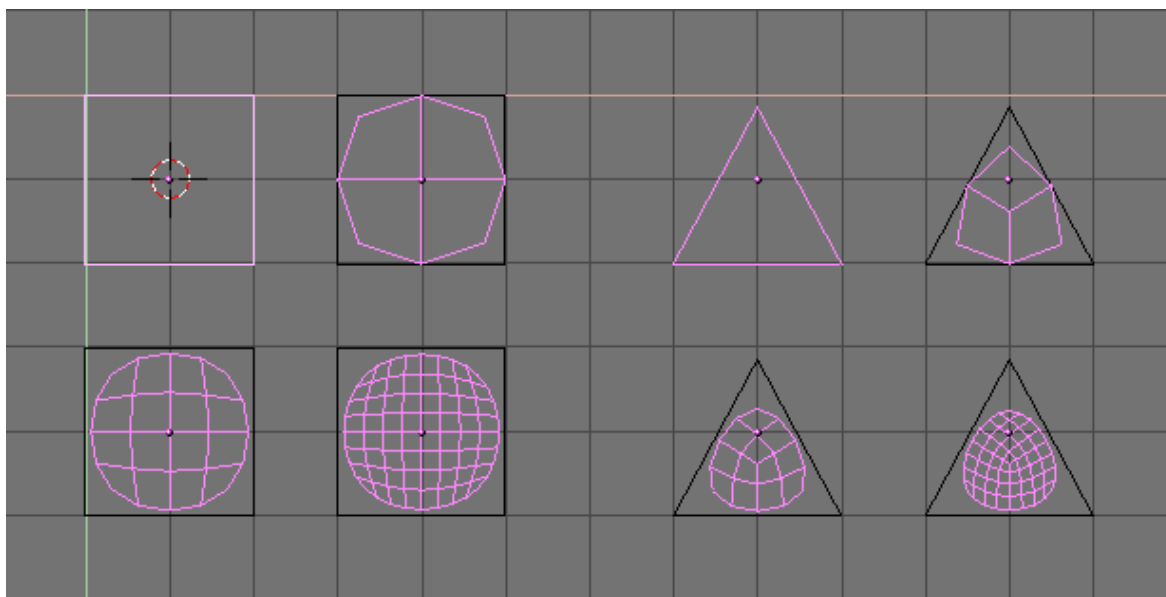




[Figure 7-3](#) muestra los niveles 0, 1, 2 y 3 de SubSurf aplicados a una cara cuadrada o a una cara triangular. Este tipo de subdivisión es realizada, en una malla genérica, en *cada* cara cuadrada o triangular.

Es evidente como cada cara cuadrada simple produce 4^n caras en la superficie subdividida. n es el nivel de SubSurf, o su resolución, mientras que cada cara triangular produce $3 \cdot 4^{(n-1)}$ caras nuevas([Figure 7-3](#)). Este aumento dramático en la cantidad de caras (y vértices) produce una baja en la velocidad de todas las acciones de Edición y Render, y hace necesario usar un nivel de SubSurf más bajo durante el proceso de Edición, que durante el Render.

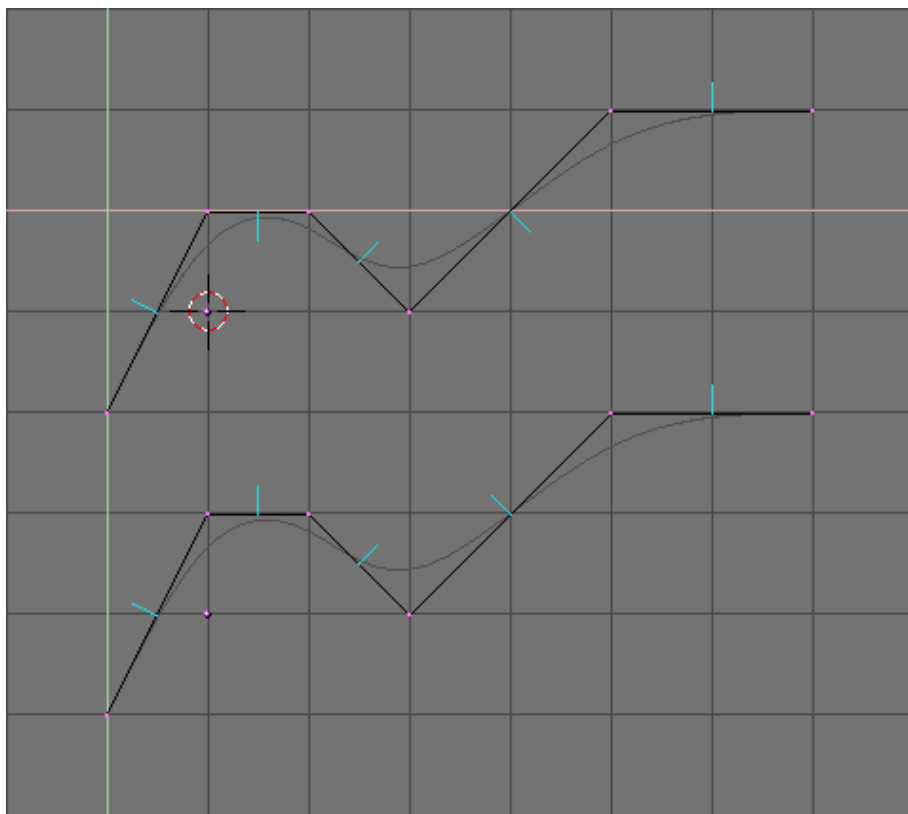
Figure 7-3. SubSurf de un cuadrado simple y caras rectangulares..



El sistema de subdivisión de Blender está basado en el algoritmo Catmull-Clarke. Esto produce una malla suavizada agradable, pero cada cara "subdividida", o sea, cada pequeña cara creada por el algoritmo a partir de una cara simple de la malla original, comparte la orientación de la normal de la correspondiente cara original.

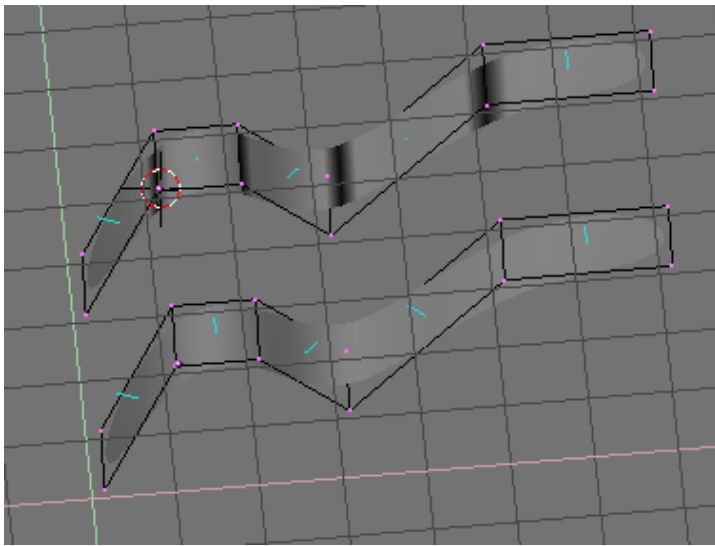
Esto no es un problema para la forma en sí, como muestra [Figure 7-4](#), pero sí lo es en la etapa de render y en el modo de visualización sólido, donde cambios abruptos de las normales pueden producir desagradables líneas negras ([Figure 7-5](#)).

Figure 7-4. Vista lateral de mallas subdivididas. Con normales al azar (arriba) y con normales coherentes (coherent normals) (abajo)



Utilizando el comando **CTRL+N** en ModoEdición, con todos los vértices seleccionados, para hacer que Blender recalcule las normales.

Figure 7-5. Vista de Sólido de mallas Subdivididas con normales inconsistentes (inconsistent normals) (arriba) y con normales consistentes (consistent normals) (abajo).

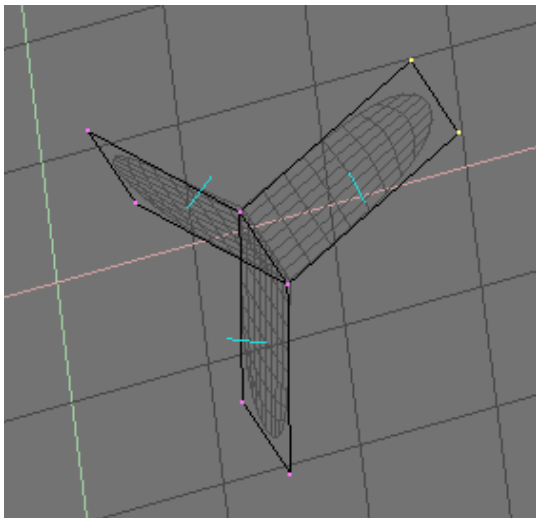


En estas imágenes las normales de las caras están dibujadas en azul claro. Se puede activar el dibujo de las normales en el menú Botones Edición (F9).

Nótese que Blender no puede recalcul las normales correctamente si la malla no es "Manifold". Una malla "Non-Manifold" es una malla para la cual un "out" no puede ser calculado inequívocamente. Basicamente, desde el punto de vista de Blender, es una malla donde existe bordes que pertenecen a *más* de dos caras.

[Figure 7-6](#) muestra un ejemplo simple de una malla "Non-Manifold". En general, una malla "Non-Manifold" ocurre cuando se tienen caras internas y por el estilo.

Figure 7-6. Una malla "Non-Manifold"



Una malla "Non-Manifold" no es un problema para mallas convencionales, pero puede producir artefactos extraños en mallas SubSurf. Además, no permite bajar el número de polígonos (decimation), por lo que es mejor evitarlo lo más posible.

Dos pistas para saber cuando una malla es "Non Manifold":

- Al recalcul las normales, quedan líneas negras en alguna parte
- La herramienta para bajar el número de polígonos (Decimator) en el panel *Malla* no funciona, argumentando que la malla es "No Manifold"

Las herramientas SubSurf permiten la creación de buenos modelos "orgánicos", pero hay que tener en cuenta que una malla regular con caras cuadradas, en lugar de caras triangulares, da mejores resultados.

[Figure 7-7](#) y [Figure 7-8](#) muestran ejemplos de lo que se puede hacer con el SubSurf de Blender.

Figure 7-7. Malla base de una Gárgola (izquierda) y su correspondiente malla SubSurf con nivel 2 (derecha) .

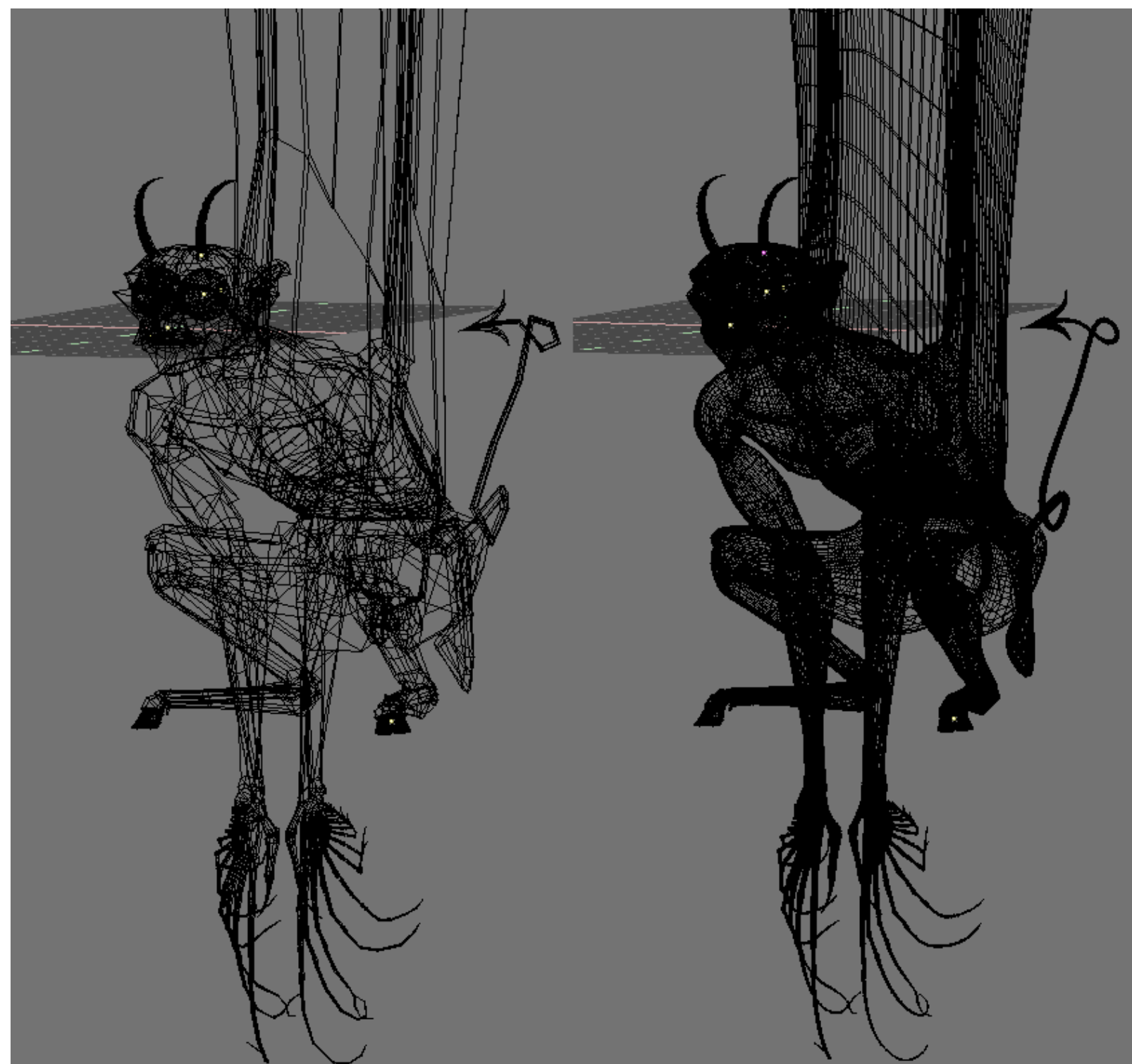
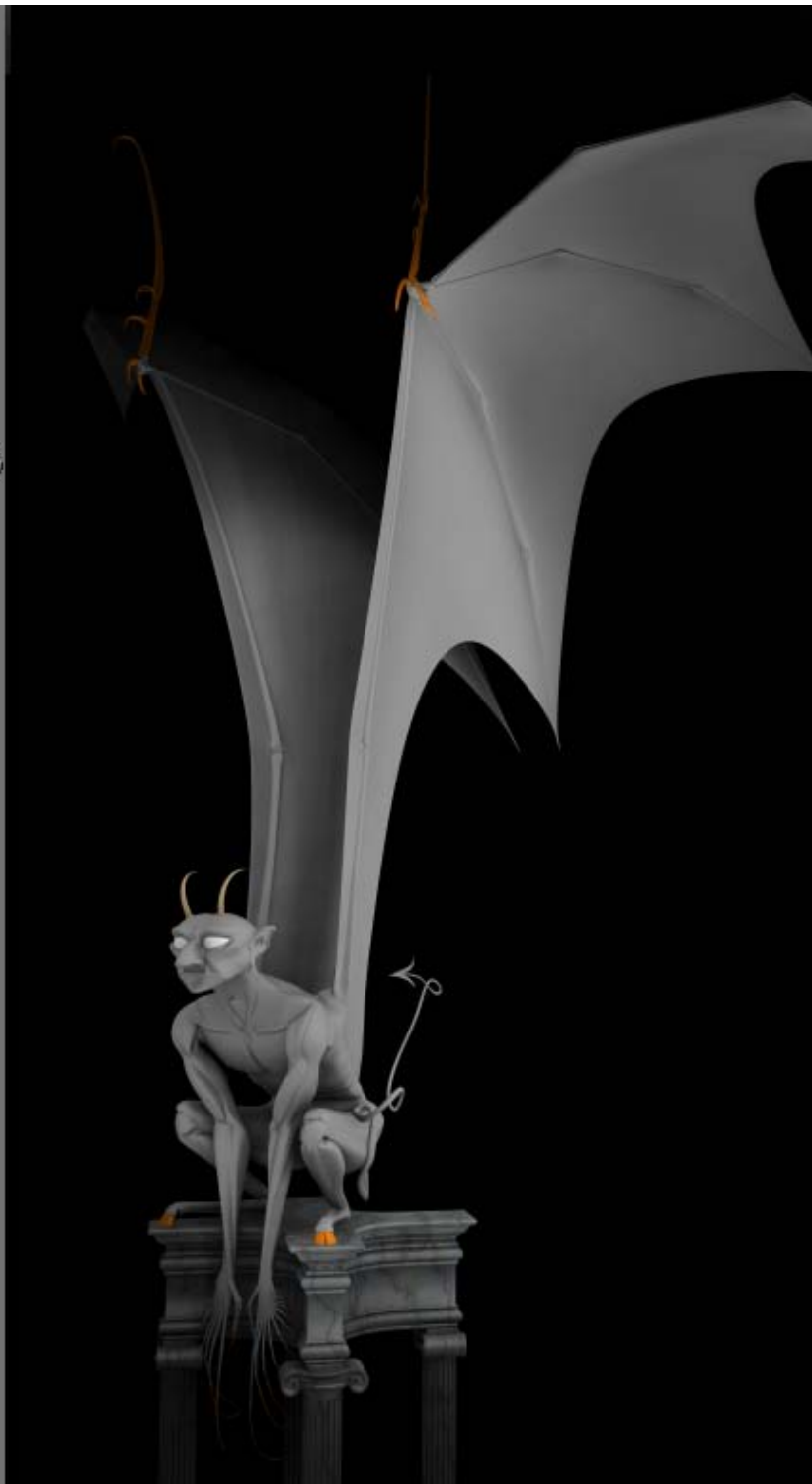
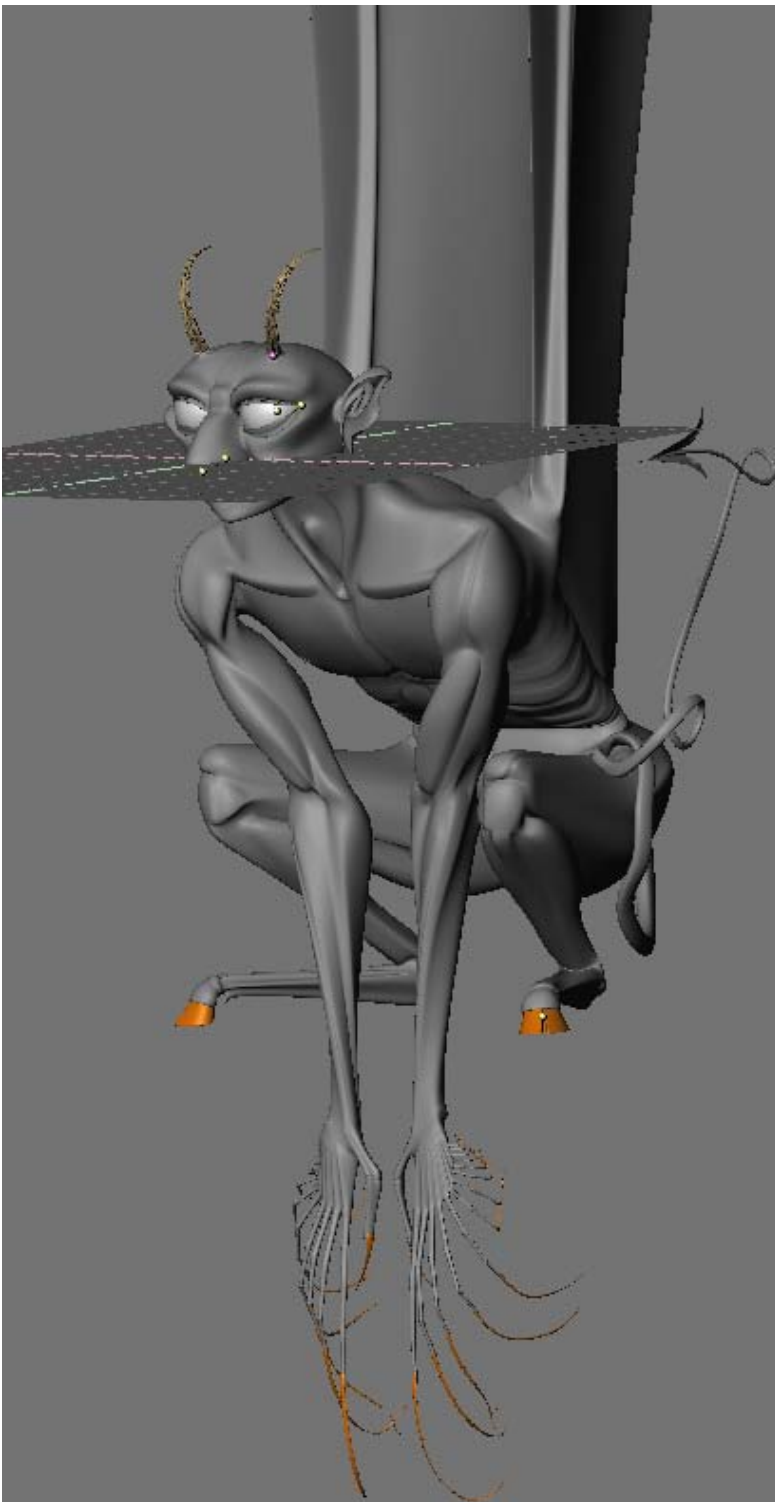


Figure 7-8. Vista de sólido (izquierda) y render final (derecha) de la Gárgola.



[Prev](#)

Garfios de Objeto (Object Hooks)

[Home](#)

[Up](#)

[Next](#)

Pliegues pesados para superficies de subdivision

7.2. Pliegues pesados para superficies de subdivision

by Kenneth Styrberg

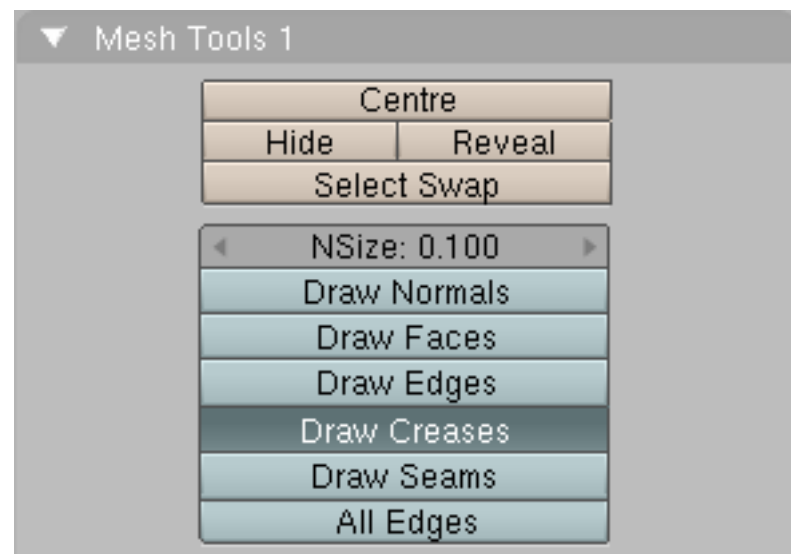
Relevante a Blender v2.34

Los pliegues pesados para superficies de subdivision permiten ajustar lo afilado de los borde

Los pliegues son una propiedad de los bodes de una malla, y pueden ser editados en una malla *Modo Edicion* cuando la malla tiene subdivision. Selecciona los bordes que quieres afilar, y presiona **SHIFT-E** para cambiar la cantidad de filo del borde.

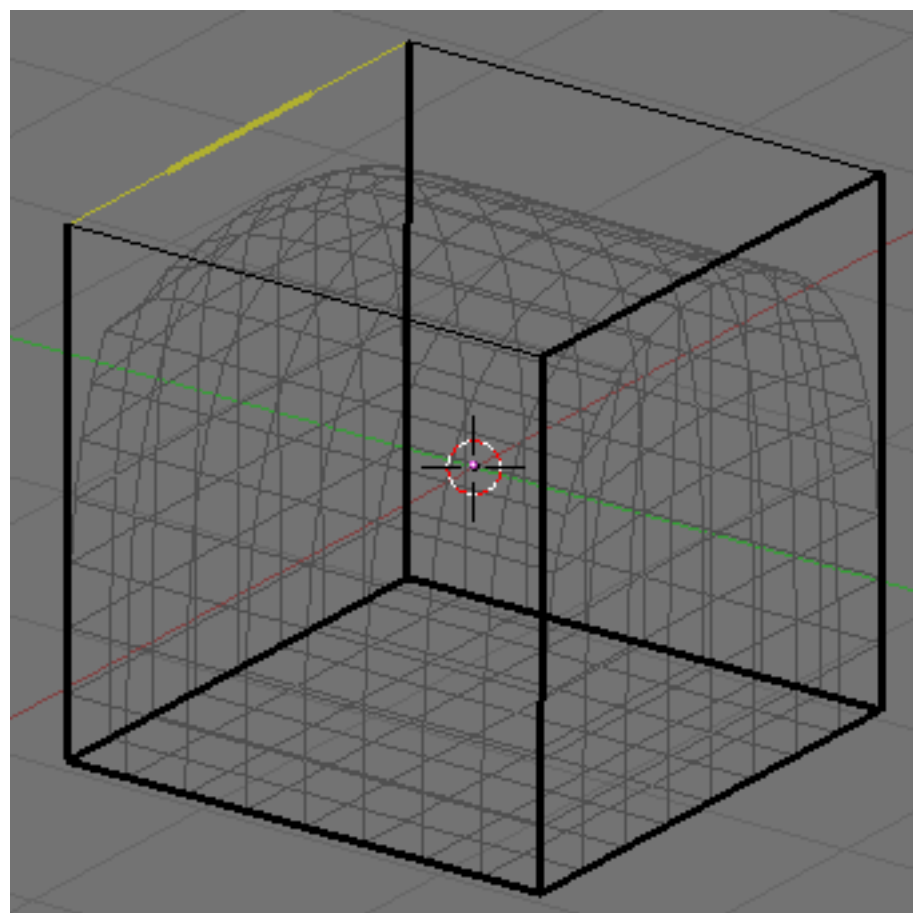
Puedes activar una indicacion del filo de tu borde al habilitar *Draw Creases*. Ver [Figure 7-9](#).

Figure 7-9. Herramientas de Mallas 1 panel



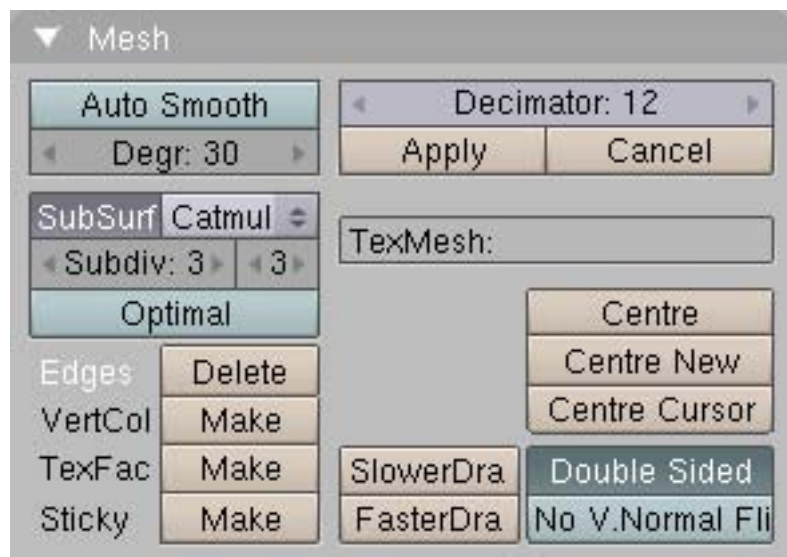
El valor de filo de un borde es indicado por una parte mas ancha en el borde. Si el borde tiene un filo de 1.0, el borde se vera bastante ancho, y si el valor de afilado es de 0.0, el borde sera delgado. Si el valor de filo esta entre 0.0 y 1.0, solo una parte del borde se vera mas ancha. Ver [Figure 7-10](#).

Figure 7-10. Afilado de un borde alrededor de 0.5



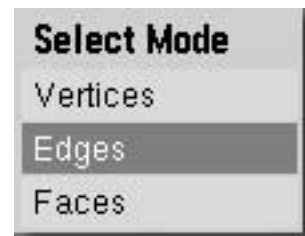
Para usar el afilado es necesario activar subdivision. Selecciona el cubo por default, si no tienes uno, añadelo! Ve al panel de Edicion (**F9**) y presiona el boton de *SubSurf*. Asegurate que el tipo de subdivision sea *Catmull-Clark* en el menu desplegable, ahora sube el nivel de subdivision a 3 en ambos, en el de desplegado y en el de render. Ver [Figure 7-11](#). El cubo se vera como una esfera.

Figure 7-11. Panel de Mallas



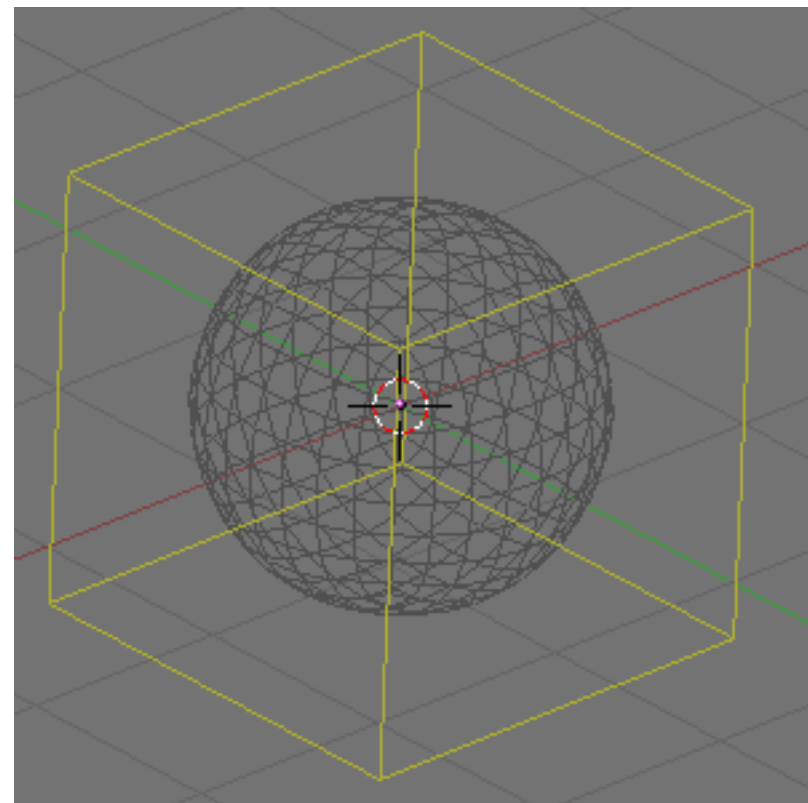
Enta a *Modo Edicion (TAB)*, con el cubo seleccionado. Por default estas en el modo de *Seleccion de Vertices*, ahora presiona **CTRL-TAB** para obtener un menu de *Seleccion de Modo*, [Figure 7-12](#), selecciona *Bordes*.

Figure 7-12. Menu Seleccion de Modo



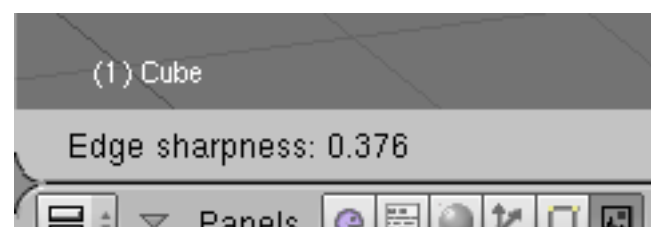
Ahora selecciona todos los bordes presionando **AKEY**. Todos los bordes deberian obtener un color amarillento, [Figure 7-13](#). Si todos los bordes son negros, entonces algo estaba previamente seleccionado. Presiona **AKEY** de nuevo para seleccionar todos los bordes.

Figure 7-13. Todos los bordes estan seleccionados



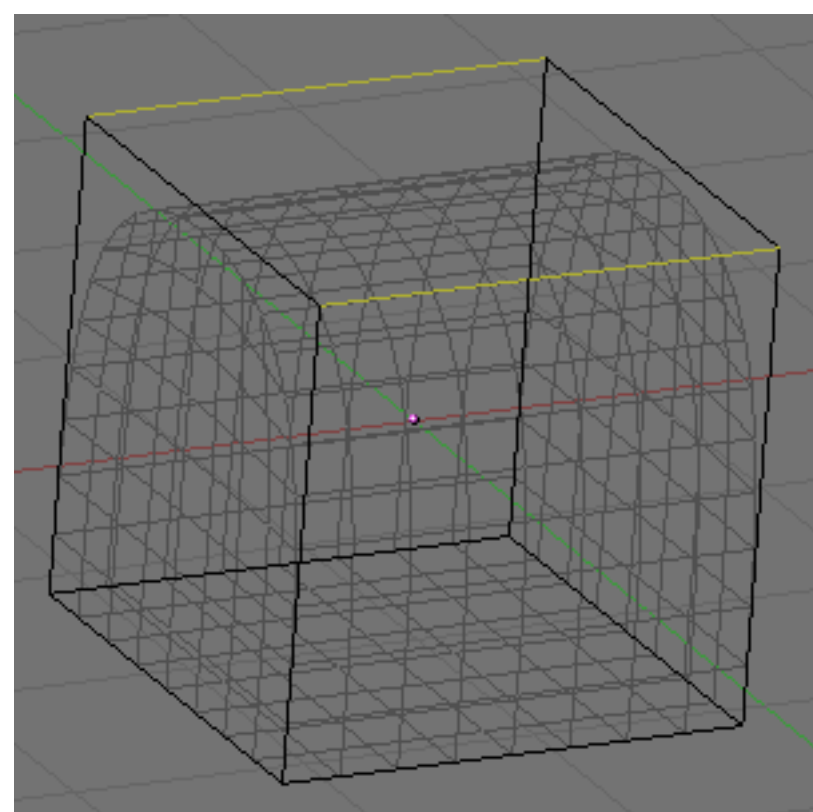
Ahora presiona **SHIFT-E** para editar el borde de filo del borde. El valor de filo se vera en tiempo real en la barra de herramientas, [Figure 7-14](#). Mueve el cursor mas cerca o mas lejos del borde para afectar su valor de afilado. Asigna el valor de afilado de todos los bordes en 1.0. El cubo regresara a su aspecto normal de cubo.

Figure 7-14. Afilado de Bordes



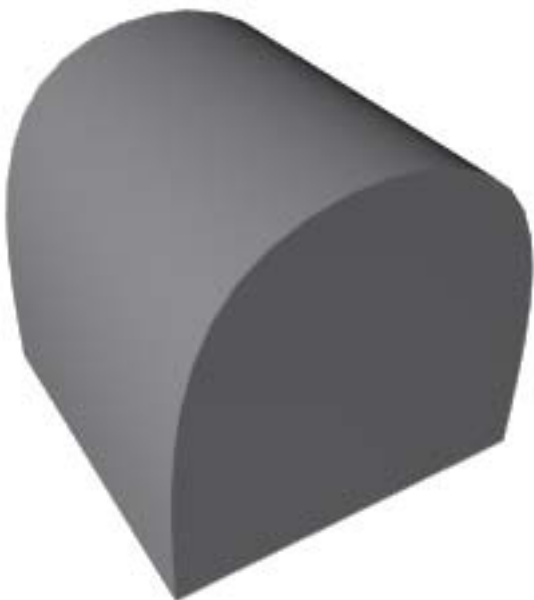
Ahora selecciona 2 bordes opuestos en la parte de arriba del cubo. Presiona **SHIFT-E** para editar el valor de afilado de los bordes. Asigna el valor de afilado a 0.0, [Figure 7-15](#).

Figure 7-15. Dos bordes seleccionados



Si renderizas obtendras un bello cubo con una parte superior redondeada, [Figure 7-16](#). Una mejora muy buena en el volumen de trabajo para lograr esta clase de malla!

Figure 7-16. Resultado



[Prev](#)

Modelado de Malla avanzado

[Home](#)

[Up](#)

[Next](#)

Herramientas de Arista

7.3. Herramientas de Arista

Relevante a Blender v2.31

Algunas herramientas de modelado nuevas se incorporaron en Blender 2.30. Las mismas están focalizadas en modelado de Aristas en lugar de modelado de vertices.

Un tema clave en Modelado es a menudo la necesidad de añadir vertices en ciertas zonas de una malla, y esto muchas veces tiene que ver con agregar o dividir aristas en una dada región. Blender nos ofrece ahora dos herramientas para ello, una *Herramienta Cuchillo (Knife Tool)* capaz de dividir aristas en los lugares deseados, y una herramienta *Bucle de Caras (Face Loop)* capaz de seleccionar hileras de caras y dividir las consistentemente.

Las herramientas de Aristas están agrupadas en un menu enlazado a la **KKEY**, pero cada herramienta individual tiene también su propia combinación de teclas.

7.3.1. Herramienta Cuchillo (Knife Tool)

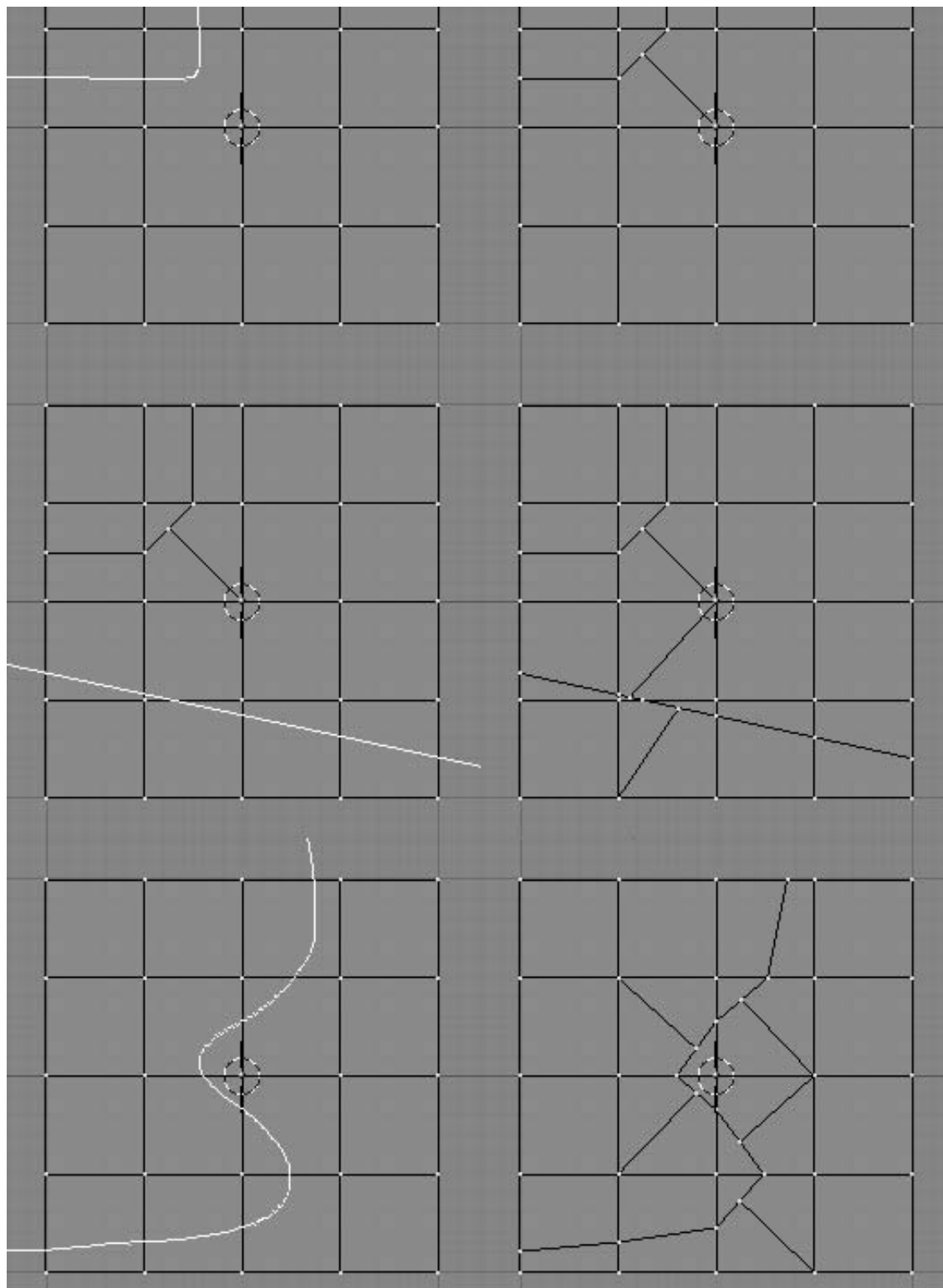
La Herramienta Cuchillo funciona subdividiendo aristas si sus dos vertices están seleccionados y la arista es intersectada por una linea "cuchillo" trazada manualmente por el usuario. Por ejemplo, si deseas cortar un agujero en solo una cara en el frente de una esfera, puedes seleccionar solamente los vertices del frente y luego trazar la linea con el ratón.

Para probar la herramienta, añade una Malla Rejilla (Grid). Estarás en Modo Edición con todos los vertices seleccionados. Pulsa **SHIFT-K** para invocar a la herramienta Cuchillo. Se te dará a elegir el tipo de corte: *Exact* dividirá las aristas exactamente por donde cruce el cuchillo, *Centers* dividirá las aristas intersectadas por su punto medio. Para este corte, elegimos *Centers*.

Ahora puedes clicar con **LMB** y comenzar a trazar. Si mueves y haces click con **LMB** trazarás segmentos rectos entre los puntos cliqueados; si mantienes presionado el **LMB** mientras arrastras el ratón, trazarás lineas a mano alzada. Las poligonales se pueden trazar con un número arbitrario de segmentos, pero la rutina de intersección solo detectará un cruce por arista. Si cruzamos otra vez una arista no se le realizarán cortes adicionales. El **MMB** restringe el trazado a un eje como se espera. El ajuste a la rejilla no se encuentra implementado actualmente pero se espera incorporarlo en futuras versiones. Cuando hayas terminado de trazar tu linea, pulsa **ENTER** para ocnfirmar el corte. **ESC** siempre cancelará la operación. [Figure 7-17](#) muestra algunos ejemplos.

Figure 7-17. Cuchillo "Center" con poligonal (arriba); Cuchillo "Exact" con un único segmento (medio) and Cuchillo "Exact" en modo mano alzada (abajo).





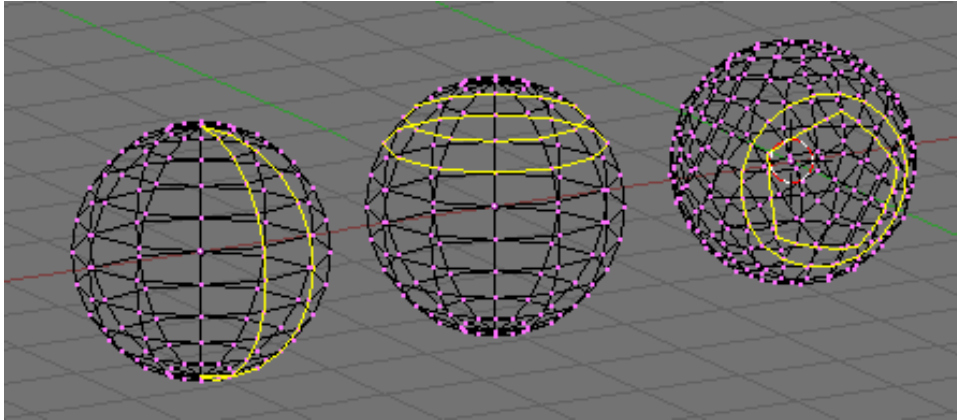
Note Con mallas grandes, será más rápido seleccionar un número pequeño de vértices, aquellos que definen las aristas que desees dividir, ya que el Cuchillo se ahorrará tiempo en verificar los vértices seleccionados contra los cruces del trazo de cuchillo.

7.3.2. Bucle de Caras (Face Loop)

La herramienta Bucle de Caras (Face Loop) te permite seleccionar, y eventualmente, dividir un *bucle o loop* de caras. Este bucle, se define comenzando con una arista bajo el puntero del ratón, y las dos caras que comparten esta arista.

Las dos caras que pertenecen al bucle y cada cara adicional opuesta a estas, son también parte del bucle. El bucle termina cuando se alcanza una cara triangular o si el bucle se cierra sobre sí mismo. La selección de bucles se activa con **SHIFT-R** ([Figure 7-18](#)).

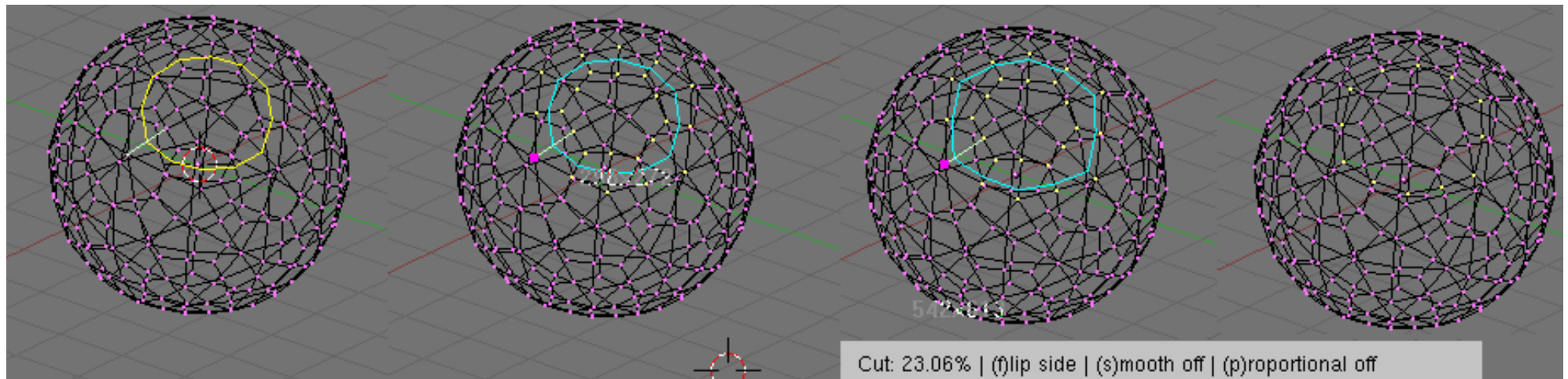
Figure 7-18. Un bucle de caras abierto (izquierda) y dos cerrados (centro y derecha).



Moviendo el ratón, se resaltarán distintos bucles de caras. Pulsando el **LMB** o **ENTER** se seleccionará el bucle deseado y con **ESC** se saldrá del comando.

Si la selección de bucle de caras se ingresa por medio de **CTRL-R** en lugar de **SHIFT-R** entonces la acción ejecutada no es una mera selección sino la división del bucle de caras a lo largo de su línea central. Esto es realmente muy útil para refinar una malla de una forma amigable con el Subsurf. ([Figure 7-19](#)).

Figure 7-19. Dividiendo un bucle de caras.



Note Ambas herramientas de bucle de caras están presentes también en el menú **KKEY**.

[Prev](#)

Pliegues pesados para superficies de subdivision

[Home](#)

[Up](#)

[Next](#)

Herramientas de sollevado (Bevel)

7.4. Herramientas de solevado (Bevel)

Relevante a Blender 2.33

Blender tiene, desde la versión 2.32, una herramienta para hacer solevados. Un solevado es algo que suaviza una esquina o arista afilada. En el mundo real las aristas de los objetos no son muy afiladas. Incluso el filo de un cuchillo no puede ser considerado una arista perfectamente afilada, si lo observas detenidamente, y la mayoría de las aristas están solevadas intencionadamente por razones prácticas o mecánicas.

La herramienta de solevado de Blender está aún bajo un profundo desarrollo y la actual implementación está lejos de ser perfecta ya que soleva todos las aristas de la malla. Por tanto no hay control sobre los vértices que se quiere mantener afilados o sobre las aristas que se encuentran en superficies que no son necesarias solevarlas.

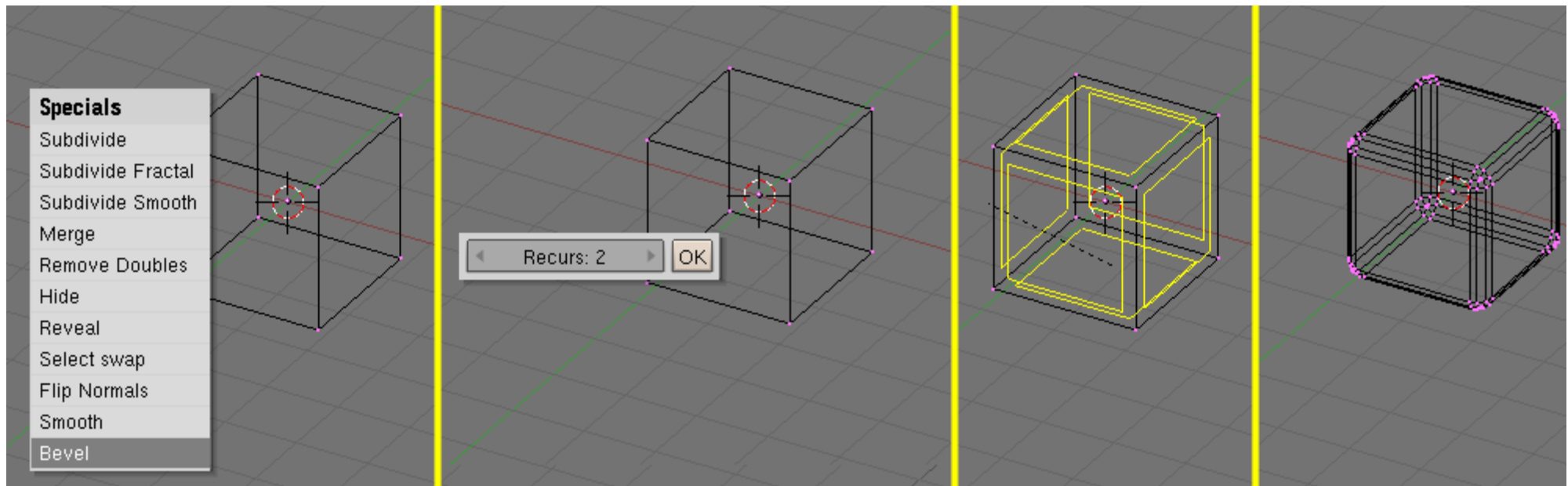
La herramienta de solevado se puede usar en modo Edición, accediendo a través del menú de la **Tecla W**, en la entrada *Bevel* ([Figure 7-20](#), izquierda). Una vez seleccionada, una ventana emergente preguntará sobre el número de recursiones en el solevado ([Figure 7-20](#), centro izquierda). Si es una recursión, entonces se reduce el tamaño de cara cara y de cada arista se crea una nueva cara. Caras de tres o cuatro vértices se crean según la necesidad. Si el número de recursiones es mayor que uno, entonces el procedimiento anterior es aplicado recursivamente ese número de veces, como por ejemplo si *Recur*: 2 cada arista es transformada en 4 aristas, y tres nuevas caras aparecerán en la arista, suavizando la original. En general el número de nuevas aristas es 2 elevado al número de *recursiones*.



Número de vértices

Recuerda que por cada nueva arista se crean dos vértices, y algunos más que se crean en la intersección de las aristas, así que el número de vértices puede crecer muy rápidamente si soleva con un número de recursión elevado!!

Figure 7-20. Solevando un cubo.



Una vez que el número de *recursion*es es seleccionado cada cara de la malla muestra un resaltado amarillo (Figure 7-20, centro derecha). Moviendo el cursor del ratón, crecerá o disminuirá el resaltado amarillo, y su correspondiente factor de disminución aparecerá en la cabecera de la ventana. Presionando **CTRL** el factor va en pasos de valor 0.1, presionando **SHIFT** puedes afinar dicho factor mucho más. Presionando **SPACE** una ventana emergente aparecerá, preguntándole cual valor de solevado desea introducir.

LMB finaliza la operación y **RMB** o **ESC** la cancela. El resultado final se puede ver en (Figure 7-20, derecha).

[Prev](#)

Herramientas de Arista

[Home](#)

[Up](#)

[Next](#)

Modelado Simétrico

7.5. Modelado Simétrico

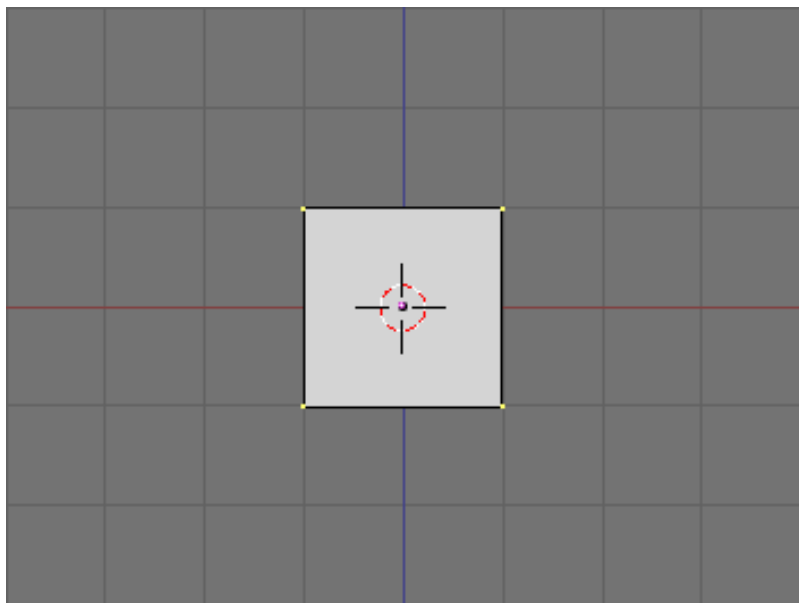
Relativo a Blender v2.34

Con frecuencia es necesario modelar objetos que poseen algún tipo de simetría. Para simetría radial, rotacional o múltiple, la mejor aproximación es modelar cuidadosamente una estructura base y luego, como paso final, duplicar la célula básica utilizando SpinDup o cualquier comando apropiado.

Para los objetos con simetría bilateral, es decir aquellos con un plano de simetría como la mayoría de los seres vivos (incluyendo al hombre) y muchas máquinas, el método supone el modelado de la mitad del objeto, y luego rebatir especularmente un duplicado de la primera mitad para obtener el objeto entero.

Como normalmente es difícil mantener las proporciones correctas modelando solo una mitad, es posible duplicar la mitad antes de que esté totalmente modelada, y actuar solo sobre ésta actualizando automáticamente la otra.

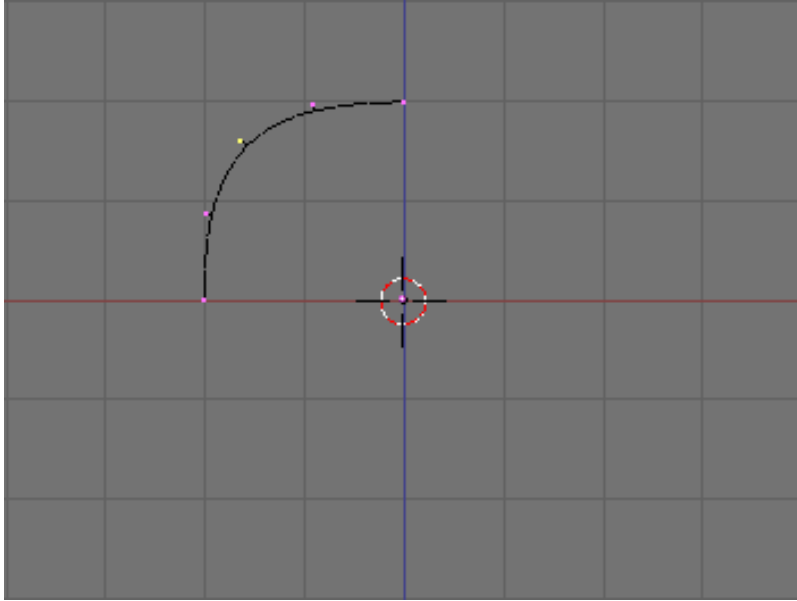
Figure 7-21. Un plano.



En la Vista Frontal, agregue un plano o cualquier otro objeto ([Figure 7-21](#)). Considérelo como el punto inicial para la mitad del objeto. Digamos la mitad

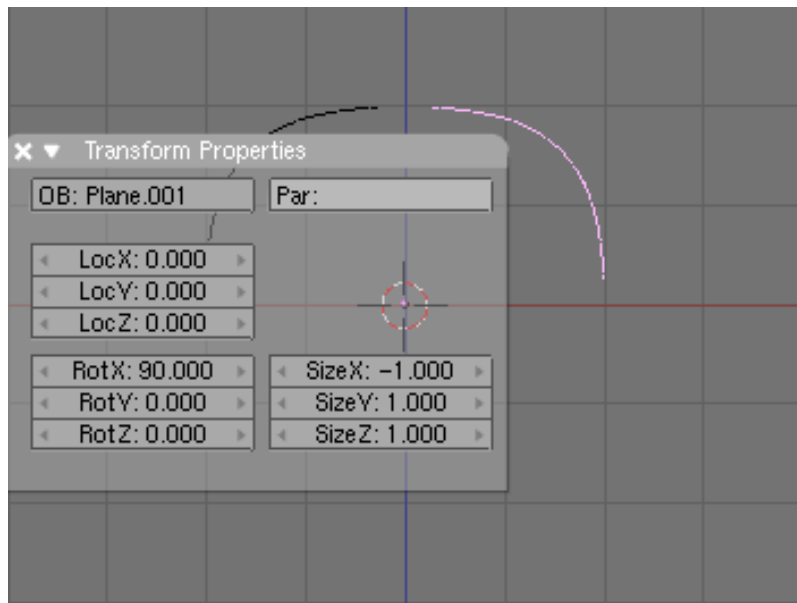
derecha del objeto, la cual para nosotros en la vista frontal, se encontrará a la izquierda de la pantalla. El plano de simetría es el plano YZ. Mueva la malla en ModoEdición, de manera que quede totalmente a la izquierda del centro. Elimine algunos vertices, y añada otros para darle la forma general como se ve en la [Figure 7-22](#).

Figure 7-22. Mitad derecha.



Ahora cambie a ModoObjeto y con la mitad seleccionada, haga un duplicado *vinculado* con **ALT-D**. Pulse **ESC** para salir del Modo Mover, y presione **NKEY**. En el panel de entrada numérica que aparece, ponga *SizeX* a -1 ([Figure 7-23](#)). esto rebatirá especularmente la copia vinculada con respecto al centro del Objeto, por ello la importancia de mantener el centro sobre un plano de simetría.

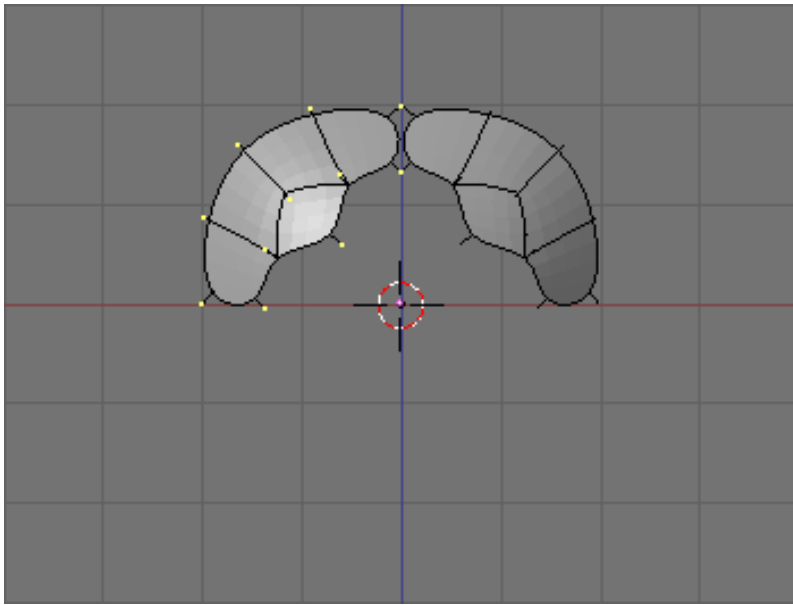
Figure 7-23. Rebatiendo especularmente el duplicado vinculado.



Haber duplicado el Objeto como un duplciado vinculado, significa que los dos objetos comparten los *mismos* datos de malla, los que están implícitamente rebatidos, por el escalado negativo unitario sobre el eje X, que es perpendicular al plano de simetría.

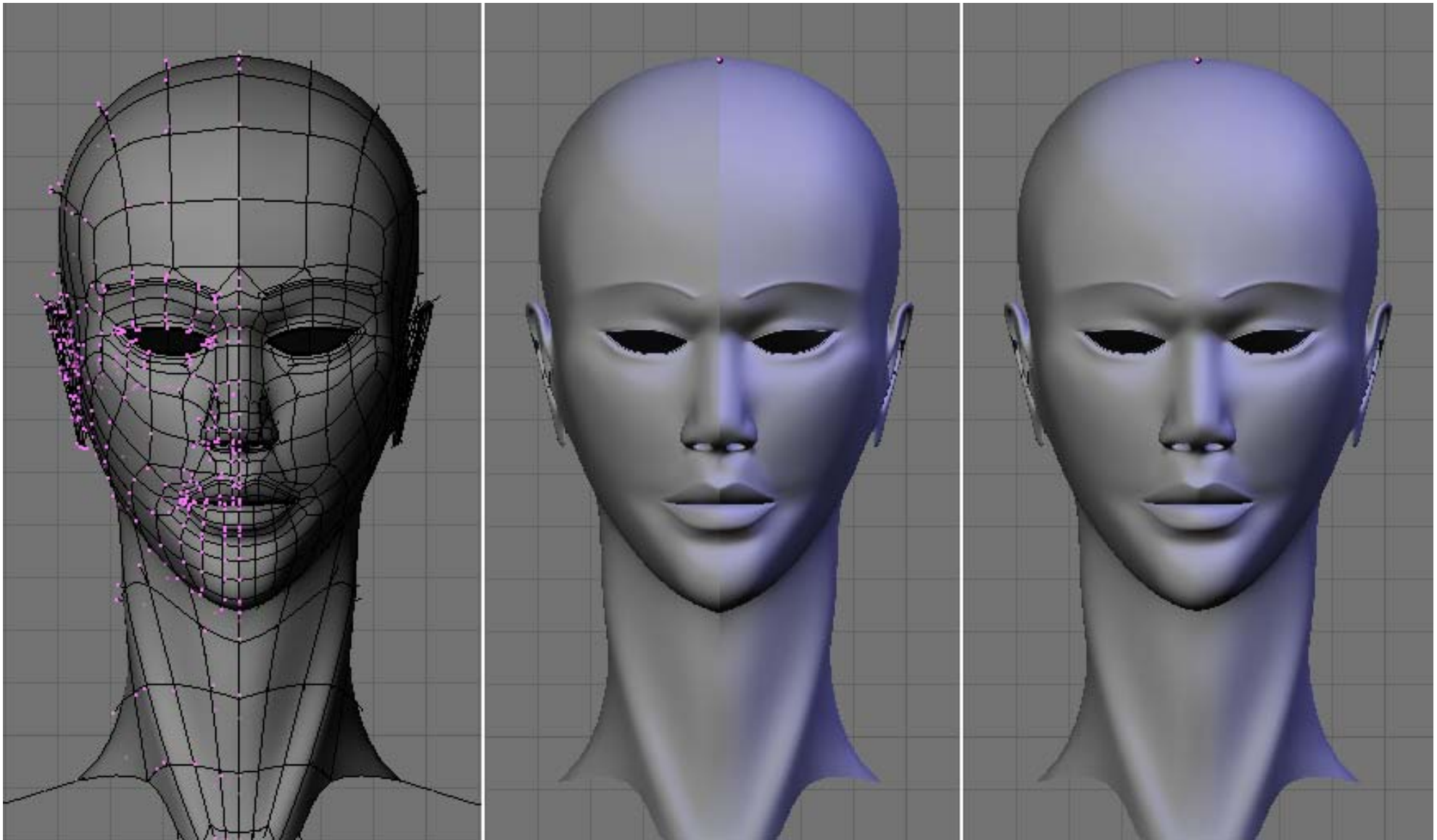
Ahora puede editar cualquier ed las dos mitades. Como comparten los datos de malla, cualquier cambio, sea un extrudado, borrado, corte en bucle de caras, etc, se reflejará inmediatamente en la otra mitad ([Figure 7-24](#)).

Figure 7-24. Editando una mitad.



Editando cuidadosamente una mitad, y posiblemente utilizando un esquema o diagrama como fondo que sirva de guía, se pueden lograr resultados muy interesantes ([Figure 7-25](#), izquierda).

Figure 7-25. Una cabeza. Izquierda: ModoEdición; Centro: ModoObjeto; Derecha: Unida.



Como último paso, cuando esté completo el modelado simétrico, se deben seleccionar las dos mitades y unirlas en un único objeto (**CTRL-J**). Esto hace desaparecer la junta (muy visible en la [Figure 7-25](#), centro). Una vez unidas en un objeto único, ([Figure 7-25](#), derecha), puede comenzar a modelar las sutiles asimetrías que toda persona tiene.

Note En Blender 2.33 y versiones anteriores, la implementación OpenGL provocaba que los duplicados vinculados rebatidos, tuviesen normales erróneas, de modo que una de las mitades era negra. Esto fue corregido en Blender 2.34, pero versiones viejas pueden aprovechar esta técnica igualmente, haciendo que la malla sea de lado simple cuando se usa el modelado simétrico.

[Prev](#)

Herramientas de sollevado (Bevel)

[Home](#)

[Up](#)

[Next](#)

Herramienta de Edición Proporcional (PET)

7.6. Herramienta de Edición Proporcional (PET)

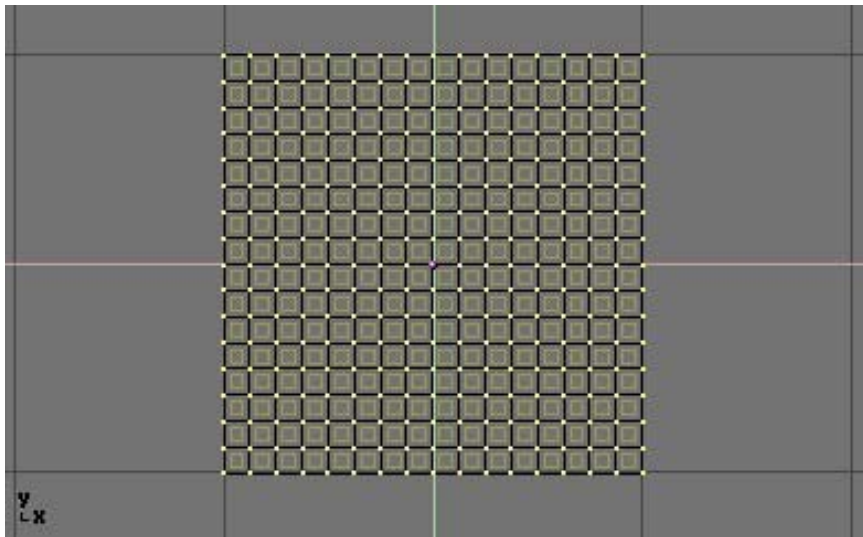
Relevant to Blender v2.37

by Kenneth Styrberg

Cuando se trabaja con mallas densas puede ser difícil hacer ajustes finos a los vértices sin provocar desagradables pliegues y pellizcos en la superficie del modelo. Cuando te encuentres con situaciones como estas, utiliza la herramienta de edición proporcional. Actúa como un imán para deformar suavemente la superficie del modelo sin crear pliegues ni bultos.

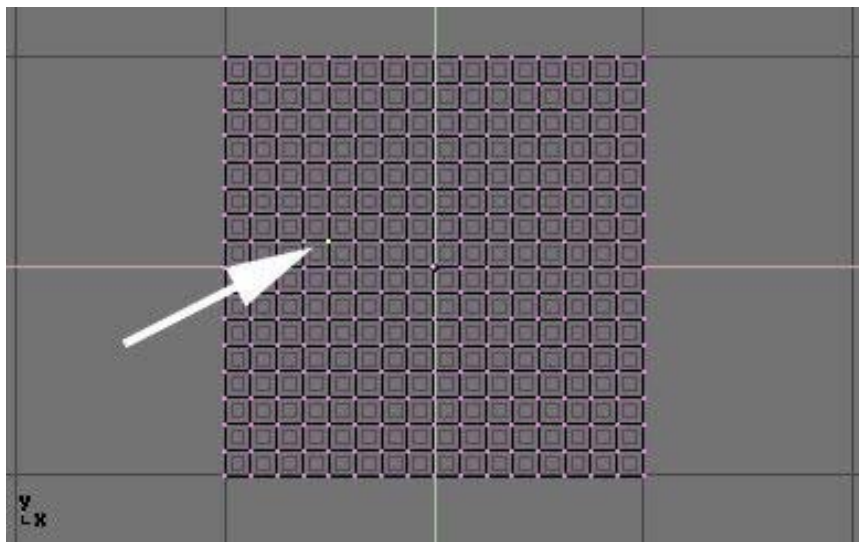
En la Vista Superior, agrega un plano a la escena con **SPACE**>>Add>>Mesh>>Plane. Subdividelo un par de veces con **WKEY**>>Subdivide (o pulsando en el botón *Subdivide* del Panel *Mesh Tools* en el Contexto Edición para obtener un malla relativamente densa ([Figure 7-26](#)). O añade una rejilla directamente por medio de **SPACE**>>Add>>Mesh>>Grid, especificando el número de vértices en cada dirección. Cuando termines deselecciona todos los vértices con **AKEY**.

Figure 7-26. Una malla plana densa.



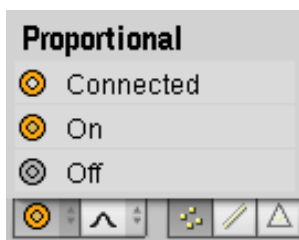
Selecciona un vertice en la malla pulsando **RMB** ([Figure 7-27](#)).

Figure 7-27. Una malla plana densa con solo un vértice seleccionado.



Mientras estés aún en modo Edición, activa la herramienta de edición proporcional pulsando la **Tecla O** o usando el icono de la barra de herramientas y seleccionando tanto a *On* como a *Connected*. ([Figure 7-28](#)). Cuando la edición proporcional esté activa el icono se pondrá de color naranja.

Figure 7-28. Icono de Edición proporcional



- **Connected** - Únicamente los vértices que estén conectados a otros en la malla, podrán verse afectados.
- **On** - Todos los vértices pueden verse afectados.
- **Off** - ¡La edición proporcional está desactivada!

Cambia el tipo de curva guía que se va a usar con el submenú *Mesh>>Proportional Falloff*, con el icono de la barra de herramientas [Figure 7-29](#) o pulsando **SHIFT-O** para cambiar entre las opciones *Constant (Constante)*, *Linear (Lineal)*, *Sharp (Afilado)*, *Root (Cuadrático)*, *Sphere (Esférico)* y *Smooth (Suave)*. ([Figure 7-30](#) [Figure 7-31](#) [Figure 7-32](#) [Figure 7-33](#) [Figure 7-34](#) [Figure 7-35](#))

Figure 7-29. Menú Falloff.



Figure 7-30. Constante - Sin Falloff.

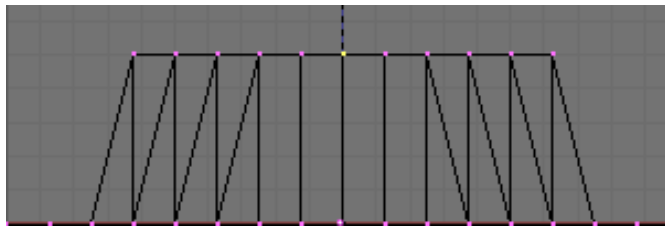


Figure 7-31. Falloff lineal.

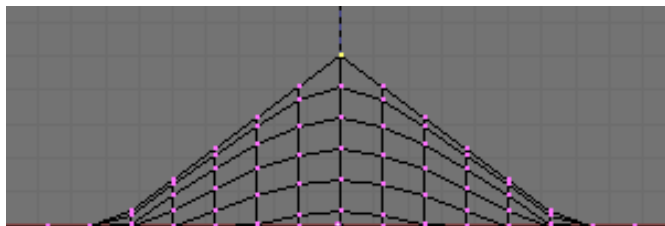


Figure 7-32. Falloff Afilado.

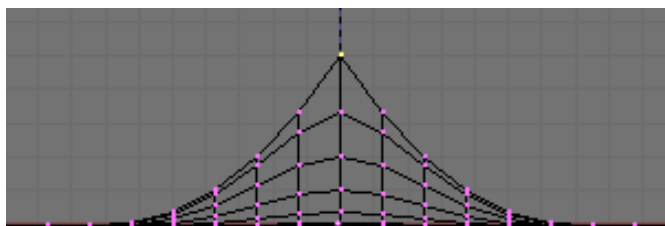


Figure 7-33. Falloff Cuadrático.

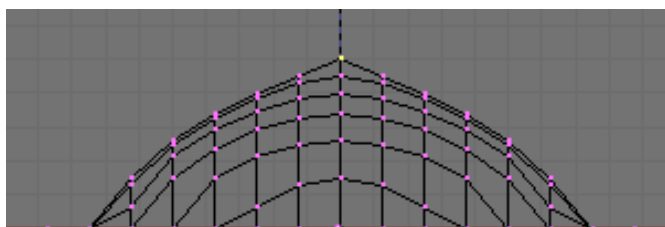


Figure 7-34. Falloff Esférico.

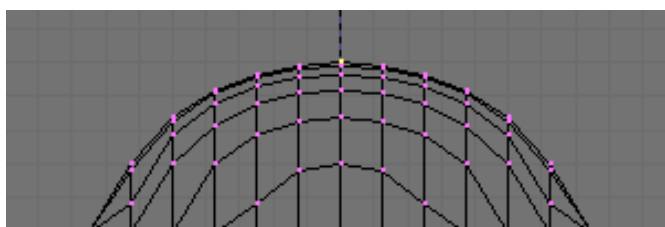
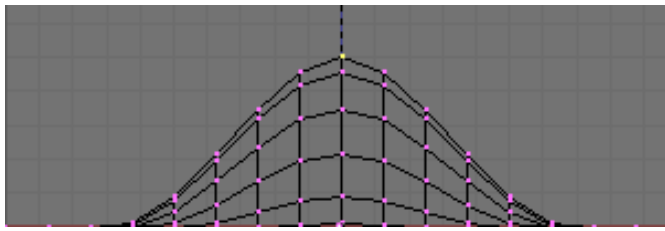


Figure 7-35. Falloff suave.

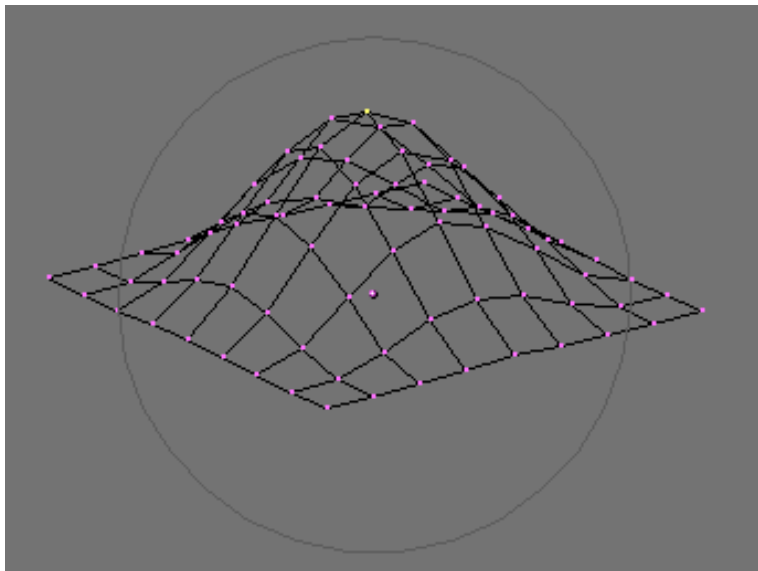


Cambia a la vista frontal (**NUM 1**) y activa la herramienta mover con **GKEY**. A medida que arrastres el puntero hacia arriba notarás que los vértices cercanos también son arrastrados con él.

Cuando estés satisfecho con la ubicación del vértice, presiona **LMB** para fijar su posición. Si no estás conforme, anula la operación y vuelve la malla a su estado inicial con **RMB** o **ESC**.

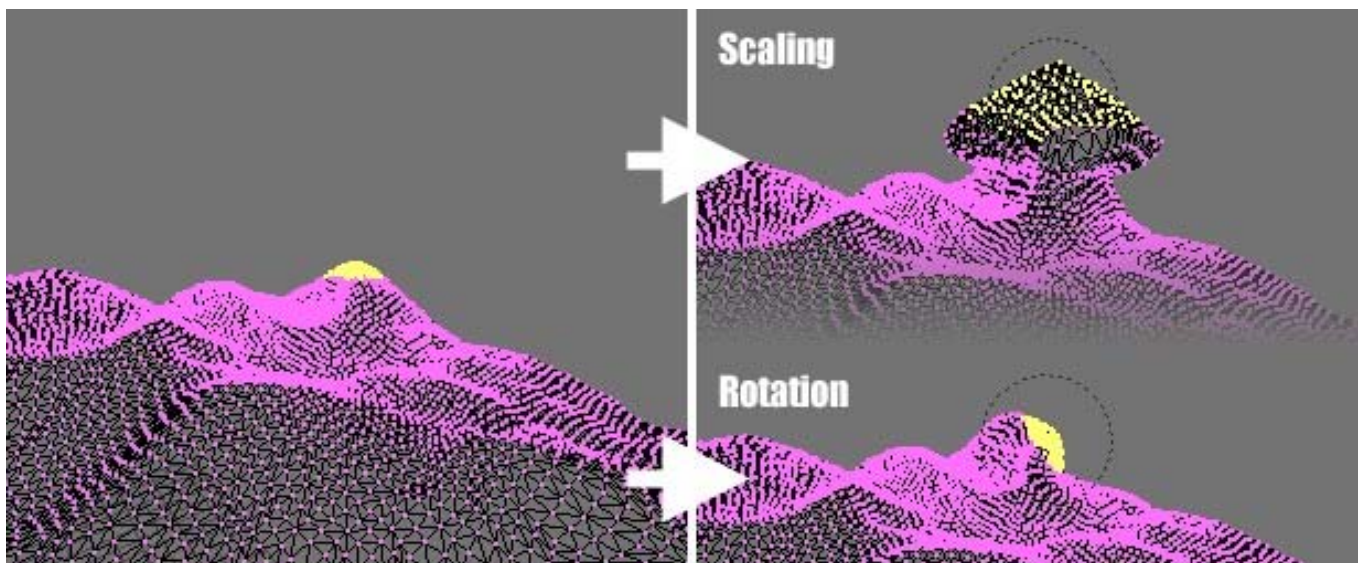
Mientras estás editando puedes aumentar o reducir el radio de influencia (mostrado como el círculo punteado en [Figure 7-36](#)) pulsando **NUM+** y **NUM-** respectivamente. A medida que modificas el radio, los puntos alrededor de la selección ajustaran sus posiciones correspondientemente. También puedes utilizar **MW** para agrandar o achicar el círculo.

Figure 7-36. Círculo de influencia.



La herramienta de edición proporcional puede utilizarse para producir grandes efectos con las herramientas de escalado (**SKEY**) y rotación (**RKEY**) como se ve en [Figure 7-37](#).

Figure 7-37. Un paisaje obtenido con Edición Proporcional



Combina estas técnicas con pintura de vértices para crear paisajes fantásticos.

[Figure 7-38](#) muestra los resultados de la edición proporcional luego de aplicar texturas e iluminación.

Figure 7-38. Paisaje final renderizado



[Prev](#)

Modelado Simétrico

[Home](#)

[Up](#)

[Next](#)

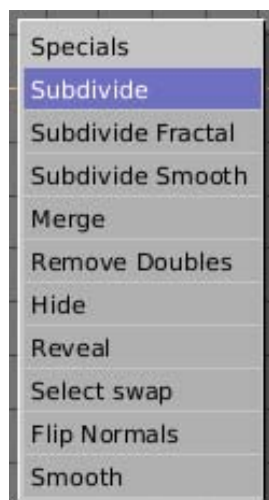
Ruido (Noise)

7.7. Ruido (Noise)

Relevante a Blender v2.31

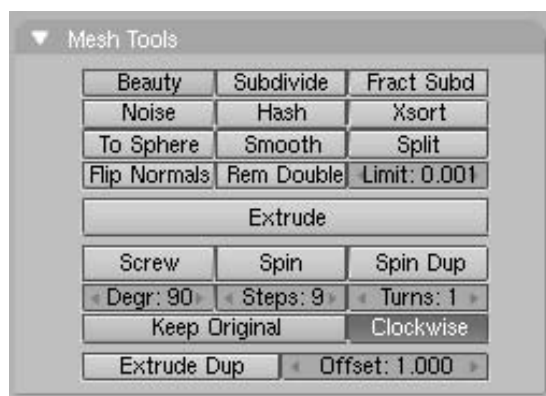
La función Ruido permite desplazar los vertices en las mallas basado en los valores de gris de una textura aplicada a las mismas. De esta manera se pueden generar grandes paisajes o texto esculpido en las mallas.

Figure 7-39. Herramienta Subdivide (Subdividir)



Agrega un plano y subdividelo al menos cinco veces con el menú Specials **WKEY**>>Subdivide ([Figure 7-39](#)). Ahora añade un material y asigne una textura Clouds. Ajusta el parametro *NoiseSize*: a 0.500. Elige blanco como color para el material y negro como el color de la textura, para dar suficiente contraste para el comando Ruido.

Figure 7-40. Botón Ruido en Botones de Edición



Verifica que estés en Modo Edición y que todos los vertices estén seleccionados, luego cambia al Contexto Edición **F9**. Pulsa el botón *Noise* en el Panel *Mesh Tools* Panel ([Figure 7-40](#)) varias veces hasta que el paisaje se vea bien. [Figure 7-41](#) muestra el plano original - texturizado - y también lo que ocurre cuando pulsas *Noise*. Ahora borra la textura del paisaje porque arruinará el aspecto. Agrega alguans luces, algo de agua, activa el suavizado y el Subsurf en el terreno. ([Figure 7-42](#)).

Figure 7-41. El proceso de aplicación de Ruido

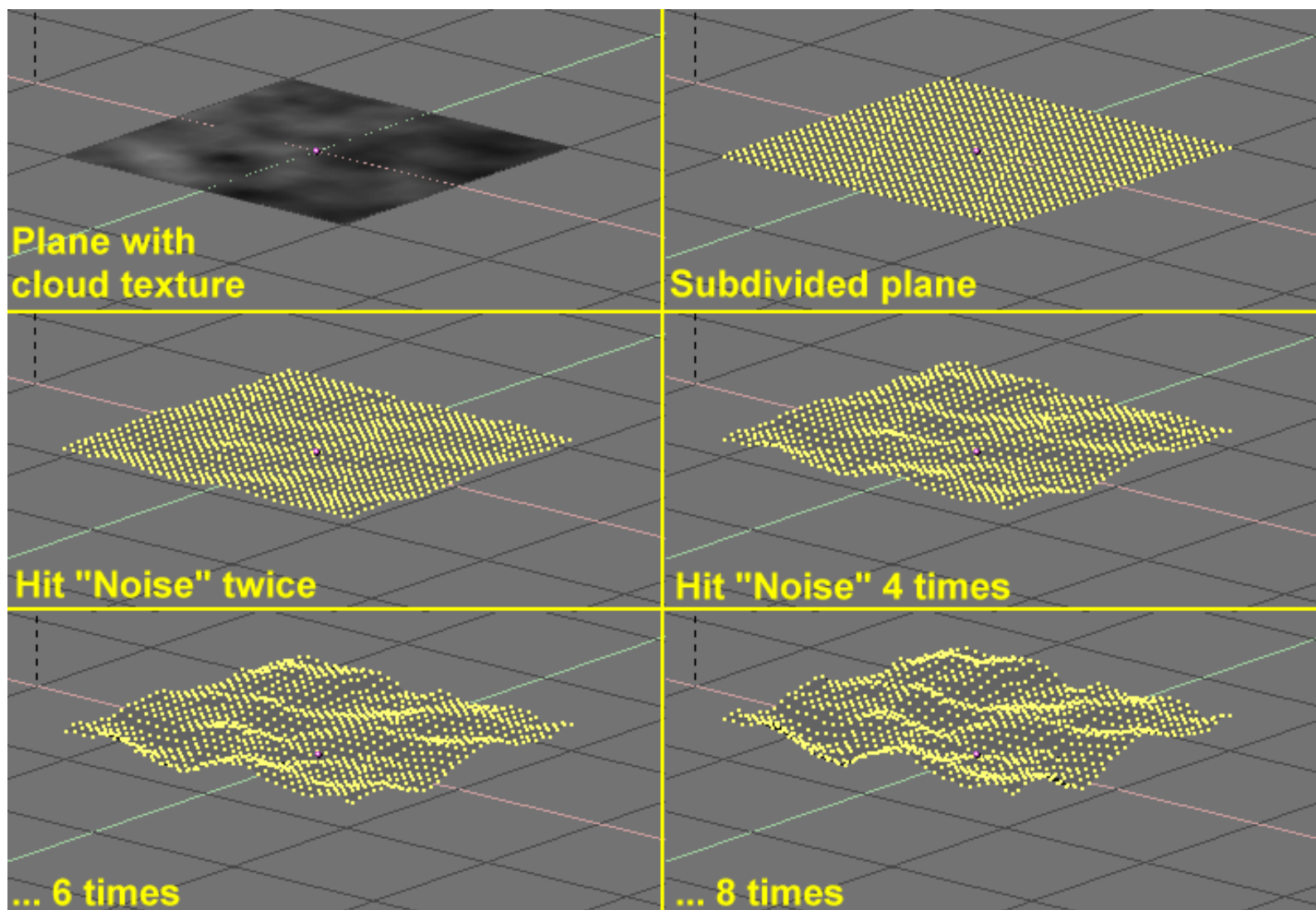
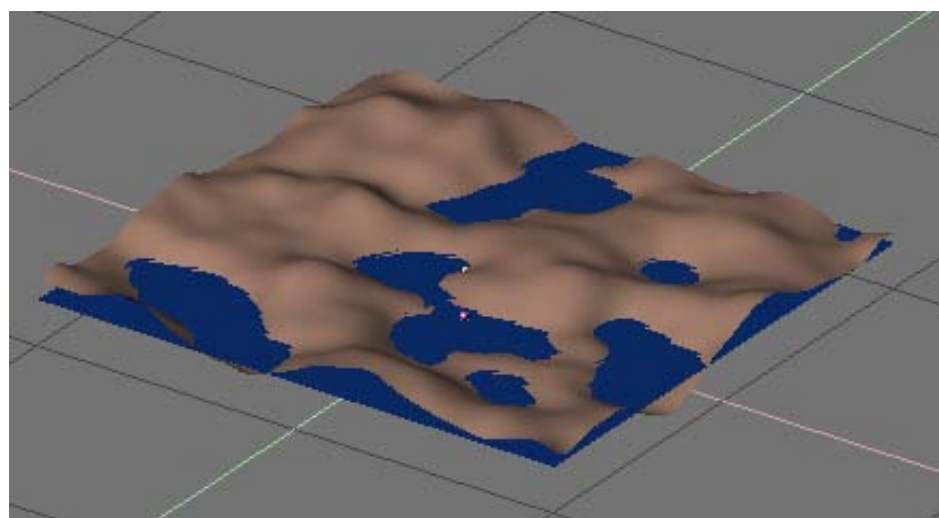


Figure 7-42. Paisaje generado con Ruido



Note El desplazamiento de ruido, siempre tiene lugar a lo largo de la coordenada z de la malla, que coincide con la orientación del eje z del objeto en referencia local.

Herramienta de Edición Proporcional
(PET)

[Up](#)

Herramienta Diezmador (Decimator)

7.8. Herramienta Diezmador (Decimator)

Relativo a Blender v2.33

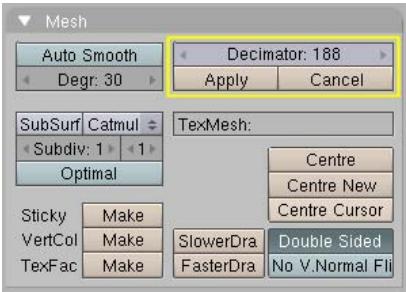
La herramienta Diezmador, es una característica a menudo pasada por alto que permite reducir la cantidad de caras/vertices de una malla con un mínimo cambio de forma.

Esto no es aplicable a mallas que han sido creadas modelando cuidadosa y económicamente, donde todos los vertices y caras son necesarios para definir correctamente la forma, pero si la malla es el resultado de un modelado complejo, con edición proporcional, reinamientos sucesivos, incluso alguna conversión de malla Subsurf a malla sin Subsurf, podría muy bien termianr con una malla con montones de vertices que no sean realmente necesarios.

Un ejemplo simple es un plano, y un objeto Cuadrícula 4x4 sin deformar. Ambos renders son exactamente iguales, pero el plano tiene una cara y cuatro vertices, mientras que la cuadrícula tiene nueve caras y 16 vertices, es decir montones de vertices y caras innecesarios.

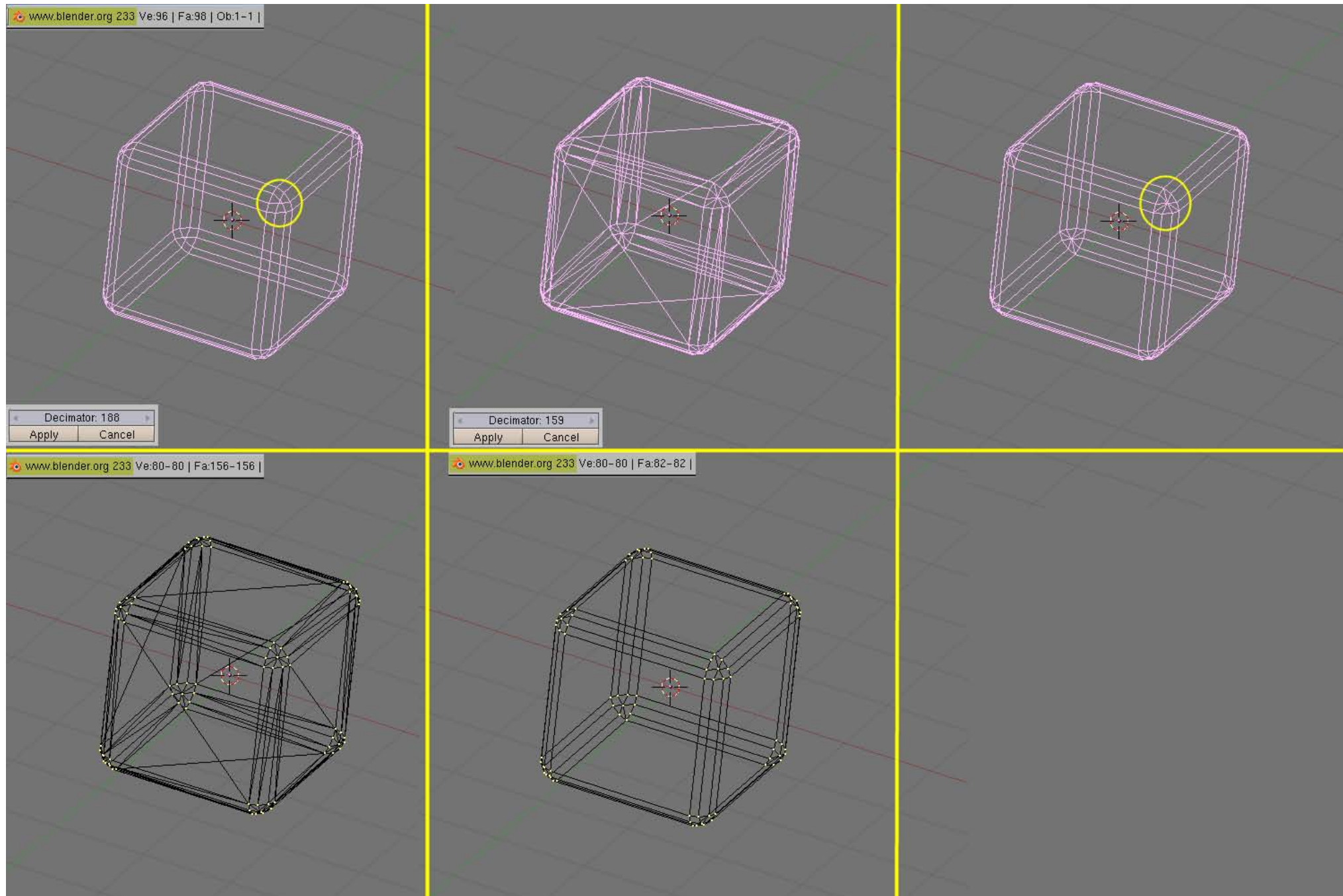
La Herramienta Diezmador ([Figure 7-43](#)) permite eliminar estas caras innecesarias. Su deslizador numérico informa la cantidad de caras de la malla seleccionada en Modo Objeto. El diezmador solo manipula triangulos, por lo que cada cara cuadrilátera es dividida implícitamente en dos triangulos para ser "diezmada".

Figure 7-43. Botones del Diezmador.



Consideremos el ejemplo utilizado en la sección Bisel. Como quizás observe, hay una pequeña cara triangular sobre cada vertice del cubo que muy probablemente sea innecesaria ([Figure 7-44](#), arriba a la izquierda). La cabecera dice que el cubo tiene 98 caras y 96 vertices. El botón Diezmador dice que el cubo tiene 188 caras triangulares, es decir 90 cuadrilateros (que serían 180 triangulos) y 8 triangulos.

Figure 7-44. El Diezmador "diezmando".

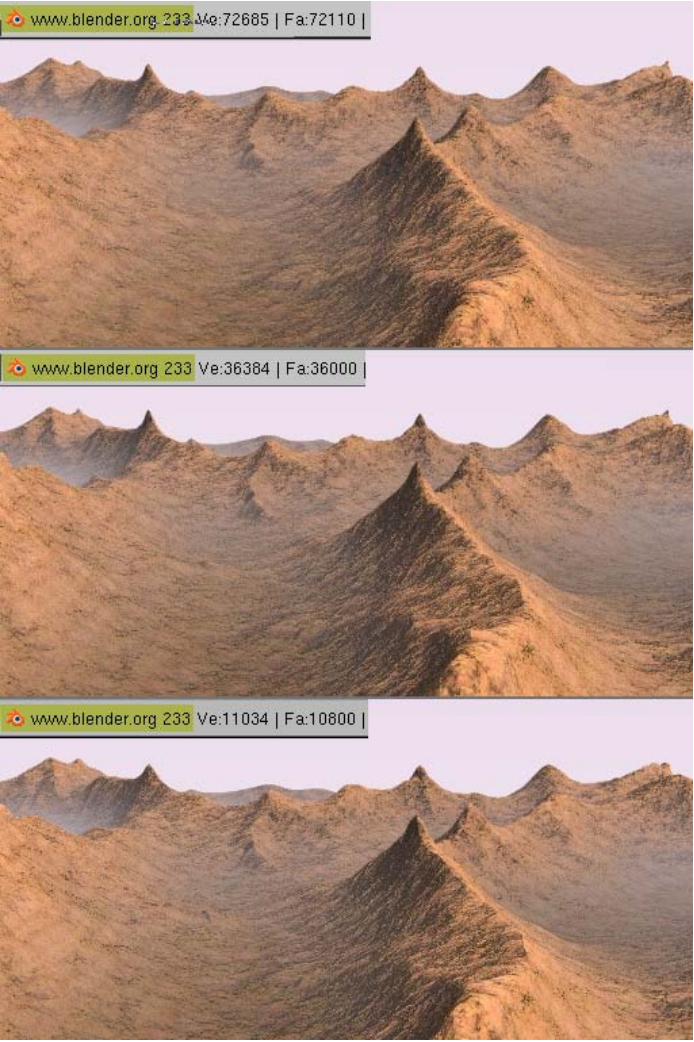


Cambiando el número en el botón numérico del diezmador, pulsando con el ratón o bien tipeando en él, la malla cambia inmediatamente a "sólo triángulos". A medida que el número disminuye, las caras van desapareciendo una tras otra. Blender hace que las caras coplanares y los vértices sobre aristas alineadas desaparezcan primero. Esto tiende a preservar la forma de la malla. A medida que más y más caras son eliminadas, menos vértices coplanares y colineales son soldados, por lo tanto podría ocurrir un cambio de forma apreciable ([Figure 7-44](#), arriba al centro).

En este caso particular, si solo deseamos que el triángulo central de cada vértice del cubo desaparezca, esperamos que la malla final sea de $2 \times 6 = 12$ caras por cada cara del cubo, $2 \times 3 \times 12 = 72$ caras por cada arista biselada, y $9 \times 8 = 72$ caras por cada vértice biselado, totalizando 156 caras. Es muy raro saber de antemano cuantas caras puede tener la malla final, comúnmente debe verificar cuidadosamente la malla en la Ventana3D para constatar que la forma aún es aceptable.

Los dos botones inferiores del Diezmador, finalizan o cancelan el diezmado. Una vez terminado, los triangulos ya no se mostrarán más (Figure 7-44, arriba a la derecha) pero la malla no obstante está hecha solamente de triangulos (Figure 7-44, abajo a la izquierda). Puede volverla a cuadriláteros si así lo desea seleccionando todos los vertices y pulsando **ALT-J** (Figure 7-44, abajo al centro). De esta manera redujimos la cuenta de vertices a 80 y la de caras a 82, sin ninguna pérdida apreciable en la forma. Podrá parecer una ganancia exigua, pero si este cubo va a ser duplicado con "Dupliverts" en un sistema de partículas con 1000 partículas, seguramente habrá valido la pena.

Figure 7-45. Paisaje diezmado, arriba: original; medio: ligeramente diezmado; abajo: fuertemente diezmado.



La Figure 7-45 muestra un paisaje generado por medio de la cuidadosa aplicación de la técnica de Ruido descrita anteriormente, sobre una cuadrícula bastante extensa. En la parte superior, el resultado de la malla original y debajo dos niveles de diezmado distintos. Para el ojo, la diferencia es efectivamente apenas apreciable, pero la reducción en la cuenta de vertices aquí representa una gran ganancia.

Chapter 8. Los Meta Objects

Relativo a Blender v2.31

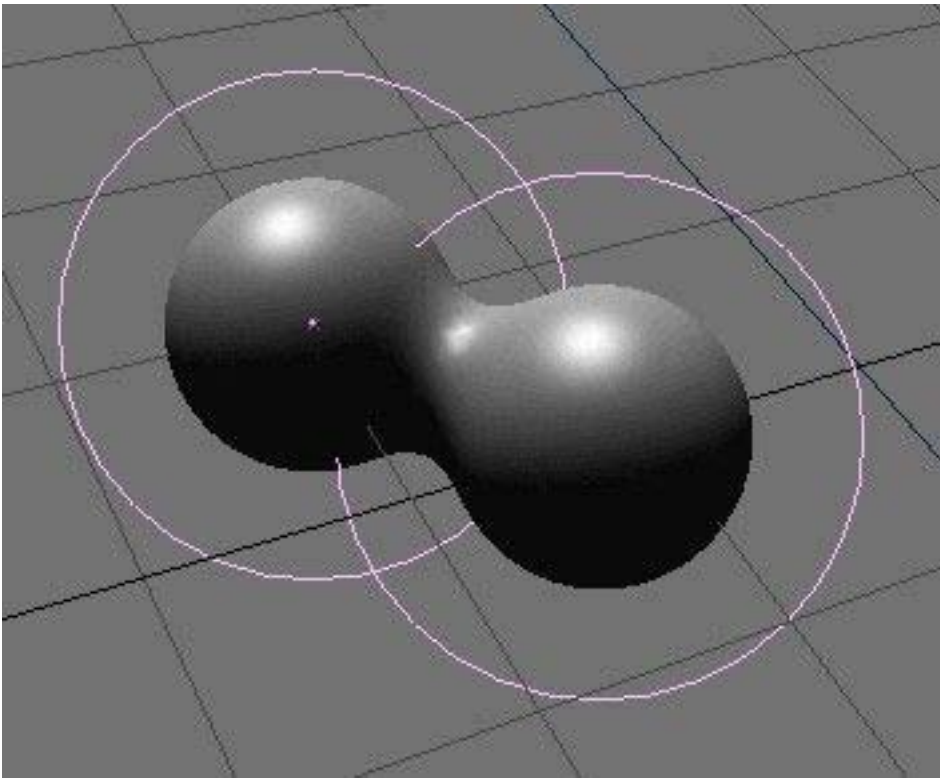
Los Meta Objects consisten en elementos esféricos, tubulares y cubicos que pueden afectar a otra forma. Puedes solo crear redondeados y líquidos 'mercuriales', o 'arcillosos', formas que existen *procesalmente*, que son computados dicamicamente. Usa los Meta Objects para crear efectos especiales o como base para modelar.

Los Meta Objects son también llamados *superficies implícitas*, para precisar que no son *explícitamente* definidos por vértices (aclopamientos) o puntos de control (como son las superficies).

Los Meta Objects definen una *estructura dirigida* que se pueda considerar como la fuente de un campo estático. El campo puede ser positivo o negativo y por lo tanto el campo generado por las estructuras vecinas que dirigen se pueden atraer o rechazar.

La superficie implícita se define como la superficie donde el campo 3D por todas las estructuras directas que asumen un valor dado. Por ejemplo un Meta Ball, es un punto que dirige la estructura, genera un campo isotrópico alrededor de él y los valores constantes de la superficie de campo son esferas centradas en el punto que dirige. Dos bolas vecinas del Meta interactuan y, si estan lo suficientemente cerca, las dos superficies se combinan en una sola surperficie ([Figure 8-1](#)).

Figure 8-1. Dos Metaballs



De hecho, los Meta Objects no son nada mas que formulas matematicas que realizan operaciones logicas unas con otras (Y, O), y que pueden ser añadidas o substraídas una con otra. Este metodo es tambien llamado CSG, Geometria Solida Constructiva. Porque de esta naturaleza matematica, CSG usa una pequeña parte de memoria, pero requiere de lotes de CPU para computar. Para optimizar esto las superficies implicitas son *poligonizadas*. El área completa de CSG esta dividida en una rejilla 3D, y para cada *borde* de la rejilla se crea un cálculo, y (donde es mas importante) la formula tiene un momento crucial, una 'cima' para que la *poligonización* sea creada.

Para crear un Meta Object presiona **ESPACIO** y selecciona **Add>>MBall**. Tu puedes seleccionar las formas de la base: Esfera, Tubo, Plano, Elipsoide y Cubo.

Las MetaBalls tienen un punto que dirige la estructura, los MetaTubes tiene un segmento que dirige la estructura, los MetaPlanes un plano y los MetaCubes un cubo. La estructura subyacente llega a ser evidente como bajas los valores *Wiresize* y aumentes el *Threshold* en el Panel de *Meta Ball*.

Cuando estas en modo edición, tu puedes mover y escalar los Meta Objects como tu desees. Esta es la mejor manera de construir formas estaticas - en oposición a las animadas- . Los Meta Objects pueden tambien influenciar *fuera* del Modo de Edición. Cuando estas fuera del Modo de Edición tu tienes mucha más libertad; las bolas pueden rotar o moverse y consiguen cada *transformación* de los Objetos Padre. Este metodo requiere más tiempo de calculo y debido a esto es algo lento.

Las siguientes reglas describen la relación entre los Meta Objects:

- Todos los Meta Objects con el mismo nombre de 'familia' (el nombre sin el numero) influencia

con los otros. Por ejemplo "MBall", "MBall.001", "MBall.002", "MBall.135". Observa aquí que no estamos hablando del nombre del bloque de MetaBall ObData.

- Los objetos con el nombre de familia *sin* un numero determinado la base, la resolución, y la transformación del poligono. Tambien tiene el area del material y la textura que será referido como *base* del Meta Object.

Solo un material puede ser usado por un sistema Meta Object . Ademas, los Meta Objects serán separados del area de la textura; esto normaliza las coordenadas de los vertices. Normalmente el area de la textura es identica a los *limites* de todos los vertices. El usuario puede forzar un area de textura con el comando **TKEY** (fuera del modo edición).

De hecho como la base dicta la poligonización del objeto implica eso, por lo que si tenemos dos Meta Objects y movemos uno de ellos veremos la poligonización de un objeto *inbase* durante el movimiento, sin importar cual de los dos objetos se está moviendo realmente.

El Panel de Control del *Meta Ball* ofrece pocas posibilidades. Si está en Modo Objeto, solo estará el Panel ahí presente. Puedes definir la dimensión media de ambas poligonizaciones en la vista 3D via Panel *Wiresize* , y la representación del tiempo via Panel *Rendersize*. Cuanto más bajo sea el suavizado del Meta Object más lento será la computación.

El Panel *Threshold* es un ajuste importante para MetaObjects. Controla el 'nivel de campo' en el cual se computa la superficie. Para tener un control mas fino, cuando estamos en modo edición, el Panel *Stiffness* de *Meta Ball Tools* permite que aumentes o reduzcas el campo de influencia del MetaObject.

En este ultimo panel puedes cambiar el tipo de Meta Object y fijarlo negativo (que es substractivo, más bien que añadido) con otro Meta Objects de las mismas características.

[Prev](#)

Herramienta Diezmador
(Decimator)

[Home](#)
[Up](#)[Next](#)

Curvas y Superficies

Chapter 9. Curvas y Superficies

Las curvas y las superficies son objetos como las mallas, pero se diferencian en el punto de que son expresadas en terminos de funciones matemáticas, como una serie de puntos.

Blender implementa curvas y superficies Bezier y Non uniform Rational B-Splines (NURBS). Ambas, aunque siguen diferentes leyes matemáticas, son definidas en términos de un conjunto de "vértices de control" que definen un "polígono de control". La manera en que las curvas y las superficies son interpoladas (Bezier) o atraídas (NURBS) por los puntos de control pueden parecer similares, en primera instancia, a la subdivisión de superficies Catmull-Clark.

Cuando se las compara con las mallas, las curvas y las superficies tienen ventajas y desventajas. Como las curvas están definidas por una menor cantidad de datos, producen buenos resultados usando menos memoria durante el modelado, mientras que la demanda de memoria se incrementa cuando renderizamos.

Algunas técnicas de modelado, tales como la extrusión de un perfil "a lo largo de un camino", son solo posibles con las curvas. Pero por otro lado el gran control disponible a través de los vértices en una malla, no es posible con las curvas.

Hay veces que es más apropiado utilizar las curvas y superficies en lugar de mallas y otras veces es al contrario. Si ha leído el capítulo previo y lee este será capaz de elegir cuando usar mallas y cuando usar curvas.

9.1. Curvas

Relevante a Blender v2.31

Esta sección describe las curvas Bèizer y las curvas NURBS, mostrando de ambas un ejemplo de como funcionan.

9.1.1. Curvas Bèziers

Las curvas Bézier son posiblemente las más usadas para el diseño de letras o de logotipos. También son usadas frecuentemente en animación, como caminos para el movimiento de los objetos o como curvas IPO para variar las propiedades de los objetos en función del tiempo.

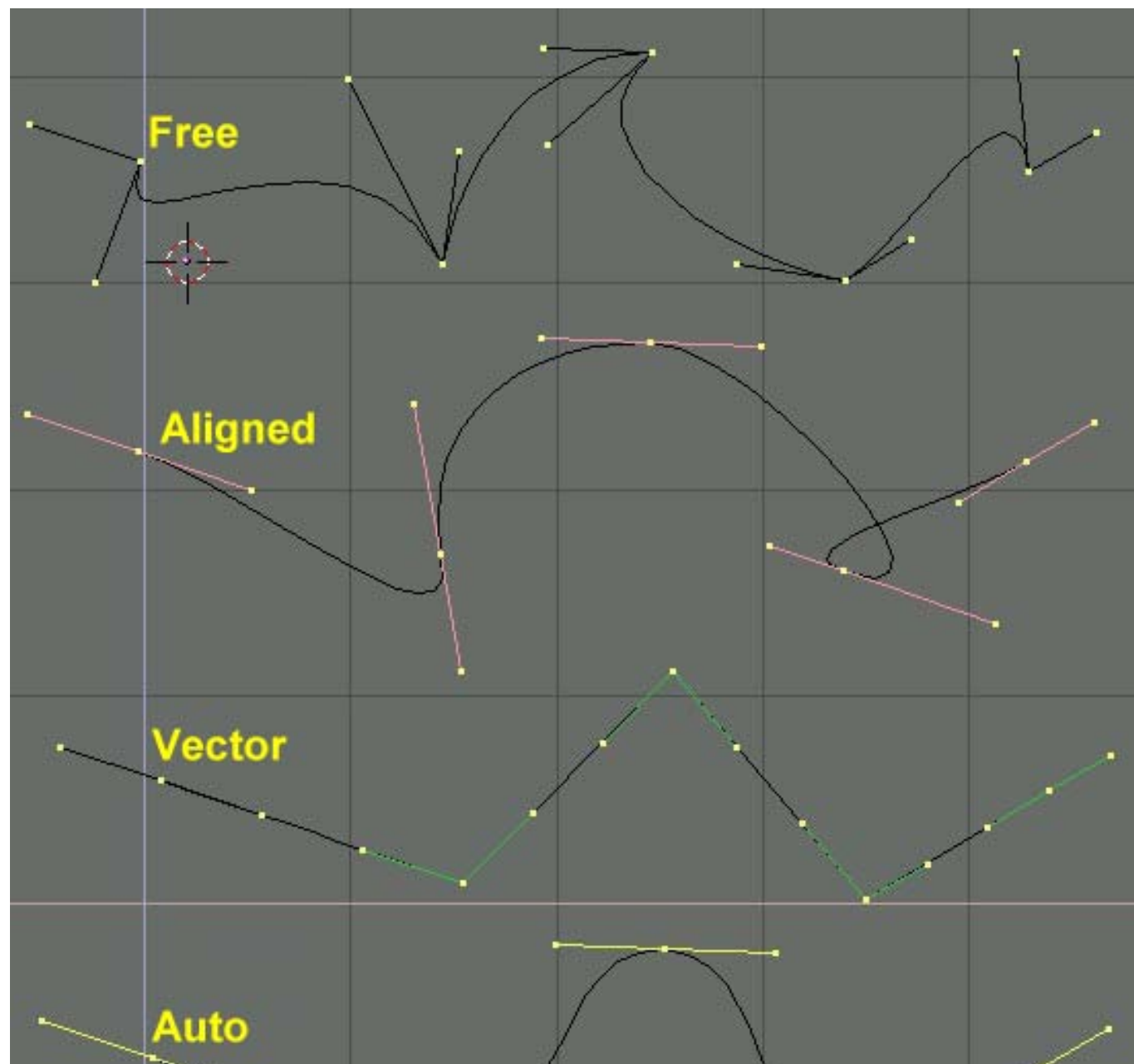
Un punto de control (vértice) de una curva Bézier consta de un punto y dos agarraderas. El punto, en el medio, es usado para mover el punto de control en su totalidad; Al seleccionar dicho punto también seleccionamos las dos agarraderas, y le permitirá mover el vértice completamente. Seleccionando y moviendo una o las dos agarraderas podrá variar la forma de la curva.

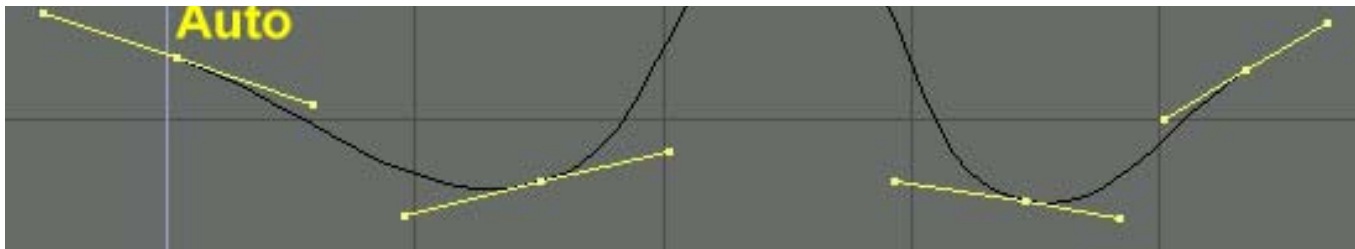
Una curva Bézier es tangente al segmento de línea que va desde el punto a la agarradera. La suavidad (steepness) de la curva es controlada con la longitud de las agarraderas.

Hay cuatro tipos de agarraderas ([Figure 9-1](#)):

- Free Handle (Agarradera libre) (en negro). Esto puede usarse de la manera en la que desee. Hotkey: la **Tecla H** (cambia entre libre y alineada);
- Aligned Handle (Agarradera alineada) (en morado). Las agarraderas siempre están en línea recta. Hotkey: la **Tecla H** (cambia entre libre y alineada);
- Vector Handle (Agarradera vector) (en verde). Ambas partes de una agarradera siempre apuntan a la agarradera previa o a la siguiente. Hotkey: la **Tecla V**;
- Auto Handle (Agarradera automática) (en amarillo). Esta agarradera tiene automáticamente asignada por Blender una longitud y dirección para asegurar el resultado más suave. Hotkey: **SHIFT-H**.

Figure 9-1. Tipos de agarraderas para curvas Bézier





Las agarraderas pueden ser *movidas*, *rotadas* y *escaladas* exactamente como se haría con un vértice en una malla.

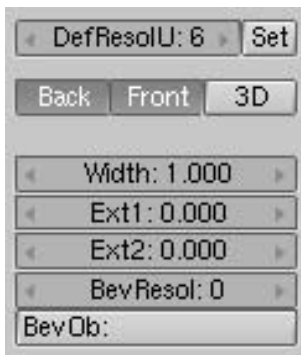
Tan pronto como las agarraderas se mueven, el tipo se modifica automáticamente:

- Las agarraderas automáticas se convierten en alineadas;
- Las agarraderas vector se convierten en libres.

Aunque la curva Bézier es un objeto matemáticamente continuo tiene que ser representado de una forma discreta desde el punto de vista del renderizado.

Esto se consigue configurando una propiedad, la *resolución*, la cual define el número de puntos que son computados entre cada par de puntos de control. La *resolución* puede ser configurada separadamente para cada curva Bézier ([Figure 9-2](#)).

Figure 9-2. Configurando la resolución de la curva Bézier.



9.1.2. Curvas NURBS

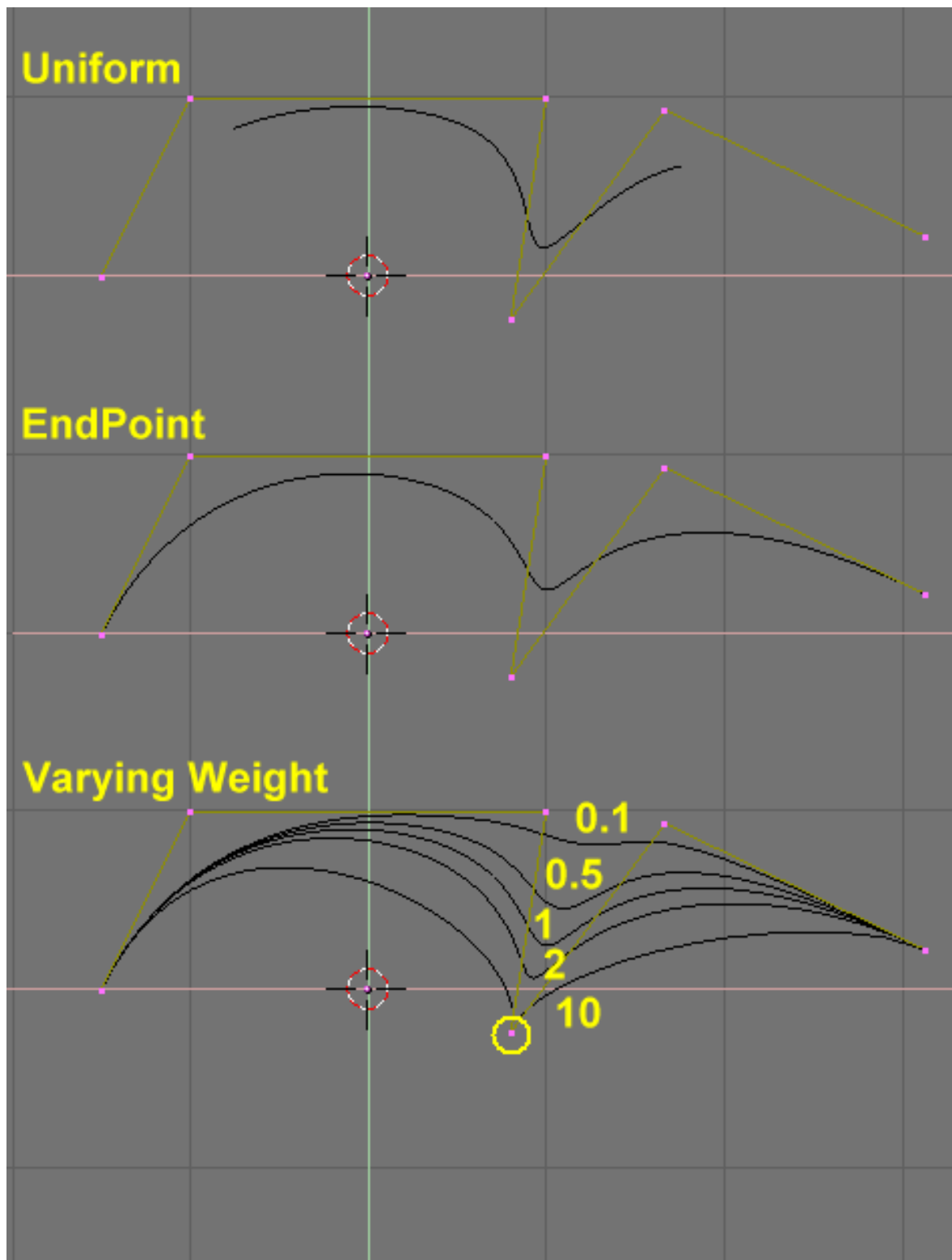
Las curvas NURBS son definidas como polinomios racionales, y son más generales, estrictamente hablando, que las curvas Bézier y las B-Splines ya que son capaces de seguir exactamente cualquier contorno. Por ejemplo un círculo Bézier es una *aproximación* polinomial de un círculo, y esta aproximación es evidente, mientras que un círculo NURBS es *exactamente* un círculo. Las curvas NURBS tienen un gran conjunto de variables, que le permitirán crear formas matemáticamente puras ([Figure 9-3](#)). Po otro lado, trabajar con ellas requiere un poco más de teoría:

Figure 9-3. Botones de control de las curvas Nurbs.



- *Knots (Nudos)*. Las curvas Nurbs tienen un vector *nudo*, es decir, una fila de números que especifica la definición paramétrica de la curva. Dos parámetros son importantes para esto. *Uniform (Uniforme)* produce una división uniforme en las curvas cerradas, pero cuando es usado con curvas abiertas se obtienen finales de curva aleatorios, que son bastante difíciles de predecir donde estarán. *Endpoint (Punto final)* configura el nudo de tal manera que el primero y el último vértice son siempre parte de la curva, lo que hace mucho más fácil situarla;
- *Order (Orden)*. El *orden* es el grado de la ecuación de la curva. Orden '1' es un punto, orden '2' es lineal, orden '3' es cuadrática, y así. Siempre use un orden 5 para curvas "camino" ya que se mostrará bastante fluido bajo todas las circunstancias, sin producir irritantes discontinuidades en el movimiento. Matematicamente hablando esto es el orden del numerador y denominador del polinomio que define la NURBS;
- *Weight (Peso)*. Las curvas Nurbs tienen un 'peso' en cada vértice - el grado en el que un vértice participa en la tensión de la curva.

Figure 9-4. Configuración de las Nurbs para controlar el polígono y los pesos.



[Figure 9-4](#) muestra la configuración del vector Nudo así como el efecto de variar el peso de un simple nudo. Como con las curvas Bézier, la resolución puede ser configurada independientemente en cada curva.

9.1.3. Un ejemplo funcional

Las herramientas de Blender para las curvas proporcionan una rápida y sencilla manera de construir asombrosos logotipos y textos extruidos. Usaremos esas herramientas para transformar un dibujo del logotipo en dos dimensiones en un modelo en 3D.

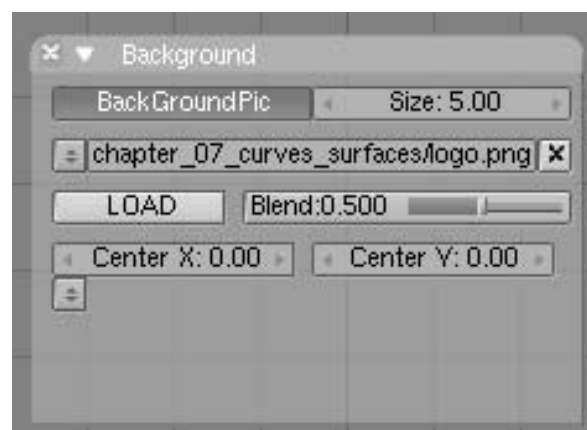
[Figure 9-5](#) muestra el diseño del logotipo que estamos construyendo.

Figure 9-5. El logotipo



En primer lugar, importaremos nuestro dibujo original para poder utilizarlo como guía. Blender soporta los formatos de imagen TGA, PNG, and JPG entre otros. Para cargar la imagen, seleccionaremos la entrada de menú *View>>Background Image...* de la ventana 3D que esté usando. Una ventana transparente aparecerá, permitiéndole seleccionar una imagen para usarla como fondo. Active el botón *BackGroundPic (Imagen de fondo)* y use el botón *LOAD (CARGAR)* para localizar la imagen que quiere usar como guía ([Figure 9-6](#)). Puede seleccionar el grado de transparencia de la imagen de fondo con el deslizador *Blend*.

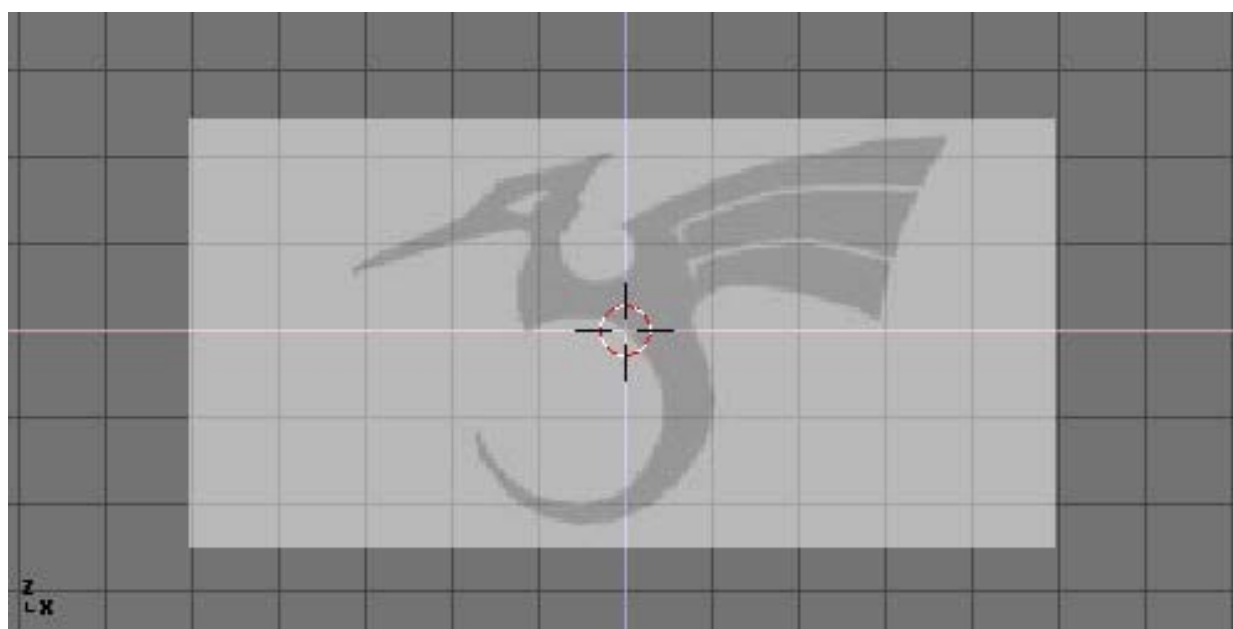
Figure 9-6. Configuración de la ventana 3D.



Haga desaparecer el panel emergente con la **Tecla ESC** o presionando el botón *X* en la cabecera del panel ([Figure 9-7](#)). Puede ocultar la imagen de fondo cuando haya terminado volviendo a sacar el panel y

deseleccionando el botón *BackGroundPic*.

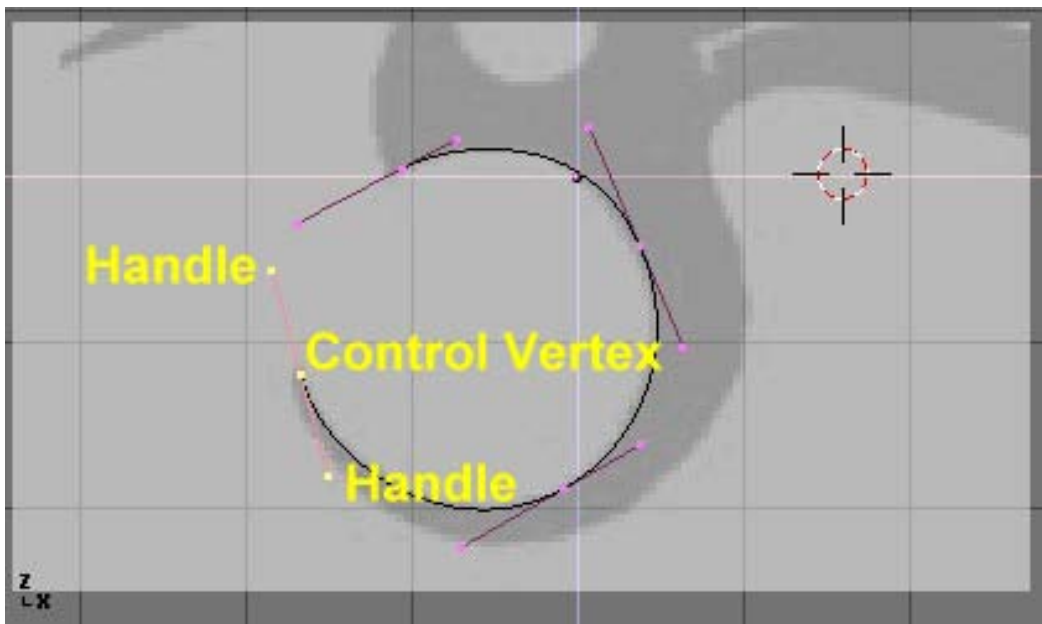
Figure 9-7. Dibujo del logo cargado de fondo



Añada una nueva curva con **SPACE**>>*Curve*>>*Bezier Curve*. Un segmento de curva aparecerá y Blender entrará en modo edición. Moveremos y añadiremos puntos para crear una forma cerrada que describa el logotipo que está intentando trazar.

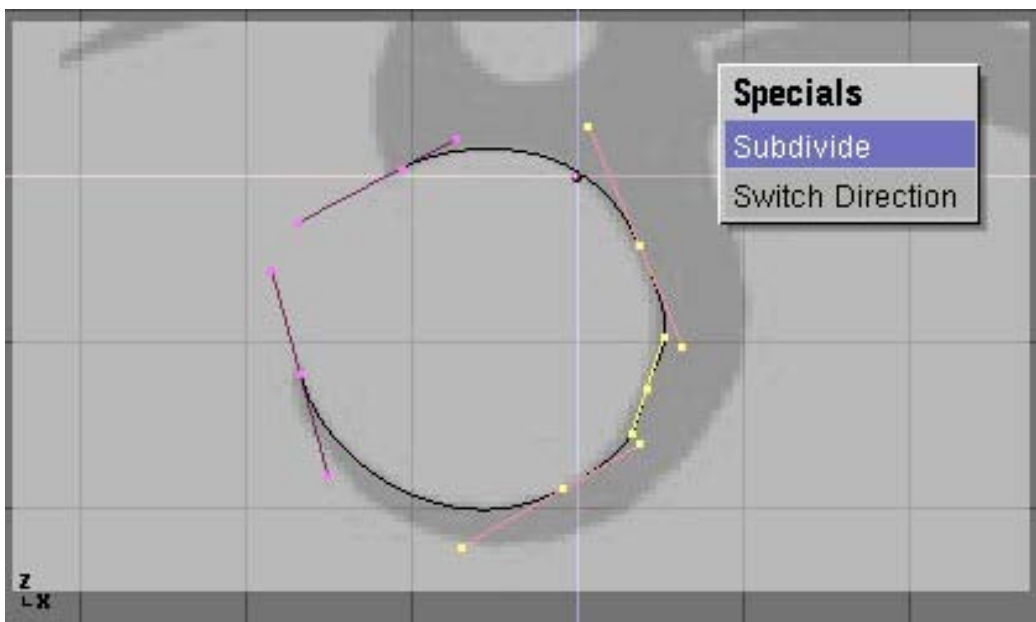
Puede añadir puntos a la curva seleccionando uno de los dos puntos finales de curva, y entonces manteniendo pulsado **CTRL** y pinchando en **BIR**. Note que el nuevo punto estará conectado al punto seleccionado anteriormente. Una vez que el punto haya sido añadido, puede moverse seleccionando un vértice de control y presionando la **tecla G**. Puede cambiar el ángulo de la curva arrastrando y moviendo las agarraderas asociadas a cada vértice ([Figure 9-8](#)).

Figure 9-8. Agarraderas Bézier



Puede añadir un nuevo punto entre dos puntos existentes seleccionando ambos y presionando la **Tecla W** > > *Subdivide* ([Figure 9-9](#)).

Figure 9-9. Añadiendo un punto de control.



Los puntos pueden ser eliminados seleccionándolos y presionando la **Tecla X** > > *Selected*. Para cortar una curva en dos, seleccione dos vértices de control adyacentes y presione la **Tecla X** > > *Segment*.

Para hacer las esquinas afiladas, seleccione un vértice de control y presione la **Tecla V**. Notará que el color de las agarraderas cambia de morado a verde ([Figure 9-10](#)). En este punto, puede mover las agarraderas para ajustar la forma en la que la curva entra y sale del vértice de control ([Figure 9-11](#)).

Figure 9-10. Agarraderas Vector (en verde).

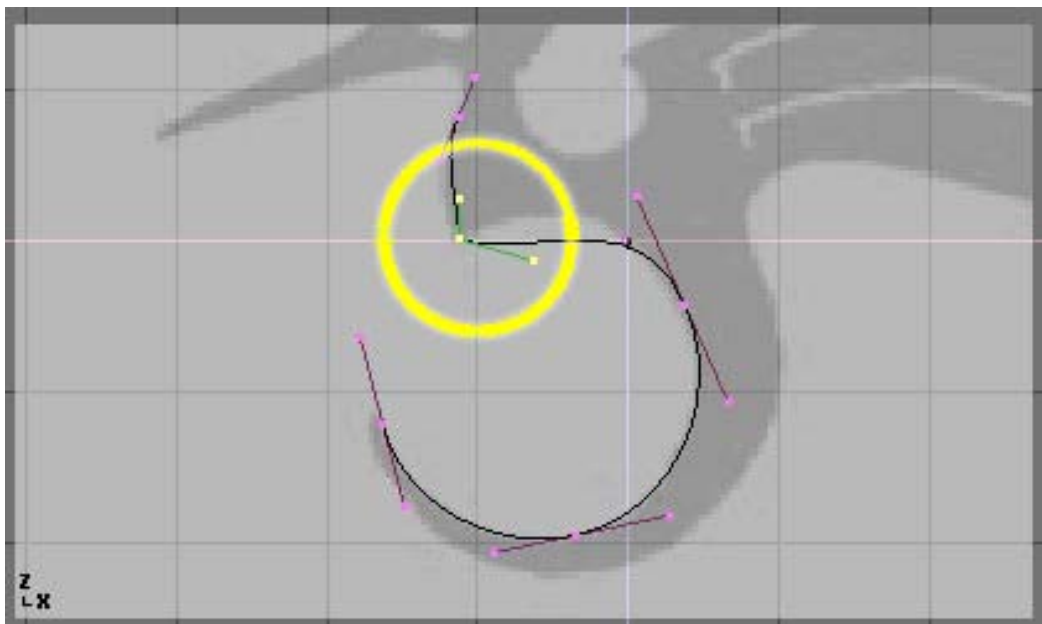
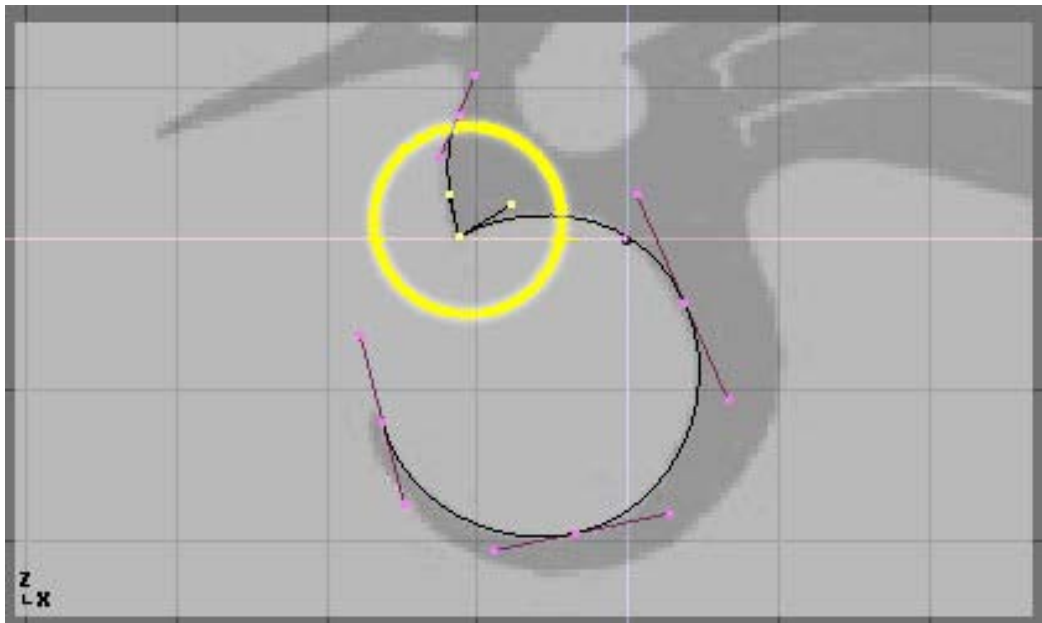
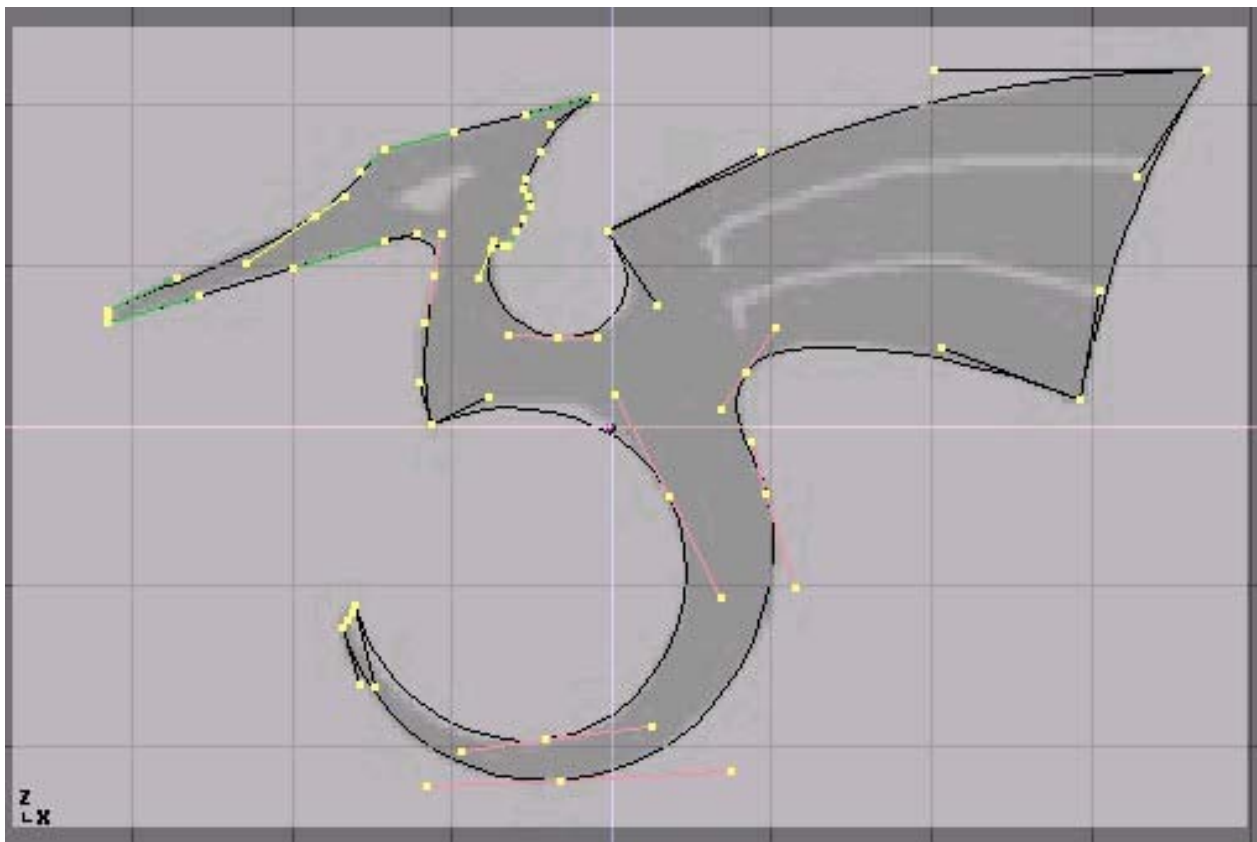


Figure 9-11. Agarraderas libres (en negro).



Para cerrar la curva , seleccione al menos uno de los vértices de control de la curva y presione la **Tecla C**. Esto conectará el último punto de la curva con el primero ([Figure 9-12](#)). Puede ser necesario manipular las agarraderas para obtener la forma que desee.

Figure 9-12. El final de linea acabado.

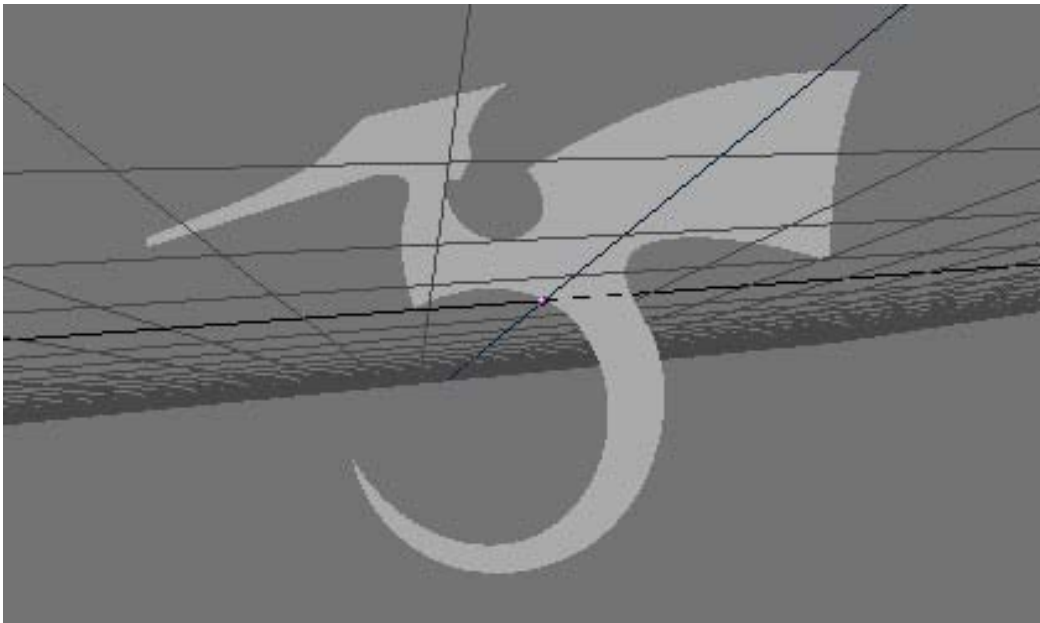


Saliendo del modo Edición con **TAB** y entrando en el modo sombreado con la **Tecla Z** debería mostrar que la curva se renderiza como una forma sólida ([Figure 9-13](#)). Nosotros queremos poner algunos agujeros dentro para representar los ojos y las alas del dragon.

Tip Superficies y agujeros

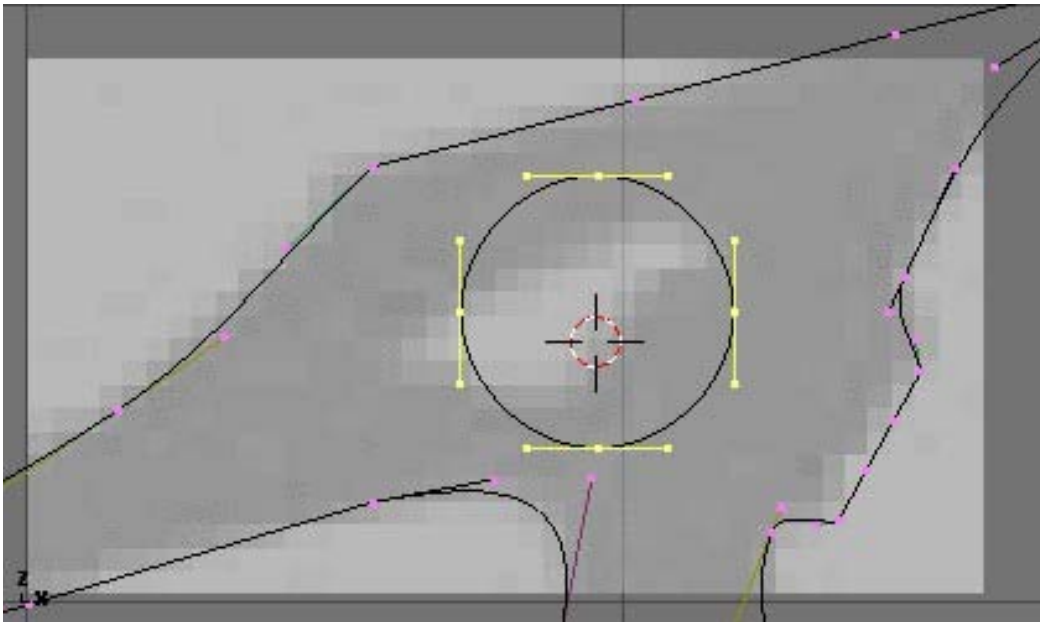
Cuando trabajamos con curvas, Blender automáticamente detecta agujeros en la superficie y los maneja de acuerdo a las siguientes reglas. Una curva cerrada siempre es considerada el límite de una superficie y será renderizado como una superficie sólida. Si una curva cerrada es incluida completamente dentro de otra, la curva interior se sustrae de la exterior, creando efectivamente un agujero.

Figure 9-13. Logotipo sombreado.



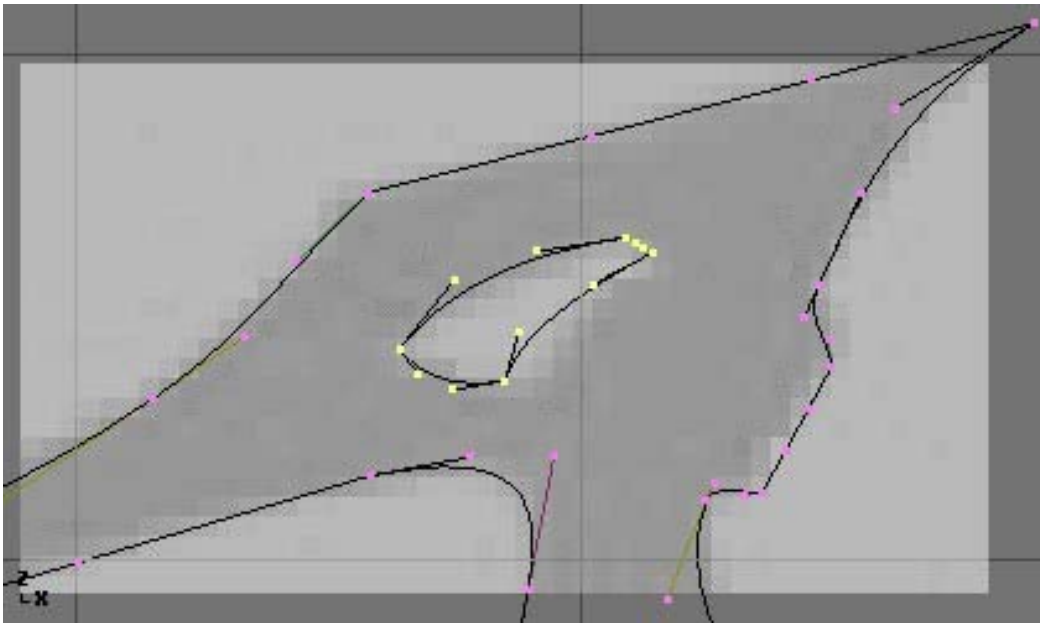
Regrese al modo de alambre con la **Tecla Z** y entre de nuevo al modo Edición con **TAB**. Mientras permanezca en el modo Edición, añada una curva 'círculo' con **SPACE>>Curve>>Bezier Circle** ([Figure 9-14](#)). Escale negativamente hasta un tamaño adecuado con la **tecla S** y muevalo con la **tecla G**.

Figure 9-14. Añadiendo un círculo.



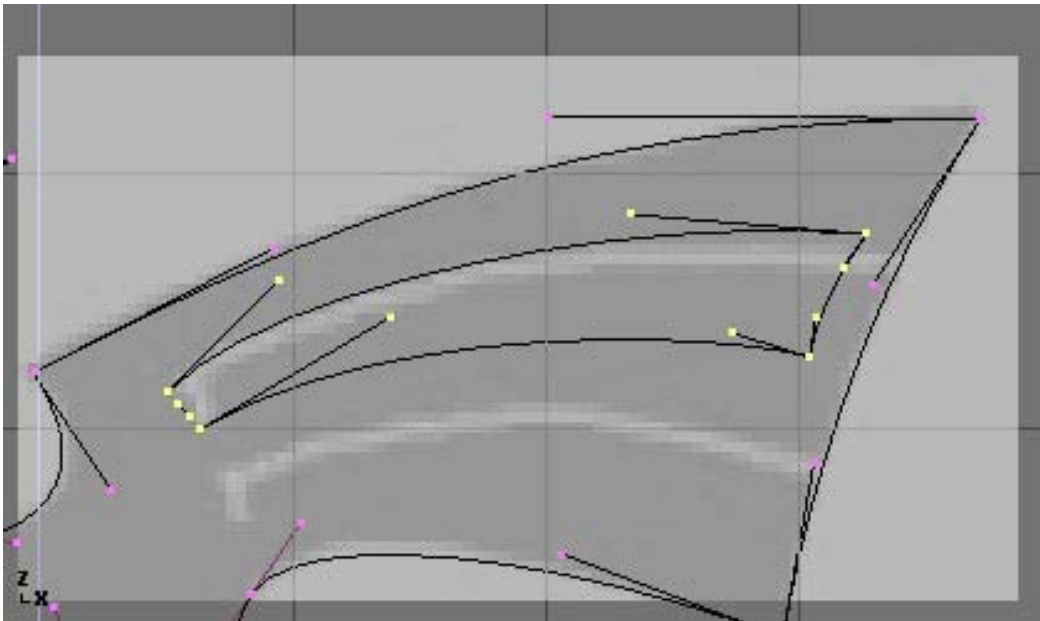
Dé forma al círculo usando las técnicas que hemos aprendido ([Figure 9-15](#)). Recuerda añadir los vértices al círculo con la **Tecla W>>Subdivide**.

Figure 9-15. Definiendo un ojo.



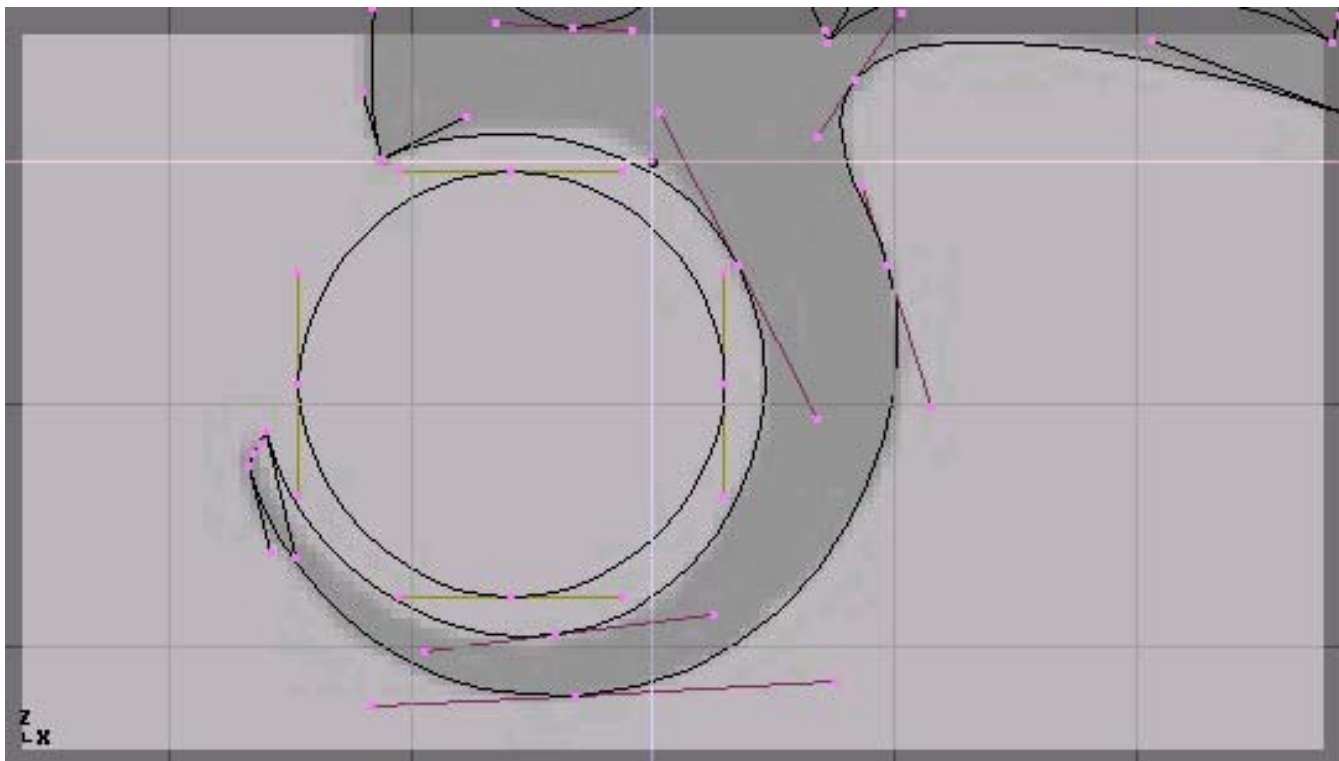
Cree un ala añadiendo un círculo Bézier, convirtiendo todos los puntos a esquinas afiladas, y ajustandolos tanto como sea necesario. Puede duplicar esta figura para crear la segunda ala y de esta forma salvar algo de tiempo. Para hacer algo así, asegurese que ningún punto esté seleccionado, entonces mueva el cursor sobre uno de los vértices en la primera ala y seleccione los puntos que estén enlazados con la **Tecla L** ([Figure 9-16](#)). Duplique la selección con **SHIFT-D** y mueva los nuevos puntos a su posición.

Figure 9-16. Definiendo las alas.



Para añadir más geometría que no esté conectada con el cuerpo principal (situando un círculo en la cola curvada del dragón, por ejemplo), use la combinación **SHIFT-A** para añadir más curvas como se muestra en [Figure 9-17](#).

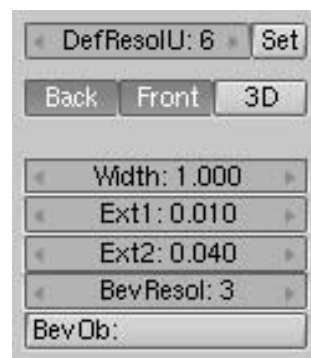
Figure 9-17. Colocación del círculo en la cola.



Ahora que tenemos la curva, necesitamos configurar su biselado (bevel) y su espesor. Con la curva seleccionada, vaya a los botones de Edición (**Tecla F9**) y localice el panel *Curves and Surface (curvas y superficies)*. El parámetro *Ext1* configura el espesor de la extrusión mientras que el parámetro *Ext2* configura el tamaño del biselado. *BevResol* configura como de afilado o curvado será el biselado.

[Figure 9-18](#) muestra la configuración utilizada para extruir esta curva.

Figure 9-18. Configuración del Bevel



Tip De la Curva a la Malla

Para optimizar operaciones de modelado más complejas, convierta la curva a malla con **ALT-C** > > *Mesh*. Note que esta es una operación de un solo sentido: No puede convertir una malla en una curva.

Cuando su logotipo esté completo, puede añadir materiales y luces y hacer un bonito render ([Figure 9-19](#)).

Figure 9-19. Renderizado final.



■

[Prev](#)
Los Meta Objects

[Home](#)
[Up](#)

[Next](#)
Superficies

9.2. Superficies

Relevante a Blender v2.31

Las Superficies actualmente son una extensión de las curvas NURBS. En Blender son consideradas como un tipo obData separado.

Mientras una curva produce una interpolación unidimensional, las Superficies presentan una segunda dimensión adicional. La primera dimensión es U, como para las curvas, y la segunda es V. Una rejilla bidimensional de puntos de control define la forma de esas superficies NURBS.

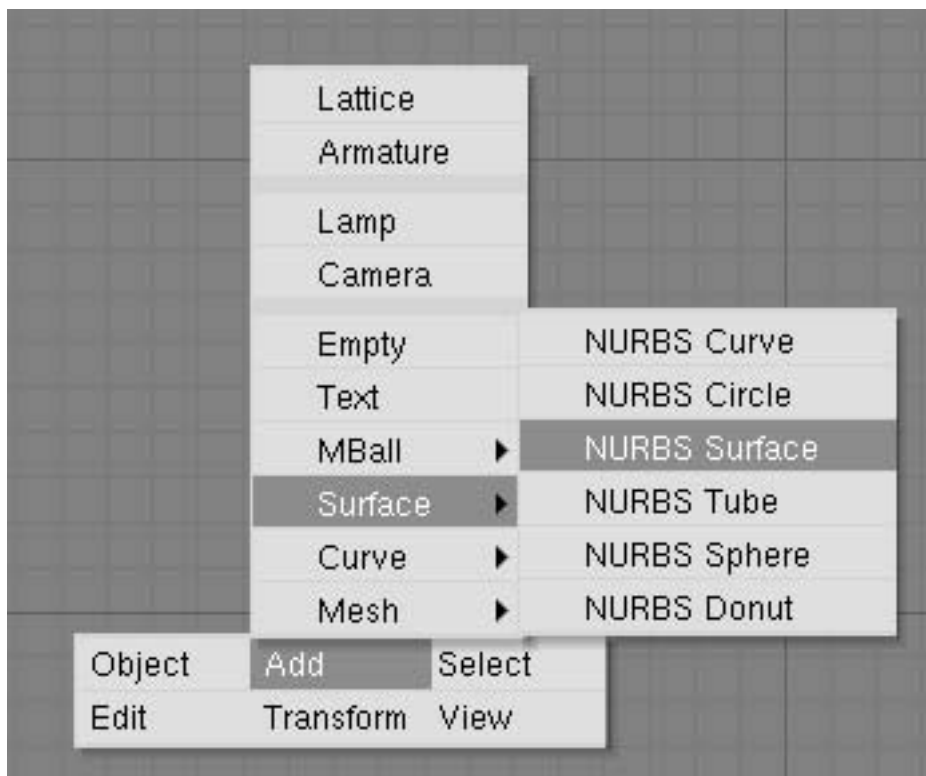
Utilice las Superficies para crear y modificar fluidamente las superficies curvadas. Las Superficies pueden ser cíclicas en ambas direcciones, permitiéndole fácilmente crear una forma de 'donut', y pueden ser dibujadas como 'sólidos' en Modo Edición (zbuffered, con iluminación OpenGL). Esto hace que trabajar con Superficies sea bastante sencillo.

Note Actualmente Blender tiene un conjunto de herramientas básicas para superficies, que permiten crear agujeros y superficies blandas. Se incrementará las funciones de esta área en futuras versiones.

Puede tomar una de las 'primitivas' de Superficie del menú ADD (Añadir) como punto de inicio ([Figure 9-20](#)). Observe que puede elegir Curve (Curva) y Circle (Círculo) en el menú Superficie. Esto es posible porque las curvas NURBS son intrínsecamente superficies NURBS, simplemente tienen una dimensión restringida.

Note Una verdadera curva NURBS y una curva 'superficie' NURBS no son intercambiables, como se dará cuenta al realizar el proceso de extrusión en la sección 'skinning' que verá más adelante.

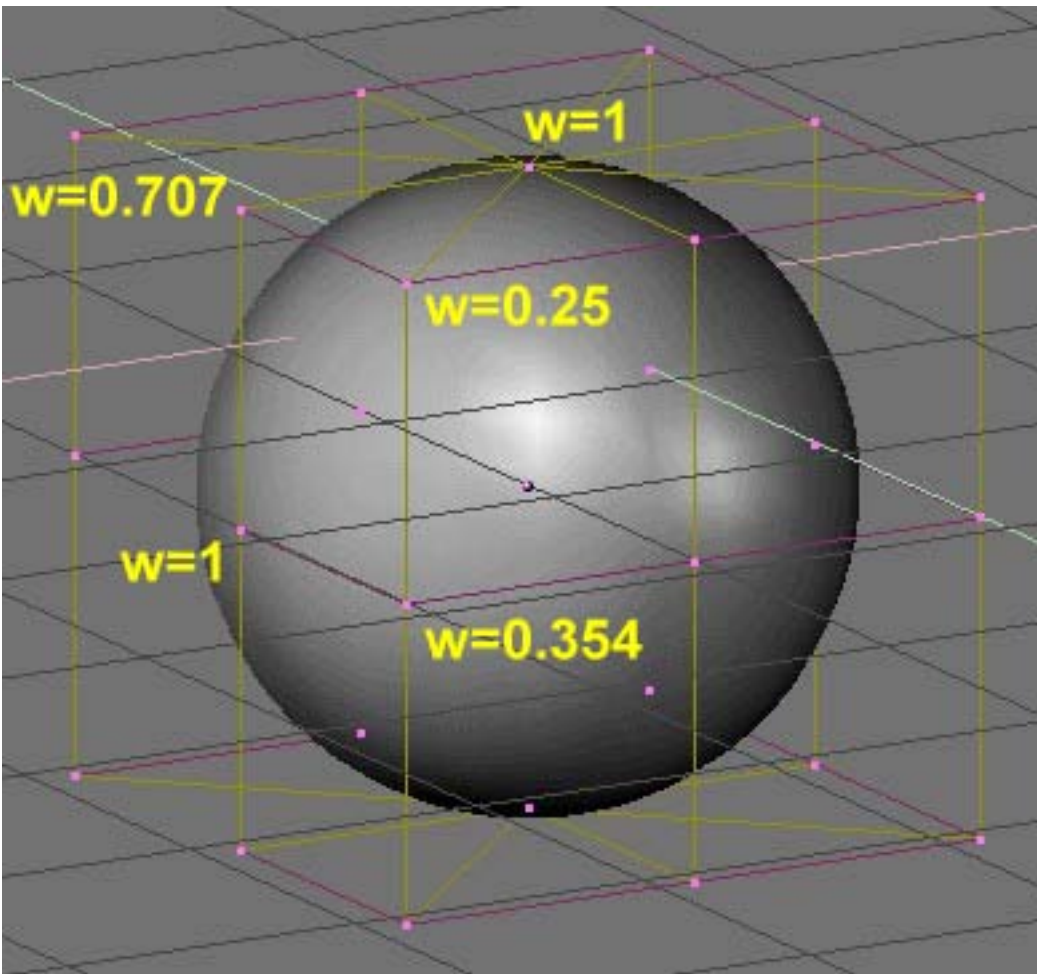
Figure 9-20. El menú añadir superficie.



Cuando añada una curva 'superficie' puede crear una verdadera superficie simplemente extruyendo la curva (**Tecla E**). Cada arista de la superficie puede entonces extruirse como desee para dar forma al modelo. Utilice la **Tecla C** para conseguir que sea ciclica la dirección U o la V. Aseguresé de configurar los nudos a *Uniform (Uniforme)* o *Endpoint (Punto final)* con uno de los botones del panel *Herramientas de curva (Curve Tools)* de los botones de Edición.

Cuando trabajamos con superficies, es práctico trabajar con una columna o fila de vértices. Blender proporciona una herramienta de selección para esto: **SHIFT-R**, "Select Row (seleccionar columna)". Comenzando por el último vértice seleccionado, una completa columna de vértices se *extiende* seleccionada en la dirección 'U' o 'V'. Elija "seleccionar columna" de nuevo con el mismo vértice para intercambiar entre la dirección 'U' y 'V'.

Figure 9-21. Una superficie esférica



Las NURBS pueden crear formas puras tales como círculos, cilindros y esferas (pero note que un círculo Bèzier no es un círculo puro). Para crear círculos puros, esferas o cilindros, debe configurar los pesos de los vértices. Esto no es intuitivo y debería leer más sobre las NURBS antes de intentar esto.

Basicamente, para producir un arco circular a partir de una curva con tres puntos de control, los puntos finales deben de tener un peso unitario, mientras el peso del punto de control central debe ser igual a la mitad del coseno de la mitad del ángulo entre los segmentos creados al ensamblar los puntos. [Figure 9-21](#) muestra esto para una esfera. Los tres números estandar son incluidos por defecto en el panel *Curve Tools* (*Herramientas de curva*) en los botones de edición ([Figure 9-22](#)).

Note Para ver cual es el peso de un vértice seleccionado, presione la **Tecla N**.

Figure 9-22. Pesos preconfigurados

Convert	Make Knots	
Poly	Uniform U	V
Bezier	Endpoint U	V
Bspline	Bezier U	V
Cardinal	Order U: 2	V: 2
Nurb	Resol U: 12	V: 12
Set Weight	Weight: 1.000	1.0
	sqrt(2)	0.25
	sqrt(0.5)	

9.3. Texto

Relevante a Blender v2.31

Figure 9-23. Ejemplos de texto.



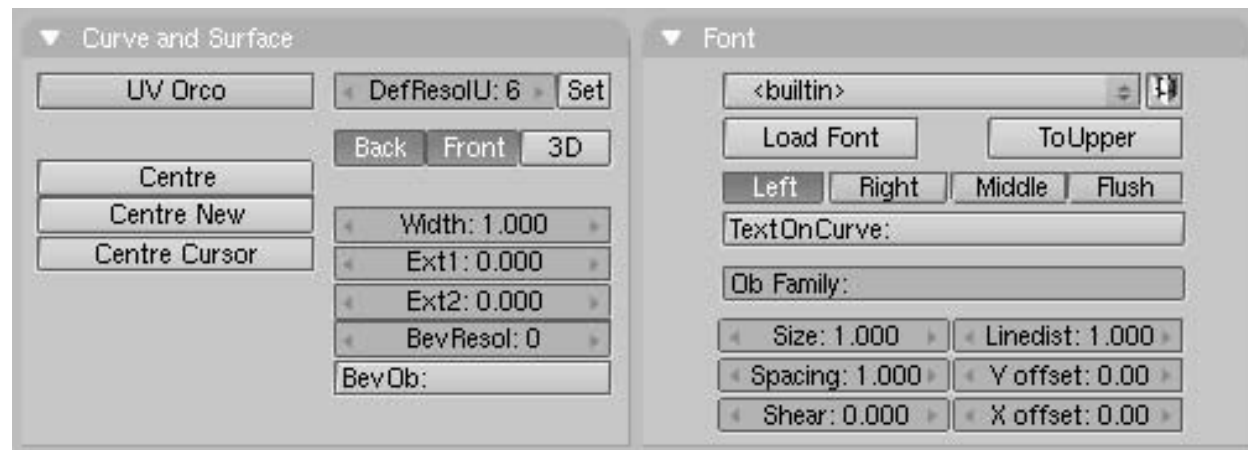
El texto es un tipo especial de curva en Blender. Blender tiene su propio tipo de fuente interna pero tambien puede usar fuentes externas, incluyendo fuentes PostScript Type 1 y fuentes True Type ([Figure 9-23](#)).

Abra Blender o comience una escena nueva presionando **CTRL-X**. Añada un objeto "Texto" con la caja de herramientas (**SPACE**>>**Add**>>**Text**). Puede editar el texto con el teclado en modo Edición; un cursor le

mostrará su posición en el texto. Cuando deje el modo Edición con **TAB**, Blender rellenará el "texto-curva", produciendo un objeto solido que es renderizable inmediatamente.

Ahora vaya a los botones de Edición (**F9**) [Figure 9-24](#).

Figure 9-24. Botones de edición de texto.



Como puedes ver en el panel *Font*, Blender, por defecto, utiliza su propia fuente *interna* cuando crea un objeto "texto". Ahora pinche en *Load Font* (*cargar fuente*). Navegue en la ventana de archivos para encontrar un directorio que contenga fuentes PostScript Type 1 o True Type y cargue una nueva fuente. (Puede descargarse varias fuentes PostScript de internet, y además Microsoft Windows incluye muchas fuentes True Type de su propiedad, aunque en este caso tenga cuidado ya que algunas de ellas ¡tienen copyright!).

Pruebe con algunas fuentes. Una vez que haya cargado una, puede usar el menú de botones para cambiar la fuente de un objeto "texto".

Hast ahora lo que tenemos es un objeto plano. Para añadir algo de profundidad, podemos usar los botones *Ext1:* y *Ext2:* en el panel *Curve and Surface* tal y como hicimos con las curvas.

Use la opción *TextOnCurve:* (*Texto en curva*) para conseguir que el texto siga una curva bidimensional. Use los botones de alineación, que se encuentran encima del campo de texto *TextOnCurve:* en el panel *Font*, para alinear el texto con la curva.

Una función de blender particularmente poderosa es la posibilidad de que un objeto "texto" sea convertido con **ALT-C** a una curva Bézier, lo cual permite le permitirá editar la forma de cada caracter en la curva. Esto es especialmente util cuando se crean logotipos o cuando se producen letras a nuestro gusto. La transformación de texto en curva es irreversible y, por supuesto, una posterior transformación de curva en malla también es posible.

9.3.1. Caracteres especiales

Normalmente, un objeto fuente cmoienza con la palabra "Text", la cual puede ser borrada facilmente con **SHIFT-BACKSPACE**. En modo Edición, el objeto texto se modifica al meter algo por teclado. Casi todas las hotkeys están deshabilitadas. El cursor puede ser movido con las flechas de dirección. Use **SHIFT-**

FLECHAIZQUIERDA y **SHIFT-FLECHADERECHA** para mover el cursor al final de la línea o al principio o al final del texto.

Casi todos los caracteres especiales están disponibles. Este es un resumen de esos caracteres:

- **ALT-c:** Copyright
- **ALT-f:** Florin Holandés
- **ALT-g:** Grados
- **ALT-l:** Libra Británica
- **ALT-r:** Marca registrada
- **ALT-s:** S Alemana
- **ALT-x:** Símbolo de multiplicar
- **ALT-y:** Yen Japonés
- **ALT-1:** Un pequeño 1
- **ALT-2:** Un pequeño 2
- **ALT-3:** Un pequeño 3
- **ALT-?:** Signo de interrogación Español
- **ALT-!:** Signo de exclamación Español
- **ALT->:** Un doble >>
- **ALT-<:** Un doble <<

Todos los caracteres de su teclado deberían funcionar, incluyendo vocales unidas y demás. Si necesita caracteres especiales (tales como letras acentuadas, que no están en los teclados estadounidenses) puede generar muchos de ellos usando una combinación de 2 caracteres. Para hacer esto, presione **ALT-BACKSPACE** junto con la deseada combinación para producir un carácter especial. Algunos ejemplos son dados a continuación.

- **Tecla A, ALT-BACKSPACE, TILDE:** ã
- **Tecla A, ALT-BACKSPACE, COMA:** à
- **Tecla A, ALT-BACKSPACE, ACENTO:** á
- **Tecla A, ALT-BACKSPACE, Tecla O:** å
- **Tecla E, ALT-BACKSPACE, COMILLAS:** ë
- **Tecla O, ALT-BACKSPACE, BARRA:** ø

También puede añadir ficheros ASCII a un objeto "texto". Guarde el fichero como **/tmp/.cutbuffer** y presione **ALT-V**.

Por otro lado puede escribir su texto desde una ventana de texto de Blender, cargando el texto en dicha ventana, o pegandolo desde el portapapeles y presionando **ALT-M**. Esto crea un nuevo objeto "texto" con el contenido del buffer (hasta 1000 caracteres)

9.4. Extruir a lo largo de una trayectoria

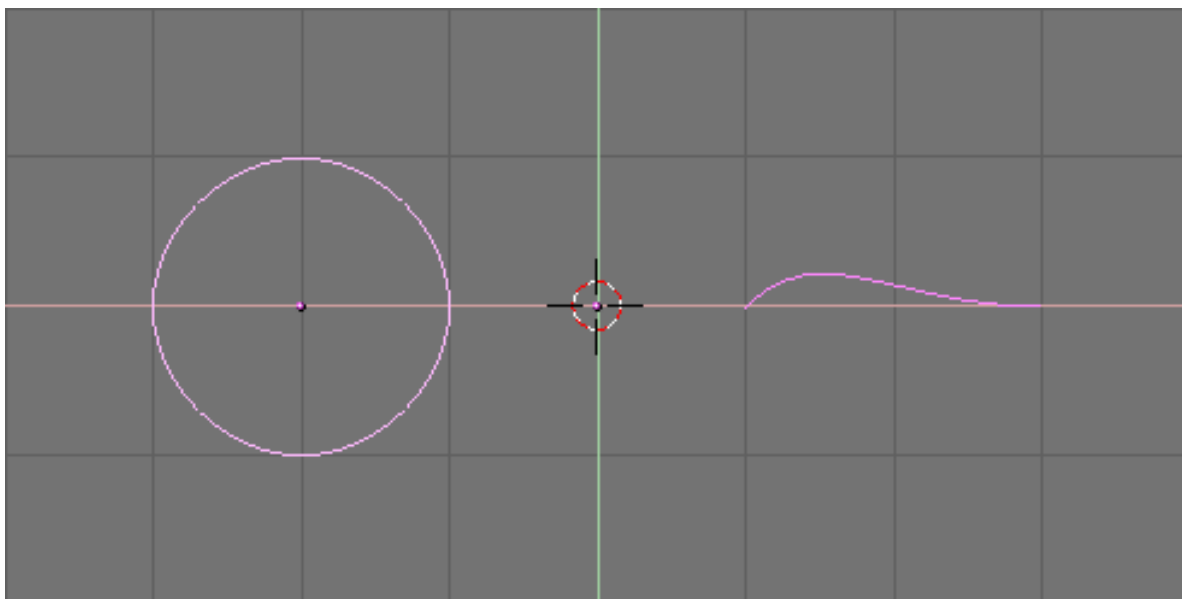
Relevante a Blender v2.31

La técnica de extrusión a lo largo de una trayectoria (Extrude along path) es una poderosa herramienta de modelado. Consiste en crear una superficie moviendo un perfil dado a lo largo de una trayectoria determinada.

Tanto el perfil como la trayectoria pueden ser una curva Bézier o una curva NURBS.

Asumiremos que ha añadido una curva Bézier y un círculo Bézier como obketos separados a su escena ([Figure 9-25](#)).

Figure 9-25. Perfil (a la izquierda) y trayectoria (a la derecha).



Juegue un poco con ambos para obtener un bonito perfil alado y una sugerente trayectoria ([Figure 9-26](#)). Por defecto, las curvas Béziere solo existen en un plano y son objetos de dos dimensiones. Para conseguir que la trayectoria abarque las tres dimensiones, como en el ejemplo, presione el botón *3D* en los botones de Edición de curva (**Tecla F9**) en el panel *Curve and Surface* ([Figure 9-27](#)).

Figure 9-26. Perfil modificado (a la izquierda) y trayectoria (a la derecha).

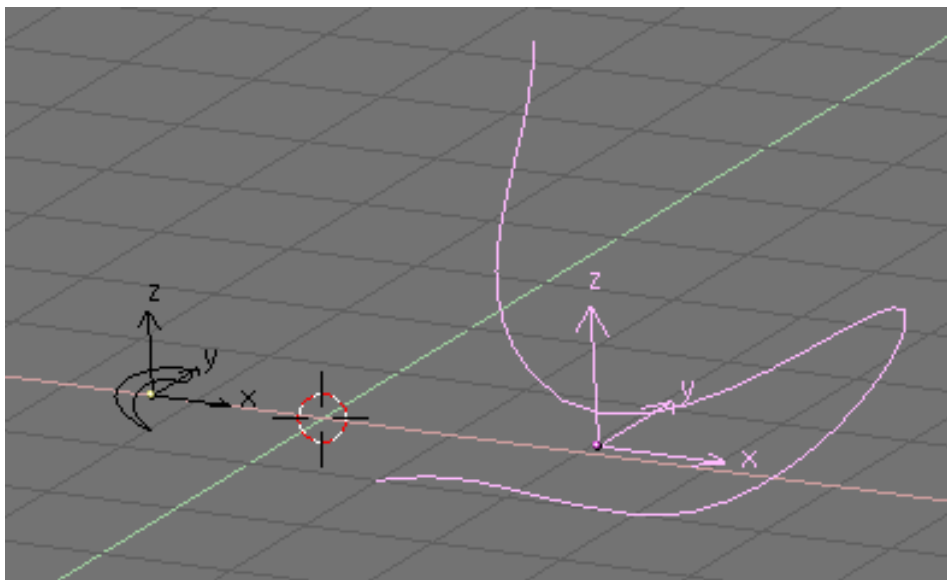
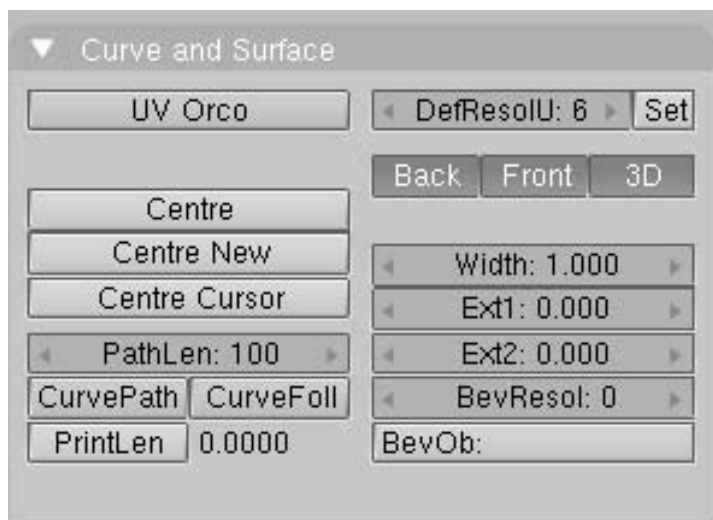
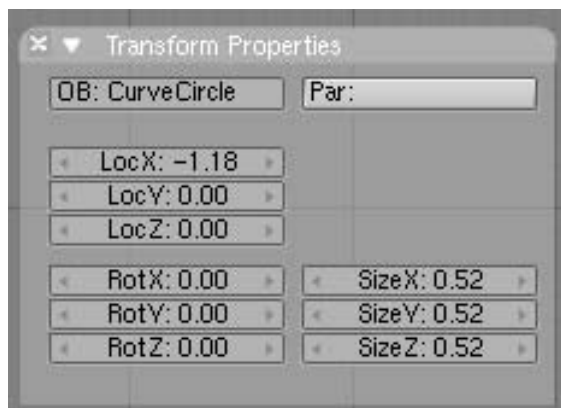


Figure 9-27. Botón 3D.



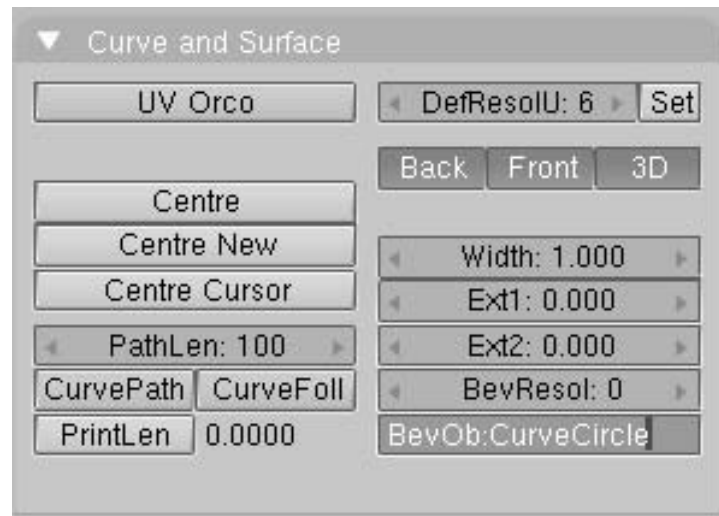
Ahora mire el nombre del objeto "perfil". Por defecto este nombre será "CurveCircle" y es mostrado en el panel emergente que aparece cuando se presiona la **Tecla N** cuando está seleccionado. Puede cambiar dicho nombre con **SHIFT-LMB** si así lo desea ([Figure 9-28](#)).

Figure 9-28. Nombre del perfil.



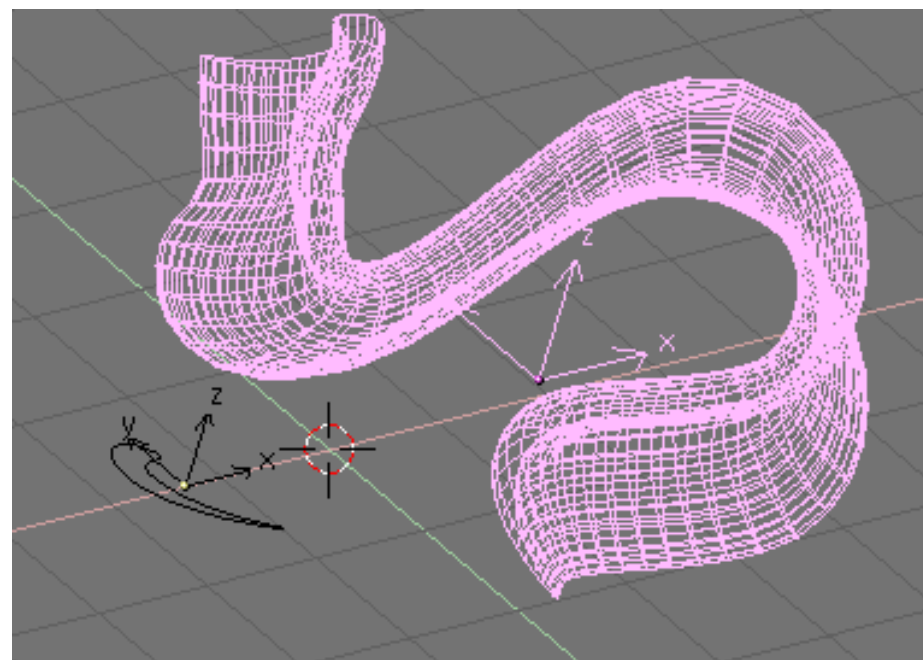
Ahora seleccione la trayectoria. En los botones de Edición localice el campo de texto *BevOb*: en el panel *Curve and Surface* y escriba allí el nombre del perfil. En nuestro caso "CurveCircle" ([Figure 9-29](#)).

Figure 9-29. Especifique el perfil que se utilizará en la trayectoria.



El resultado es una superficie definida por el perfil, que cubre todo el camino ([Figure 9-30](#)).

Figure 9-30. Resultado de la extrusión.



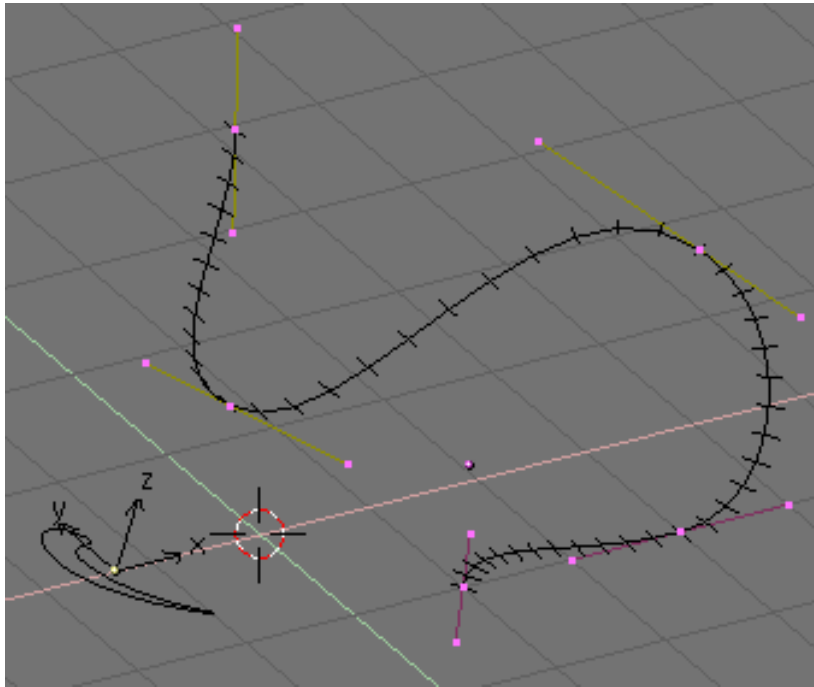
Para comprender los resultados, y obtener los efectos deseados es importante comprender los siguientes puntos:

- El perfil esta orientado de tal manera que su eje z es tangente (*en este caso* dirigido hacia delante) a la trayectoria y su eje x está en el plano de la trayectoria; consecuentemente el eje y es ortogonal al plano de la trayectoria;
- Si la trayectoria es tridimensional el "plano de la trayectoria" esta definido localmente en vez de globalmente

y es mostrado, en modo Edición, por varios segmentos cortos perpendiculares a la trayectoria ([Figure 9-31](#));

- El eje y del perfil siempre apunta hacia arriba. Esto es a menudo una fuente de resultados inesperados y de problemas, como explicaremos más tarde.

Figure 9-31. Plano local de la trayectoria.



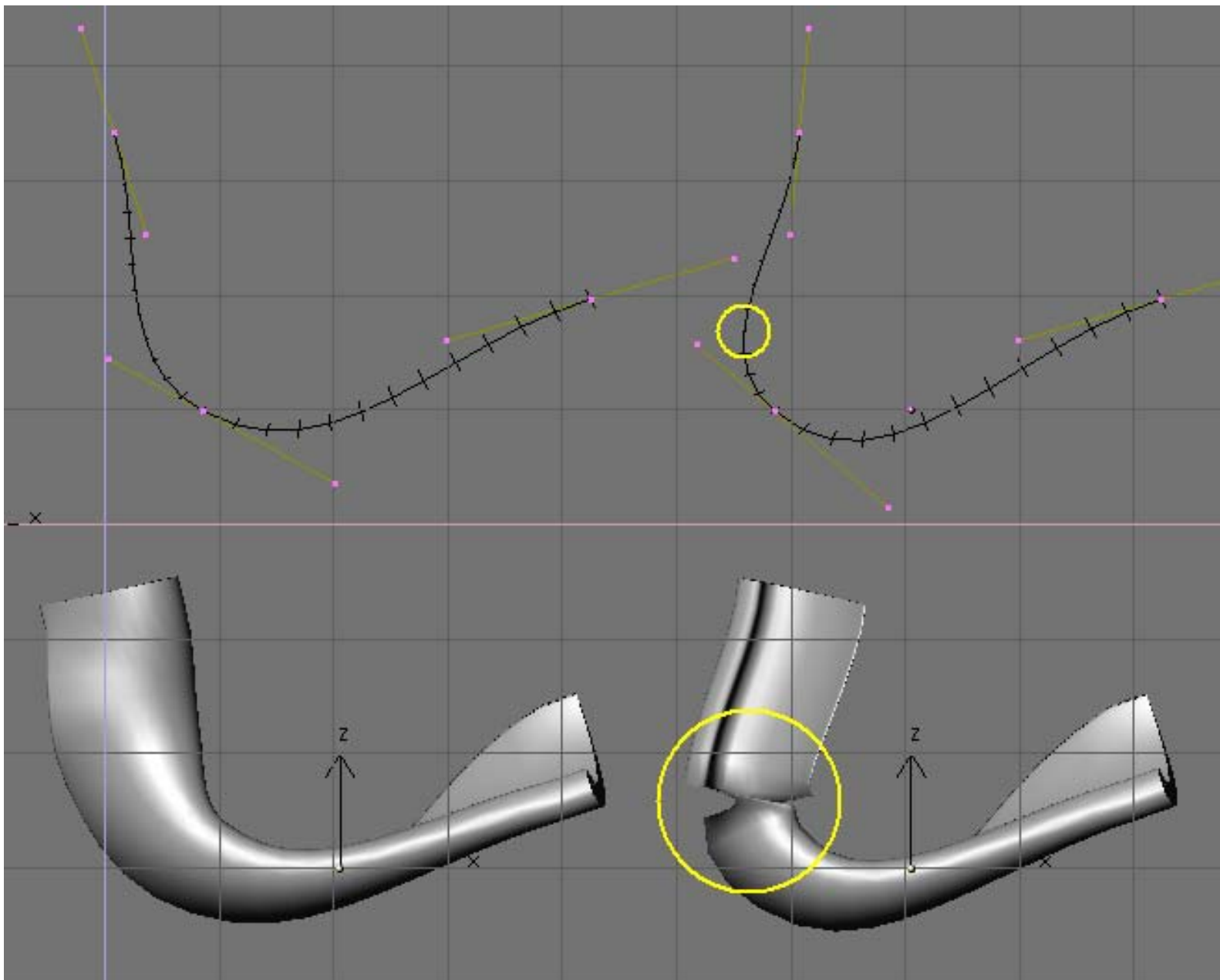
Tip **Inclinación (Tilting)**

Para modificar la orientación del plano local de la trayectoria seleccione un punto de control y presione la **Tecla T**. Entonces mueva el ratón para cambiar la orientación de los segmentos suavemente en las proximidades del punto de control. **LMB** fija la posición, y **ESC** vuelve al estado anterior.

Con el eje y obligado a apuntar hacia arriba, ocurrirán resultados inesperados cuando la trayectoria es tridimensional y el perfil que está siendo extruido llega a un punto donde la trayectoria es exactamente vertical. De hecho si la trayectoria es vertical y entonces continua doblandose hay un punto donde el eje y del perfil comenzaría a apuntar hacia abajo. Si esto ocurre, como el eje y está forzado a apuntar hacia arriba hay una rotación muy brusca, 180 grados, del perfil, para que de nuevo el eje y apunte hacia arriba.

[Figure 9-32](#) muestra el problema. A la izquierda hay una trayectoria cuya inclinación es tal que la normal al plano local de la trayectoria está siempre apuntando hacia arriba. A la derecha vemos una trayectoria donde, en el punto rodeado de amarillo, la normal comienza a apuntar hacia abajo. el resultado de la extrusión presenta un giro brusco allí.

Figure 9-32. Problemas con la extrusión debido a la restricción del eje y.



Las únicas soluciones a este problema son: Usar multiples trayectorias o cuidadosamente colocar la trayectoria de tal forma que la normal siempre apunte hacia arriba.

Tip Cambiando la orientación del perfil

Si la orientación del perfil a lo largo de la curva no es como esperabas, y quieres rotarla para la longitud total de la trayectoria, hay una forma mejor que inclinar todos los puntos de control de la trayectoria.

Simplemente puede rotar el perfil en modo Edición en su plano. De esta manera el perfil cambiará pero su referencia local no.

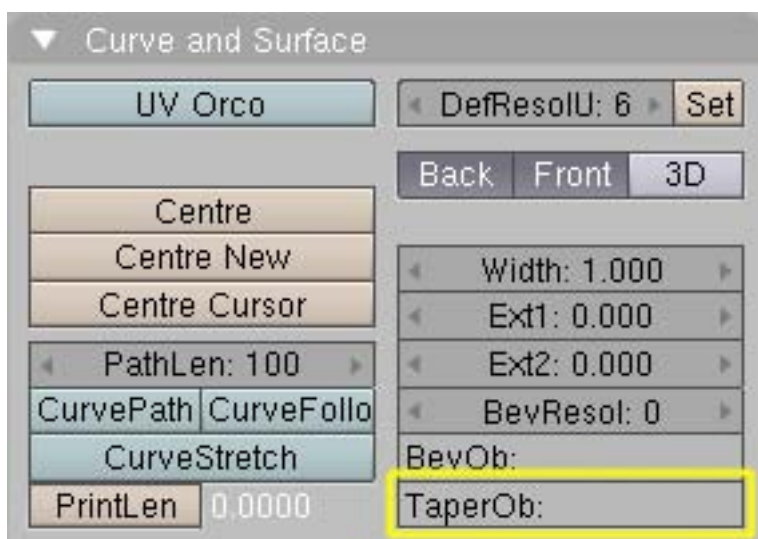
9.5. Curva Taper

por Kenneth Styrberg

Relevante a Blender v2.35

Taper es una herramienta para curvar objetos. En el panel de Edición (**F9**) tiene un campo *TaperOb* donde tiene que poner el nombre de la curva que definirá la anchura del objeto curvado.

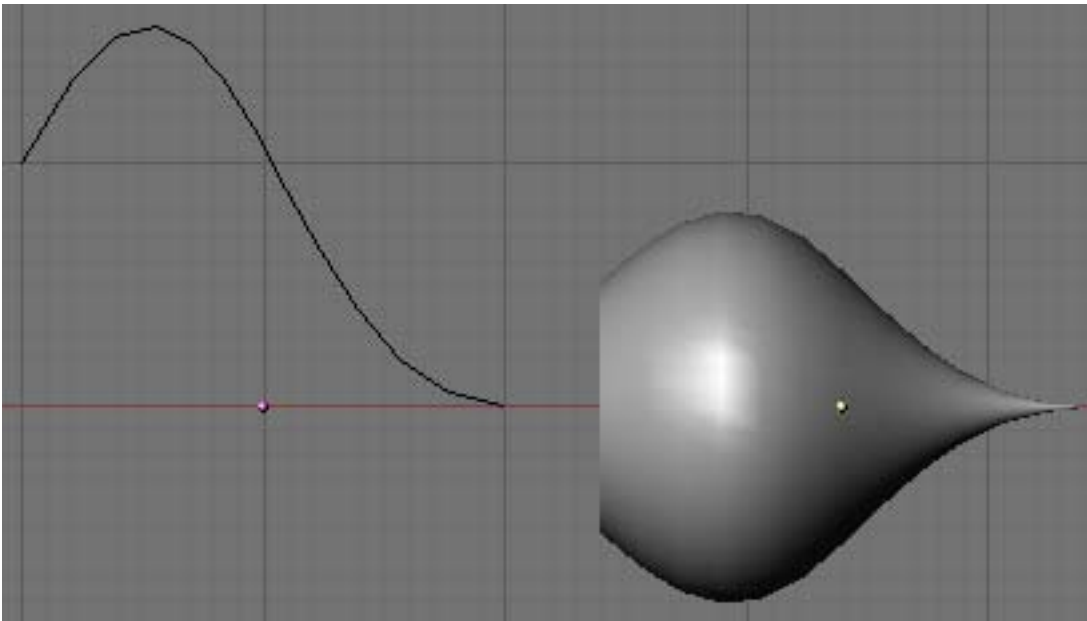
Figure 9-33. Panel de curvas y superficies



Note Reglas importantes

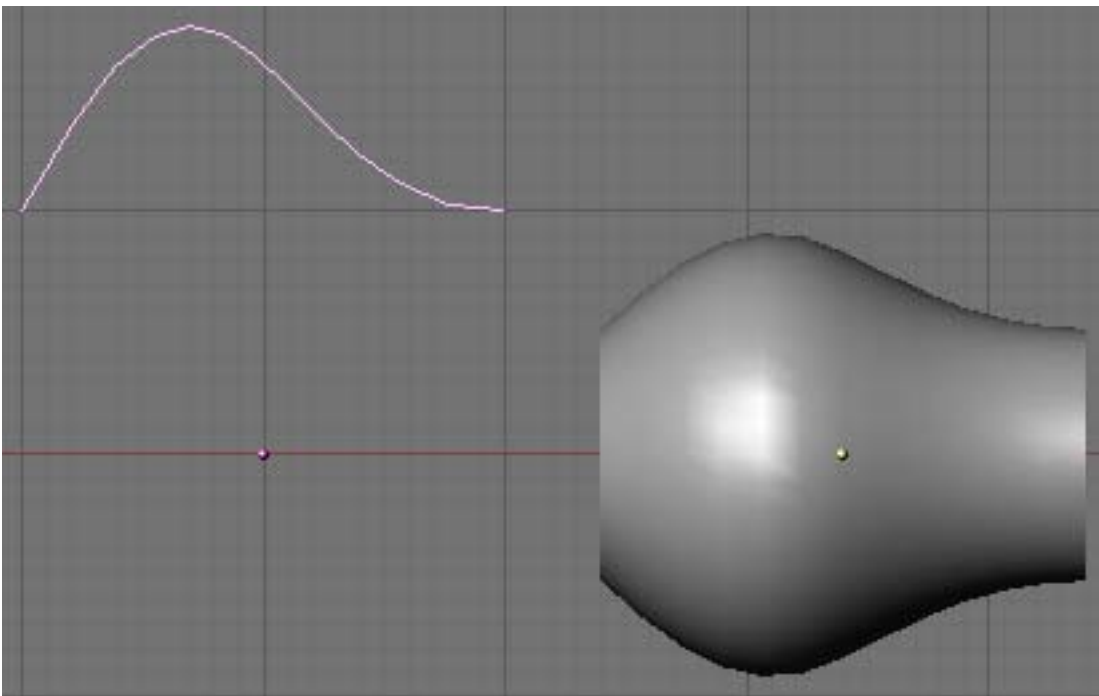
- Solo la primera curva en *TaperOb* es evaluada.
- la anchura de inicio es tomada de izquierda a derecha.
- Anchuras negativas también son posibles, pero en el renderizado se pueden crear pequeños artefactos.
- Escala la anchura de las extrusiones normales basadas en la evaluación de la curva taper, lo cual significa que las esquinas afiladas en la curva taper no serán fácilmente visibles.

Figure 9-34. Ejemplo 1 de Taper



En [Figure 9-34](#) puede claramente ver el efecto que la curva taper de la izquierda tiene sobre el objeto curvado de la derecha. Aquí la curva taper de la izquierda está más cercana al centro del objeto lo que produce una curvatura más pequeña en el objeto de la derecha.

Figure 9-35. Ejemplo 2 de Taper

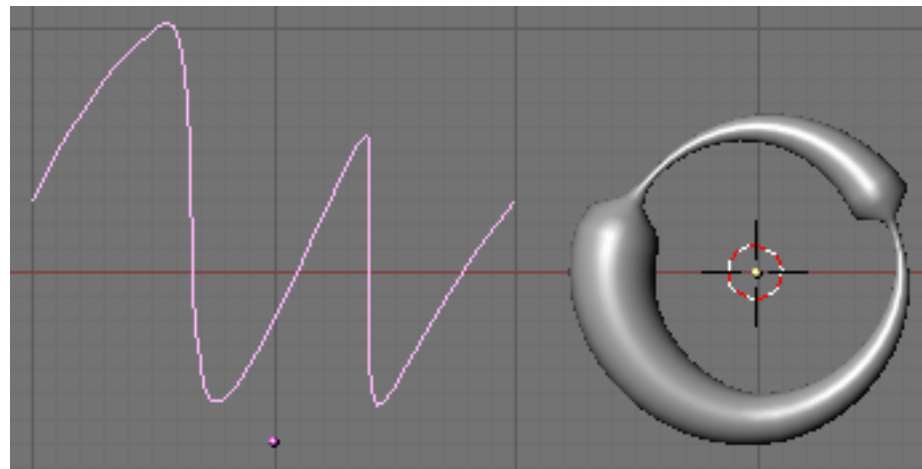


En [Figure 9-35](#) un punto de control en la curva taper de la izquierda se saca del centro lo que da un resultado más ancho al objeto curvado de la derecha.

Note El objeto curvado es extruido con un círculo. (Vea [Section 9.4](#) para más sobre extrusión de curvas).

En [Figure 9-36](#), vemos el uso de una curva taper irregular añadida a un círculo.

Figure 9-36. Ejemplo 3 de Taper



[Prev](#)

Extruir a lo largo de una
trayectoria

[Home](#)

[Up](#)

[Next](#)

Curva Deformar

9.6. Curva Deformar

por Kenneth Styrberg

Relevante a Blender v2.35

9.6.1. Introducción

La *Curva Deformar* proporciona un método sencillo pero eficiente de definir una deformación en una malla. Emparentando un objeto malla a una curva, puede deformar la malla estirando o encogiéndola a lo largo, u ortogonalmente, al eje dominante.

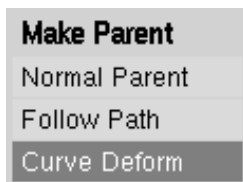
La *Curva Deformar* funciona en un eje dominante, X, Y, o Z. Esto significa que cuando mueva su malla en la dirección dominante, la malla seguirá el movimiento de la curva. Moviéndola en una dirección ortogonal moverá la malla más cerca o más lejos de la curva. La configuración por defecto de Blender hace que el eje dominante sea el eje Y. Cuando mueva el objeto más allá del final de la curva, el objeto continuará deformándose en la dirección del vector del final de la curva.

Tip Intente posicionar su objeto sobre la curva y mueva esta a su alrededor. Esto le proporcionará un mejor control sobre cómo funciona la deformación.

9.6.2. La Interfaz

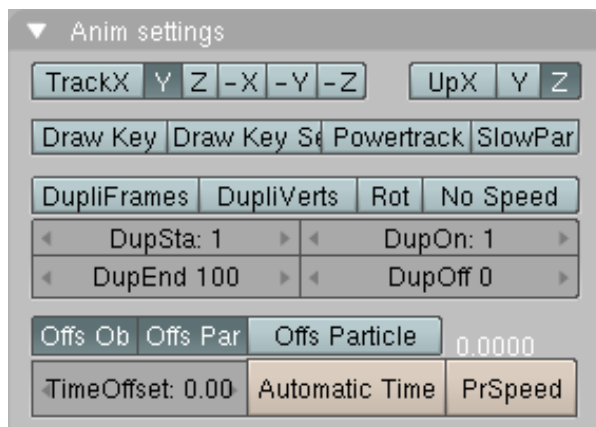
Cuando emparente una malla a una curva (**CTRL-P**), se le mostrará un menú, [Figure 9-37](#). Seleccionando **Curve Deform** activará la función *Curva Deformar* en el objeto malla.

Figure 9-37. El menú que aparece cuando se emparenta.



La propiedad del eje dominante se encuentra activa en el objeto malla. Por defecto el eje dominante en Blender es Y. Esto puede cambiarse seleccionando uno de los botones **Track X**, **Y** o **Z** en el Panel *Anim*, [Figure 9-38](#), en contexto *Objeto* (**F7**).

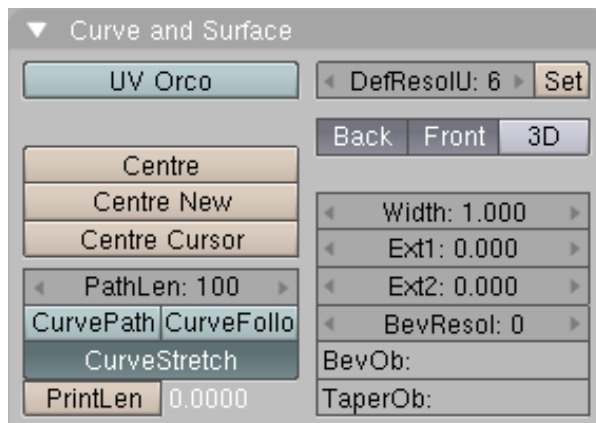
Figure 9-38. La configuración del Panel Anim.



Las curvas cíclicas trabajan tal y como se espera en las deformaciones de un objeto a lo largo de una trayectoria.

CurveStretch proporciona una opción que permite que la malla del objeto se extire, o se encoja, sobre la curva. Esta opción está en el contexto *Edición (F9)* de la curva. Ver [Figure 9-39](#).

Figure 9-39. Panel de Curvas y superficies.

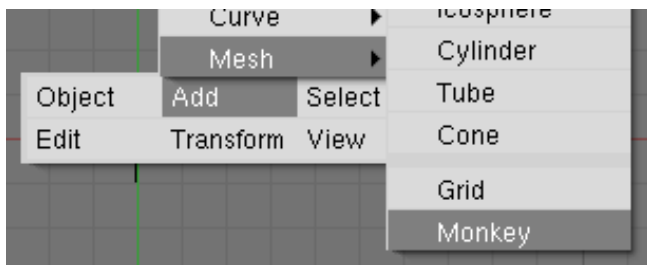


9.6.3. Ejemplo

Vamos a realizar un sencillo ejemplo.

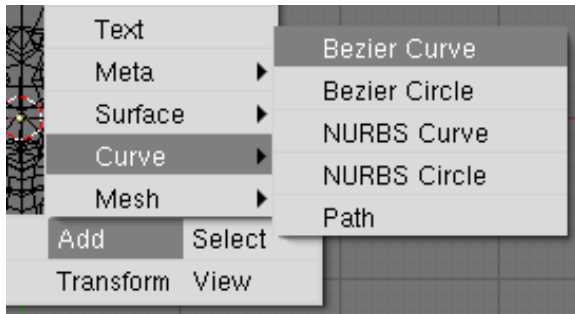
1. Elimine el cubo que hay por defecto en la escena y añada un Mono (Monkey)!! (**SHIFT-A** -> **Add** -> **Mesh** -> **Monkey**, [Figure 9-40](#)).

Figure 9-40. ¡Añada un mono!



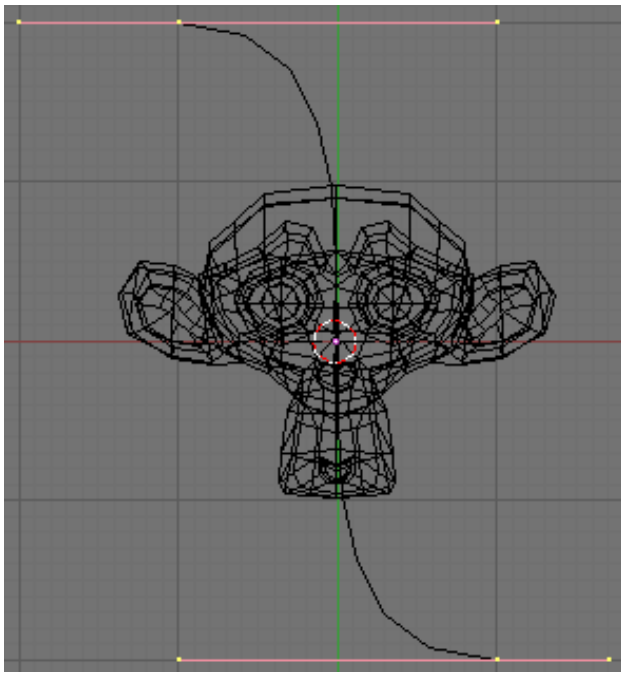
2. Ahora pulse **TAB** para salir del *Modo de Edición*. A continuación añada una curva. (**SHIFT-A** -> **Add** -> **Curve** -> **Bezier Curve**, [Figure 9-41](#)).

Figure 9-41. Añada una curva.



3. Mientras estemos en el Modo edición, mueva los puntos de control de la curva como se muestra en [Figure 9-42](#), y entonces salga del *Modo Edición*, (**TAB**).

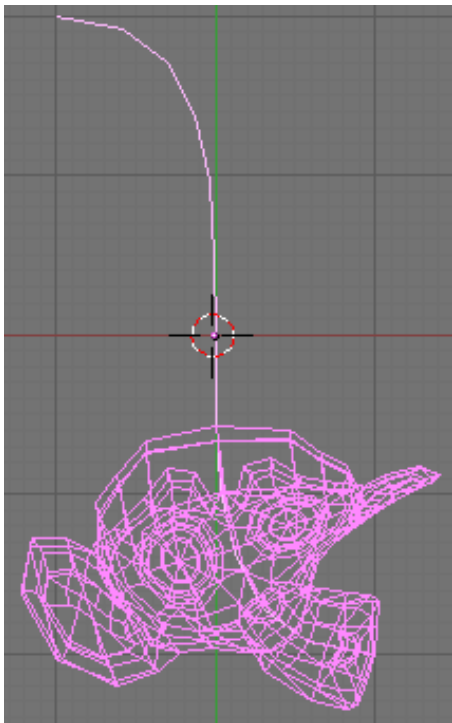
Figure 9-42. Editando la curva.



4. Seleccione el Mono, (**RMB**), y con shift (mayúsculas) pulsado seleccione la curva, (**SHIFT-RMB**). Presione **CTRL-P** para abrir el menú *Make Parent*. Seleccione la opción **Curve Deform**. ([Figure 9-37](#)).

El mono adoptará la posición de la curva tal y como se muestra en [Figure 9-43](#).

Figure 9-43. Mono en la curva.



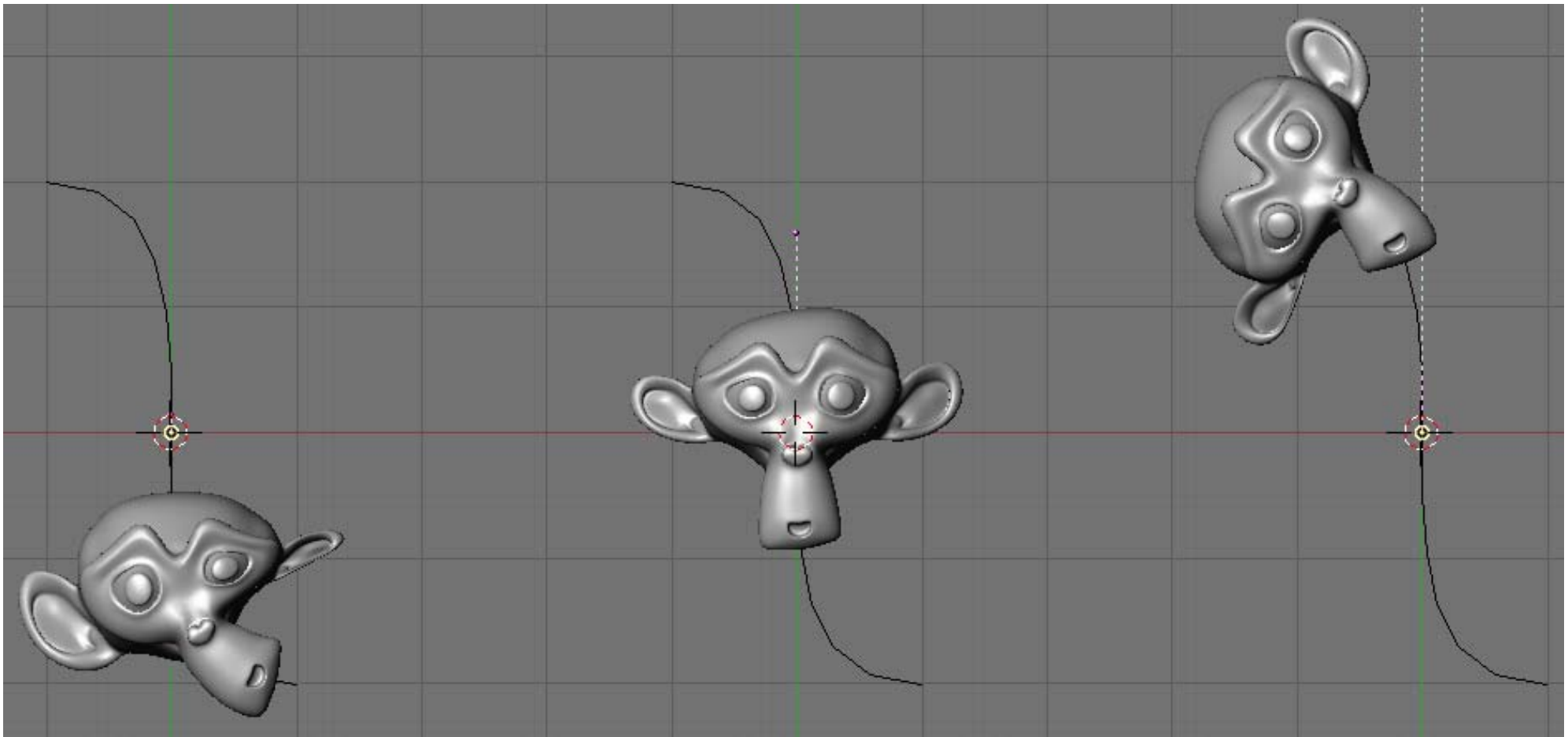
5. Ahora si seleccionas el mono, (**RMB**), y lo mueves, (**G**), en la dirección Y, (por defecto, el eje dominante), el mono se deformará a lo largo de la curva.

Tip Si presionas **MMB** mientras estés moviendo el mono restringirás el movimiento a un único eje.

6. En [Figure 9-44](#), puedes observar el mono en diferentes posiciones a lo largo de la curva. Para obtener una visión más clara de la deformación se ha activado **SubSurf** con **Subdiv 2** y **Set Smooth** en la malla del mono. (**F9** para mostrar las opciones de *Edición*).

Tip Cuando se mueva el mono en cualquier dirección distinta a la del eje dominante crearás algunas deformaciones raras. Algunas veces es lo que querrás conseguir, así que necesitarás experimentar un poco para dominarlo.

Figure 9-44. Deformaciones del mono.



[Prev](#)

Curva Taper

[Home](#)

[Up](#)

[Next](#)

Skinning

9.7. Skinning

Relativo a Blender v2.31

Skinning es el arte de definir una superficie usando 2 ó más perfiles. En Blender puede hacer eso preparando algunas curvas con la forma de la superficie deseada para después convertirlas a una única superficie NURBS.

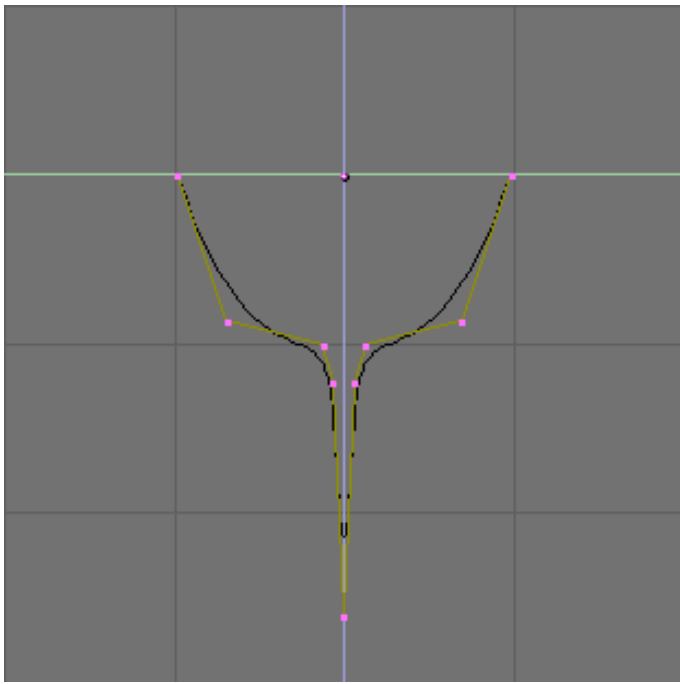
Como ejemplo crearemos un bote de pesca. La primera cosa que haremos, en la vista lateral (**NUM3**), es añadir una "Surface Curve". Asegurate de que realmente sea una curva *Surface* y no una curva Beizer o NURBS, o el truco no funcionará ([Figure 9-45](#)).

Figure 9-45. Una "curva superficie" para realizar skinning.



Dé a la curva la forma de la sección transversal de la mitad del barco, añadiendo tantos vértices como sean necesarios con el botón "Split" (partir) y, posiblemente, configurando el 'Endpoint' (punto final) de la curva a 'U' y/o a 'V' ([Figure 9-46](#)).

Figure 9-46. Perfil del barco.



Ahora duplique (**SHIFT-D**) la curva, tantas veces como sea necesario, hacia la izquierda y hacia la derecha ([Figure 9-47](#)). Ajuste las curvas para que coincidan con las diferentes secciones transversales que tiene el barco a lo largo de su eslora. Para llevar esto a cabo, unos blueprints (planos técnicos con el perfil) ayudarán bastante. Puede cargar un blueprint de fondo (como hicimos para el diseño del logo en este mismo capítulo) para ajustar los perfiles de las secciones transversales ([Figure 9-48](#)).

Note que la superficie que hemos creado pasa de un perfil a otro suavemente. Para crear cambios bruscos necesitará situar los perfiles muy cerca los unos a los otros, como en el caso del perfil seleccionado en [Figure 9-48](#).

Figure 9-47. Varios perfiles a lo largo del eje del barco.

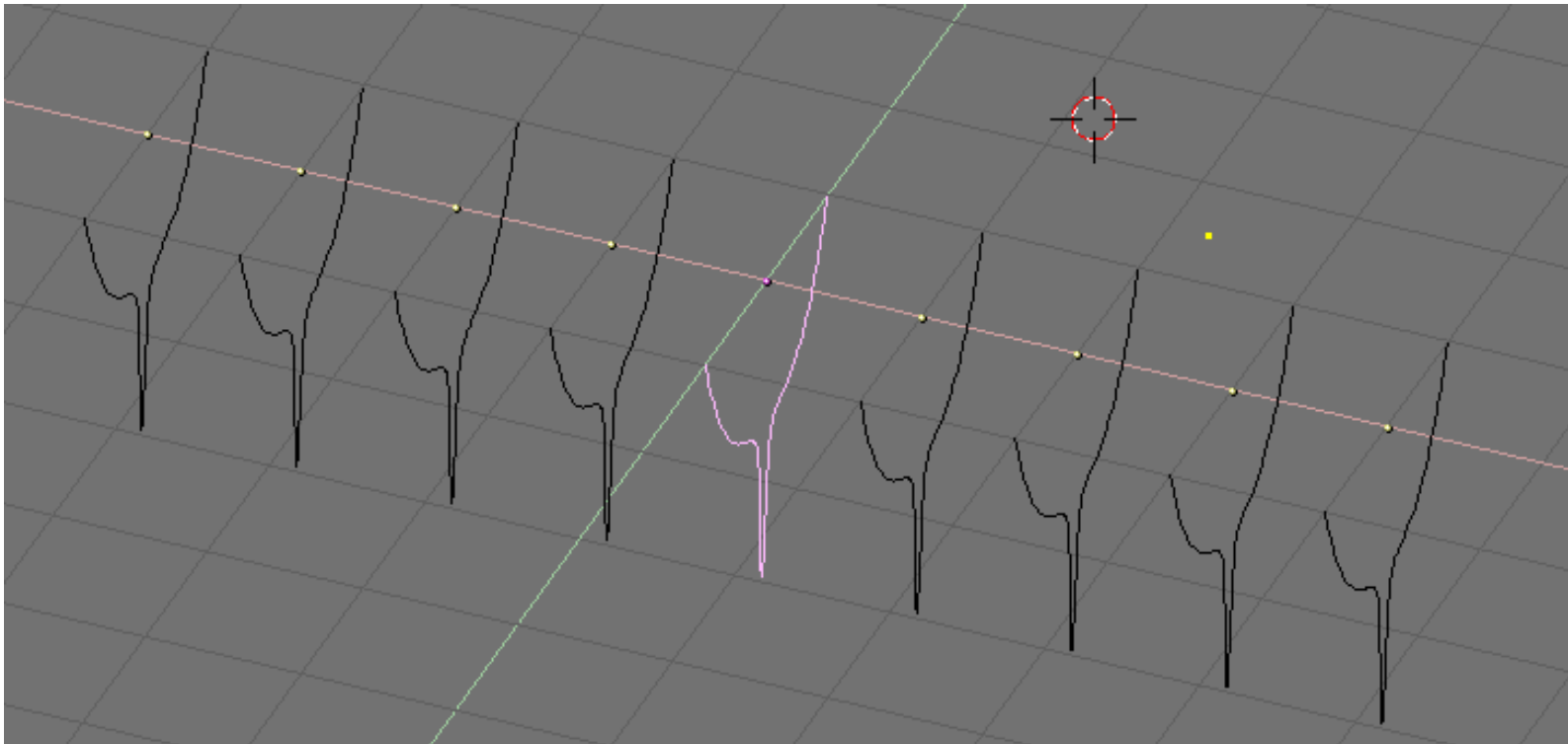
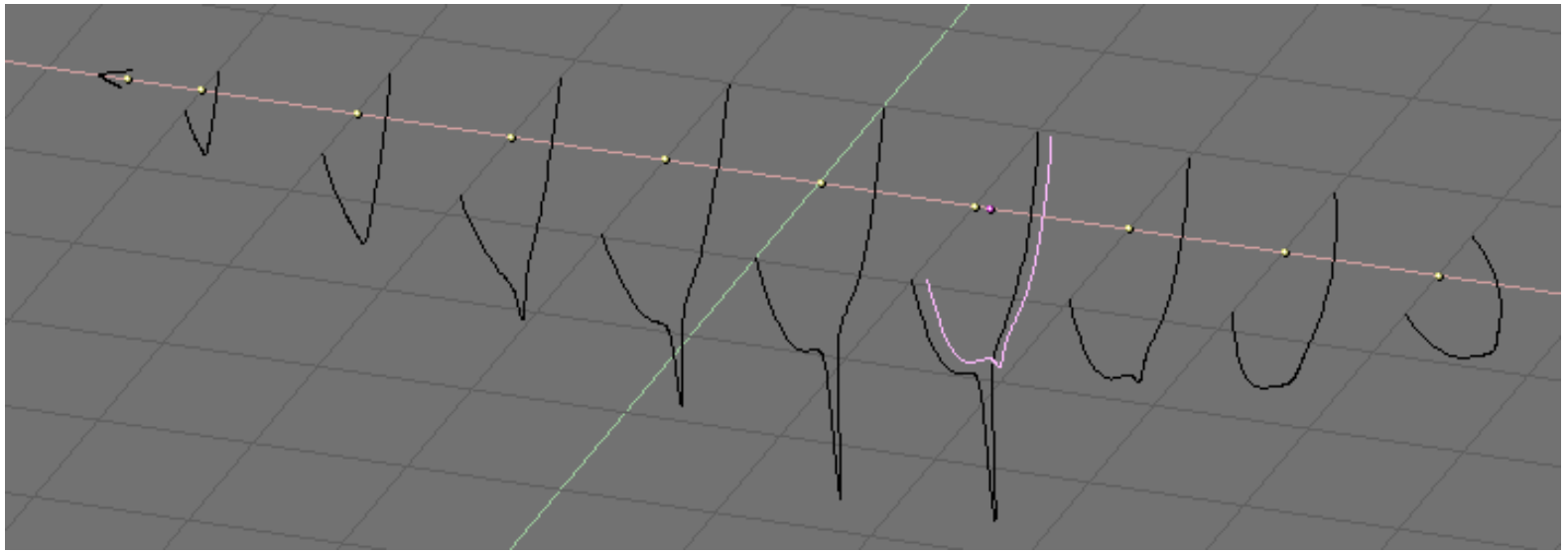
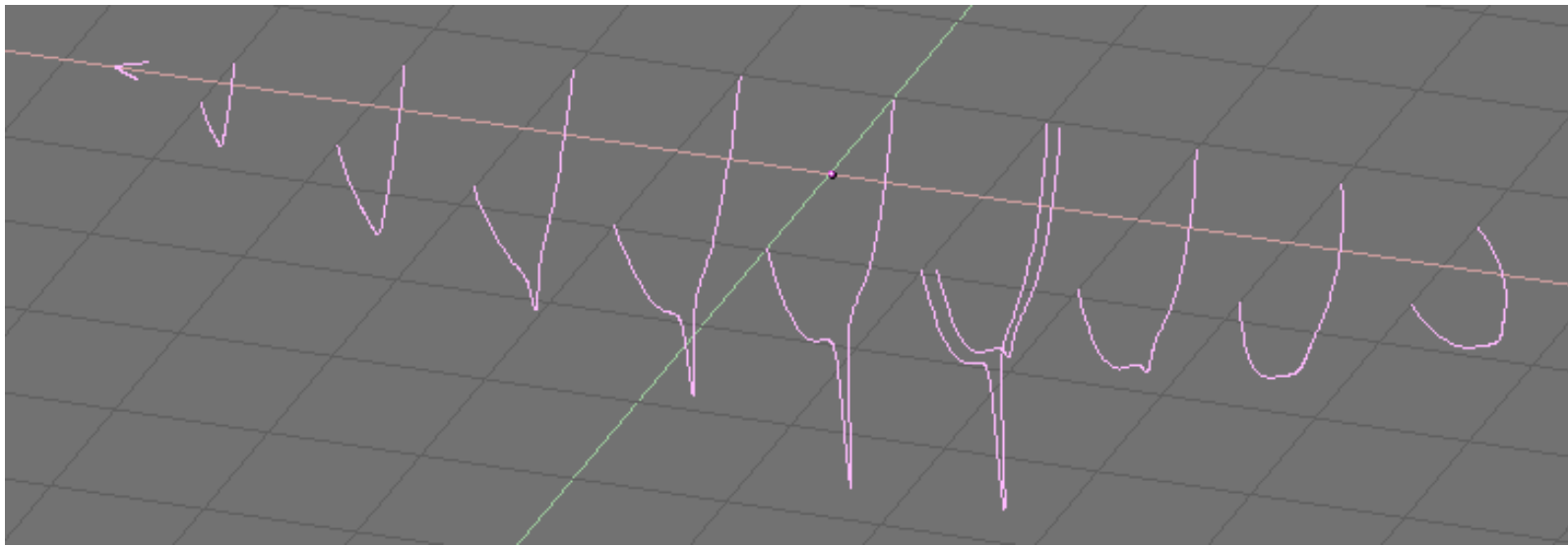


Figure 9-48. Varios perfiles de la forma correcta.



Ahora seleccione todas las curvas (con la **Tecla A** o la **Tecla B**), y unalas presionando **CTRL-J** y contestando 'Yes' a la cuestión 'Join selected NURBS?' (¿Unir NURBS seleccionados?). Se resaltarán todos los perfiles [Figure 9-49](#).

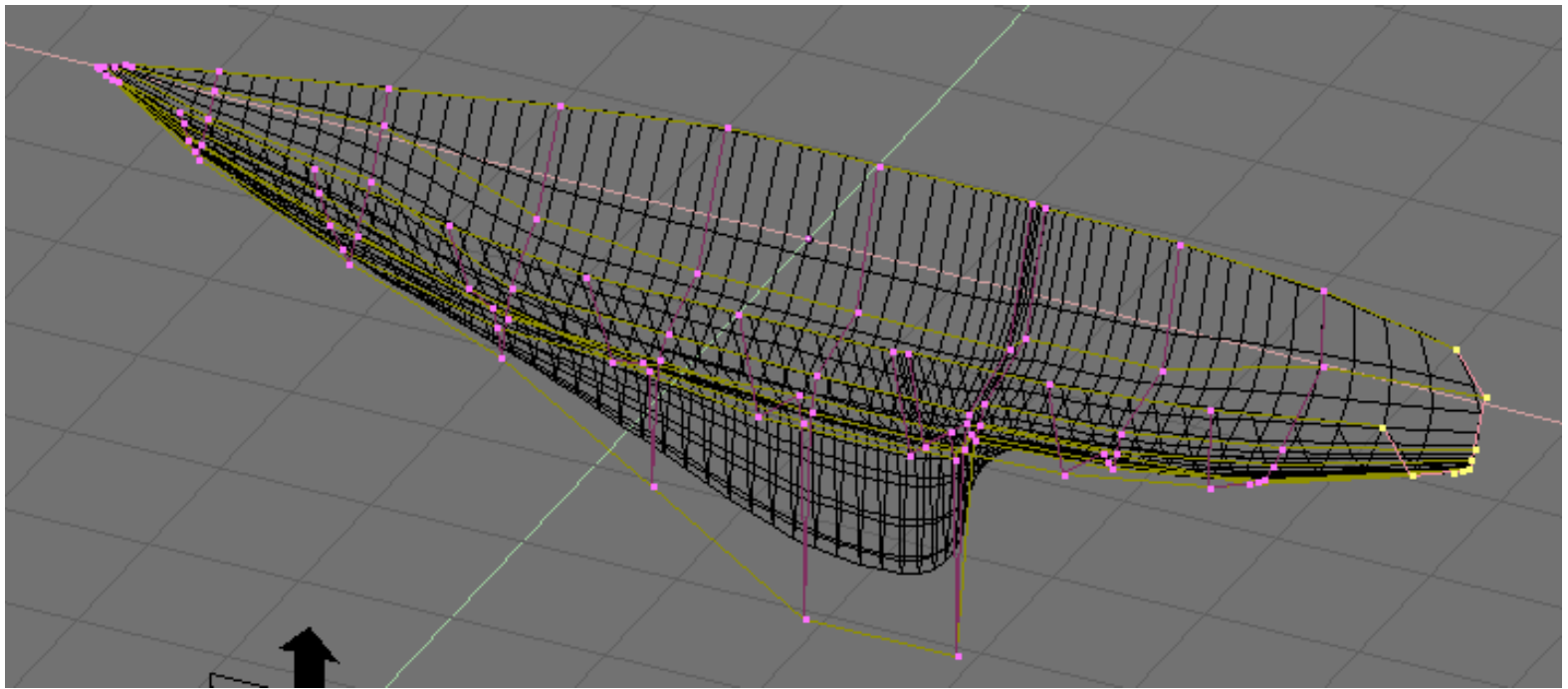
Figure 9-49. Perfiles unidos.



Ahora entre en el modo edición (**TAB**) y seleccione todos los puntos de control con la **Tecla A**; entonces presione la **tecla F**. Los perfiles se convertirán en una superficie ([Figure 9-50](#)).

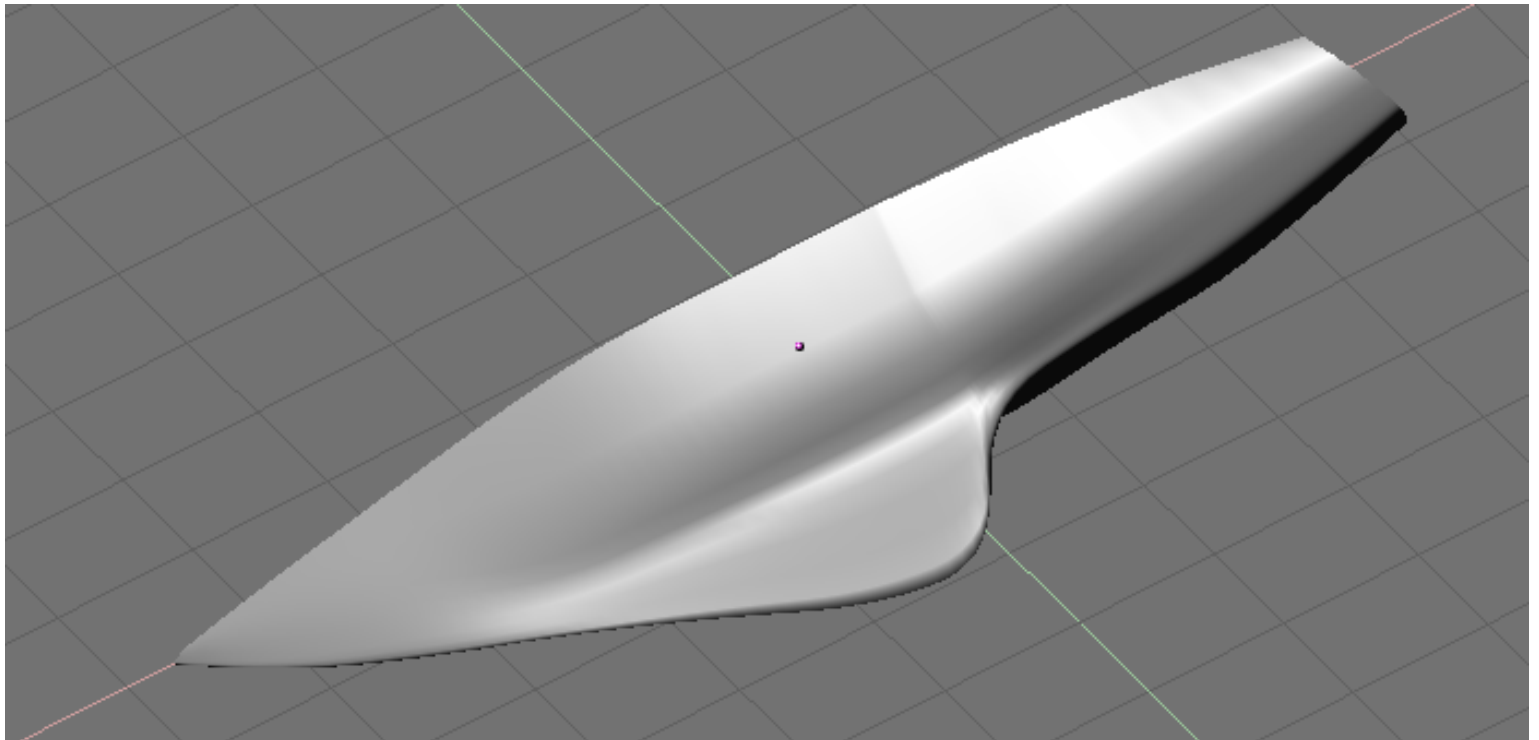
Note Como debería ser evidente desde el primer al último perfil en este ejemplo, las secciones transversales no necesitan ser definidas en un grupo de planos ortogonales.

Figure 9-50. Superficie en modo edición.



Modifique la superficie, si fuese necesario, moviendo los puntos de control. [Figure 9-51](#) muestra una vista en modo sombreado. Probablemente necesitará incrementar *ResolU* y *ReloIV* para obtener una forma más óptima.

Figure 9-51. El casco finalizado.



Tip Configuración de los perfiles

La única limitación de esta poderosa técnica es que todos los perfiles tienen que tener el mismo número de puntos de control. Esto es por lo que es bastante buena idea modelar primero la sección transversal más compleja y después duplicarla, moviendo los puntos de control como sea necesario, sin añadir o quitar dichos puntos, como se muestra en el ejemplo.

[Prev](#)

Curva Deformar

[Home](#)

[Up](#)

[Next](#)

Materiales

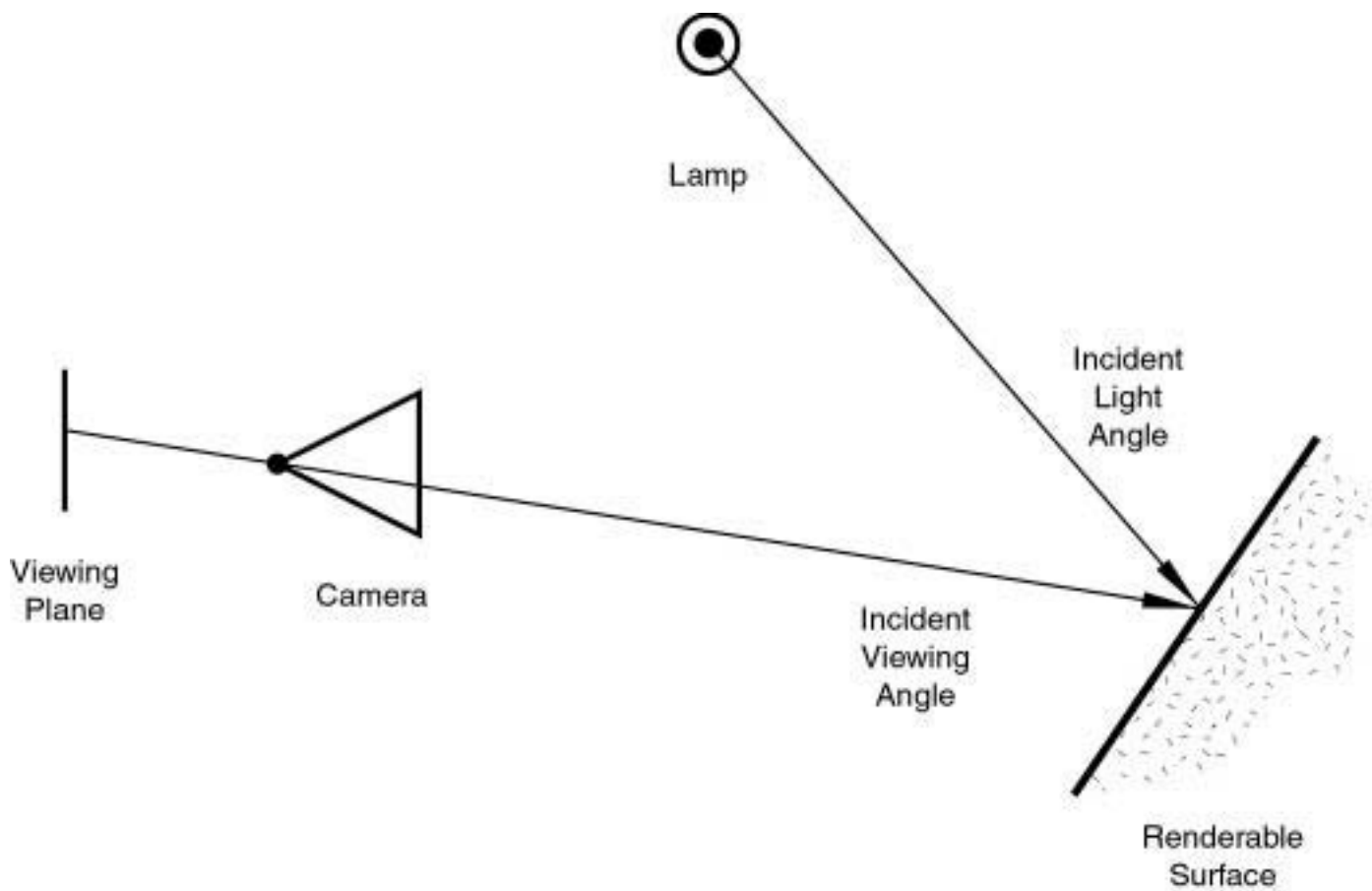
Chapter 10. Materiales

Antes de que pueda entender como diseñar de forma eficiente con materiales, debe entender como interactuan con el motor de render de Blender las luces simuladas y las superficies, y como los parámetros del material controlan éstas interacciones. Una profundización en el entendimiento del motor le ayudará a entender todo ello.

La imagen renderizada que usted crea con Blender es una proyección de la escena en una superficie imaginaria llamada *plano de proyección (viewing plane)*. El plano de proyección es análogo a la película en una cámara tradicional, o a la córnea en un ojo humano, excepto que recibe luz simulada en lugar de luz real.

Para renderizar una imagen de una escena primero debe determinar que luz de la escena está llegando a cada punto del plano de proyección. La mejor manera de responder a esta pregunta es seguir una línea recta (el rayo de la luz simulada) desde dicho punto en el plano de proyección y el punto focal (la posición de la cámara) hasta que impacte sobre una superficie representable en la escena, en este punto podremos determinar que luz debería impactar ese punto. Las propiedades de la superficie y el ángulo de incidencia de la luz nos dirán qué cantidad de luz debería ser reflejada por el ángulo de visión incidente ([Figure 10-1](#)).

Figure 10-1. Principio básico del motor de render.



Cuando un rayo de luz impacta sobre cualquier punto de una superficie, se dan lugar dos tipos básicos de fenómenos: difusión y reflexión especular. La difusión y la reflexión especular se distinguen la una de la otra principalmente por la relación entre el ángulo de luz incidente y el ángulo de la luz reflejada.

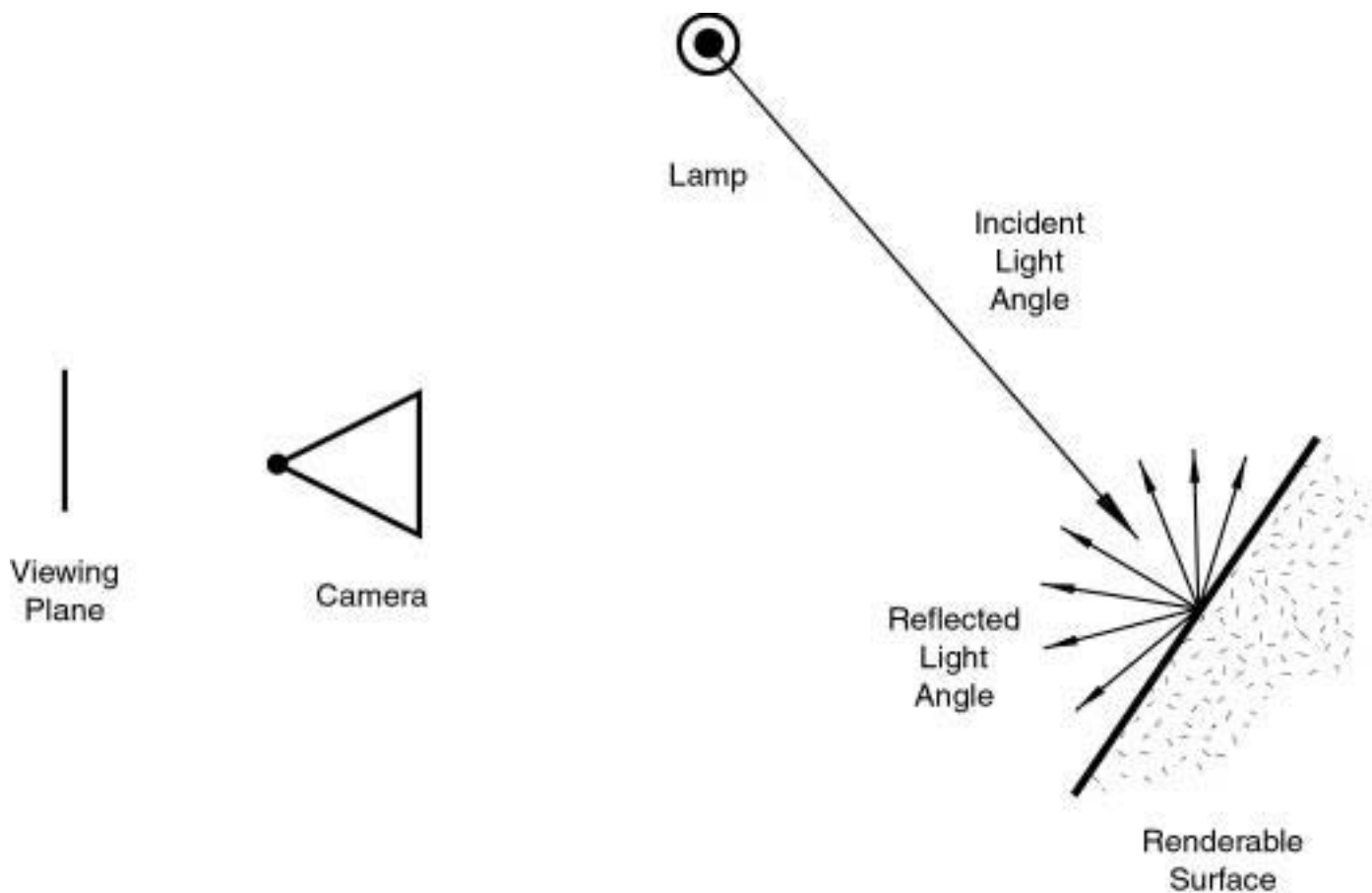
10.1. Difusión

Relevante a Blender v2.31

El impacto de la luz sobre una superficie y la posterior reirradiación por fenómeno de difusión puede ser muy dispersa, por ejemplo, la re-irradiación en todas las direcciones de forma isotrópica. Esto significa que la cámara verá la misma cantidad de luz desde ese punto de superficie sin importar cual sea su *ángulo de visión incidente*.

Es esta cualidad la que hace a la luz difusa *independiente del punto de visión*. Por supuesto la cantidad de luz que impacta una superficie depende del ángulo de luz incidente. Si la mayoría de la luz que golpea una superficie es reflejada de forma difusa, la superficie tendrá una apariencia mate ([Figure 10-2](#)).

Figure 10-2. Luz re-irradiada con fenómeno de difusión.



Desde la versión 2.28, Blender ha implementado tres fórmulas matemáticas diferentes para calcular la difusión. Y, de forma más notable, el fenómeno de difusión y reflexión especular, que suelen ir limitados en un solo tipo de material, y han sido separados para que sea posible elegir la implementación de la difusión y reflexión especular de forma separada.

Las tres implementaciones de difusión, o *sombreadores (shaders)*, usan dos o tres parámetros cada una. Los dos primeros parámetros son compartidos por todos los sombreadores de difusión y son los *Colors de Difusión* o simplemente *color*, del material, y la cantidad de energía de la luz incidente que es difundida en realidad. Esta última cantidad, dada en un rango de $[0,1]$, es normalmente llamada *Refl* en el interfaz.

Los sombreadores implementados son:

- *Lambert* - Éste fué el sombreador de difusión por defecto de Blender hasta la versión 2.27. Y como tal, todos los tutoriales antiguos se refieren a él, y todas las imágenes pre-2.28 han sido creadas con él. Este sombreador tiene únicamente los parámetros por defecto.
- *Oren-Nayar* - Este sombreador introducido por primera vez en Blender 2.28. Toma una especie de aproximación 'física' al fenómeno de la difusión. Además de los dos parámetros por defecto, tiene un tercero que es usado para determinar la cantidad de rugosidad microscópica de la superficie.
- *Toon* - Este sombreador fué introducido por primera vez en Blender 2.28. Es un sombreador muy poco 'físico' ya que no está pensado para imitar la realidad, sino para producir renders de 'dibujos

animados', con límites muy claros de luz-sombra y regiones uniformes de iluminadas-en sombra. Aunque es relativamente simple, también requiere dos parámetros más que definen el tamaño de las áreas iluminadas y el suavizado de los límites de sombra.

Una sección siguiente, dedicada a la implementación actual del material, analizará todo esto y sus parámetros relativos.

[Prev](#)

Skinning

[Home](#)

[Up](#)

[Next](#)

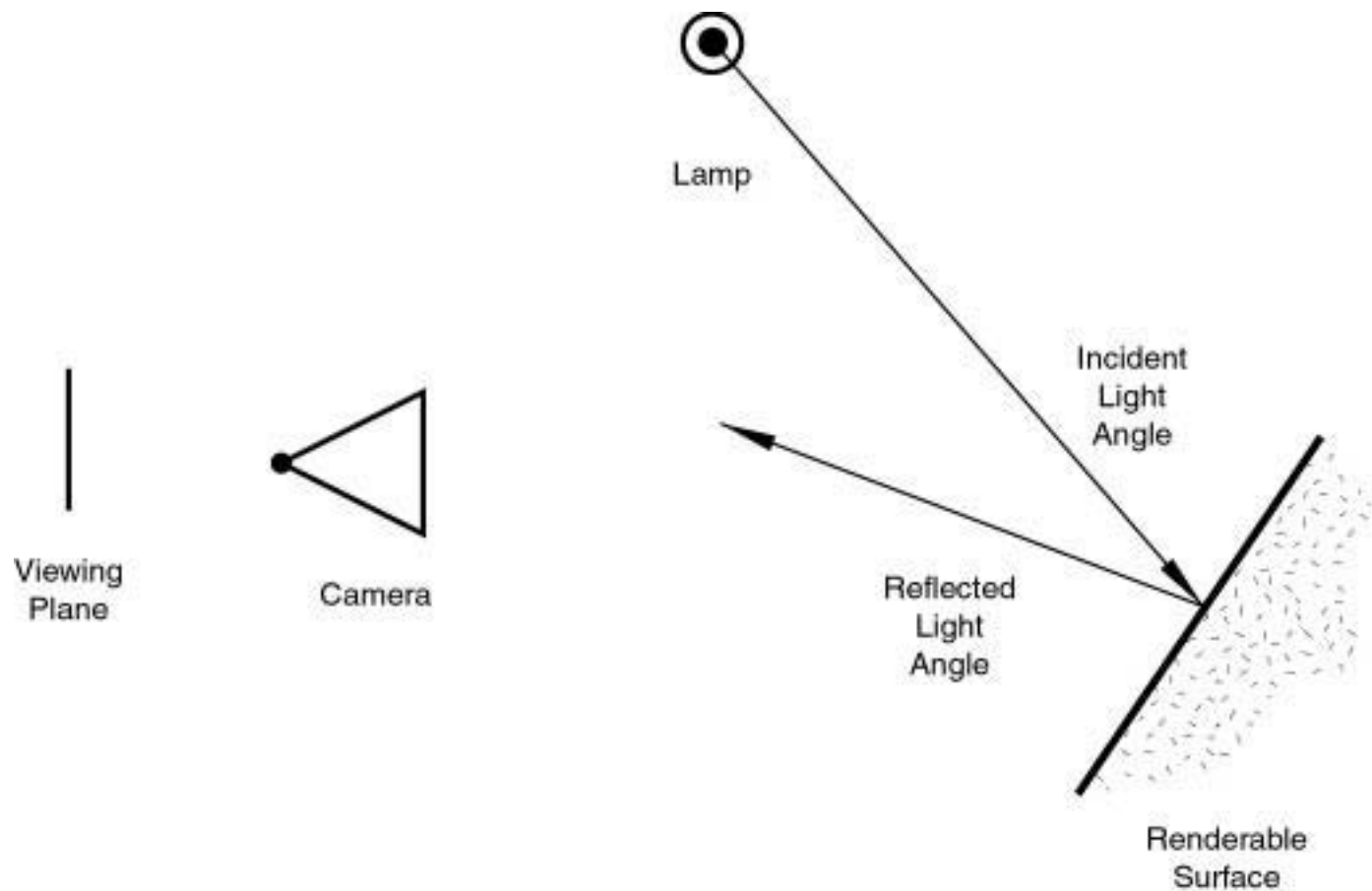
Reflexión especular

10.2. Reflexión especular

Relevante a Blender v2.31

Al contrario que la difusión, la reflexión especular es *dependiente del punto de visión*. De acuerdo con la ley de Snell, la luz que impacta sobre una superficie especular se verá reflejada con un ángulo espejado al ángulo de luz incidente (NdT: Un ángulo igual a ángulo incidente +90 grados), lo cual hace que el ángulo de visión cobre mucha importancia. La reflexión especular crea finos y brillantes reflejos, haciendo que la superficie parezca pulida ([Figure 10-3](#)).

Figure 10-3. Reflexión especular.



En realidad, la Difusión y la reflexión especular son generadas por el mismo proceso exacto de dispersión de la luz. La difusión es dominante en una superficie que tiene una pequeña escala de rugosidad en ella, con respecto a la longitud de onda, de forma que la luz se ve reflejada en muchas

direcciones por cada pequeño fragmento de superficie, con cambios muy pequeños en el ángulo de la superficie.

Por otro lado, la reflexión especular, predomina en una superficie que es suavizada, con respecto a la longitud de onda. Esto implica que la dispersión de los rayos de cada punto de la superficie serán direccionados en su mayoría en la misma dirección, más que al ser dispersada de forma difusa. Es simplemente un problema de la escala de detalle. Si la rugosidad de la superficie es mucho menor que la longitud de onda de la luz incidente parecerá plano y actuará como un espejo.

Note Es importante incidir especialmente en que el fenómeno de la reflexión especular discutido aquí no es la reflexión que deberíamos ver en un espejo, pero sí los destellos de luz que deberíamos ver en una superficie pulida. Para conseguir reflexiones como las de un espejo debería usar un trazado de rayos como tal (raytracer), pero ésto puede producir superficies estilo espejo convincentes, mediante una aplicación cuidadosa de texturas, como veremos más tarde.

Como la difusión, la reflexión especular tiene bastantes implementaciones diferentes, o *sombreadores especulares*. De nuevo, cada una de dichas implementaciones comparte dos parámetros comunes: el *Color Especular* y la energía de la especularidad, en un rango de $[0,2]$. Esto permite que se pueda drenar más energía con eficiencia como reflexión especular como si fuera energía incidente. Como resultado, un material tiene al menos dos colores diferentes, uno difuso y uno especular. El color especular suele ser blanco puro, pero pueden ser cambiados sus valores para conseguir efectos interesantes.

Los cuatro sombreadores especulares son:

- *CookTorr* - Éste fué el único sombreador especular en Blender hasta la versión 2.27. De hecho, hasta dicha versión no era posible aplicar de forma separada sombreadores de difusión y especulares, y no había más que una implementación de un material llano. Además de los dos parámetros estándar, este sombreador usa un tercero, *dureza (hardness)*, que regula la anchura de los resalte especulares. Cuanto menor dureza tenga, más anchos serán los reflejos.
- *Phong* - Éste es un algoritmo matemático diferente, usado para calcular los reflejos especulares. No es muy diferente de CookTor, y está configurado por los mismos tres parámetros.
- *Blinn* - Éste es un sombreador especular más 'físico', pensado para combinarlo con el difuso Oren-Nayar. Es más físico debido a que añade un cuarto parámetro, un *índice de refracción (IDR - IOR)* a los tres anteriores. Este parámetro en realidad no se usa para calcular la refracción de los rayos (para ello se necesita un trazado de rayos), pero para calcular correctamente en la reflexión especular la intensidad y extensión por la ley de Snell. La dureza y los parámetros especulares dan más grados de libertad.
- *Toon* - Este sombreador coincide con el de difusión Toon. Está diseñado para producir la agudeza y reflejos uniformes de los dibujos. No tiene dureza pero añade un par de parámetros. Tamaño y suavizado, que dictaminan la extensión y rugosidad de los reflejos especulares.

Gracias a esta flexible implementación, la cual trata de separar los fenómenos de difusión y de reflexión especular, nos permite tener un sencillo control de blender de cuánta luz incidente que impacta sobre un punto en una superficie se dispersa por difusión, cuanta se refleja de forma especular, y cuánta es absorbida. Esto, alternativamente, determinar en qué direcciones (y en qué cantidades) la luz es reflejada por un punto de luz dado; esto es, desde qué fuentes (y en qué cantidades) la luz es reflejada hacia un punto dado en el plano de proyección.

Es muy importante recordar que el color del material es solo un elemento en el proceso de render. El color normalmente es el producto del color de la luz y el color del material.

[Prev](#)

Materiales

[Home](#)

[Up](#)

[Next](#)

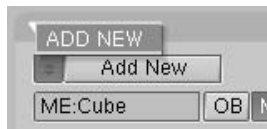
Materiales en la práctica

10.3. Materiales en la práctica

Relativo a Blender v2.31

En esta sección veremos como configurar los distintos parámetros de materiales en Blender, y qué esperar como resultado.

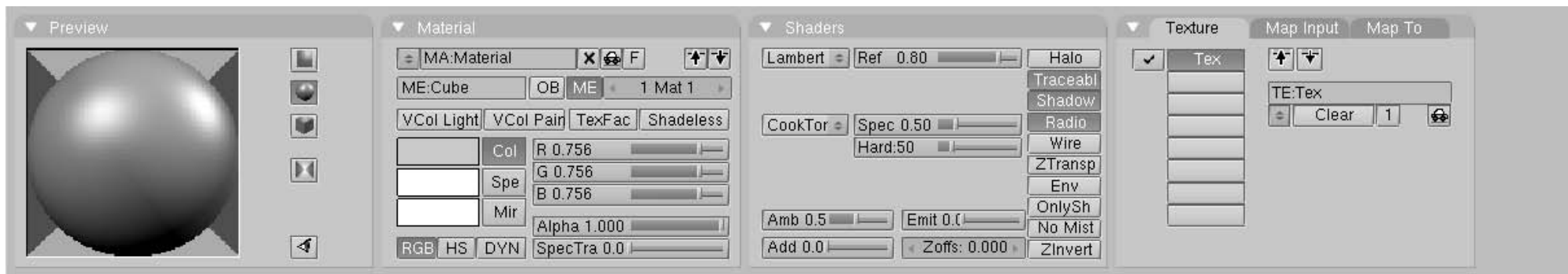
Figure 10-4. Agregar nuevo material.



Una vez que un objeto es seleccionado, presionando la tecla **F5** o , se cambia al contexto de Sombreado y aparece la ventana Botones de Material. Esta ventana aparecerá completamente vacía, a menos que el Objeto seleccionado ya tenga un material asignado. Si no hay un material asignado, agregar uno nuevo con el botón ([Figure 10-4](#)).

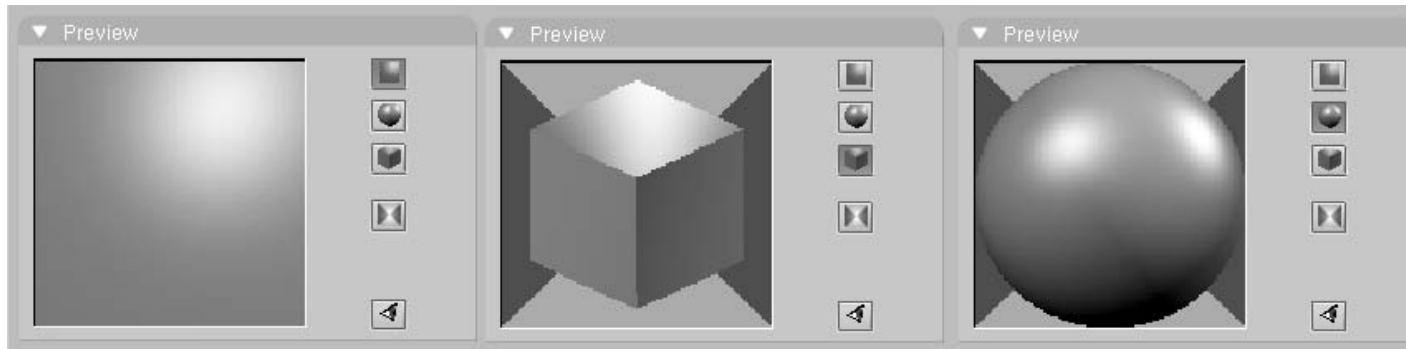
Una vez agregado un material, los botones aparecerán como se muestra en [Figure 10-5](#). Existen cuatro paneles. De izquierda a derecha: un panel de *Previo* (*Preview*), un panel de *Material*, un panel de *Shader* y un panel de *Textura*. Por ahora nos concentraremos en los primeros tres.

Figure 10-5. Botones de Material.



El panel *Previo* (*Preview*) muestra una previsualización del material. Por defecto muestra un plano visto desde arriba, pero puede ser cambiado a una esfera o un cubo con los botones que se encuentran a la derecha del panel ([Figure 10-6](#)).

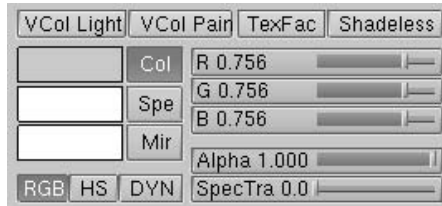
Figure 10-6. Previsualización del Material, plano (izquierda), esfera (centro) y cubo (derecha).



10.3.1. Colores del Material

El panel de *Material* ([Figure 10-7](#)) permite, entre otras cosas, elegir los colores del material.

Figure 10-7. Botones de Color de Material.



Cada material puede mostrar hasta tres colores:

- *El color básico del material* o Color de Difusión (Diffuse color), o, en resumen, el *Color* (botón *Col* en la interfase), que es el color usado por el sombreador de difusión (diffuse shader).
- *El color Especular*, indicado por el botón *Spe* en la interfase, es el color usado por el sombreador especular (specular shader).
- *El color de Espejo (The Mirror color)*, indicado por el botón *Mir* en la interfase, es el color usado por texturas especiales, para simular los reflejos especulares (mirror reflections). (Mas información sobre este tema en la sección de Mapeado de Ambiente.

Los botones ya mencionados seleccionan el color pertinente, que es mostrado en la vista previa inmediatamente a la izquierda de cada uno. Los tres deslizadores (sliders) a la derecha, permiten cambiar los valores para el color *activo* mediante un esquema RGB o HSV. Se pueden seleccionar estos esquemas con los botones *RGB* y *HSV* que se encuentran abajo.

El botón *DYN* se usa para configurar las propiedades Dinámicas del Objeto dentro del motor de TiempoReal (que queda fuera del alcance de este libro), mientras que los cuatro botones de arriba se relacionan con *Pintura de Vértices (Vertex Paint)* y *Textura UV (UV Texture)*.

10.3.2. Los Sombreadores (The Shaders)

El panel *Sombreador (Shader)* ([Figure 10-8](#)) muestra dos Botones de Menú que permiten seleccionar un sombreador de difusión (diffuse shader) ([Figure 10-9](#)) y un sombreador de especularidad (specular shader) ([Figure 10-10](#)).

Figure 10-8. Botones del Sombreador de Material (Material Shader buttons).

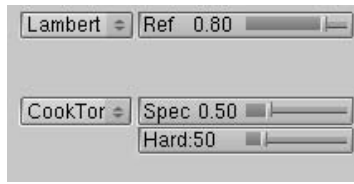
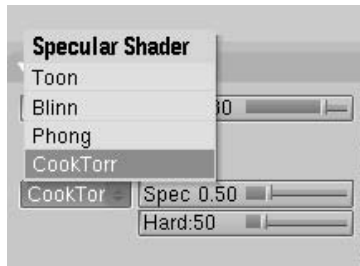


Figure 10-9. Sombreadores de Difusión del Material (Material Diffuse shaders).



Figure 10-10. Sombreadores de Especificidad del Material (Material Specular shaders).

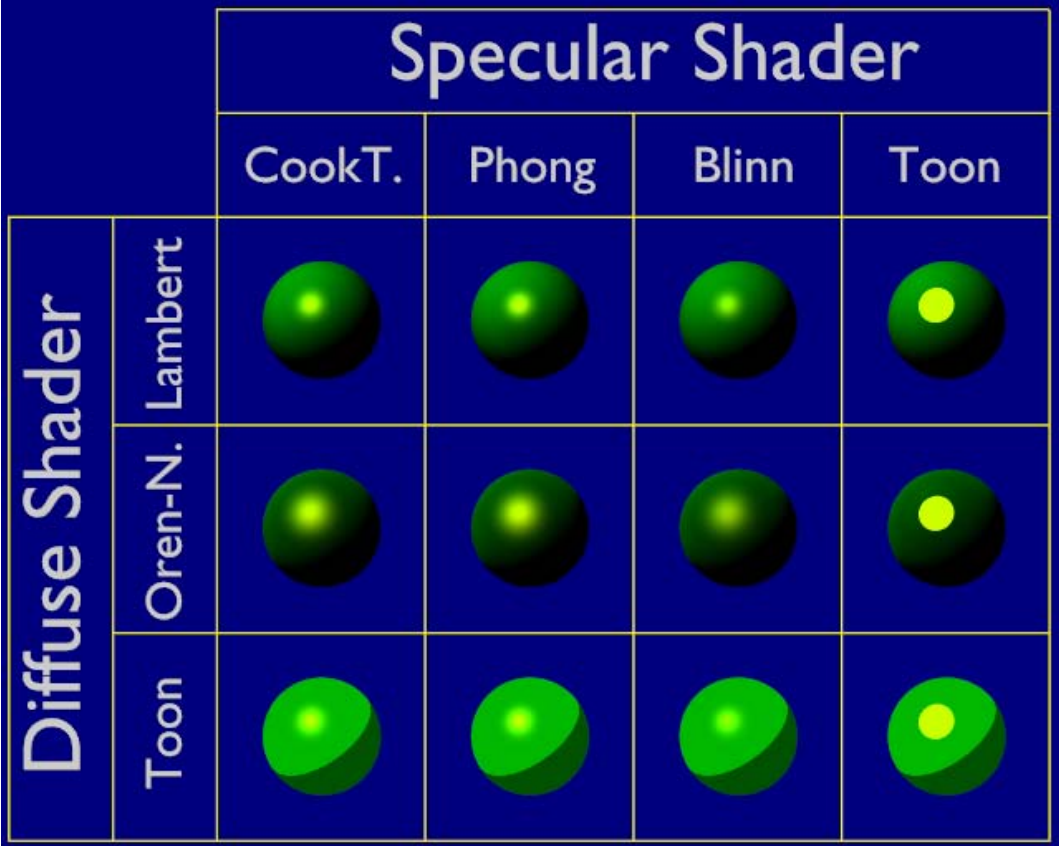


Los dos deslizadores (sliders) al costado, valido para todos los sombreadores, determinan la intensidad de los fenómenos de Difusión y Especificidad. El deslizador (slider) *Ref* tiene un rango de 0 a 1, mientras que *Spec* posee un rango de 0 a 2. Hablando en términos puramente físicos, si A es la energía lumínica que afectando al objeto, Ref multiplicado por A es la energía difundida (diffused) y $Spec$ multiplicado por A es la energía reflejada especularmente. Para que sea físicamente correcto, esto debe ser $Ref + Spec < 1$, o el objeto irradiaría más energía de la que recibe. Pero esto es CG, así que no es necesario ser tan estricto sobre cuestiones físicas.

Dependiendo del sombreador seleccionado, pueden aparecer otros deslizadores (sliders), permitiendo configurar los variados parámetros discutidos en la introducción.

Con la intención de ser completo, [Figure 10-11](#) muestra todas las combinaciones posibles. Por supuesto, ya que existen tantos parametros, esto es sólo una pequeña muestra.

Figure 10-11. Descripción de Sombreadores (Shader overview).



10.3.3. Manipulando Materiales (Tweaking Materials)

Los botones de material restantes, ambos en los paneles de *Material* y *Sombreadores (shaders)*, producen algunos efectos interesantes.

Figure 10-12. Deslizadores adicionales de Material (Additional material sliders).

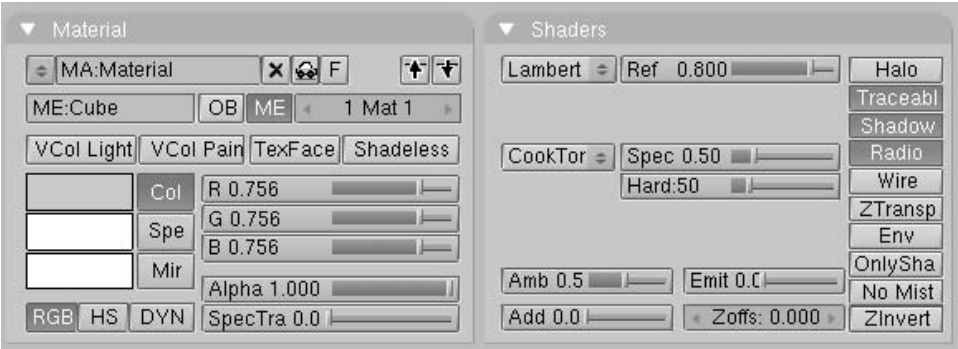
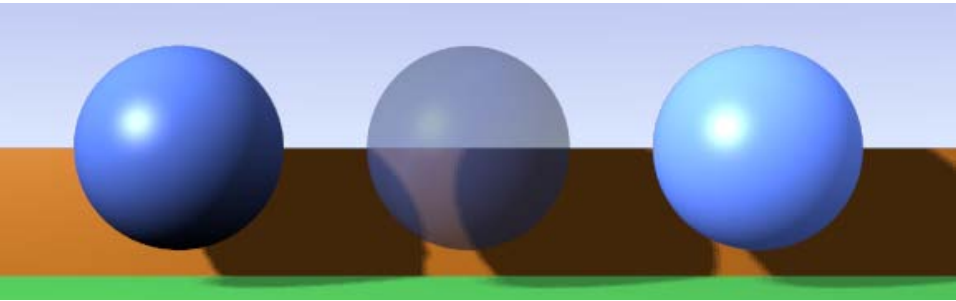


Figure 10-12 muestra algunos deslizadores (sliders) interesantes. *Alpha* maneja la opacidad del material; 1 es completamente opaco y 0 es completamente transparente. *SpecTra* obliga a los brillos especulares (specularity highlights) en objetos transparentes a ser opacos. *Shadeless* hace al material insensible a su sombreado (shading), dándole un color de difusión uniforme.

En el panel *Sombreadores (Shaders)*, el deslizador (slider) *Emit*, da como resultado, si es distinto de cero, un carácter emisor al material. Esto hace visible al material, aún sin luces, y puede ser él mismo una fuente de luz si se utiliza el motor de Radiosidad (Radiosity) ([Figure 10-13](#)).

Figure 10-13. Material común (izquierda), material con Alpha < 1 (centro) y material con Emit > 0 (derecha).



La columna de botones restante ([Figure 10-14](#)) activa algunas funciones especiales. El botón de más arriba, *Halo*, convierte al material en un material ‘Halo’, que será explicado más adelante. *Traceable*, *Shadows* y *Radio* están activos por defecto. El primero permite al material *proyectar* sombras, mientras que el segundo le permite *recibir*las; el tercer botón permite al material ser tenido en cuenta si se hace un render con Radiosidad (Radiosity).

Figure 10-14. Botones especiales de Material.



Wire renderiza el Objeto como una estructura de alambre (wireframe). *ZTransp* es necesario para activar el efecto de transparencia por Alpha.

Los otros botones no son usados muy seguido y se describen en la sección de referencia al final de este libro.

10.4. Sombreadores de Rampa (Ramp Shaders)

por Kenneth Styrberg

Relativo a Blender v2.34

10.4.1. Introducción

En muchas situaciones de la vida real - como en la piel o los metales - el color de las reflexiones especulares y difusas pueden variar levemente, basados en la cantidad de energía que recibe una superficie o del ángulo de incidencia de la luz.

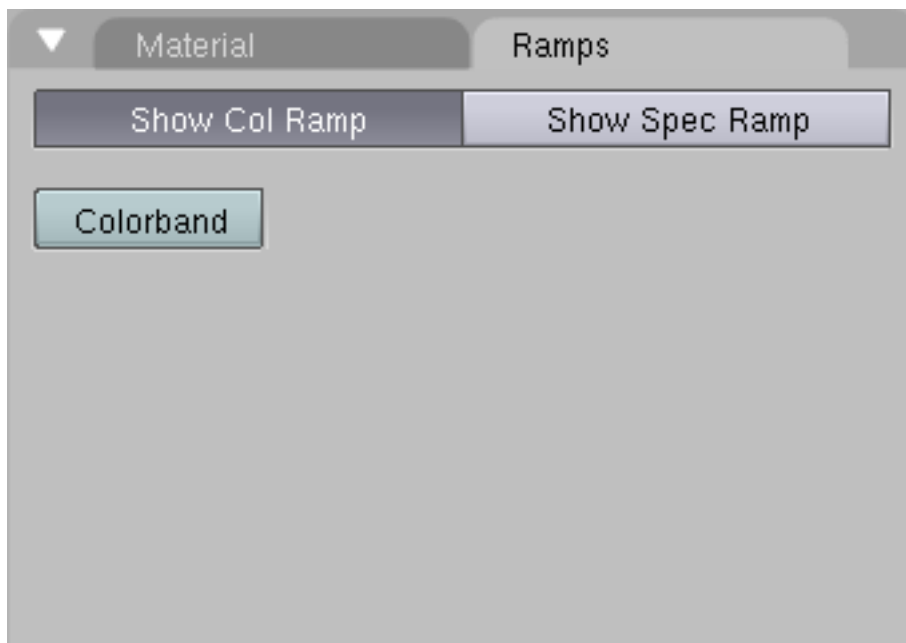
Las nuevas opciones de *Ramp Shader* (*Sombreador de Rampa*) en Blender, permiten establecer un rango de colores para un *Material*, y definen de qué manera variará este rango sobre la superficie y como se mezclará con el "color real" (comúnmente desde el material o como salida de textura).

Como el calculo de texturas en Blender, tiene lugar antes que el sombreado, el *Ramp Shader* puede reemplazar completamente las texturas o el color del material. Pero mediante el uso de las opciones de mezcla y valores Alfa, es posible crear una capa adicional de sombreado en los materiales de Blender.

10.4.2. Interfaz

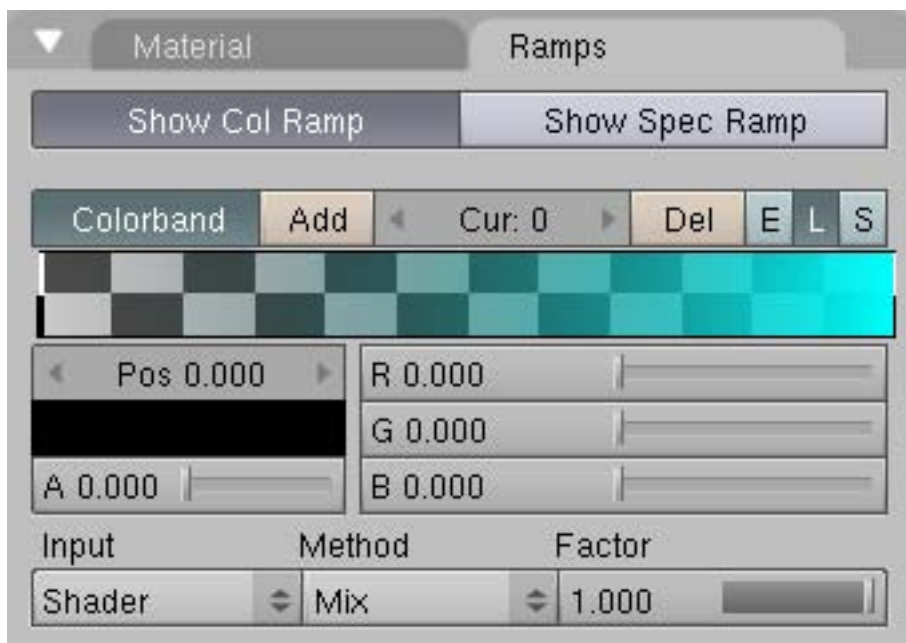
El panel *Ramps* (*Rampas*) está localizado en el Contexto de *Materiales* (**F5**). Aquí puede usar los dos botones superiores para mostrar tanto la configuración de las rampas difusas pulsando **Show Col Ramp** o de las rampas especulares con **Show Spec Ramp**. ([Figure 10-15](#)).

Figure 10-15. Panel Ramps.



Presionando el botón **Colorband** se habilitan los *Sombreadores de Rampa*. Por omisión, se abre con dos colores, el primero con **Alpha** = 0, y sin color y el segundo con **Alpha** = 1 y un color cian. ([Figure 10-16](#)).

Figure 10-16. Panel Colorband de las Rampas.



- **Add** - Agrega un nuevo color en el medio de la colorband en gris neutro por omisión.
- **Cur** - Muestra el número de color de la colorband seleccionada actualmente
- **Del** - Borra la posición actual.
- **E/L/S** - Define el tipo de interpolación entre el color del *Ramp Shader* y el color del *Material*. **E** - Ease o Cubico, **L** - Lineal y **S** - B-Spline.
- **Alpha** - Define en qué medida se verá el efecto del *Ramp Shader*. Un valor para *Alpha* de **0**

significa que el *Ramp Shader* es totalmente transparente y no será visible en el material final. Un valor de **1** hace que el *Ramp Shader* sea opaco. Si definió colores con diferentes valores de *Alpha*, se los interpolará entre sí para lograr una transición suave entre los distintos valores de transparencia. Puede previsualizar los valores de *Alpha* sobre la colorband con el patrón en forma de damero detrás de la colorband. Si el patrón es visible entonces la transparencia es menos que **1**.

- **R/G/B** - Los valores RGB del color actual. Puede pulsar con **LMB** sobre el campo color bajo el campo *Pos* para elegir un color usando el *Selector de Colores*.

El color actual es indicado con una barra ligeramente más gruesa en la colorband. Para seleccionar una posición de color puede pulsar con **LMB** sobre la posición de color deseada o avanzar el número del color actual en el campo *Cur* arriba y abajo con las flechas izquierda y derecha. También puede pulsar **SHIFT-LMB** en el campo e ingresar el número de color requerido manualmente.

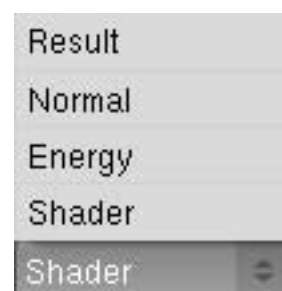
Puede mover la posición de un color pulsando simplemente con **LMB** y arrastrándolo sobre la colorband. También dispone de un campo *Pos* donde puede mover la posición con las flechas de cursor izquierda y derecha o pulsar **SHIFT-LMB** en el campo e ingresar la posición manualmente.

Note Si reordena las posiciones de los colores, serán reenumerados de modo que siempre empiecen con **0** desde la izquierda e incrementándose hacia la derecha.

Los dos botones emergentes y el deslizador en el fondo del panel definen como funcionan los *Sombreadores de Rampa*:

Input

Figure 10-17. Menú emergente Input.



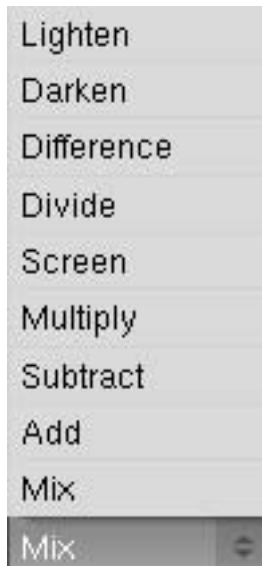
- **Shader** - El valor como sale de la fórmula de sombreado básico (como Lambert o Phong) define el color. Aquí sólo importa la dirección de la luz y no la cantidad de la misma.
- **Energy** - Como *Shader*, pero ahora también se consideran la energía, el color y la distancia de las luces. esto hace que el material cambien su color a medida que más luzbrilla sobre él.
- **Normal** - La normal de la superficie, relativa a la cámara, es empleada por el *Sombreador de Rampa*. Esto también es posible lograrlo por medio de una textura, pero se agrega por

conveniencia.

- **Result** - Las tres opciones anteriores funcionan por la luz, esta opción lo hace al final de todos los cálculos de sombreado. Esto permite control total sobre el sombreado completo, incluyendo resultados estilo "Caricatura". Utilizar valores de Alpha aquí es fundamentalmente útil para ajustar un retoque final del *Material*.

Method (Método)

Figure 10-18. Menú emergente Método.



El menú *Method* tiene varias opciones para el tipo de mezclado: **Mix (Mezclar)**, **Add (Sumar)**, **Subtract (Restar)**, **Multiply (Multiplicar)**, **Screen (Pantalla)**, **Divide (Dividir)**, **Difference (Diferencia)**, **Darken (Oscurecer)** y **Lighten (Aclarar)**.

Las opciones de *Method* ofrecen una opción para elegir cómo se realizará el mezclado entre el color de *Input* y los *Sombreadores de Rampa*.

Factor

Figure 10-19. Deslizador Factor.



El deslizador *Factor* denota el factor global del efecto del *Sombreador de Rampa* : **0** significa ningún efecto y **1.0** significa un efecto total.

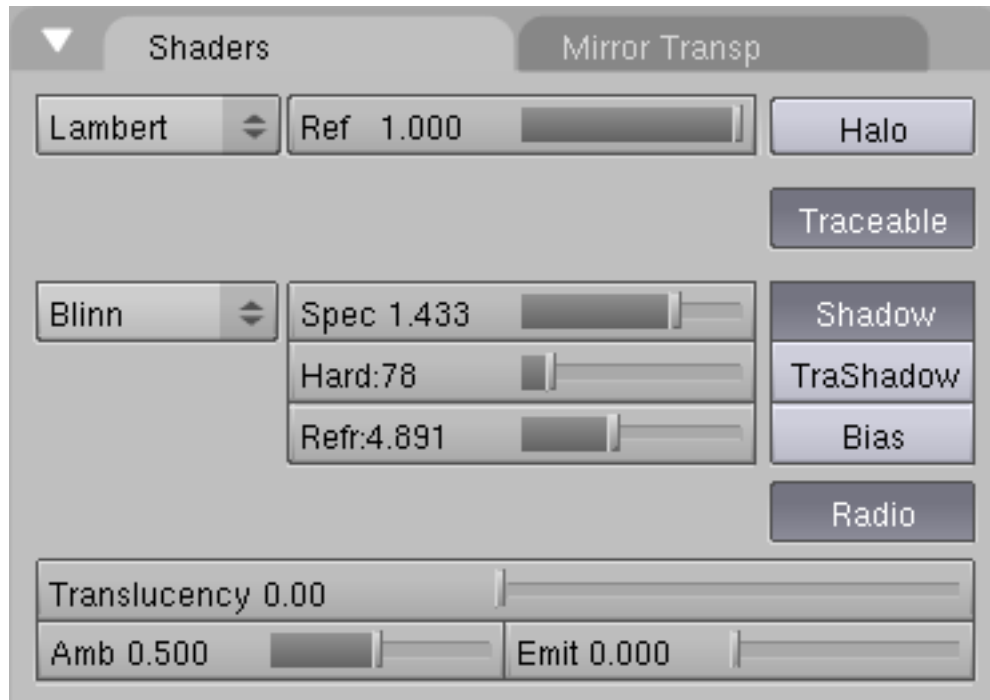
10.4.3. Un ejemplo

Veamos un sencillo ejemplo de uso de los *Sombreadores de Rampa*.

1. Elimine el objeto cubo inicial de la escena y cree una malla Mono ! (**SHIFT-A** -> **Add** -> **Mesh** -> **Monkey**). Presione **Subsurf** y ponga el nivel de *Subdivisión Subsurf* para pantalla y para render a **2**. Presione **Set Smooth** para obtener un Mono agradablemente suavizado! Todos estos botones están en el Contexto *Edición*. (**F9**).

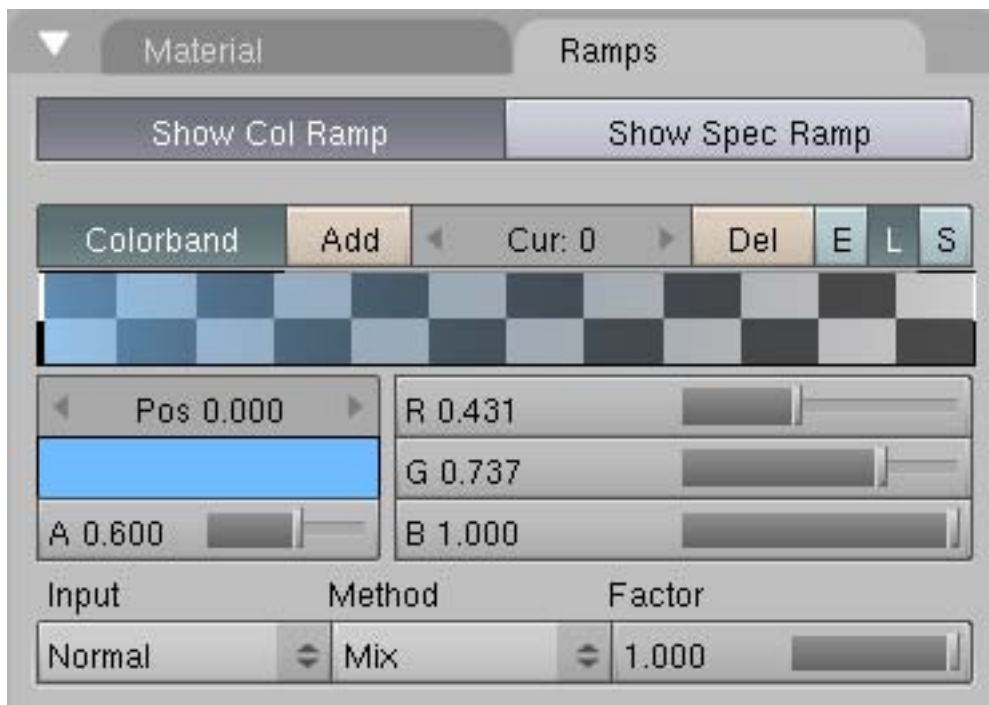
2. Ahora pulse **TAB** para salir de *ModoEdición*. Pressione **F5** para entrar al contexto *Materiales*. En el panel *Material* presiones **Add New** para crear un nuevo material. Modifique los parámetros en la pestaña *Shaders* como en la [Figure 10-20](#).

Figure 10-20. Valores del Sombreador.



3. Pulse sobre la pestaña **Ramps** para abrir el panel *Ramp Shader*. Presione el botón **Colorband** para activar el efecto *Sombreador de Rampa*. Ahora ponga los valores de los parámetros lo más parecido posible a la [Figure 10-21](#). Recuerde poner *Input* a **Normal**. El segundo color a la derecha tiene puesto *Alpha* = **0** y el color es negro puro.

Figure 10-21. Valores del Sombreador de Rampa.



4. En la pestaña *Ramps* presione **Show Spec Ramp** y ajuste los parámetros lo más fielmente posible a la [Figure 10-22](#) y la [Figure 10-23](#).

Figure 10-22. Color 0 del Sombreador de Rampa Especular.



Figure 10-23. Color 1 del Sombreador de Rampa Especular.



Aquí se ve el resultado renderizado con los valores que acabamos de ingresar. En la [Figure 10-24](#) no hay ningún *Sombreador de Rampa* activo. En la [Figure 10-25](#) está activada la *Rampa de Color* y finalmente en la [Figure 10-26](#) ambas rampas, la *Rampa de Color* y la *Rampa Especular* están activadas.

Tenga presente que acabamos de demostrar solo un efecto de los *Somrbeadores de Rampa* . Hay mucho más para explorar, pruebe cambiando los parámetros *Input* y *Method* para ver resultados totalmente diferentes de los que hemos visto recién en el ejemplo.

Figure 10-24. Ningún Sombreador de Rampa.



Figure 10-25. Rampa de Color.



Figure 10-26. Rampas de Color y Especular.



[Prev](#)

Materiales en la práctica

[Home](#)

[Up](#)

[Next](#)

Reflexiones con el Trazador de rayos

10.5. Reflexiones con el Trazador de rayos

Aún sin escribir

10.6. Transparencias con el trazador de rayos

Aún sin escribir

10.7. Materiales Múltiples

Relativo a Blender v2.31

La mayoría de los objetos están armados de modo que puedan modelarse por partes, con cada una de las mismas hecha de un material diferente. Pero en algunas situaciones, podría ser útil tener un objeto modelado como una malla única y que no obstante posea distintos materiales.

Considere la imagen de la seta como se la ve en la [Figure 10-27](#). Este objeto es una malla única a la que se quieren asignar dos materiales: uno para el tronco y otro para el sombrero. Veremos cómo hacerlo.

Figure 10-27. Malla de una Seta

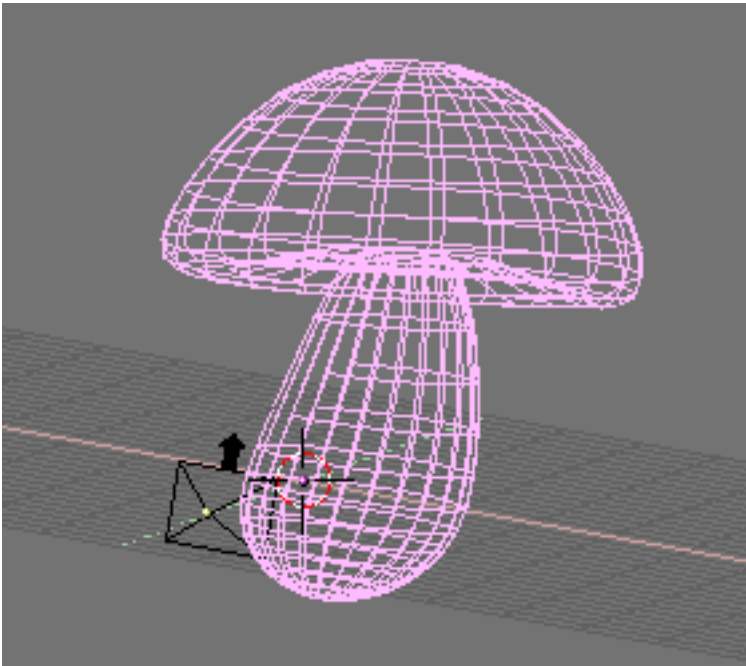
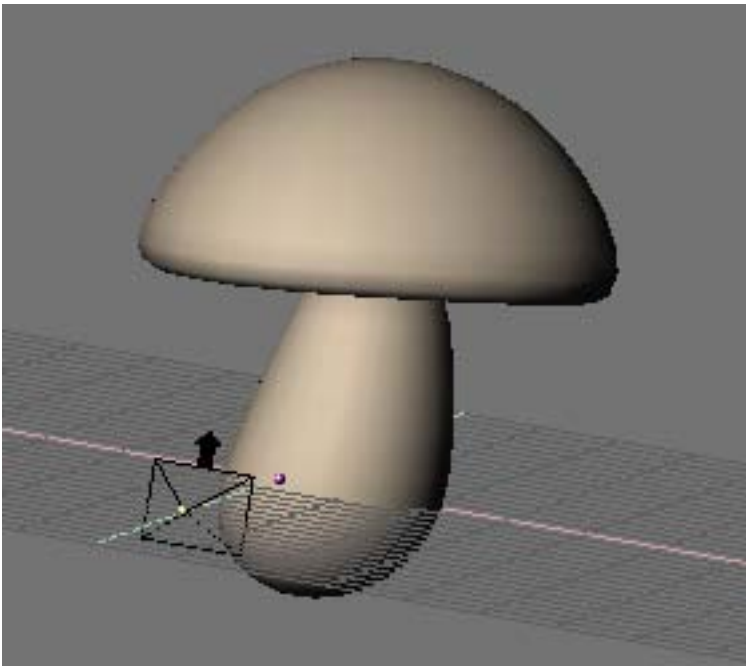
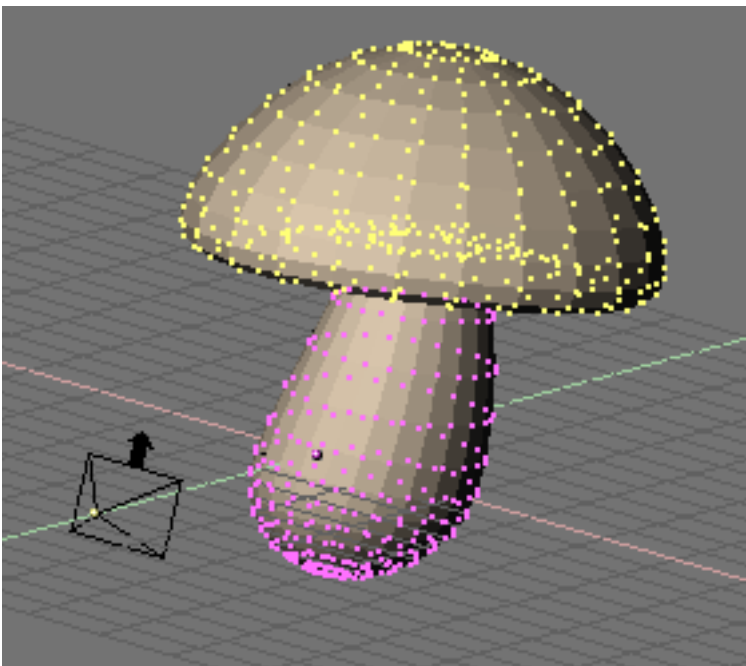


Figure 10-28. Seta con un material



1. Cree un material cremoso apropiado para el tronco de la seta y asígnelo a la seta entera. ([Figure 10-28](#)).

Figure 10-29. Seta con los vertices del sombrero seleccionados.



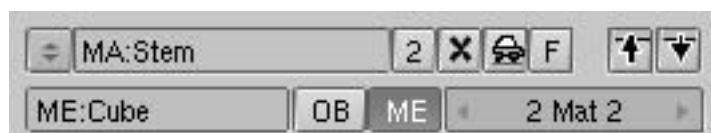
2. En una Ventana3D, ingrese en ModoEdición para la seta, y seleccione todos los vertices que pertenecen al sombrero de la misma ([Figure 10-29](#)).
3. En el Panel *Link and Material* de los Botones de Edición de Malla (**F9**) presione *New* ([Figure 10-30](#)).

Figure 10-30. Añadiendo un nuevo material a la malla.

4. La malla debería tener ahora dos materiales. En la etiqueta debería leerse *2 Mat: 2* lo que significa que el material número 2 de los 2 posibles, está activo. Las caras seleccionadas serán asignadas a este nuevo material cuando presione el botón *Assign*; las caras no seleccionadas mantendrán la asignación de material previa.

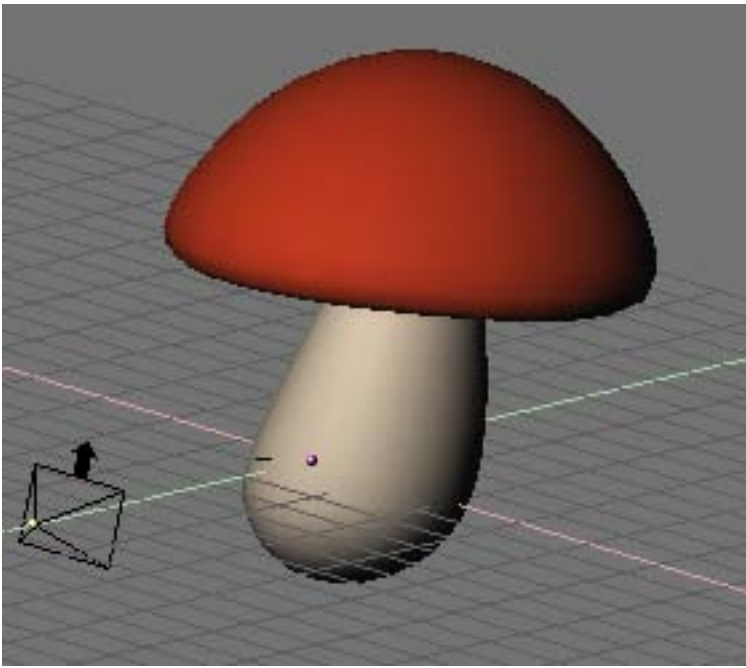
Para ver qué caras pertenecen a qué material, utilice los botones *Select* y *Deselect*. Navegue entre los materiales con el botón numérico *Mat:*. Puede disponer de hasta 16 materiales por malla.

5. En cualquier caso, ambos materiales de malla ¡ son instancias del mismo material ! Entonces, manteniendo activo el material que quiera cambiar, cambie a los Botones de Materiales (**F5**) donde encontrará un botón "2 Mat 2" idéntico al anterior. El material tiene en este momento dos usuarios, como se aprecia por el color azul en el nombre del material, y el botón que luce un "2" ([Figure 10-31](#)).

Figure 10-31. Material con usuarios múltiples

Pulse sobre el "2" y confirme la pregunta *OK? Single user*. Ahora tiene un duplicado exacto del material. El material original aún se llama "Tallo" ("Stem") y el duplicado es "Tallo.001" ("Stem.001"). Renombre el duplicado a "Sombrero" ("Cap"). Puede editar ahora el material a su gusto para obtener un sombrero de seta atractivo. ([Figure 10-32](#)).

Figure 10-32. Seta con dos materiales.

**Tip Texturas**

Si su material utiliza texturas, seguirán vinculadas aún después de hacer "usuario-único" al material. Para desvincularlas, de modo que pueda editar las texturas de ambos materiales individualmente, vaya a los botones de Texturas del material, y haga también a la textura "usuario único".

[Prev](#)

Transparencias con el trazador de rayos

[Home](#)
[Up](#)[Next](#)

Materiales Especiales

10.8. Materiales Especiales

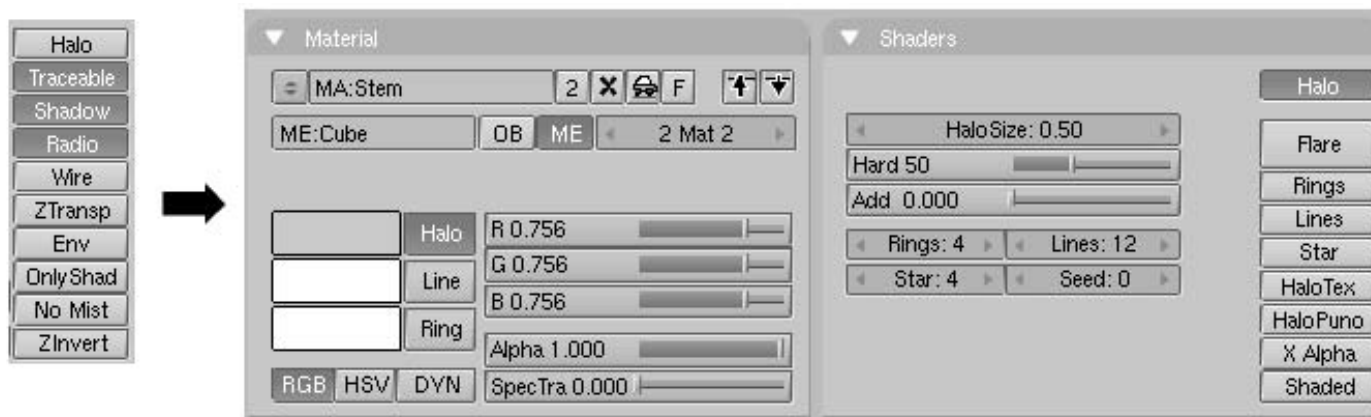
Relativo a Blender v2.31

Blender provee una gama de materiales que no obedecen al paradigma del sombreador y que son aplicados por-vértices en lugar de por-caras.

10.8.1. Materiales Halo

Pulsar **F5** para mostrar los botones de Material, y luego pulsar el botón *Halo* en el Panel *Shaders*. Los paneles cambian, como se muestra en [Figure 10-33](#).

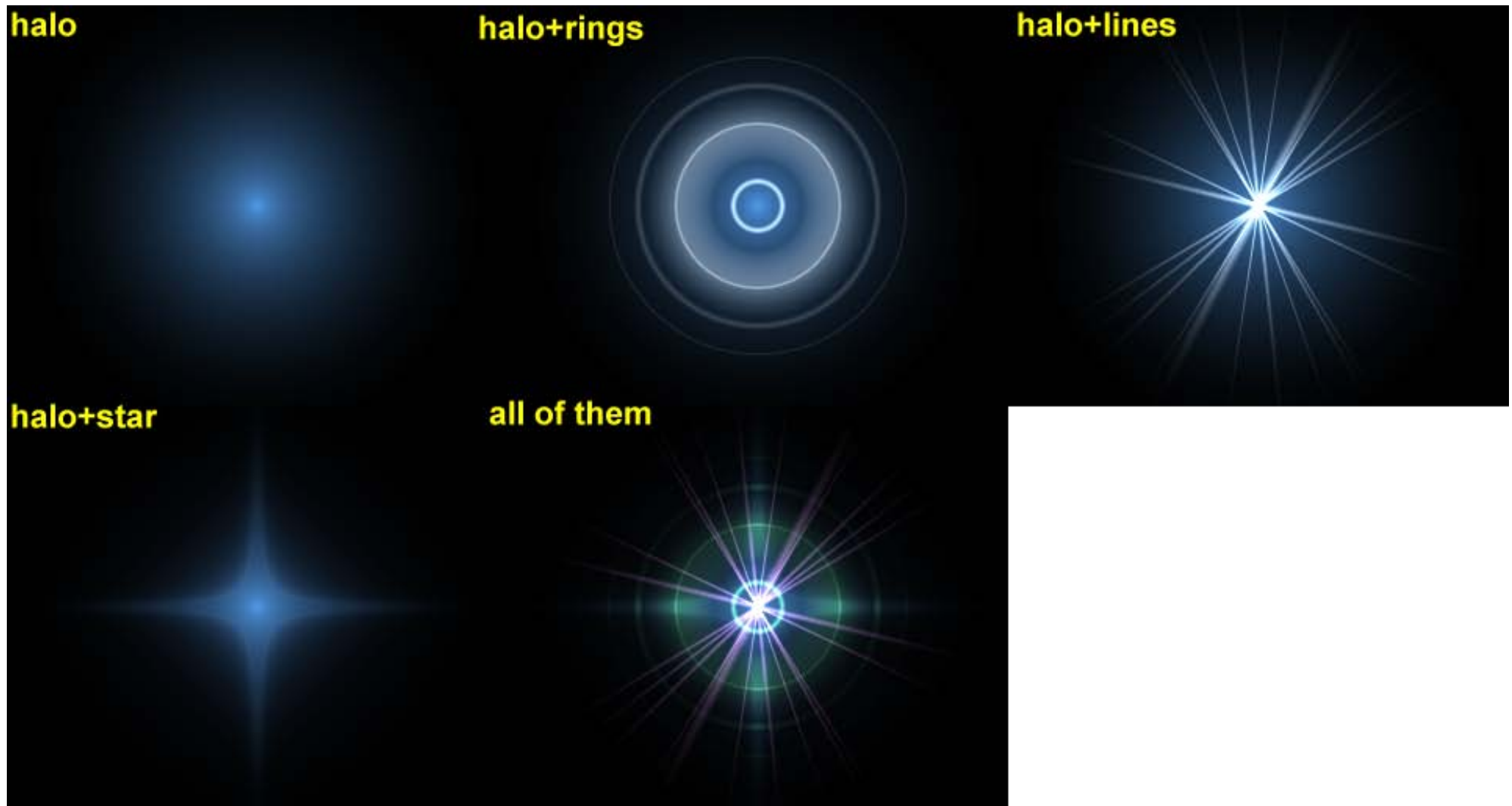
Figure 10-33. Botones Halo



Como se ve, las caras de la Malla ya no se renderizan; en cambio, se renderiza un "halo" en cada vértice. Esto es muy útil para sistemas de partículas ya que estos generan vértices libres, pero también pueden ser muy útiles para crear cierto tipo de efectos especiales, al crear un glow de un objeto, o al crear una fuente de luz visible.

Como se puede ver en los tres colores que, en un material estándar eran color Difuso (Diffuse), Especular (Specular) y Espejo (Mirror), ahora son relativos a tres características diferentes: el color del halo, el color de algún posible anillo y el color de alguna posible línea que se quiera agregar con los botones correspondientes en [Figure 10-33](#).

Figure 10-34. Resultados del Halo



[Figure 10-34](#) muestra el resultado de aplicar un material halo a una malla de un solo vértice. El tamaño del halo, su transparencia y su dureza pueden ser ajustados con los deslizadores correspondientes en [Figure 10-33](#). Los deslizadores *Añadir* (*Add*) determinan cuanto de los colores del halo son 'agregados a' (en lugar de mezclados con) los colores de los objetos que se encuentran atrás y junto con otros halos.

Para configurar el número de anillos, líneas, y puntos de estrella independientemente, una vez que se han activado los Botones correspondientes, utilizar los Botones Num *Anillos:* (*Rings:*), *Líneas:* (*Lines:*) y *Estrella:* (*Star:*). Los anillos y líneas son ubicados y orientados al azar. Para cambiar su patrón se puede modificar el valor del Botón *Seed:*, que configura el generador de números al azar.

Usaremos un material halo para crear un display de matriz de puntos.

1. Para comenzar, agregar una grilla de 32 x 16 unidades. Luego, agregar una cámara y ajustar la escena de manera de tener una buena vista de la cartelera.

2. Usar un programa de 2D para crear algún texto rojo sobre fondo negro, usando una tipografía simple y en negrita. [Figure 10-35](#) Muestra una imagen de 512 píxeles de ancho, por 64 de alto, con algo de espacio en negro a ambos lados.

Figure 10-35. Textura de la Matriz de Puntos.



3. Agregar un material para la cartelera, y convertirlo en tipo *Halo*. Asignar un valor de 0.06 a *HaloSize*. Al renderizar la escena debería verse una grilla de puntos blancos.
4. Agregar una Textura, cambiar a Botones de Textura y convertirla en textura de imagen. Cuando se carga la imagen y se renderiza nuevamente, deberían verse algunos puntos teñidos de rojo sobre la grilla.
5. Volver a Botones de Material y ajustar el parámetro *sizeX* a 0.5 aproximadamente, luego renderizar; el texto debería estar centrado en la cartelera.
6. Para eliminar los puntos blancos, ajustar el color del material para que sea de un rojo oscuro y renderizar. Deberían verse solamente los puntos rojos, pero la cartelera aún se ve demasiado oscura. Para corregir esto, entrar en Modo Edición de la cartelera y copiar todos los vértices utilizando el atajo de teclado **SHIFT-D**. Luego, ajustar el brillo mediante el valor de *Add* en los Botones de Material.

Figure 10-36. Visualización de la Matriz de Puntos (Dot Matrix display)



Se puede animar la textura para que se mueva sobre la cartelera, usando el valor *ofsX* en el panel *Textura (Texture)* de Botones de Material. (Se puede usar una resolución mayor para la grilla, pero habrá que achicar el tamaño de los halos o de otra manera se superpondrán.) ([Figure 10-36](#)).

Tip Texturizando el Halo

Por defecto, las texturas son aplicadas a los objetos con las coordenadas de Objeto, y actúan sobre los halos afectando su color, sobre las bases del color del vértice que origina al halo.

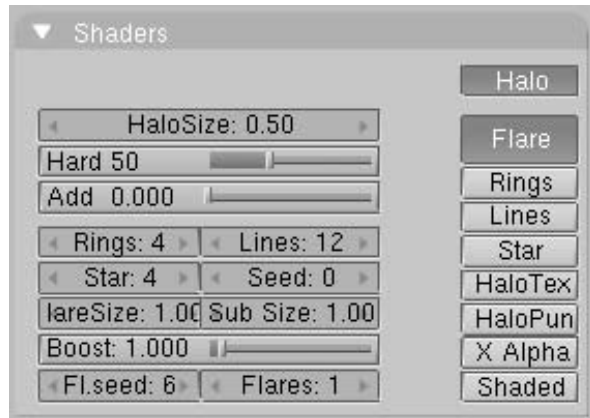
Para que la textura tenga efecto *dentro* del halo, y por lo tanto, para que varíe los colores o transparencias, presionar el botón *HaloTex*; esto hará que se mapee la textura completa sobre *cada* halo. Esta técnica es muy útil cuando se quiere crear un efecto de lluvia realista, o algo por el estilo, utilizando un sistema de partículas.

10.8.2. Lens Flares

El ojo humano ha sido entrenado para creer que una imagen es real si muestra artefactos que resultan del proceso mecánico de la fotografía. *Desenfoque de Movimiento* (*Motion blur*), *Profundidad de Campo* (*Depth of Field*), y *lens flares* son sólo tres ejemplos de estos artefactos. Los dos primeros se explican en [Chapter 17](#); el último se puede producir mediante halos especiales.

Un lens flare simulado le dice al espectador que la imagen fue creada con una cámara, lo que hace que el espectador crea que es auténtica. Los lens flares se crean en Blender a partir de una malla, utilizando primero el botón *Halo* y luego las opciones *Flare* en el Panel *Sombreadores* (*Shaders*) dentro de la configuración del material. Prueba activando *Anillos* (*Rings*) y *Líneas* (*Lines*), pero mantiene sutiles sus colores. Juega con los valores de *Flares*: y *Fl.seed*: hasta lograr un resultado que sea agradable a la vista. Puede ser necesario jugar con *FlareBoost*: para obtener un efecto mas fuerte ([Figure 10-37](#)). (Esta herramienta no simula la física de los fotones viajando a través de un lente de vidrio; es sólo para engañar al ojo.)

Figure 10-37. Configuración de Lens Flare



El lens flare de Blender se ve bien en movimiento, y desaparece cuando un objeto oculta la malla que contiene el flare. ([Figure 10-38](#)).

Figure 10-38. Lens Flare



[Prev](#)

Materiales Múltiples

[Home](#)

[Up](#)

[Next](#)

Texturas

Chapter 11. Texturas

Las texturas son...

11.1. Texturas

Relativo a Blender v2.31 FALTAN TODAS LAS COSAS NUEVAS!

Las configuraciones de material vistas hasta el momento producen objetos lindos, suaves y *uniformes*. Por supuesto, este tipo de objetos no existen en la realidad, donde lo común es la no uniformidad.

Blender simula estas irregularidades, ya sean de color, intensidad de reflejos o brillos, rugosidad, etc, por medio de *texturas*.

11.1.1. Texturas desde el Punto de Vista del Material

En Blender, los Materiales y las Texturas forman bloques separados con la intención de mantener la interface simple y permitir una integración universal entre bloques de Texturas, Luces y Mundo.

Esta relación entre un Material y una Textura, llamado "mapeado", es bidireccional. Primero, la información que es enviada a la Textura debe ser especificada. Luego, se especifica el efecto que la Textura tiene sobre el Material. El panel *Textura*, a la derecha (y paneles similares existentes para los botones de Luces y Mundo), definen todos estos cálculos.

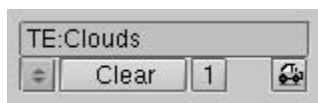
Para un Material sin texturas, el panel muestra una columna de 8 *canales* de textura vacios ([Figure 11-1](#)). Seleccionando uno de estos canales y presionando *Add New*, o seleccionando una textura existente con el Boton de Menú justo debajo ([Figure 11-2](#)), se agrega una textura y el Panel muestra dos etiquetas mas: *Map Input* y *Map To*. Los botones de las Etiquetas están organizados en la secuencia en que se realiza el "pipeline de la textura".

Figure 11-1. Canales de Textura



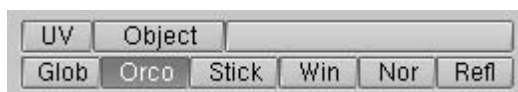
Cada canal tiene su propio mapeado individual. Por defecto, las texturas se ejecutan una después de la otra y son luego superpuestas. Como resultado, una textura agregada en el segundo Canal de Textura puede remplazar completamente a una en el primer Canal! Al lado de cada Canal de Textura que no está vacío hay un botón que permite activar o desactivar un determinado canal. Canales desactivados simplemente son eliminados de la línea de trabajo (pipeline).

Figure 11-2. Bloque de selección de Textura



La Textura en sí es designada por su nombre, que puede ser editado en el Botón de Texto que se encuentra arriba en el Menú de Botones de selección de Textura.

Figure 11-3. Coordenadas de entrada del Material



[Figure 11-3](#) muestra el panel de *Entrada de Mapa (Map Input)*. Cada Textura tiene una coordenada 3D (la coordenada de textura) como entrada (input). Los valores, pasados a la textura como coordenadas, para cada pixel de la imagen renderizada pertenecientes a un material dado, son calculados de acuerdo a estos botones:

- *UV* Utiliza un tipo especial de mapeado llamado "mapeado UV". Esto es especialmente útil cuando se utilizan imágenes como texturas, como se ve en [Section 11.5](#).
- *Objeto (Object)* Utiliza un objeto como fuente de las coordenadas, normalmente un Empty. El nombre del Objeto debe ser especificado en el botón de texto de la derecha. Esta es la forma preferida de ubicar una pequeña imagen, como un logo, sobre un punto determinado del objeto.
- *Glob* Utiliza las coordenadas 3D Globales de Blender..
- *Orco* Utiliza las coordenadas locales originales del Objeto.
- *Stick* Utiliza las coordenadas locales "sticky" del Objeto.
- *Win* Utiliza las coordenadas de la ventana de la imagen renderizada.
- *Nor* Utiliza la dirección de los vectores de normal como coordenadas.
- *Refl* Utiliza la dirección del vector de reflejo como coordenadas.

Figure 11-4. Mapeado de Textura (Texture mapping)



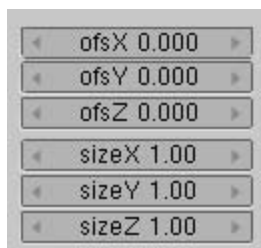
Si la textura es una imagen, es 2D, y se debe mapear el espacio 3D sobre esta. La forma más flexible de hacerlo es con mapeado UV (UV mapping), de lo contrario, se proveen cuatro posibles mapeados pre-configurados ([Figure 11-4](#)).

Figure 11-5. Transformación de Coordenadas



Las coordenadas X, Y y Z enviadas a la textura pueden ser mezcladas para obtener efectos especiales. Los botones en [Figure 11-5](#) permiten cambiar X por Y o por Z, y así, o apagarlos todos.

Figure 11-6. Factores de Offset y Escala de coordenadas



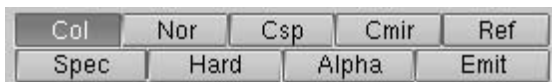
Las coordenadas se pueden escalar y trasladar asignandoles un offset ([Figure 11-6](#)).

Figure 11-7. Entradas de Textura (Texture Inputs)



Cambiando a la etiqueta *Map To*, [Figure 11-7](#) muestra la configuración de la entrada (input) de la textura. Los tres botones determinan si una textura debe ser usada como *Stencil* (una máscara para los siguientes canales de textura); una textura *Negativa* (*Negative*) (asignándole valores negativos en lugar de positivos); o solamente como una textura de intensidad, en blanco y negro (*No RGB*). Tres deslizadores (sliders) debajo de estos botones definen el color base de la textura, que pueden ser anulados por las especificaciones de color dentro de la definición de la textura. Finalmente, el último deslizador (slider) define la intensidad del efecto de la textura.

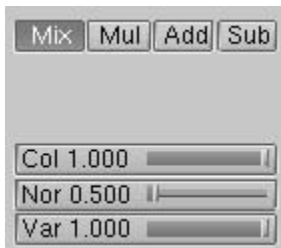
Figure 11-8. Salidas de Textura (Texture Outputs)



[Figure 11-8](#) muestra los botones que determinan qué características del material estará afectando la textura. Algunos de estos botones poseen tres estados, lo que significa que la textura puede ser aplicada como positiva o negativa. Todos estos botones son independientes entre si.

- *Col* (activado/desactivado) (on/off) Utiliza la textura para alterar el color del Material.
- *Nor* (desactivado/positivo/negativo) Utiliza la textura para alterar la dirección de la normal local. Esto se usa para simular imperfecciones o desigualdades en la superficie, mediante mapeado de relieve (bump mapping).
- *Csp* (activado/desactivado) (on/off) Utiliza la textura para alterar el color Especular.
- *Cmir* (activado/desactivado) (on/off) Utiliza la textura para alterar el color de Espejo (Mirror).
- *Ref*, *Spec*, *Hard*, *Alpha*, *Emit* (desactivado/positivo/negativo) Utiliza la textura para alterar el Correspondiente valor del Material.

Figure 11-9. Configuración de Salida (Output settings)



La configuración de salida (ouput settings) ([Figure 11-9](#)) determina la intensidad del efecto de la salida de Textura (Texture Ouput). Se pueden mezclar mediante valores estándar, incluyendo adición, substracción o multiplicación. Las texturas dan tres tipos de salidas.

- Texturas RGB: retornan tres valores que siempre afectan al color.
- Texturas de Relieve (Bump): devuelven tres valores, que siempre afectan el vector de la normal. Solamente "Stucci" e "Imágen" pueden dar normales.
- Texturas de Intensidad: retornan un único valor. Esta intensidad puede controlar el "Alpha", por ejemplo, o determinar la intensidad de un color específico utilizando los botones de *mapeado*

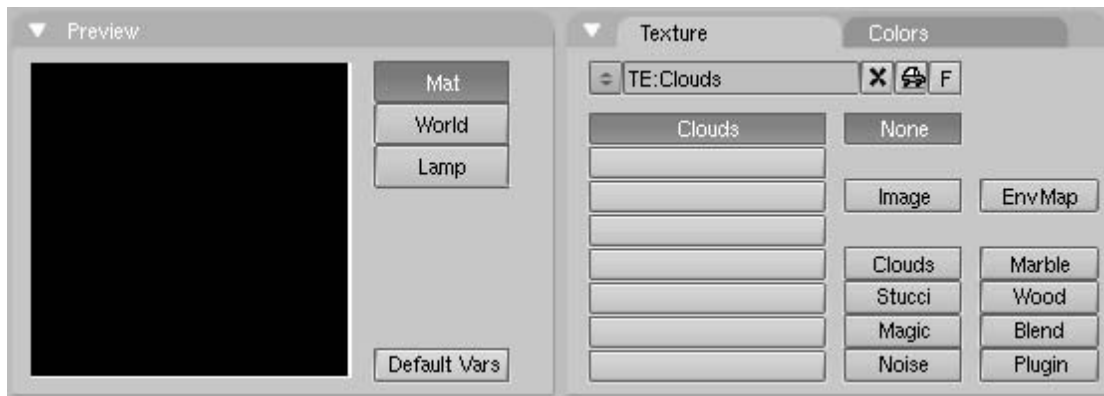
Se puede ajustar la intensidad de estas configuraciones por separado, utilizando los deslizadores (sliders) pertinentes ([Figure 11-9](#)).

11.1.2. Texturas (Textures themselves)

Una vez que una nueva textura ha sido agregada a un material, puede ser definida cambiando al sub-contexto de Botones de Textura (**F6**) o

 del contexto de Sombreado, para obtener [Figure 11-10](#).

Figure 11-10. Botones de Textura



Una ventana de Boton de Textura nueva, vacía, muestra dos paneles: una *Previsualización* de la Textura y un panel de *Textura*, este último, con dos pestañas.

En el panel de *Previsualización*, los botones definen si es una textura de Material, Luz o Mundo, y el botón *Default Var* permite volver los parámetros de la textura a los valores por defecto.

La pestaña *Texture* muestra los canales de textura y el Menu de Botones de Textura del material conectado. Las dos columnas de Botones seleccionan el tipo de Textura. El botón *Imágen* permite cargar una imagen y usarla como textura (el primer botón es simplemente "sin textura"). El tercer botón permite utilizar un tipo especial de textura, Mapa de Ambiente (Environment Map) (*EnvMap*). El último botón (*Plugin*), permite cargar un trozo de código externo para definir una textura. (Estos tres botones son más bien especiales y serán explicados por separado más adelante). En cuanto se elige un tipo de textura, un nuevo Panel aparece, con su nombre idéntico al del tipo de textura, donde se pueden configurar los parámetros de la textura.

Los botones restantes definen texturas 3D *procedurales*, que son texturas definidas matemáticamente. Generalmente son más fáciles de usar, y dan resultados increíbles una vez se les toma la mano. Se explicará sólo una de estos, el botón *Madera (Wood)*, dejando el resto para ser investigado por el usuario. (El capítulo de referencia de este libro contiene todos los detalles sobre cada tipo de textura).

Wood es un *procedural* simple, lo que significa que cada coordenada 3D puede ser traducida directamente a un color o un valor. Este tipo de texturas son 3D "real". Esto quiere decir que encajan perfectamente entre ellas en los bordes, y seguirán viendose como deben verse aún cuando son cortadas, como si un bloque de madera hubiera sido realmente cortado en dos.

Las texturas procedurales *no* son filtradas o tratadas con anti-aliasing. Esto rara vez es un problema: el usuario puede fácilmente mantener las frecuencias específicas dentro de límites aceptables.

Las texturas procedurales pueden bien producir texturas coloreadas o solamente texturas de intensidad. Si sólo se usan como texturas de intensidad, el resultado es una textura en blanco y negro, que puede ser mejorada utilizando las bandas de color (colorbands). La banda de color (colorband) es una herramienta generalmente menospreciada que se encuentra en la pestaña *Colores* en el Panel *Textura*, que permite un impresionante nivel de control sobre la forma en que la textura será renderizada. En lugar de simplemente renderizar cada textura como una progresión lineal de 0,0 a 1,0, se puede usar la banda de color (colorband) para crear un gradiente que pasa por una serie de variantes de color y transparencia (alpha) a gusto del usuario ([Figure 11-11](#)).

Figure 11-11. Banda de Color de Textura (Texture Colorband).



El uso experto de las bandas de color (colorbands) lleva a texturas de mármol y nubes realmente buenas. Para utilizarlas, seleccionar una textura procedural, como por ejemplo *Wood* y presionar el botón *Colorband*.

Colorband es el editor de gradientes de Blender. Cada punto sobre la banda puede ser ubicado en cualquier posición, y se le puede asignar cualquier color y transparencia. Blender hará una interpolación de los valores entre un punto y el siguiente. Para usarlo, seleccionar el punto que se quiere editar con el botón numérico *Cur:*, y luego agregar o borrar puntos con los botones *Add* y *Del*. Se muestran los valores de RGB y Alpha del punto activo, junto con la posición del punto sobre la banda. Para cambiar la posición del punto activo, arrastrar el puntero del ratón mientras se mantiene apretado el botón izquierdo del mismo.

Se pueden usar dos texturas *Madera (Wood)* para generar patrones en dos escalas diferentes, cada una produciendo un efecto diferente sobre la apariencia de la madera. Las texturas *Madera (Wood)* son idénticas excepto por la forma en que son mapeadas en la ventana de botones de material, y por las diferentes bandas de color (colorbands) que utilizan.

También se puede usar una textura *Nubes (Clouds)* para hacer un patrón de grano. Para ver el resultado de una sola textura, aislada de las demás, se pueden usar los botones que se encuentran en el panel *Textura*, en los Botones de Material.

Figure 11-12. Copiando y Pegando Texturas



Tip Copiando la configuración de la textura

Al agregar una textura existente, se conecta esa textura con el material, pero los parámetros de mapeado del material se mantienen intactos. Para copiar la configuración completa de la textura, incluso la configuración del mapeado, se debe copiar un determinado canal de textura y pegarlo en otro utilizando los dos botones-flecha que se encuentran [Figure 11-12](#).

[Figure 11-13](#), [Figure 11-14](#) y [Figure 11-15](#) muestran las tres texturas individuales que, cuando se combinan en un solo material y son mapeadas sobre varios parámetros del mismo, crean buenas texturas de madera ([Figure 11-16](#)).

Figure 11-13. Primera textura del anillo de la Madera

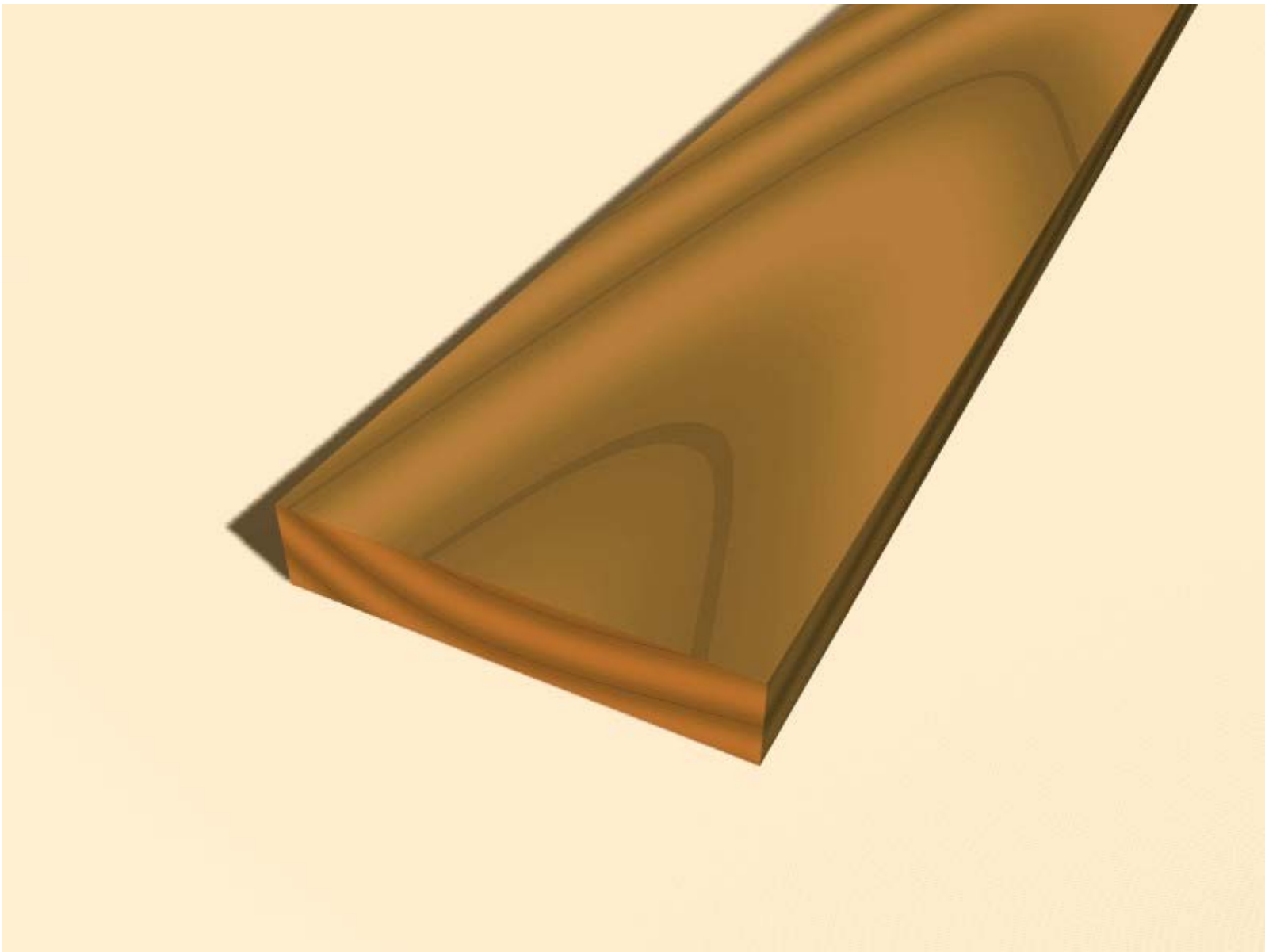


Figure 11-14. Segunda textura del anillo de la Madera

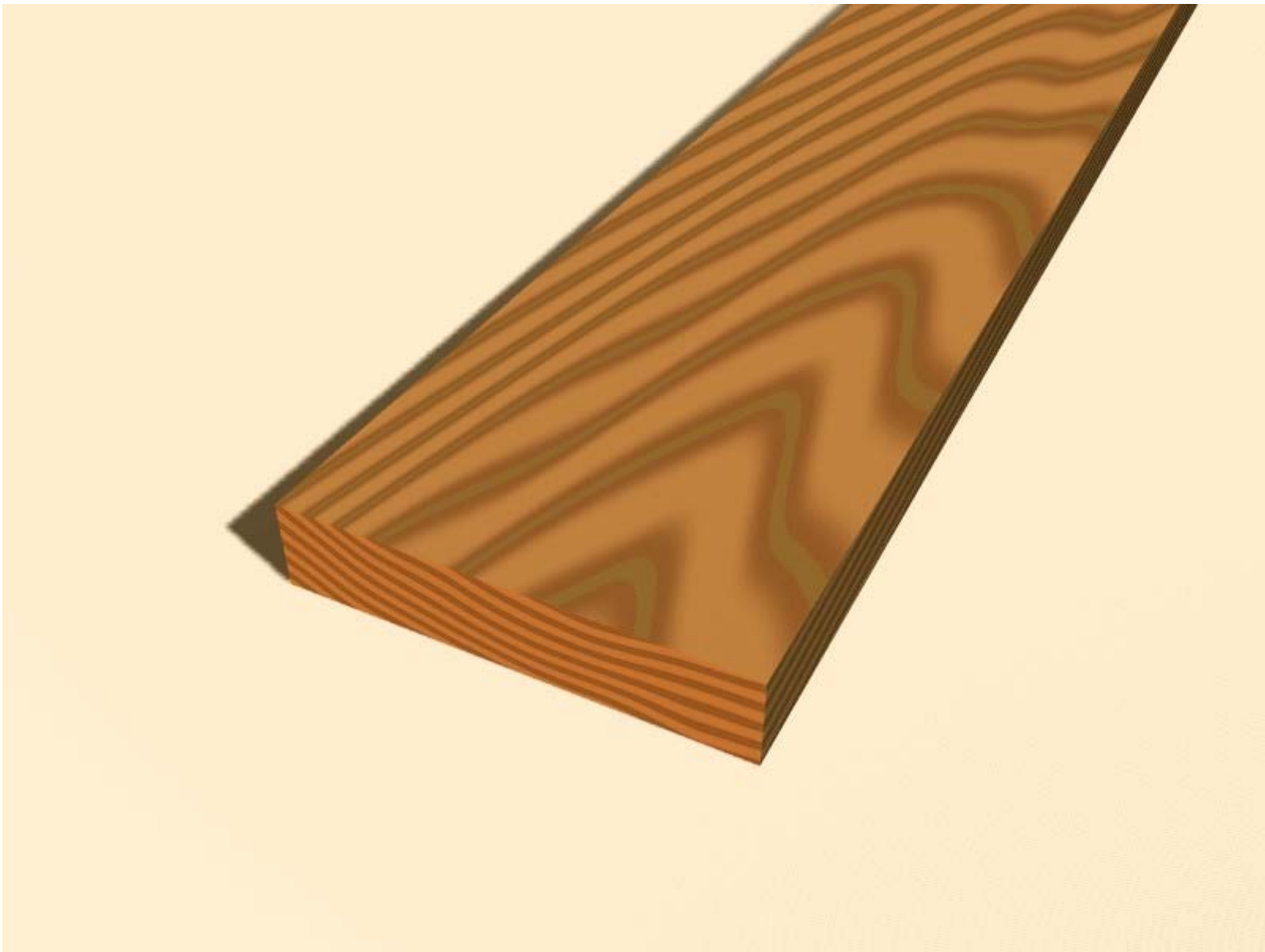
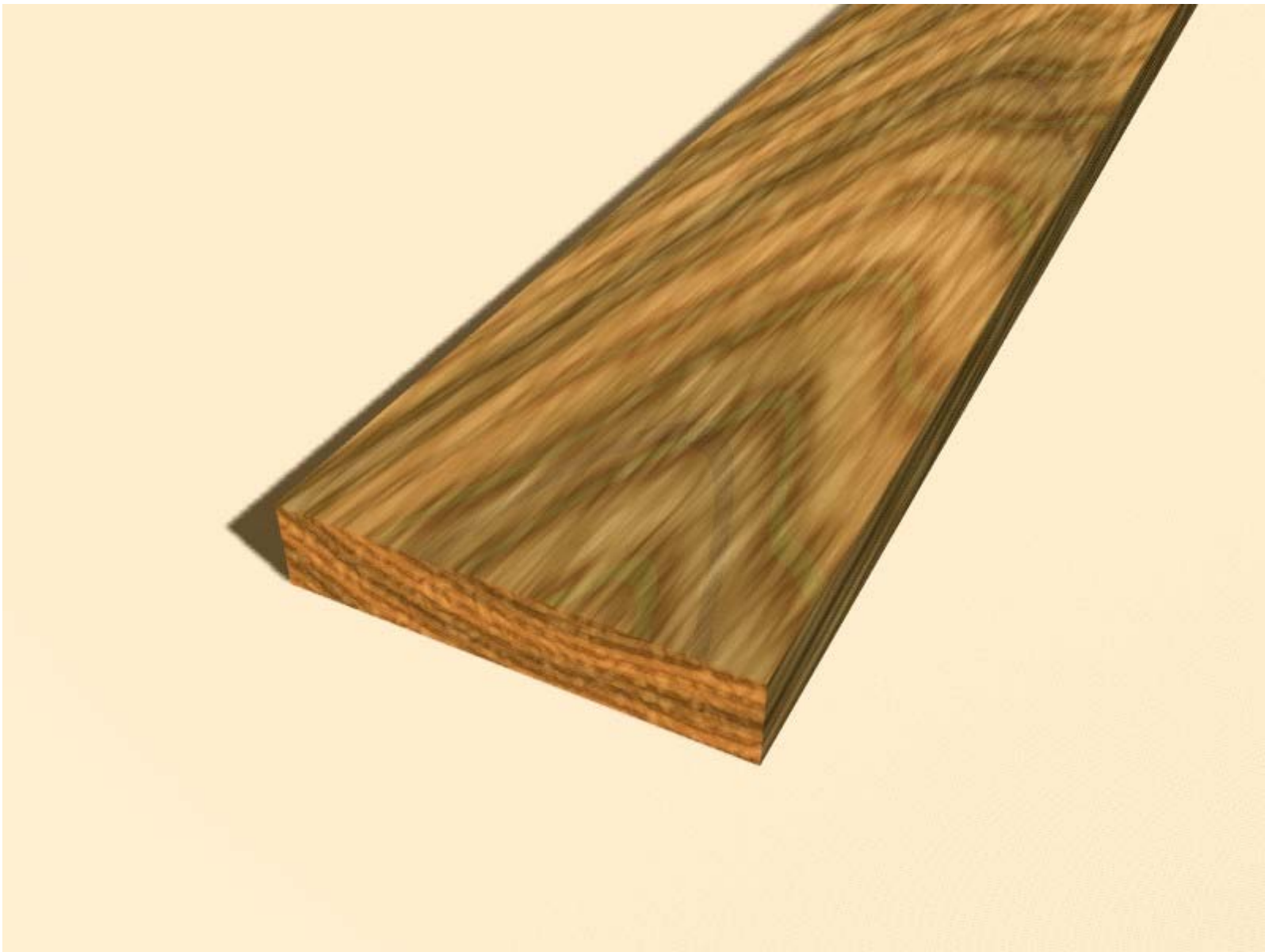


Figure 11-15. Textura Nubes (Clouds texture)



Figure 11-16. Resultado final



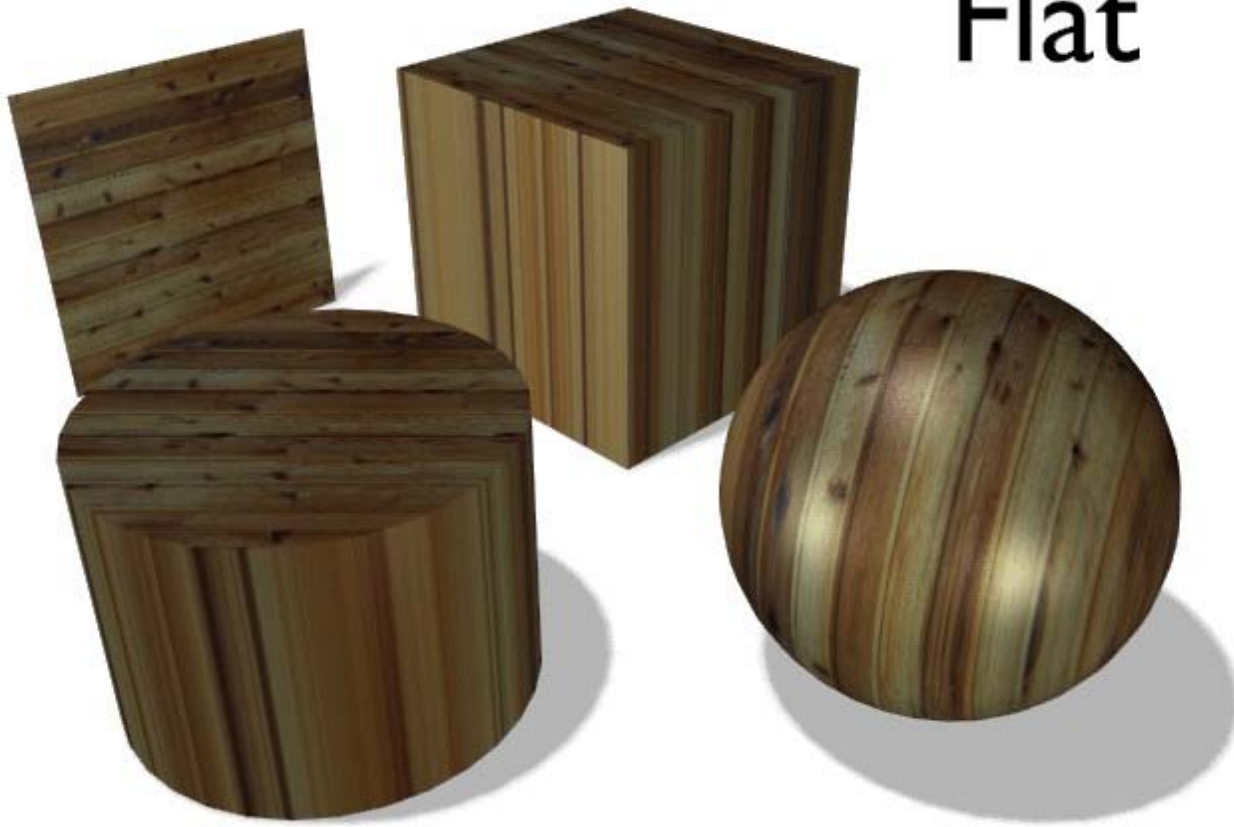
11.1.3. ImageTexture

La textura de imagen es la única textura realmente 2D, es la más frecuentemente usada, y una de las texturas más avanzadas que posee Blender. Las características incorporadas, como mapeado de relieves y perspective-corrected mip-mapping, filtrado, y anti-aliasing, garantizan imágenes asombrosas (Para esto hay que tener activado el botón *OSA*, que se encuentra en *DisplayButtons*). Debido a que las imágenes son bi-dimensionales, la forma en que la coordenada 3D es traducida a 2D se especifica en los botones de *mapeado* ([Figure 11-4](#)).

Los cuatro tipos de mapeado estándar son: *Plano (Flat)*, *Cúbico (Cube)*, *Tubular (Tube)* y *Esférico (Sphere)*. El tipo más adecuado de mapeado para un modelo determinado, dependerá de la forma general (simplificada) del objeto.

Figure 11-17. Mapeado Plano (Flat Mapping).

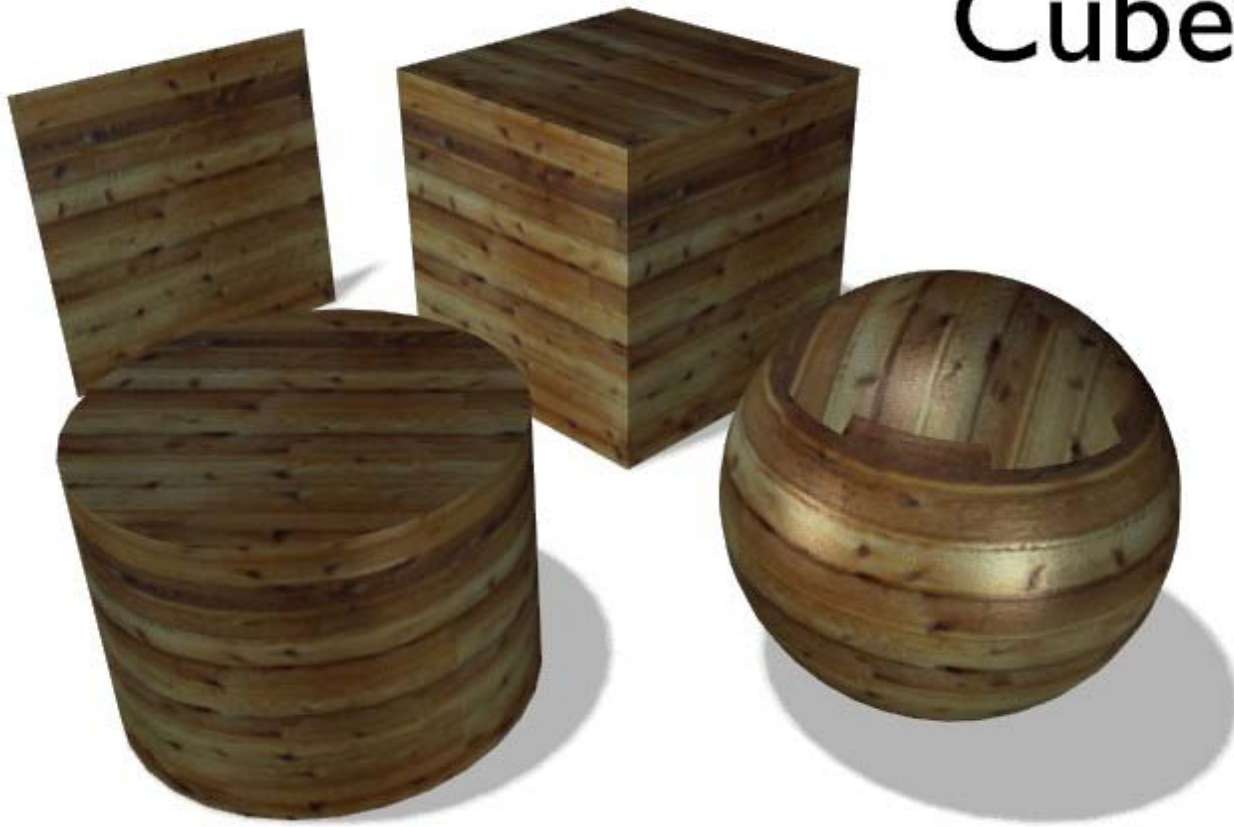
Flat



El mapeado Plano (Flat) ([Figure 11-17](#)) da los mejores resultados para caras planas simples. Produce efectos interesantes al aplicarlo sobre una esfera, pero comparado con una esfera mapeada con el mapeado Esférico (Sphere mapping), el resultado se ve chato. Sobre caras que no están sobre el plano del mapeado, el último píxel de la textura se repite, lo que produce bandas al usarlo sobre un cubo o un cilindro.

Figure 11-18. Mapeado Cúbico (Cube Mapping).

Cube



El mapeado Cúbico (cube-mapping) ([Figure 11-18](#)) normalmente da los mejores resultados si se usa sobre objetos que no son demasiado curvos u orgánicos (nótese las "costuras" sobre la esfera).

Figure 11-19. Mapeado Tubular (Tube Mapping).

Tube



El mapeado Tubular (tube-mapping) ([Figure 11-19](#)), posiciona la textura alrededor del objeto como una etiqueta sobre una botella. Por lo tanto, la textura se ve mas estirada sobre el cilindro. Este tipo de mapeado, obviamente, es muy útil para crear la etiqueta de una botella, o para poner pegatinas sobre objetos redondeados. Sin embargo, este tipo de mapeado no es cilíndrico, por lo que las "tapas" del cilindro no están definidas.

Figure 11-20. Mapeado Esférico (Sphere Mapping).

Sphere



El mapeado Esférico (sphere-mapping) ([Figure 11-20](#)) es el mejor tipo de mapeado para aplicar a una esfera, y es perfecto para la creación de planetas y cosas similares. Es a veces útil para la creación de objetos orgánicos. También produce efectos interesantes al aplicarlo sobre un cilindro.

Tip Moviendo una Textura

Como se explicó en la sección anterior, se puede manipular la textura en la sección de Botones de Material correspondiente a texturas. Existe una última característica importante para la manipulación de texturas.

Cuando se selecciona un objeto y se pulsa **TKEY**, se presenta la opción de escalar y mover el espacio de la textura de una manera visual, pero no se puede rotar la misma. Sin embargo, el mapeado de coordenadas de *Objeto* es mucho más flexible.

[Prev](#)[Materiales Especiales](#)[Home](#)[Up](#)[Next](#)[Mapas de Ambiente \(Environment Maps\)](#)

11.2. Mapas de Ambiente (Environment Maps)

Relativo a Blender v2.31

Las superficies brillantes que blender genera, muestran reflejos especulares. Lo irónico de estos sombreadores especulares es que son sólo sensibles a las luces (lamps). Específicamente, las superficies con sombreado especular muestran un punto de brillo como si fueran una reflexión de tipo espejo de la luz (lamp).

Todo esto tiene sentido excepto que si uno gira la cámara hacia la luz (lamp), ésta no se ve. La cámara ve esta luz sólo si está siendo reflejada por un sombreador especular, no directamente. Por otra parte, los objetos que aparecen muy brillantes en la escena (que reflejan much luz hacia la cámara), pero que no son luces (lamps), no aparecen en estos reflejos.

Es fácil hacer una luz (lamp) que este directamente visible para la cámara poniendo algún objeto renderizable en la escena, que se vea como algún tipo apropiado de lámpara, llama, sol, o algo por el estilo. Sin embargo, no hay una solución inmediata para el hecho que, los objetos alrededor no se ven en los reflejos especulares.

En una palabra, faltan *reflejos*. Este es el tipo de problema que trataremos utilizando la técnica de mapas de ambiente.

Así como se renderiza la luz que alcanza el plano visible utilizando la cámara para definir un punto de vista, se puede renderizar la luz que alcanza la superficie de un objeto (y por lo tanto, la luz que puede ser reflejada hacia la cámara).

El mapeo de ambiente de Blener renderiza un mapa cúbico de la escena en los seis puntos cardinales desde algún punto. Cuando los seis bloques de la imagen son mapeados sobre un objeto utilizando las coordenadas de *Refl*, crean la complejidad visual que el ojo espera ver en reflexiones brillantes.

Note Es útil recordar que el verdadero objetivo de esta técnica es lograr *credibilidad*, no *exactitud*. El ojo no necesita una simulación física exacta de la trayectoria de la luz; solamente necesita creer que la escena es real, viendo la complejidad que espera ver. La cosa menos creíble en la mayoría de las imágenes renderizadas es la esterilidad o pulcritud de la escena, no la exactitud.

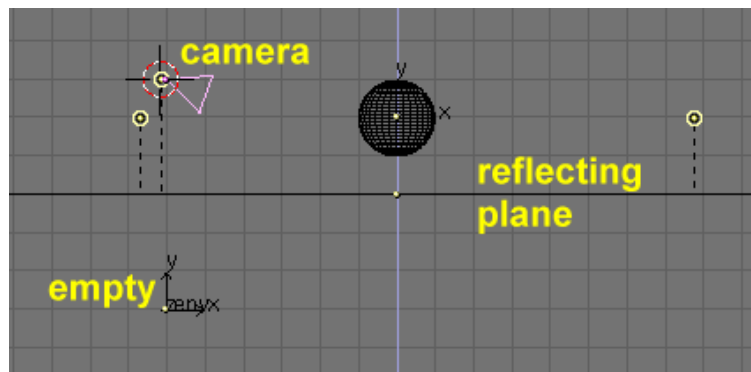
El primer paso cuando se crea un mapa de ambiente es definir el punto de vista para el mapa. Para empezar, hay que agregar un empty a la escena y ubicarlo *en la posición del reflejo de la cámara con respecto a la superficie reflejante*. (Esto es posible, estrictamente hablando, solo para superficies reflejantes planas.)

En el caso ideal, la posición del empty copiaría inversamente a la posición de la cámara a través del plano del polígono sobre el que está siendo mapeado. Sería realmente difícil crear un mapa de ambiente único para cada polígono de una malla detallada, por lo que hay que aprovechar que el ojo humano es fácil de engañar.

En particular, para objetos relativamente pequeños y complejos, se puede solucionar simplemente ubicando el empty cerca del centro. Nombramos al empty *env*, de manera que podamos referirnos a el por su nombre en la configuración del mapa de ambiente.

Crearemos una esfera reflejante sobre un plano reflejante, usando la configuración mostrada en [Figure 11-21](#).

Figure 11-21. Ejemplo del uso del Mapa de Ambiente

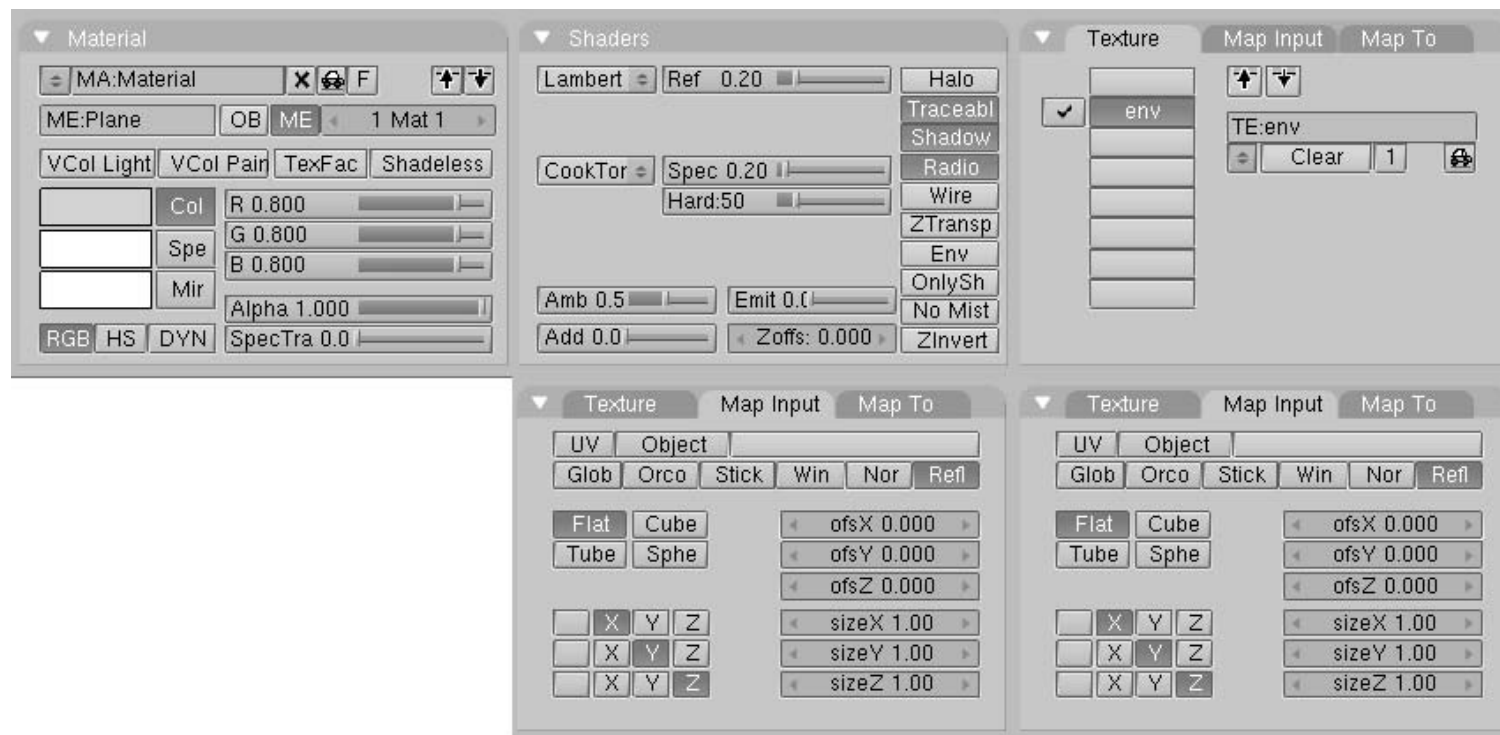


Nótese que el Empty "env" está ubicado exactamente debajo de la cámara, a una distancia de 3 unidades de blender desde el plano reflejante, que es igual a la altura de la cámara sobre el mismo plano.

A continuación, pongamos algunas luces, dejemos la esfera sin un material asignado, y movamos el plano a una *capa diferente*. Por ejemplo, digamos que todo está en la capa 1, excepto el plano, que está en la capa 2.

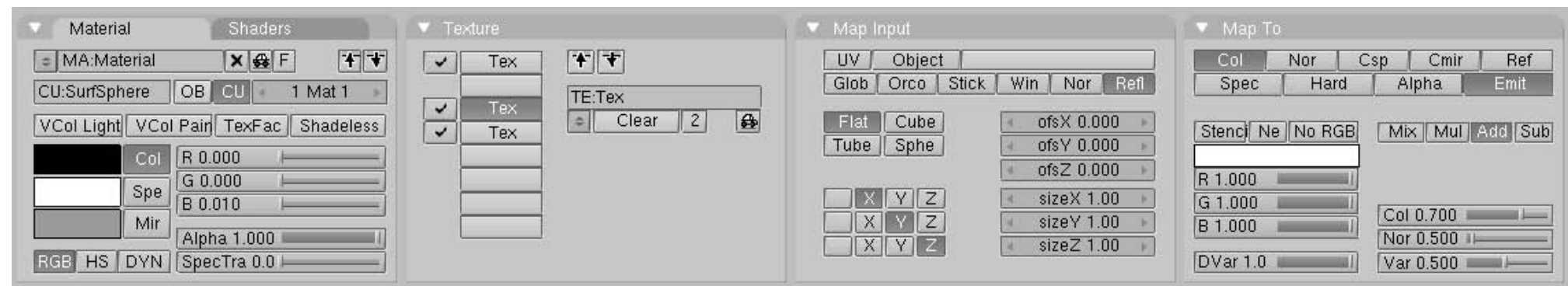
Le asignamos al plano un material bajo en *Ref* y *Spec* y agregamos una textura en el canal dos (2) con los parámetros que se muestran en [Figure 11-22](#).

Figure 11-22. Material del plano Reflejante.



Nótese el mapeado *Ref* y el efecto *Cmir* Utilizamos el canal 2 en lugar del 1, porque necesitaremos el canal 1 más adelante en este ejemplo.

Figure 11-23. Configuración EnvMap del plano reflejante.

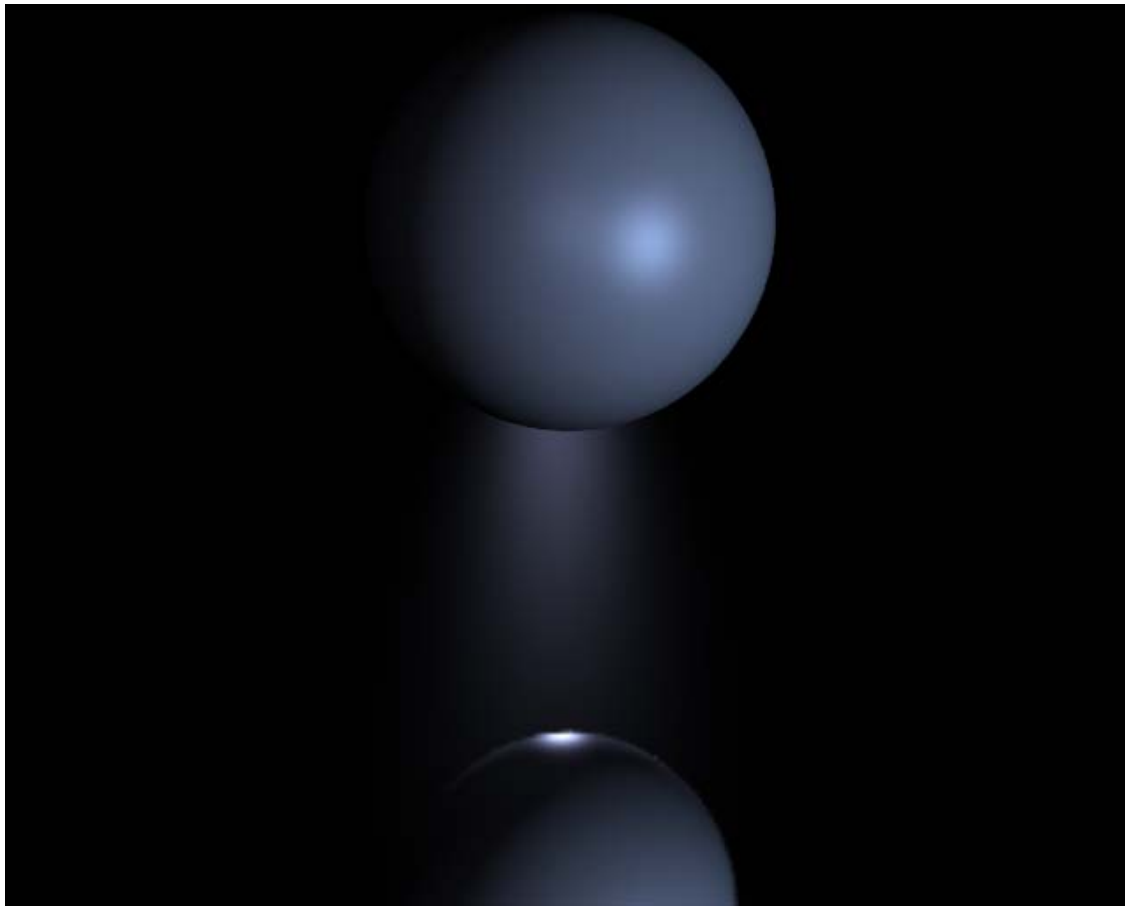


Ahora definimos la nueva textura como *EnvMap*, en los Botones de Textura (**F6**) ([Figure 11-23](#)). En el Panel *Envmap*, observar que el campo *Ob*: contiene el nombre del Empty con respecto al cual vamos a computar el Mapa de Ambiente (EnvMap). Nótese también la resolución del cubo sobre el cual será computado el EnvMap y, lo más importante, los botones *No renderizar capa*: (*Don't render layer:*) .

Debido a que el EnvMap es computado desde la posición del Empty, debe tener una visión de la escena libre de obstáculos. Ya que el plano reflejante escondería completamente a la esfera, debe estar en su propia capa, que debe ser activada como "No renderizable" para el cálculo del EnvMap.

Presionando **F12** comienza el proceso de render. Primero se computan las seis imágenes cuadradas correspondientes al EnvMap, a partir de las cuales se produce la imagen final de la esfera reflejada sobre el plano.

Figure 11-24. Esfera sobre una superficie reflejante.

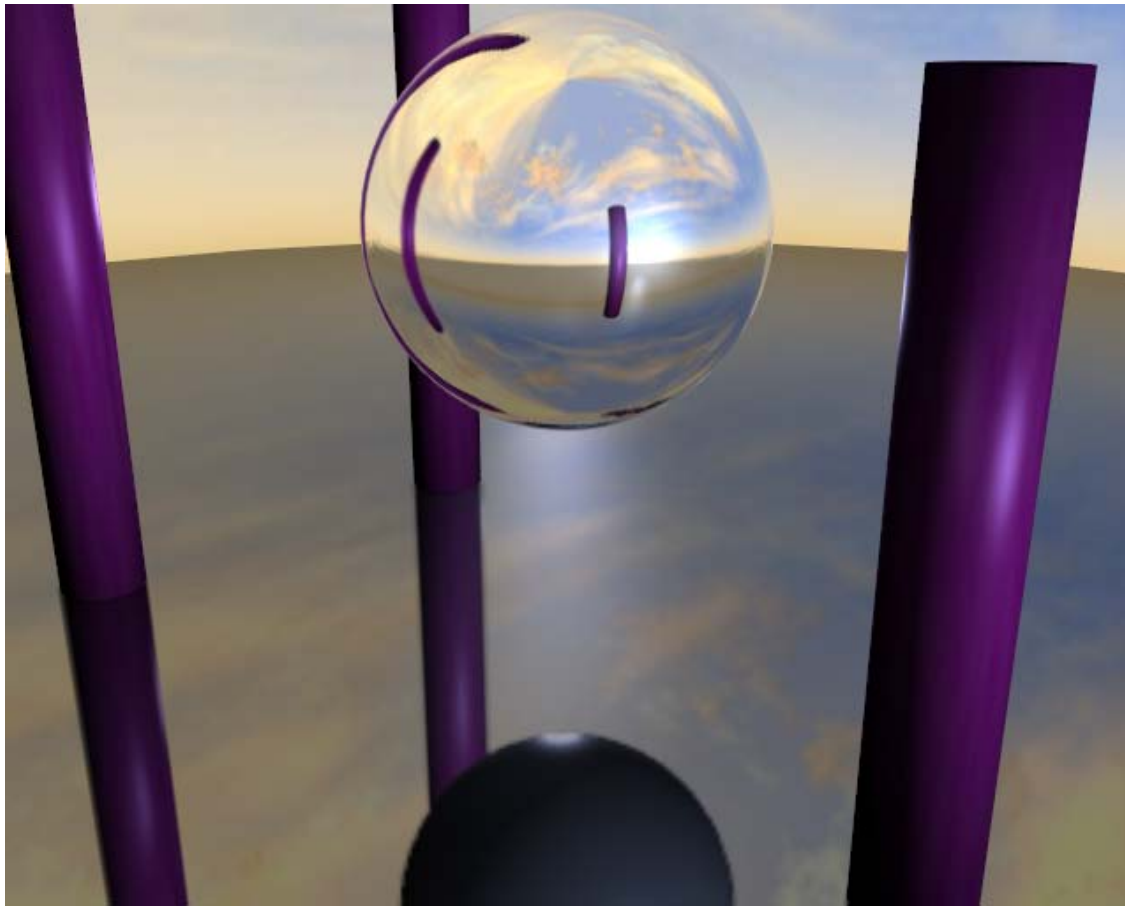


Para mejorar la apariencia de la escena, agregamos una gran esfera conteniendo la escena completa y mapeamos una imagen de cielo sobre ésta para simular un mundo nublado. Luego agregamos un nuevo Empty en el centro de la esfera y movemos la esfera a la capa 3. A continuación le asignamos a la nueva esfera un EnvMap, exactamente como hicimos para el plano (pero esta vez, es la capa 3 la que no debe ser renderizada!).

Ahora agregamos algunos cilindros, para hacer el entorno aún más interesante y, antes de pulsar **F12**, volvemos a la textura del plano y presionamos sobre el botón *Liberar Datos (Free Data)*. Esto obliga a Blender a recalcular el EnvMap para el nuevo entorno.

Esta vez, durante el proceso de render, se computan doce imágenes, seis por cada EnvMap. El resultado se ve en [Figure 11-25](#). La esfera es más brillante que el plano debido a pequeñas diferencias en la configuración de los materiales.

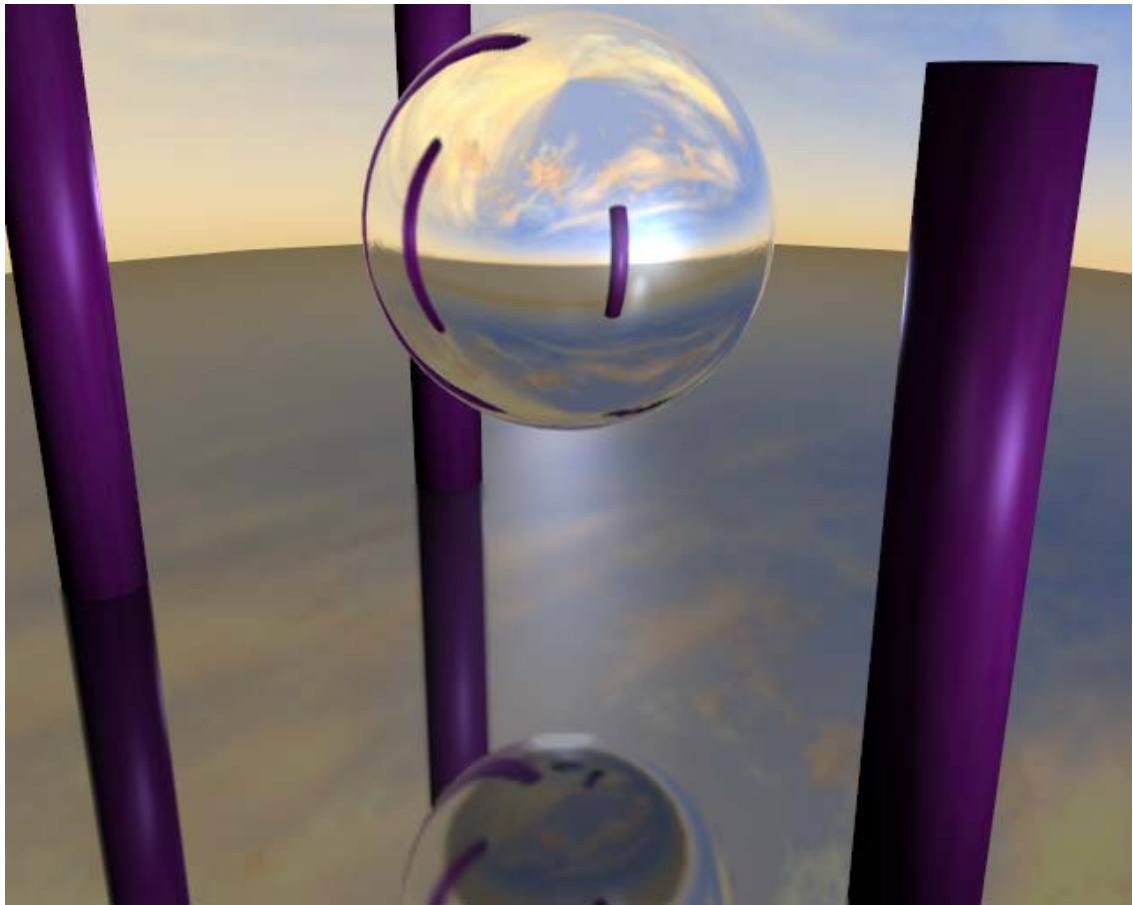
Figure 11-25. Esfera reflejante sobre una superficie reflejante.



Pero, un momento, hay un problema! La Esfera refleja al Plano, pero el Plano refleja una Esfera gris! Esto pasa porque el EnvMap del Plano es calculado *antes* que el EnvMap de la Esfera. Por lo tanto, cuando lo calcula, la esfera aún es gris, mientras que cuando se calcula el EnvMap de la esfera, el plano ya tiene sus Reflejos.

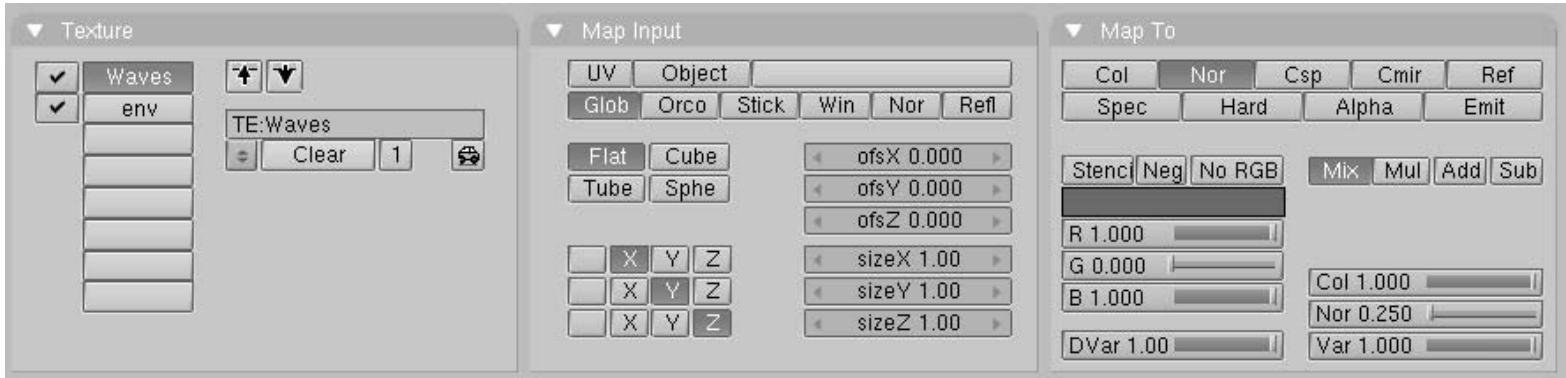
Para corregir esto, localizamos el Botón Numérico *Profundidad (Depth)* en el panel *Envmap*, en los botones de Textura, y le asignamos un valor de 1 tanto para la textura EnvMap del plano como de la esfera. Esto fuerza el cálculo recursivo de EnvMaps. Cada EnvMap es computado y luego es recalculado tantas veces como diga "Profundidad (Depth)", siempre uno luego del otro. El resultado se ve en [Figure 11-26](#).

Figure 11-26. Esfera reflejante sobre superficie reflejante, con múltiples reflejos.



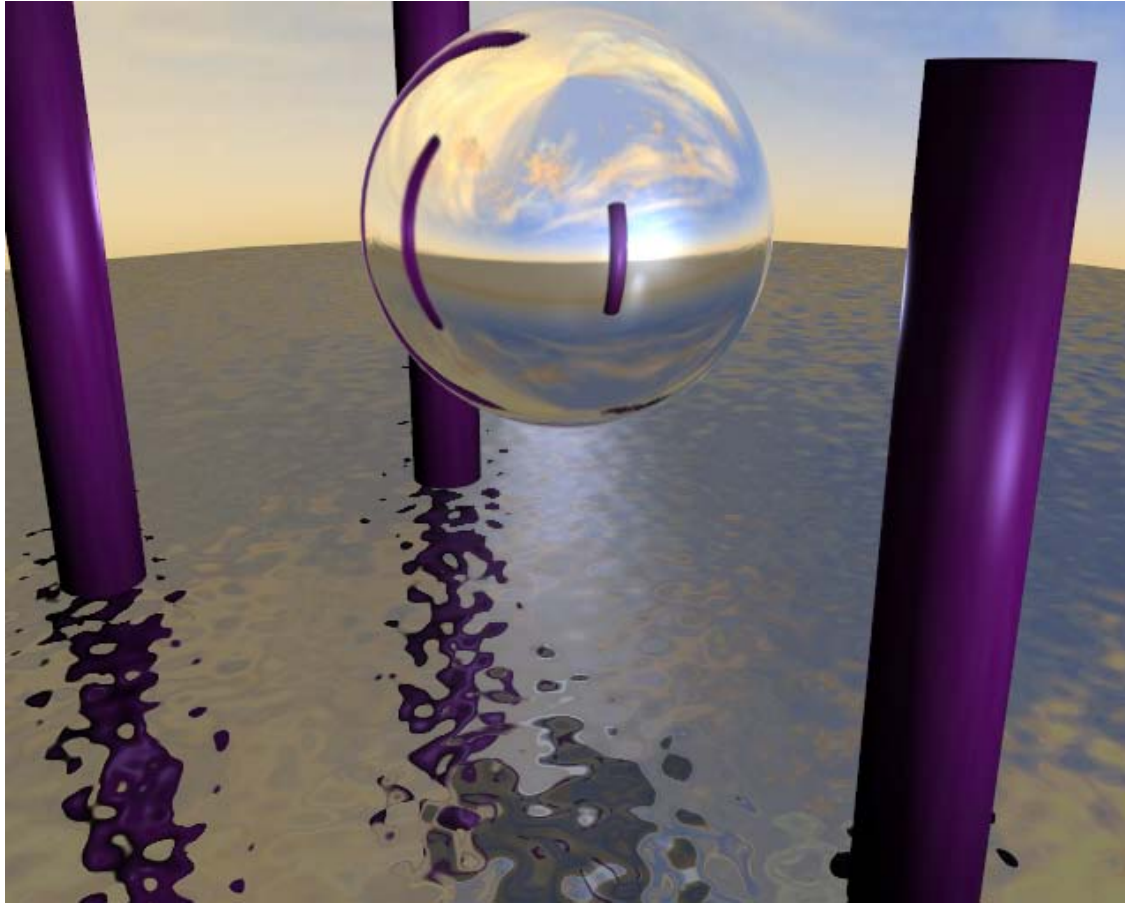
Ahora, si todavía quedan dudas acerca de por qué quedó en blanco el primer canal de textura del material del Plano... Agregamos una nueva textura al primer canal del material del plano. Hacemos que sea *Glob*, afectando *Nor*, con una intensidad de 0,25 ([Figure 11-27](#)).

Figure 11-27. Configuración adicional de la textura para Mapeado de Relieves (BumpMapping).



Esta nueva textura debe ser de tipo *Stucci*; ajustamos el *Tamaño del Ruido* (*Noise Size*) a 0,15 aproximadamente. Si renderizamos la escena, ahora el plano luce como agua con ondas ([Figure 11-28](#)).

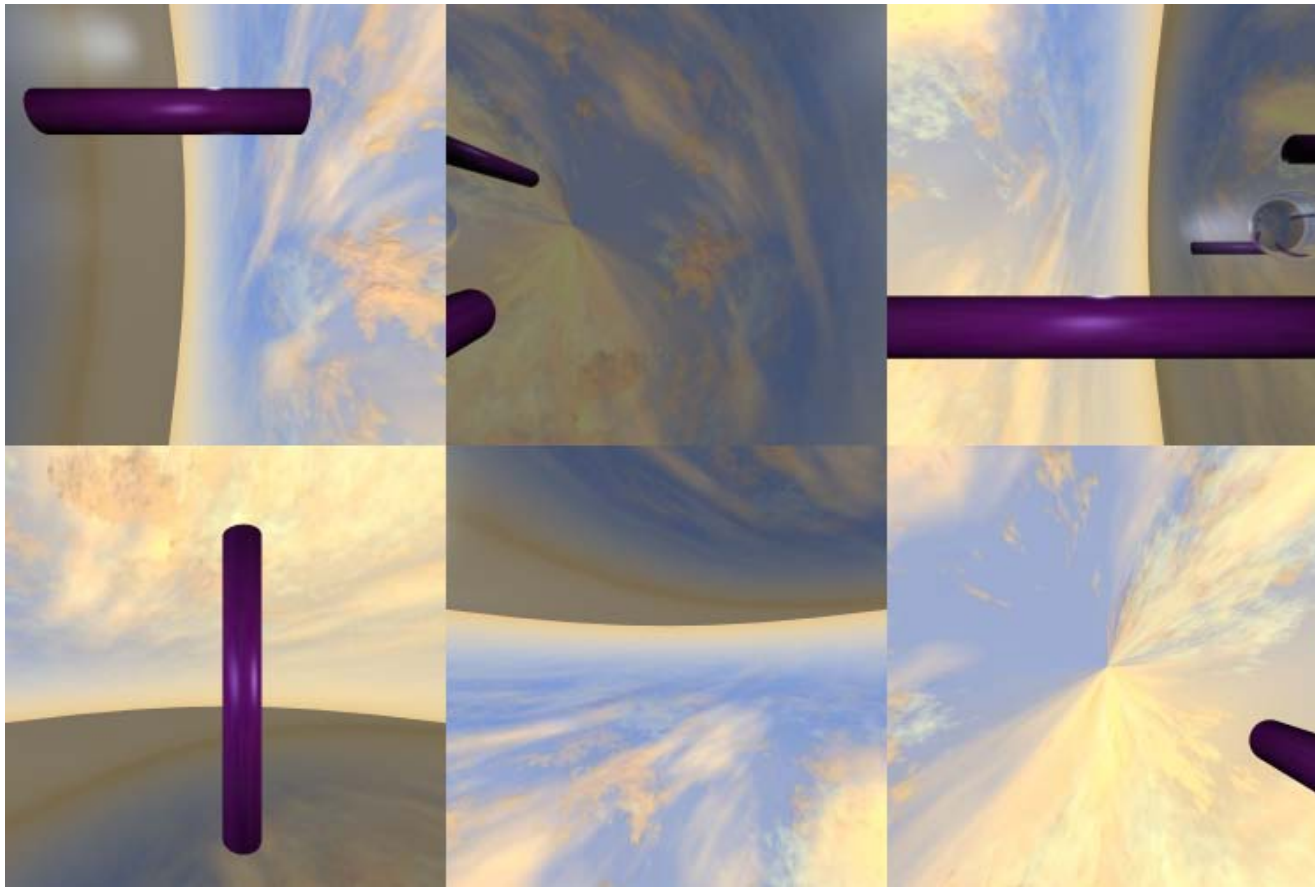
Figure 11-28. Esfera reflejante sobre agua reflejante, con múltiples reflejos.



Se debe tener el Mapa de Relieve (BumpMap) en un canal que preceda al EnvMap porque las texturas se aplican secuencialmente. Si se hiciera al revés, los reflejos parecerían estar rotos por las ondas.

Se pueden salvar los EnvMaps para ser usados mas tarde y cargarlos con los botones correspondientes en los Botones de Textura. Tambien se pueden construir mapas de ambiente. El estándar es ubicar las seis imágenes mapeadas sobre el cubo en dos filas de tres imágenes cada una, como se muestra en [Figure 11-29](#).

Figure 11-29. El EnvMap como queda al ser salvado.



Blender permite tres tipos de mapas de ambiente, como se puede ver en [Figure 11-23](#):

- *Estático (Static)* - El mapa sólo se calcula una sola vez durante la animación, o después de cargar un archivo.
- *Anim* - El mapa es calculado cada vez que se hace un render. Esto significa que los objetos en movimiento se verán correctamente sobre las superficies reflejantes.
- *Cargar (Load)* - Cuando se salva como un archivo de imagen, los mapas de ambiente pueden ser cargados desde el disco. Esta opción permite renders con mapas de ambiente mucho más rápidos.

Note Se puede animar el agua del ejemplo previo configurando una curva IPO para los valores de posición de textura *ofsX* y *ofsY* en Botones de Material. Al renderizar la animación se verán ondas en movimiento sobre la superficie, con reflejos que cambian de manera acorde!

Note El EnvMap del plano necesita ser calculado solamente una vez al principio siempre y cuando nada más se mueva! Por lo tanto, puede ser *Estático (Static)*. El EnvMap de la esfera es otro tema, ya que no reflejará los cambios en los reflejos del agua, a menos que se lo compute en cada cuadro de la animación. Por consiguiente, debe ser de tipo *Anim*.

Si la cámara es el único objeto que se mueve, y existe un plano reflejante, el Empty también debe moverse y deben usarse EnvMaps de tipo *Anim*. Si el objeto es pequeño, y el Empty está en su centro, el EnvMap puede ser *Estático (Static)*, aún si el objeto mismo rota, ya que el Empty no lo hace. Si, por otro lado, el Objeto se traslada, el Empty debe seguirlo y el EnvMap debe ser de tipo *Anim*.

Otras configuraciones son:

- *Filtro (Filter)*: - Con este valor se puede ajustar la dureza o suavidad de los reflejos.
- *Clipsta*, *ClipEnd* - Estos valores definen los límites de truncamiento (clipping) cuando se renderizan imágenes con mapas de ambiente.

Note

El cálculo de EnvMap puede ser desactivado globalmente mediante el botón *EnvMap* en el Panel de *Render*, dentro de los Botones de Render.

[Prev](#)

Texturas

[Home](#)

[Up](#)

[Next](#)

Mapas de Desplazamiento

11.3. Mapas de Desplazamiento

por Kenneth Styrberg

Relativo a Blender v2.35

El mapeado de desplazamiento es una técnica poderosa que permite que una textura , sea procedural o una imagen, manipule la posición de las caras renderizadas . El desplazamiento se controla como un mapa de *NOR*, a una textura más brillante le corresponderá un desplazamiento mayor. A diferencia que el mapeado de *Normal* o "*Bump*", donde las normales son distorsionadas para dar la ilusión óptica de un relieve, esto crea un relieve real. Este relieve puede proyectar sombras, ocluir otros objetos y hacer todo lo que uan geometría real puede hacer.

El mapeado de Desplazamiento es configurado para comportarse como un canal de textura, con una diferencia muy importante. Con el objeto de manipular las posiciones de las caras renderizadas en forma suave, las caras deberán ser muy pequeñas y esto consume mucha memoria y tiempo de CPU.

Para objetos distantes o en segundo plano, se debería seguir usando un mapeado *NOR* . Los mapas *NOR*, comparados con los mapas de *Desplazamiento*, agregan muy poco costo computacional adicional por cara renderizada, y se puede mapear *NOR* independientemente de la cuenta de caras renderizables. La utilización de mapas de *Desplazamiento* lleva rápidamente a tener millones de caras en una escena.

En la [Figure 11-30](#) vemos la diferencia de usar mapas *NOR* en contraposición con los *Mapas de Desplazamiento*. Al momento del renderizado, el objeto de la izquierda tiene 240 caras, mientras que el de la derecha tiene ¡ 60717 caras !

Figure 11-30. Diferencia entre Mapas de NOR y de Desplazamiento.



Note Use los *Mapas de Desplazamiento* cuando necesite que su geometría sea más precisa.

11.3.1. Mapas de Desplazamiento en Objetos

En esta lista se muestran, de mejor a peor, cómo funcionan los distintos tipos de objeto con los *Mapas de Desplazamiento*.

- **Mallas con Subsurf** (*Catmull-Clark*) el tamaño se controla con el nivel de subsurf de renderizado. ¡ El desplazamiento funcionará de maravilla !
- **Mallas con Subsurf Simple** Las caras renderizables se controlan con el nivel de subsurf. El desplazamiento funcionará, pero habrá problemas en las aristas filosas si la textura no tiene un gris neutro.
- **Mallas subdivididas manualmente** Las caras renderizables se controlan con el número de subdivisiones. Puede ralentizar la edición porque no puede desactivar el nivel de subdivisión para editar mallas densas.
- **Metaballs** Se controla con el tamaño de alambre para render. Un tamaño de alambre pequeño da más caras.

El uso de *Mapas de Desplazamiento* sobre los siguientes tipos de objeto es posible, pero pueden dar errores en las normales y juntas visibles al renderizar.

La cantidad de caras está vinculada directamente con la resolución U/V de las superficies. Resoluciones más altas dan más caras renderizables.

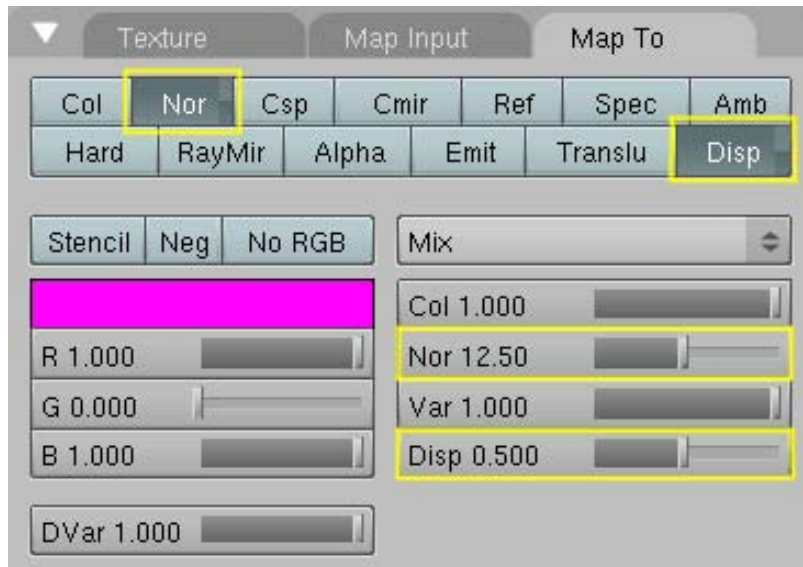
- **Superficies Nurbs abiertas**
- **Superficies Nurbs cerradas**
- **Curvas y Texto**

Note Es recomendable que convierta los tipos de objeto Curvas y Superficie a Mallas antes de aplicar desplazamiento.

11.3.2. Interfaz

La interfaz de los *Mapas de Desplazamiento* consiste en dos botones y dos deslizadores de valor en el Panel *Map To* del contexto de *Materiales*. ([Figure 11-31](#)).

Figure 11-31. Map To panel.



La intensidad del desplazamiento se controla con el deslizador *Disp* y el desplazamiento normal se controla con el deslizador *Nor*.

11.3.3. Utilización de un Mapa de desplazamiento

El desplazamiento trabaja de dos maneras :

- Desplazando vertices renderizados por intensidad, los vertices se mueven en la direccion de las normales de vertices.
- Desplazando vertices renderizados según la normal de textura, los vertices se mueven según la entrada de *NOR* de la textura.

Estos dos modos no son mutuamente excluyentes. La cantidad de cada tipo puede mezclarse utilizando los deslizadores en el panel *Map To* del contexto de *Materials* (F5), [Figure 11-31](#).

Sin embargo, no todas las texturas ofrecen ambos tipos de entrada. La *Stucci*, por ejemplo, solo ofrece *Normal*, en cambio que *Magic* solo ofrece *Intensidad*.

Las de tipo *Cloud*, *Wood* y *Marble* ofrecen ambas, *Normal* e *Intensidad*. *Image* ofrece tanto *Intensidad* y una *Normal derivada*.

Note El OSA de textura, no funciona correctamente aún en el caso de imágenes mapeadas a desplazamiento.

El desplazamiento por *Intensidad* da una superficie más continua y suave, debido a que los vértices solo son desplazados hacia afuera. El desplazamiento *Normal*, da una superficie más aglomerada debido a que los vértices son desplazados en muchas direcciones.

Aquí se muestra el ejemplo del objeto en la [Figure 11-30](#), pero con un valor para el deslizador *Nor* de alrededor de **2.0**. Observe que el botón *Nor* sigue desactivado! Puede ver claramente el desplazamiento más aglomerado cuando se utiliza un valor para *Nor* junto con *Disp*.

Figure 11-32. Valores para *Disp* y *Nor*.



La profundidad del desplazamiento es escalada con la escala del objeto, pero no con el tamaño relativo de los datos. Esto significa que si se duplica el tamaño de un objeto en *Modo Objeto*, la profundidad del desplazamiento también es duplicada, por lo que el desplazamiento relativo parece el mismo.

Si por el contrario, escala en *Modo Edición*, la profundidad del desplazamiento no se modifica, y por ende la profundidad relativa aparecerá más pequeña o más grande.

La intensidad de la textura define el desplazamiento. Un gris neutro, RGB = 128,128,128, representa un desplazamiento nulo. Para obtener un desplazamiento positivo, [Figure 11-33](#), el blanco es un pico, y el negro es una depresión.

Figure 11-33. Desplazamiento Positivo seleccionado.

Para el desplazamiento negativo, [Figure 11-34](#), es al revés.

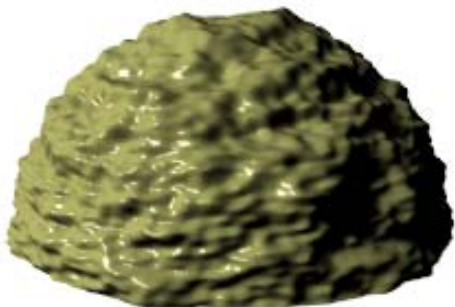
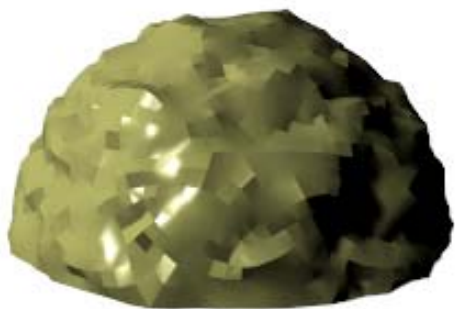
Figure 11-34. Desplazamiento Negativo seleccionado.

11.3.4. Ejemplo

En este ejemplo se muestra el efecto del nivel de subdivisión sobre el resultado final del mapa de desplazamiento. Es un tipo de subdivisión *Catmull-Clark* y la textura es una simple *Cloud* añadida a una *Icosphere*. Vea la [Section 7.1](#) para más información sobre superficies de subdivisión.

Los niveles de subdivision van de **0** - (ninguna) a **6** - (máxima subdivisión).

Figure 11-35. Niveles de Subsurf 0, 1, 2, (arriba) 3, 4, 5 (medio) y 6 (abajo).



[Prev](#)

Mapas de Ambiente (Environment Maps)

[Home](#)

[Up](#)

[Next](#)

Cristal Solido y Hueco

11.4. Cristal Solido y Hueco

Relativo a Blender v2.31

Los materiales tipo cristal y transparentes son generalmente mas dificiles de renderizar porque exhiben *refracción*; esto es, la flexión de los rayos de luz suaves debido a la diferente *densidad optica*, o al *indice de refracción* de los diversos materiales. Desafortunadamente, explicar completamente la refracción en un generador de rayos (raytracer) es obligatorio. Aún, podemos producir resultados convincentes en Blender usando EnvMaps y avanzadas tecnicas de texturizado.

Considerar una escena con algunas geometrias basicas, incluyendo un cubo, un cono, una esfera y un toroide. Como primer ejemplo haremos una esfera como una bola solida de cristal y, como segundo ejemplo, la misma esfera será como una burbuja de cristal.

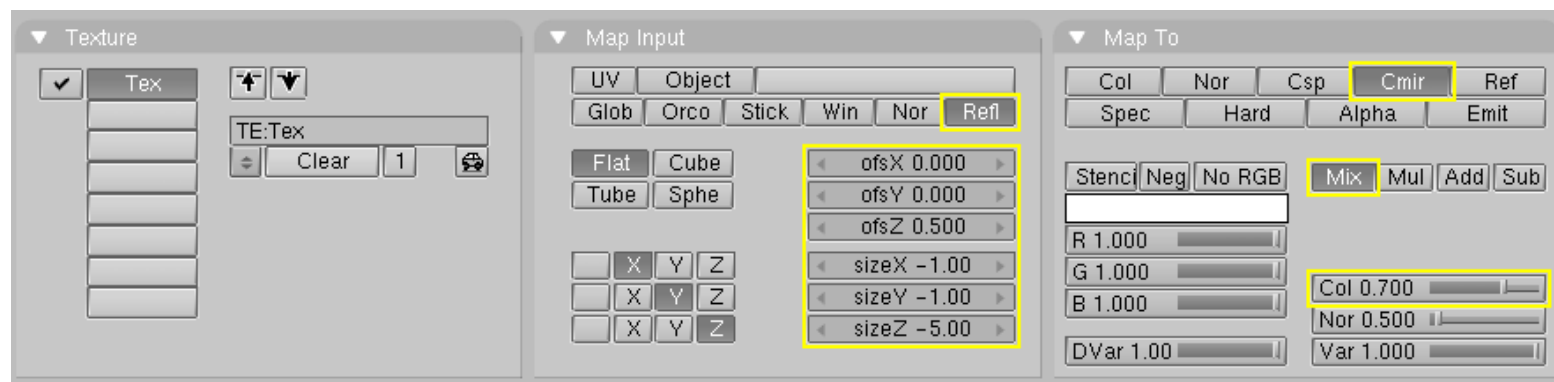
Para crear este efecto, necesitamos hacer que parezca que la luz se doble cuando pasa a traves de la esfera, puesto que esperaríamos que los objetos de detrás de la esfera de cristl solida aparecieran pesadamente combados, como si se viera a través de una lente gruesa. Por otra parte, el centro de la esfera de cristal hueca debe ser casi transparente mientras que los lados deben desviar la luz.

11.4.1. Cristal Solido

1. Para comenzar, instalamos un mapa del ambiente para el material de la esfera justo como hicimos para la bola en la sección anterior, con un vacío que localiza la perspectiva del EnvMap en el centro de la esfera.

2. Para falsificar la Refracción moveremos la salida del mapeado con los desplazadores *ofsZ*, *sizeX*, *sizeY*, *sizeZ* y *Col* para combar el mapa de una manera que cree el efecto de la refracción. Para hacerlo, usa los ajustes como estos [Figure 11-36](#).

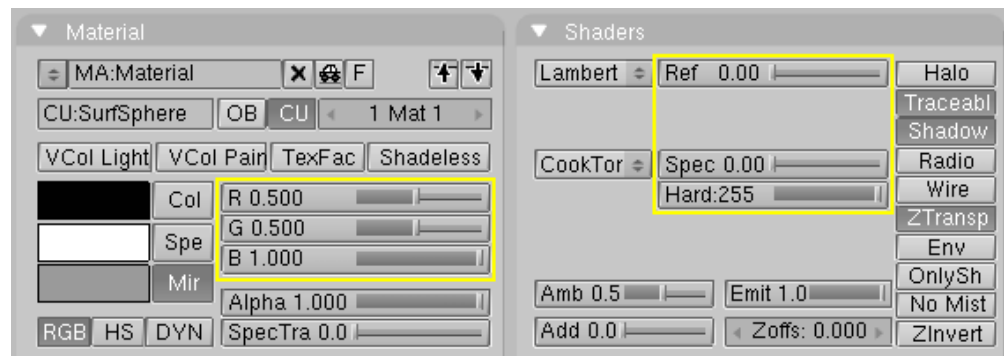
Figure 11-36. .



3. Selecciona los desplazadores de material RGB de *Mir* y bajaremos *R* y *G* un poco para dar a la textura un tintado azul. (Nuestra experiencia con las idiosincrasias de Blender la dirección de los colores reflejados dicta este desintuitivo acercamiento cuando se combinan reflexiones y refracciones en el mapa-envolvente como un solo material.)

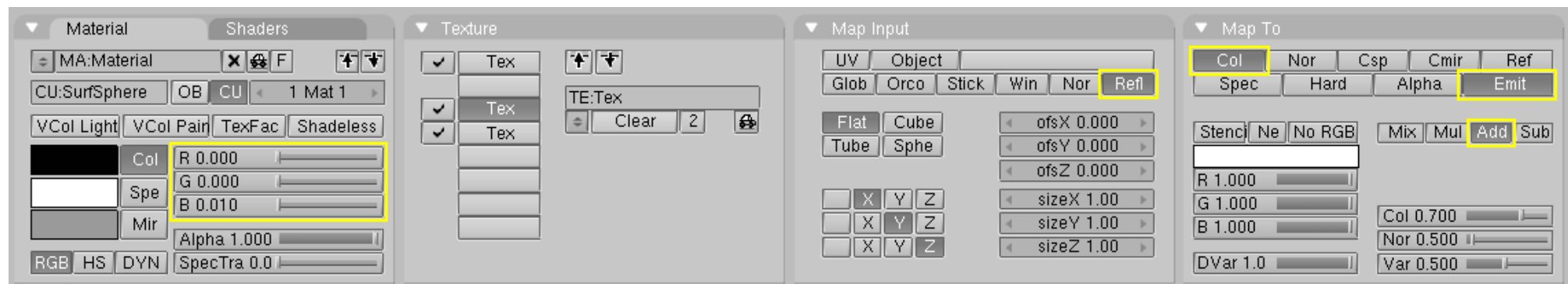
4. Mueve el desplazador de *Ref* hacia abajo. ([Figure 11-37](#)). Tendras que haber producido una refracción tintada de azul en el ambiente.

Figure 11-37. Ajustes de los materiales



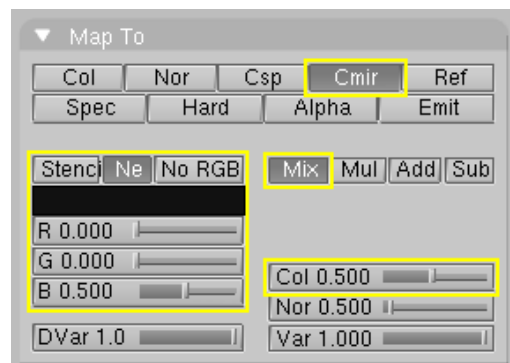
5. El cristal brillante también necesita un mapa de refracción, así que pondremos la textura *same* dentro de otro canal de textura. Pulsa los botones *Add*, *Col*, y *Emit*, y usa el botón *Ref* para las coordenadas. Coje el material negro *Color* y gira *Emit* todo hacia arriba. (Figure 11-38).

Figure 11-38. Mapa de Reflexión



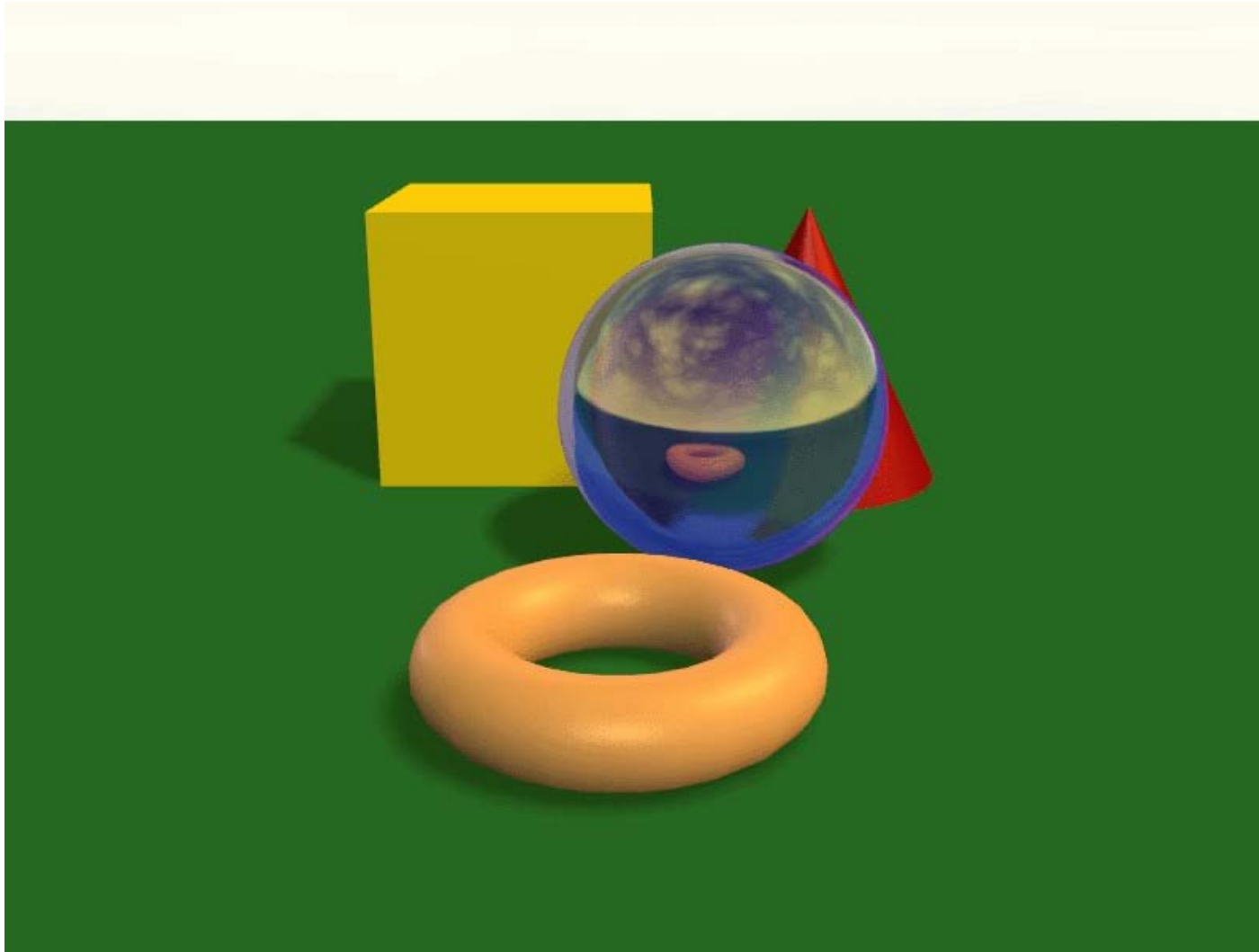
6. Esto cambia nuestra primera textura considerablemente. Para volver la textura de la refracción a un tinte azul agradable, tenemos que añadir una nueva textura, saliendo del un tipo de textura fijado a *None*. Selecciona los botones *Mix* y *Cmir*, y fija el desplazador de *Col* de medio hacia arriba. Clicka el botón *Negy* fija los valores de la entrada de la textura RGB con los desplazadores a un azul marino (Figure 11-39).

Figure 11-39. Toques Finales



El resultado final se debe parecer a [Figure 11-40](#). El efecto de refracción es mas notable cuando se anima la escena.

Figure 11-40. Renderizando



11.4.2. Cristal Hueco

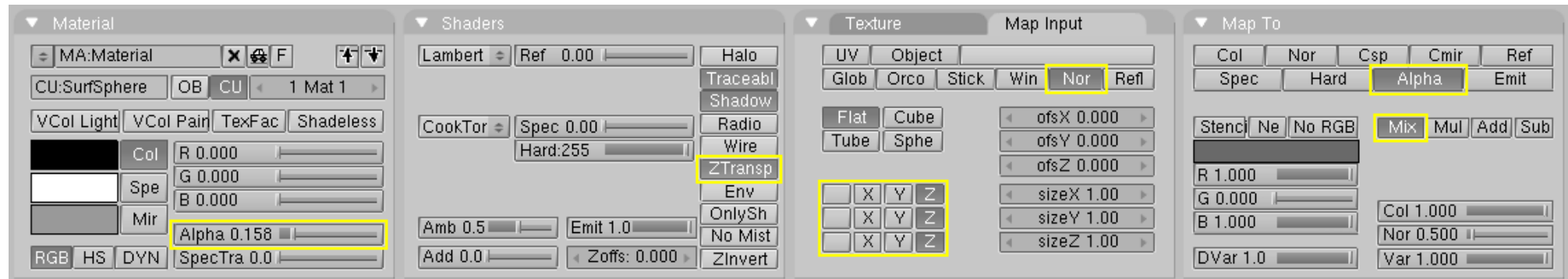
El procedimiento que hemos descrito va muy bien sobre trabajos con un terrón sólido de cristal, pero ¿cómo producimos una apariencia de cristal hueco, como un florero?

El cristal fino tiene refracción fuerte solo donde está más inclinado lejos del ojo. Podemos fácilmente representar este efecto usando en texturas de objetos *Blend* en Blender para controlar la transparency, tan bien como otra textura de la transparencia para mantener los toques de luz brillantes visibles.

1. Añade una nueva textura al material. Selecciona *Blend* como el tipo y selecciona la opción de *Sphere*.

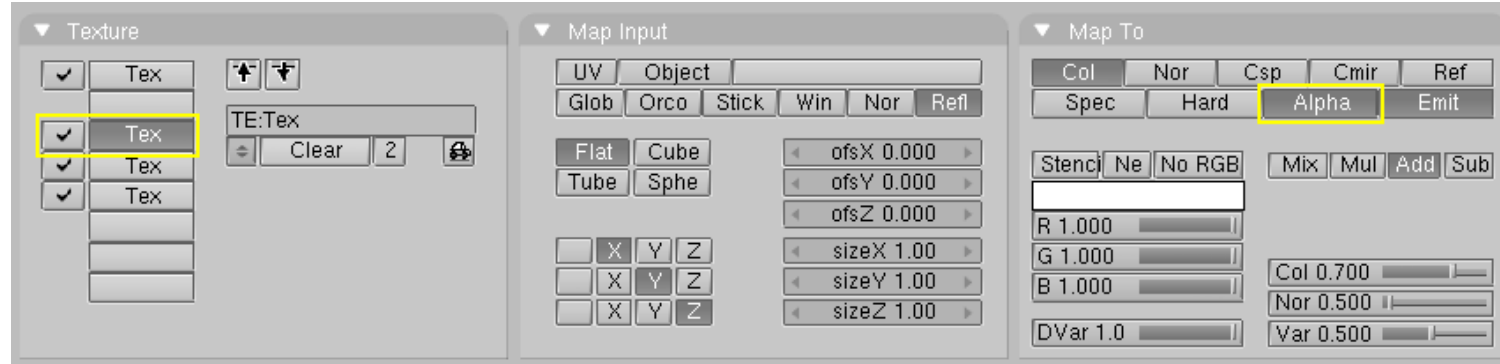
2. Vuelve a los botones de material. Selecciona *Nor* como tipo de mapa, e inhabilita las X e Y en el eje de coordenadas.
3. *Mix* la textura con *Alpha*, cuando mueves el desplazador de material *Alpha* a 0.0 y seleccionas la opción *ZTransp* ([Figure 11-41](#)).

Figure 11-41. Fijar la transparencia.



Esto produce el efecto de la transparencia agradable como los ángulos superficiales hacia el ojo, pero queremos que las reflexiones del mapa-envolvente superiores en estas áreas sean sem-transparentes. Por ejemplo, si miras una ventana de cristal, veras que la luz brillante que refleja la superficie es visible, evitando que veas a través de un cristal que de otra manera sería transparente. Podemos producir este efecto facilmete seleccionando el mapa de ambiente la textura de reflexión en la ventana de material y permitiendo la opción *Alpha* ([Figure 11-42](#)).

Figure 11-42. Fijar Reflexiones



Esto es todo. El resultado debe quedar como [Figure 11-43](#).

Figure 11-43. Cristal Hueco



[Prev](#)

Mapas de Desplazamiento

[Home](#)

[Up](#)

[Next](#)

EditorUV y Selección de Caras

11.5. EditorUV y Selección de Caras

Relativo a Blender 2.33

11.5.1. Introducción

El mapeado UV es una manera de mapear texturas de tipo Imagen sobre modelos tridimensionales. Se puede usar para aplicar texturas a formas arbitrarias y complejas como cabezas humanas o animales. A menudo estas texturas son imágenes pintadas o dibujadas, creadas con programas como Gimp, Photoshop, o cualquiera que sea tu programa de dibujo favorito.

Las texturas Procedurales, como hemos visto en los capítulos anteriores, constituyen una buena manera de texturizar un modelo. Lo que es realmente bueno de este tipo de texturas es que siempre "calzan", se ajustan perfectamente al modelo. Las imágenes 2D en cambio, no siempre se ajustarán tan bien. Crear texturas procedurales es relativamente sencillo, y ofrecen una manera rápida de obtener buenos resultados.

Sin embargo, hay situaciones en las que este tipo de texturas no es suficiente. Por ejemplo, la piel de una cabeza humana nunca se verá lo suficientemente bien cuando es generada proceduralmente. Las arrugas en un rostro humano, o las saltaduras de pintura en un automóvil no aparecen en lugares al azar sino que dependen de la forma del modelo y de su uso. Las imágenes pintadas manualmente le dan al artista control total sobre el resultado final. En lugar de andar jugando con deslizadores numéricos, los artistas pueden controlar cada pixel sobre la superficie. Esto comúnmente significa más trabajo, pero los resultados valdrán la pena.

Un mapa UV es una forma de asignar una parte de una imagen a un polígono en el modelo. Cada vertice del polígono es asignado a un par de coordenadas 2D que definen que parte de la imagen es mapeada. Estas coordenadas 2D se llaman UVs (comparar con las coordenadas XYZ en 3D). La operación de crear estos mapas UV se conoce también como "despliegue" ("unwrap" en inglés), debido a que todo ocurre como si la malla fuera desenvuelta o desplegada sobre un plano 2D.



El mapeado UV es también fundamental en el motor de juego de Blender como en cualquier otro juego tridimensional existente. Es de hecho el estándar para la aplicación de texturas sobre modelos; casi todo modelo que encuentre en un juego se hallará mapeado UV.

11.5.2. El Editor UV

El mapeado UV se hace en Blender a través de la ventana del Editor UV y de un modo especial de la Ventana3D llamado el Modo Selección de Caras (Face Select). El Editor UV nos permite mapear las texturas directamente sobre las caras de las mallas.

Cada cara puede tener coordenadas de textura individuales y una imagen distinta asignada a la mismas, puede combinarse con colores de vertices para hacerla más brillante o más oscura o incluso darle color.

Por medio del editor UV, a cada cara se le asignan dos propiedades adicionales:

- *cuatro coordenadas UV* Estas coordenadas definen la forma en que se mapean las imágenes sobre la cara. Son coordenadas bidimensionales, por lo que se las llama UV, para distinguirlas de las coordenadas XYZ. Estas coordenadas pueden utilizarse para renderizado o para uso en la pantalla OpenGL en tiempo real.
- *un enlace a una Imagen* Cada cara en Blender puede tener un enlace a una imagen distinta. Las coordenadas UV definen como se mapea esta imagen sobre la cara. La imagen puede renderizarse o ser mostrada en tiempo real.

Para poder asignar Imágenes o modificar las coordenadas UV del Objeto Malla activo, la Ventana3D debe estar en el Modo "Selección de Caras".

Figure 11-44. Entrando al Modo Seleccion de Caras.



Agregue a la escena un Objeto Malla, luego ingrese en el Modo Selección de Caras eligiendo la opción "UV Face Select" del menú Modo.

La malla será visualizada en modo Z-Buffer. Si entra en el modo de visualización Texturizado (**ALT-Z**, también llamado "modo papa") verá la malla en color blanco, lo que indica que actualmente no hay ninguna imagen asignada a sus caras. Puede controlar la manera en que se representan las caras usando los botones Draw Edges (Dibujar Aristas) y Draw Faces (Dibujar Caras) en el Panel *UV Calculation*. Si se activa Draw Edges, todas las caras se mostrarán con un delineado. Con Draw Faces activado, todas las caras seleccionadas se visualizarán en un tono rosado claro (o del color del Tema).

Figure 11-45. Modo Selección de Caras.

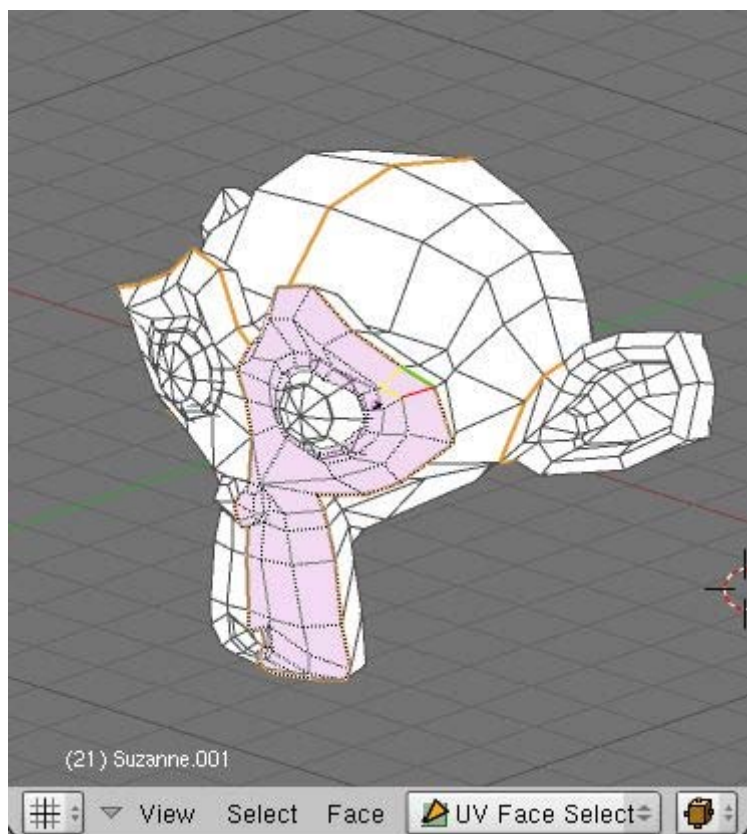
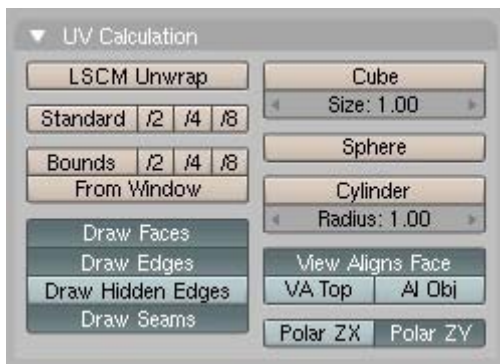


Figure 11-46. Panel Cálculo de UV.

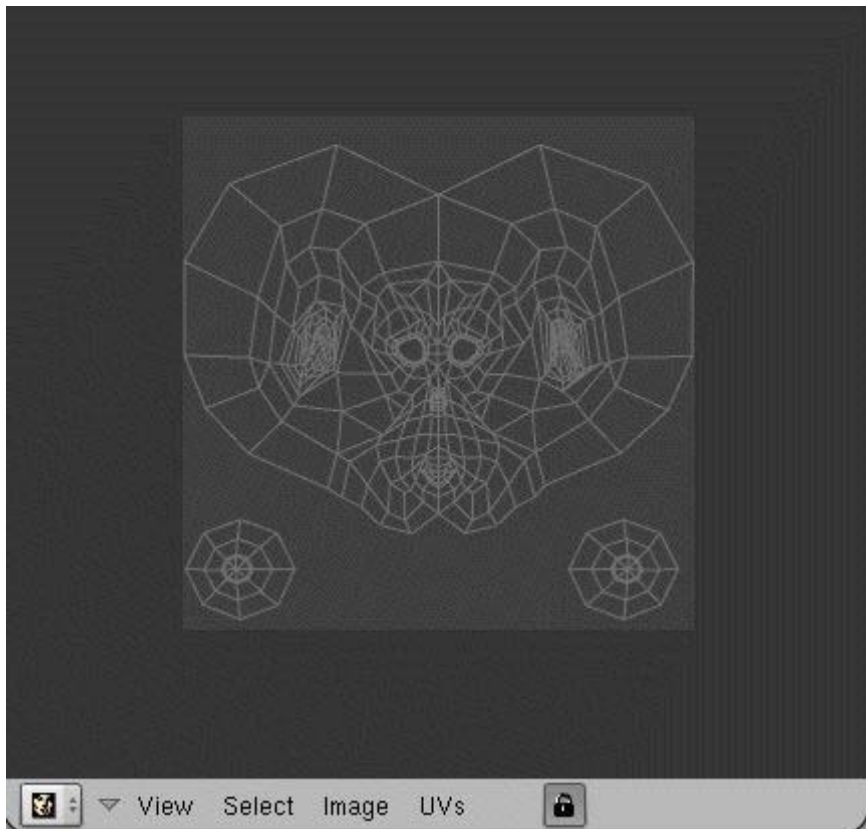


Presione **AKEY** y todas las caras de la Malla serán seleccionadas y resaltadas con líneas punteadas. En la Ventana3D puede seleccionar caras con **RMB**, o la Selección por Area (**BKEY**). Si tiene dificultad para seleccionar las caras deseadas, puede ingresar en Modo Edición y seleccionar los vertices que quiera. Luego, al salir de ModoEdición, las caras definidas por los vertices seleccionados deberían también permanecer seleccionadas.

Solamente una cara es la activa. O en otras palabras: La Ventana de Imagen, solamente mostrará la imagen de la cara activa. Como se acostumbra en Blender, solamente la ultima cara seleccionada es la activa y la selección se realiza con **RMB**.

Cambie una ventana a la Ventana de Imagen/Editor UV con **SHIFT-F10**. Aquí puede cargar o buscar una imagen con el botón *Load*. Las imagenes deben tener una resolucion multiplo de 64 pixels (64x64, 128x64 etc.) para poder ser representadas en tiempo real (nota: la mayoría de las tarjetas 3D no soportan imagenes más grandes que 256x256 pixeles). Sin embargo, Blender puede renderizar todas las imagenes asignadas sin importar su tamaño cuando se crean imagenes estáticas o animaciones.

Figure 11-47. El Editor UV.



Cargar o buscar una imagen en Selección de Caras, asigna automáticamente la imagen a las caras seleccionadas. Esto puede observar inmediatamente en la Ventanas 3D (cuando se encuentra en Modo Vista Texturizada).

11.5.3. Herramientas de Desplegado (Unwrapping)

En la Ventana3D estando en Modo Selección de Caras, puede presionar **UKEY** para hacer aparecer un menú para calcular las coordenadas UV de las caras seleccionadas. También puede realizar este desplegado utilizando el Panel de Cálculo UV en los Botones de Edición. Dicho panel nos ofrece un mayor control sobre el proceso de despliegue.

Figure 11-48. UV pre-sets.



Los algoritmos de Despliegue UV disponibles son:

- *Cube (Cúbico)* Determina mapeado cúbico.
- *Cylinder (Cilindrico), Sphere (Esférico)* Mapeado cilíndrico/esférico, calculado desde el centro de las caras seleccionadas.
- *Bounds to 1/8, 1/4, 1/2, 1/1* Las coordenadas UV se calculan utilizando la proyección de lo mostrado en la Ventana3D, y luego escalado a la fracción de la textura imagen indicada.
- *Standard 1/8, 1/4, 1/2, 1/1* Cada cara obtiene un conjunto de coordenadas UV cuadradas, que luego son escaladas la fracción requerida de la textura imagen..
- *From Window (Desde Ventana)* Las coordenadas UV se calculan utilizando la proyección de lo que se muestra en la Ventana3D.
- *LSCM* Las coordenadas se calculan empleando el algoritmo Mapa Conforme a Cuadrados Mínimos (Least Squares Conforming Maps). Uselo conjuntamente con las costuras marcadas.

En el panel de mapeado UV, puede ajustar la forma en que se realiza el mapeado, y cómo se lo muestra en la Ventana 3D cuando el modelo esté en Modo Selección de Caras.

Con *View Aligns Face* habilitado, el despliegue Cilíndrico y Esférico es afectado por la vista actual. La vista es considerada como el frente del Cilindro/Esfera, con las tapas arriba y hacia el fondo de la Vista. El cilindro/esfera es *cortado* del lado opuesto de la vista.

Size y *Radius* definen el escalado del mapa cuando se usan los mapeados Cúbicos y Esférico/Cilindrico respectivamente.

Con el botón *VA Top* (View Aligns Top) habilitado, la Vista debe verse como a través del Cilindro/Esfera. Es cortada en parte superior de la vista. Con esto activado también puede definir como se rotará la vista respecto de los polos utilizando las opciones *Polar ZX* y *Polar ZY*.

Si se habilita *Al Obj*, el Cilindro/Esfera se gira basado en el giro del Objeto.

Los botones *Draw Edges* y *Draw Faces* activan la visualización de aristas u caras en la Ventana3D cuando se está en Modo Selección de Caras. Las caras seleccionadas en este modo se ven en color púrpura transparente (o el color indicado en el tema), similarmente a ModoEdición. La visualización de costuras en ModoEdición se puede encender y apagar con el botón *Draw Seams*. Los colores de las costuras también pueden modificarse en las opciones del Tema.

11.5.4. Editando las coordenadas UV

Dentro del Editor UV verá una representación de las caras seleccionadas en la forma de vertices amarillos o rosados, conectados por líneas punteadas. Pueden usarse aquí las mismas técnicas que en ModoEdición de Mallas para seleccionar, mover, rotar, escalar y así sucesivamente. Si pulsa el botón *Lock* puede obtener una vista en 3D en tiempo real de los cambios producidos. Si lo necesita, puede realizar El escalado y traslación de los vertices en los ejes locales X e Y del mapa. Pulse **XKEY** o **YKEY** una vez iniciado el comando escalar (**SKEY**). También se encuentra disponible la Herramienta de Edición Proporcional (PET) que funciona exactamente igual que lo visto para ModoEdición de Mallas. Al igual que en ese modo, los vertices pueden ocultarse o mostrarse utilizando **HKEY** y **ALT-H** respectivamente.

Figure 11-49. Menú de transformaciones UV.

Grab/Move	G
Rotate	R
Scale	S
Weld/Align	W
Mirror...	M

Dentro del Editor UV se encuentran disponibles varios modos de selección. Como cada vertice de la malla es dibujado en el Editor por cada cara a la que pertenece, a veces es difícil distinguir si hemos seleccionado el mismo vertice o no.

Habilitando la opción *Pegar UV a Vertice de Malla*, al hacer click con **RMB** no solo seleccionará un vertice UV sino todos los vertices UV que correspondan al mismo vertice de la malla. Este modo es accesible incluso si no se lo ha activado desde el menú, simplemente manteniendo pulsada la tecla **CTRL** cuando se selecciona el vertice.

La opción *Stick Local UVs to Mesh Vertex (Pegar UV Local a Vertice de Malla)* funciona de la misma manera, pero solo selecciona los UVs que están "conectados", es decir, que están dentro de un rango de 5 píxeles alrededor del primer UV seleccionado. También puede acceder a este modo aún cuando no esté activado por defecto manteniendo pulsada **SHIFT** cuando se seleccione un vertice.

Estas dos opciones pueden ser respectivamente activadas o desactivadas pulsando **CTRL-C** y **SHIFT-C**.

Con *Active Face Select (Seleccionar Cara Activa)* activada, un click **RMB** seleccionará una cara y la convertirá en la cara activa. Esto puede ser activado y desactivado pulsando **CKEY**.

En todos estos tres casos, aparecerá un icono especial en la esquina inferior izquierda del editor UV. Observe que *Active Face Select* y *Stick UVs to Mesh Vertex (Pegar UVs al Vertice de Malla)* pueden también combinarse.

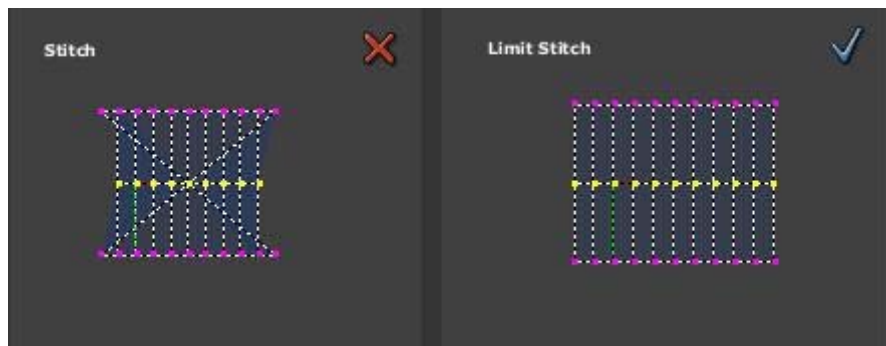
La opción *Unlink Selection (Desconectar Selección)* basará la selección actual, es decir solo dejará seleccionados esos vertices, si las caras estuvieran totalmente seleccionadas. Como lo implica el nombre, esto es particularmente útil para desconectar caras y moverlas a otra parte. La combinación de teclas es **ALT-L**.

Select Linked UVs (Seleccionar UVs Conectados) funciona similarmente a *Select Linked* en la Vista3D. Seleccionará todos los UVs que estén "conectados" a los UVs seleccionados actualmente. La diferencia con la vista 3D es que en el Editor UV, los vertices UV están conectados "implícitamente". Dos vertices UV se consideran conectados si la distancia entre ellos no es mayor de 5 píxeles. La combinación de teclas es **LKEY**.

Utilizando el comando "Coser" ("Stitch" **VKEY**), distintas partes de una mapa UV pueden ser "cosidas" o pegadas entre sí si los vertices UV de los bordes, corresponden a los mismos vertices de malla. El comando Coser funciona uniendo contornos irregulares, seleccione los vertices de la línea fronteriza usando "Stick UVs to Mesh Vertex".

Limit Stitch (Limitar Costura) funciona de manera parecida. La diferencia es que solo pega UVs dentro de un rango dado. El límite por omisión es de 20 píxeles. La ventaja sobre "Coser" es que evita que aquellos UVs que se supone deben permanecer separados, se unan entre sí. En [Figure 11-50](#) puede verse como Limitar Costura evita que las dos partes de un cilindro se superpongan una sobre otra.

Figure 11-50. "Stitch" and "Limit Stitch".

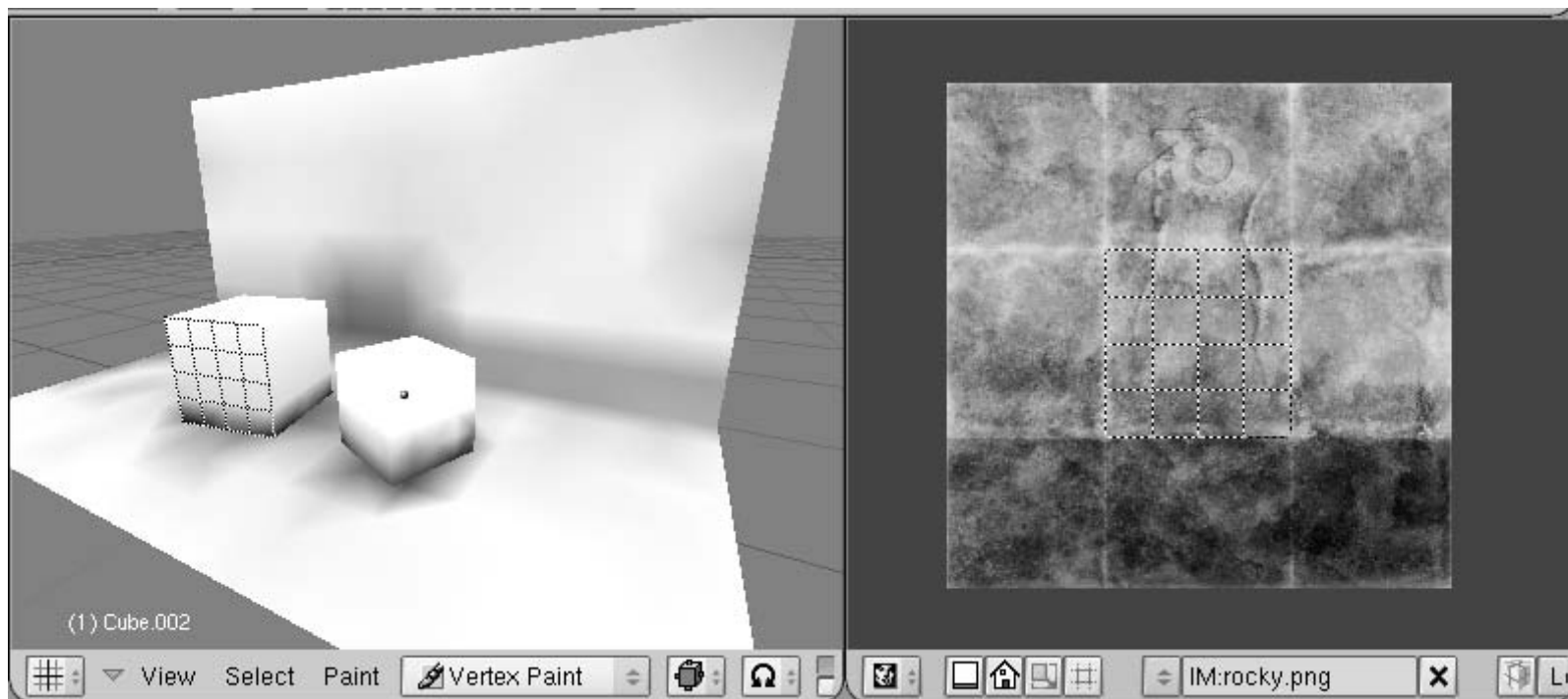


Puede soldar UVs que no correspondan al mismo vertice de malla usando el comando Weld ("Soldar" **WKEY**). También puede usar este comando para alinear varios vertices en X o Y. Después de pulsar **WKEY** pulse **XKEY** o **YKEY** para elegir a cual eje quiere alinear.

Algunas pistas:

- Pulse **RKEY** en la ventana 3D para rotar las coordenadas UV.
- A veces es necesario mover los ficheros de imagenes a otro lugar en el disco rígido. Presione **NKEY** en la Ventana de Imagen para acceder al menu *Replace Image name*. Puede completar el viejo nombre de directorio y el nuevo. Pulse **OK** para cambiar las direcciones de todas las imagenes utiizadas en Blender que apunten al viejo directorio. (Nota: use como nuevo directorio el código `"/"` para referirse al directorio donde se encuentra el fichero .blend).
- Puede usar simultaneamente Selecccion deCaras y Pintura de Vertices (**VKEY**). La Pintura de Vertices solo actuará sobre las caras seleccionadas en este caso. Esta caracteristica es especialmentne util para pintar como si no compartieran vertices. Observe que los colores de vertice se usan para modular el brillo o el color de la textura de imagen aplicada.

Figure 11-51. Los colores de vertice modulan la textura.



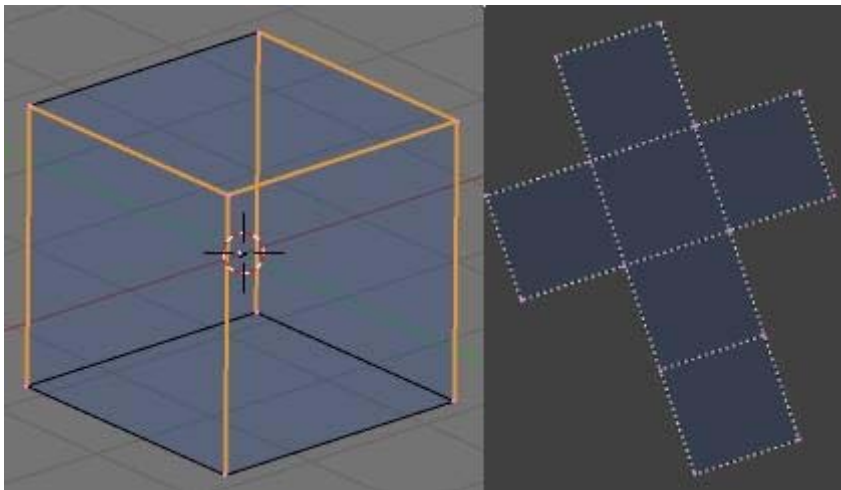
11.5.5. Despliegue LSCM

LSCM es una sigla que significa "Least Squares Conforming Map" o "Mapa Conforme a Cuadrados Mínimos". Este es un método matemático avanzado para crear automáticamente un mapa UV manteniendo al mínimo el estirado y deformación de la textura. Funciona manteniendo los angulos locales. Como cualquier otro modo de despligue UV, desplegará las caras seleccionadas en el Modo Selección de Caras UV. Se puede acceder a él pulsando la tecla **UKEY** y eligiendo la opción *LSCM*, o directamente pulsando *LSCM Unwrap* en el Panel *UV Calculation*.

Para poder desplegar correctamente una malla empleando LSCM, debe asegurarse de que la malla pueda ser aplanada sin demasiada deformación (en términos matemáticos, debería ser equivalente a un disco). Esto se hace definiendo "costuras" (seams), es decir, lugares por donde será cortada la malla. No necesita definir costuras si la malla puede ser desplegada sobre un plano directamente.

En ModoEdición, las aristas seleccionadas pueden ser marcadas como costuras o aristas simples usando las teclas **CTRL-E**. Aquí puede verse un cubo con costuras, y el mapa que resulta luego de aplicar LSCM.

Figure 11-52. Método de despliegue LSCM.



A menudo una malla no puede desplegarse como un único grupo de caras, sino que debe cortarse en varios grupos. Si las costuras dividen a las caras seleccionadas en varios grupos, el despliegue LSCM los desplegará separadamente ubicándolos en el Editor UV de manera que los grupos de caras no se superpongan. Para facilitar la selección de grupos de caras, el comando Seleccionar Conectados (**LKEY**) en el Modo Selección de Caras UV seleccionará todas las caras conectadas siempre que no haya ninguna costura que las divida. De esta manera puede seleccionar un grupo de caras seleccionado una sola de las caras del grupo y luego pulsando **LKEY**.

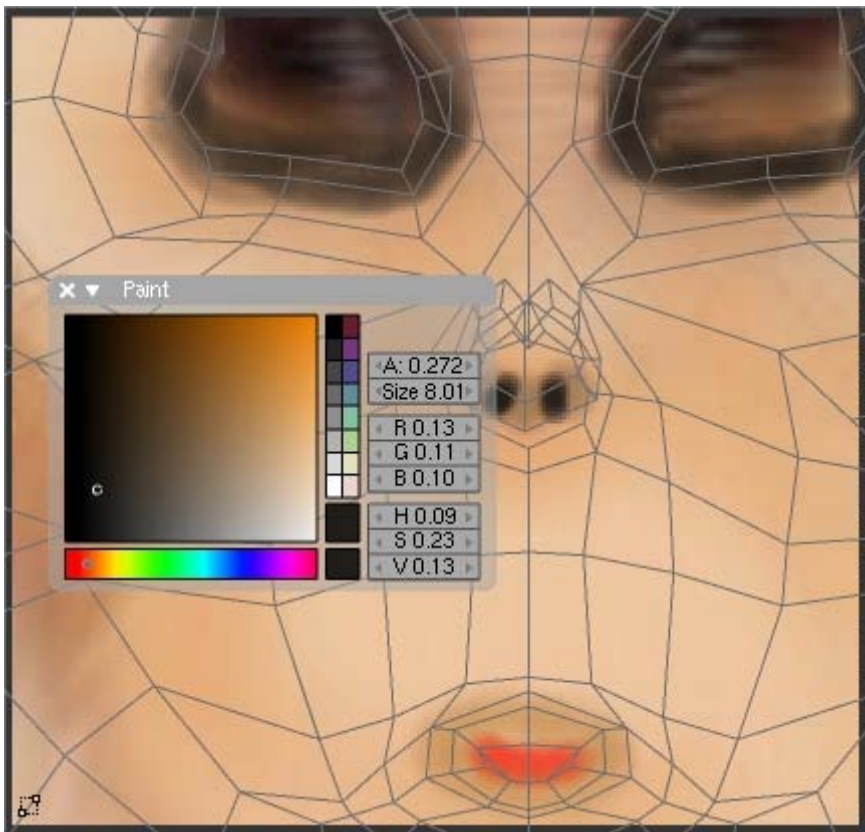
Para retocar aún más el resultado, los UVs en el Editor UV pueden ser fijados o "clavados" en una posición determinada. Si se ejecuta LSCM, estos UVs permanecerán en su lugar y el mapa UV resultante se adaptará a los UVs fijos. En el Editor UV, los UVs seleccionados se fijarán o soltarán pulsando **PKEY** o **ALT-P** respectivamente. Pulsando **EKEY** en el Editor UV se iniciará el despliegue LSCM sobre las caras visibles en el Editor. Los UVs fijos son marcados en color rojo.

11.5.6. Pintado de Texturas (Texture Paint)

Una vez que ha cargado una imagen en el Editor UV, puede modificarla utilizando el Modo de Pintado de Texturas. Use la opción *Paint Tool* en el menú *View* para cambiar el tamaño de pincel, Opacidad y Color. Actualmente solo hay un pincel por omisión, pero se está trabajando para ofrecer más pinceles.

Todos los cambios realizados se reflejarán inmediatamente en la Vista 3D si el modelo se encuentra en "modo papa". Sin embargo la textura modificada no se guardará hasta que explícitamente sea ordenado. Use la opción *Save Image* en el menú *Image* para guardar su trabajo con un nombre distinto o reemplazar la imagen original.

Figure 11-53. La herramienta "Paint" en acción.



Observe que la opción *Draw Shadow Mesh (Dibujar Sombra de Malla)* se vuelve muy útil para mantener la referencia del mapa UV mientras se pintan las texturas.

11.5.7. Renderizado y coordenadas UV

Aún sin una imagen asignada a las caras, puede renderizar usando coordenadas UV. Para ello, utilice el botón *UV* verde en el menú de Botones de Materiales (**F5**).

Si también desea renderizar la imagen asignada, deberá pulsar el botón *TexFace* en los botones de Materiales. Combinando esto con la opción *VertexCol* hará que se usen también los colores de vertice.

[Prev](#)

Cristal Solido y Hueco

[Home](#)

[Up](#)

[Next](#)

Desenrollando a Suzanne

11.6. Desenrollando a Suzanne

Relevant to Blender 2.34

por Claudio 'Malefico' Andaur

Cuando trabajamos con modelos complejos como por ejemplo personajes, se hace evidente la necesidad de herramientas más poderosas para hacer nuestros mapas UV. Desde la versión 2.34 se han incorporado al código fuente de Blender varias herramientas nuevas como las Costuras (Seams) y el método de desenrollado (unwrapping) LSCM.

En este tutorial voy a utilizar dos enfoques para realizar un desenrollado completo de Suzanne para mostrar cómo utilizar estas nuevas herramientas.

11.6.1. Organizandose con "Costuras"

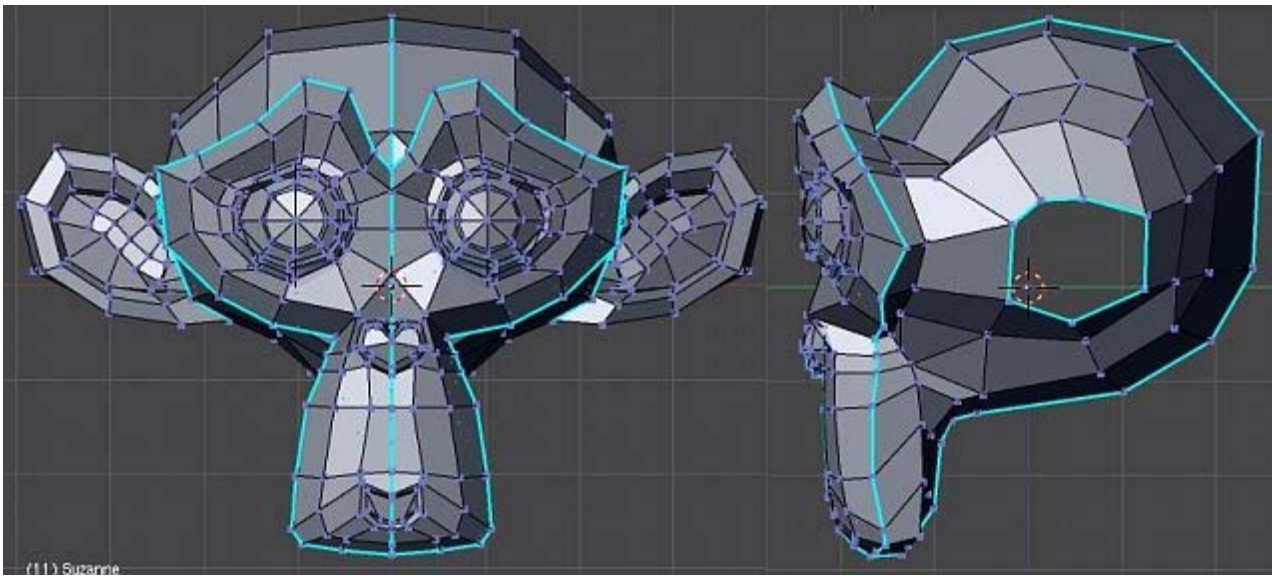
Una malla puede organizarse con la ayuda de "costuras" que proveen una manera cómoda de crear "Grupos de Caras" antes de hacer un mapeado UV. Estos "grupos" son solo una manera de nombrar a una seleccion de vertices y no una entidad separada como lo son los Grupos de Vertices. Las costuras se crean seleccionando un loop de vertices en Modo Edición y pulsando **CTRL-E** o si usamos los menues, seleccionando la opcion *Edge Menu->Mark Seam*. Una linea continua gruesa se dibujará en la ventana 3D representando esta costura. Es posible visualizar esta costura tanto en Modo Edición como en Modo Seleccion de Caras activando la opción *Draw Seams* en los botones de edición.

Cuando se define una costura cerrada, podemos seleccionar cualquiera de las partes en que se ha dividido el modelo en Modo Seleccion de Caras, seleccionando primero una cara y luego pulsando **LKEY**. Se seleccionarán todas las caras conectadas entre si que han sido aisladas del resto del modelo por esta costura. Creando costuras estratégicamente en nuestro modelo podremos luego trabajar luego cómodamente con este grupo de caras solamente simplificandose nuestra tarea.

Idealmente deberíamos marcar una costura donde sea que queramos que nuestro mapa UV tenga cortes. Por ejemplo, para aislar los brazos de nuestro personaje del torso, marcaríamos una costura cerrada en cada hombro. Habrá casos tambien donde necesitaremos marcar una costura abierta en lugar de una cerrada.

Hagamos algunas pruebas. Agreguemos una Suzanne a nuesra escena y seleccionemos el loop de vertices vertical central, asegurémonos de seleccionarlo completamente. Presionemos **CTRL-E** para marcar este loop como una costura. Probemoslo, entremos al Modo Seleccion de Caras, deberíamos ver la costura. Seleccionemos una cara y pulsemos **LKEY**, todas las caras de este lado deberian haberse seleccionado. Si se hubieran seleccionado todas las caras del modelo significaría que hemos omitido algún vertice del loop central. Si ese fuera el caso, por favor selecciona los vertices faltantes. ¿ Fácil no ? Esa es la idea.

Figure 11-54. Suzanne con las costuras marcadas.



Creemos algunas costuras mas, intenta aislar las zonas mas complejas de las otras, por ejemplo las orejas del resto de la cabeza. Cuando hagamos el volcado UV, estas costuras actuarán como cortes en la superficie y también en el mapa UV.

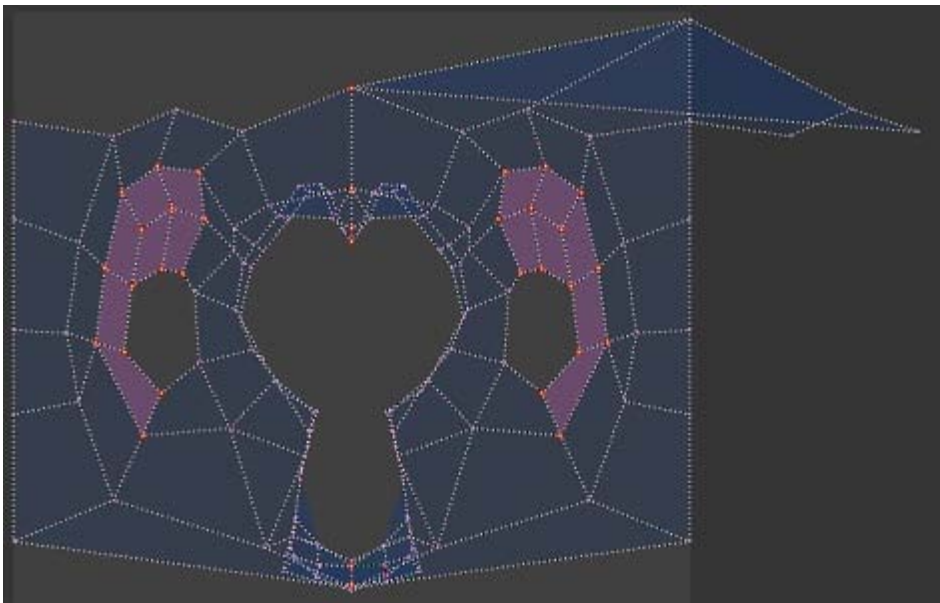
No olvides usar la herramienta de seleccion de loop de vertices **ALT-B** para ayudarte en la seleccion de loops para crear costuras. Un loop interesante para marcar una costura es el loop principal de la cara de Suzanne, que nos permitirá separar la cara frontal de la cara posterior.

11.6.2. Desenrollando la malla

Abramos un ventana de editor UV junto a la ventana 3D. Seleccionemos el modelo y entremos en Modo Seleccion de Caras.

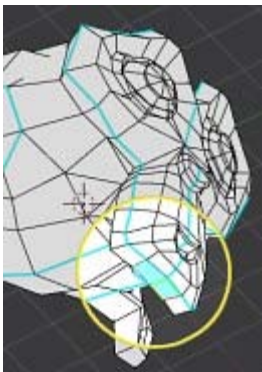
Seleccionemos las caras de los lados izquierda y derecho de la cabeza, incluyendo las orejas. Vamos a realizar el volcado UV de estos grupos usando el método Sphere que nos dará un buen punto de partida para aplicar LSCM más tarde. Es muy probable que obtengas algo "casi" perfecto. sin embargo yo siempre obtengo un par de caras en el lado equivocado del mapa. Esto puede arreglarse facilmente a mano, pero yo soy tan perezoso que no voy a hacer nada en absoluto. Presiona **CKEY** en el editor UV para entrar en Modo de Seleccion por Caras (este Modo está muy pero muy bien). Selecciona algunas de las caras alrededor de las orejas, solo aquellas que se vean desplegadas radialmente. Salgamos de este modo y selecciona los vertices en la zona superior central. Los de la parte inferior central son un poquito más dificles de seleccionar. Los seleccionaremos en un rato. Ahora con todos estos vertices seleccionados pulsa **PKEY**. Este es el comando Pin (alfiler). Esto fijará los evrtice seleccionados en la posicion actual, de forma que nada pueda alterar nuestro mapa. Más sobre esto pronto.

Figure 11-55. Fijando UVs.



Ahora que hemos fijado los UV "bonitos" del mapa, deseccionemos todo en la ventana 3D excepto las dos caras en el mentón de Suzanne. En el editor UV, fijemos los dos vertices UV centrales de estas caras que antes eran inaccesibles. Casi estamos.

Figure 11-56. Accessing occluded UVs.

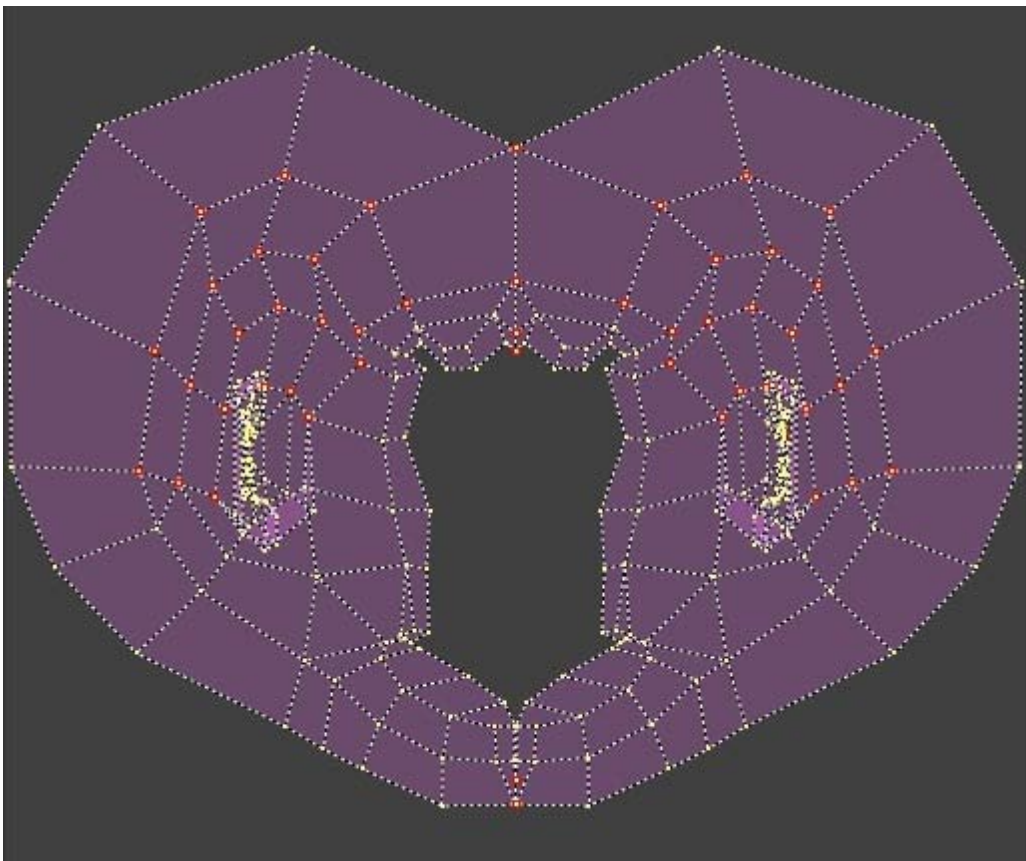


Ahora, llegó el momento de un poco de magia LSCM. En la ventana 3D selecciona todas las caras unidas a las del mentón (usando **LKEY**, te acuerdas ?). Esta vez NO seleccionaremos los vertices de las orejas. Las caras seleccionadas se deberían ver como un mapa en el editor UV. Ahora selecciona los vertices en el editor UV y presiona **EKEY**. Blender te preguntará si debe hacer un cálculo LSCM. Acepta el ofrecimiento. Cosas maravillosas ocurrirán. De pronto las caras que no fueron fijadas se reubicarán en mejores posiciones, caras que antes se solapaban se irán desolapando. Cameron Díaz te llama por teléfono.

Si seleccionas las caras de las orejas en la ventana 3D. verás que aun están ahí, no te preocupes por ellas ahora.

Ahora deberíamos hacer algo de espacio en el mapa para las caras frontales. Podemos hacerlo seleccionando los vertices del borde y escalandolos luego. El editor UV soporta la herramienta de Edición Proporcional (PET) (**OKEY**) que funciona igual que con las mallas. Es muy útil para hacer algo de lugar aquí sin producir solapamiento de caras.

Figure 11-57. LSCM in action.



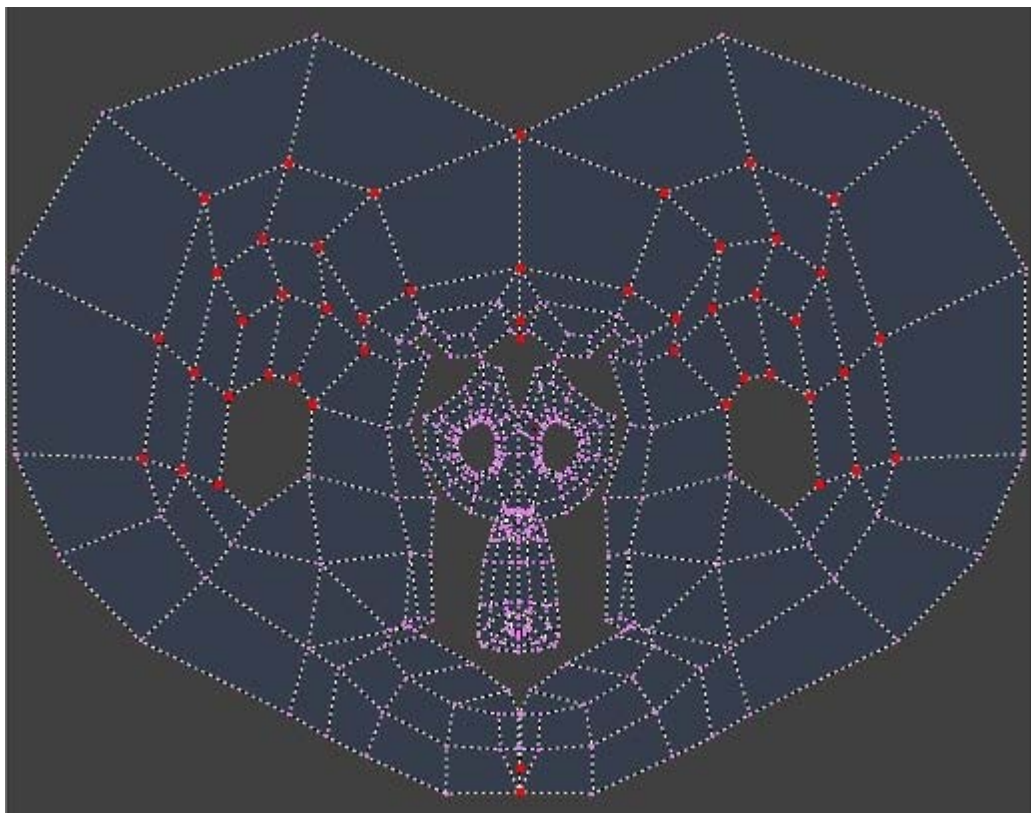
Una vez que hayamos despejado un poco el mapa, seleccionemos las caras frontales, configuremos la vista 3D en Vista Frontal, y volquemos las caras usando la opción "From Window". Reduzcamos un poco en X y en Y para que estas caras quepan en el espacio que hemos reservado en el mapa. Solo presiona **SKEY** seguido por **YKEY** o **XKEY** como si se tratara de mallas.

Tenemos ahora nuestras caras frontales volcadas prolijamente en el Editor UV.

Figure 11-58. Grupo frontal mapeado con "From Window".



Figure 11-59. Dos islas en el mapa.



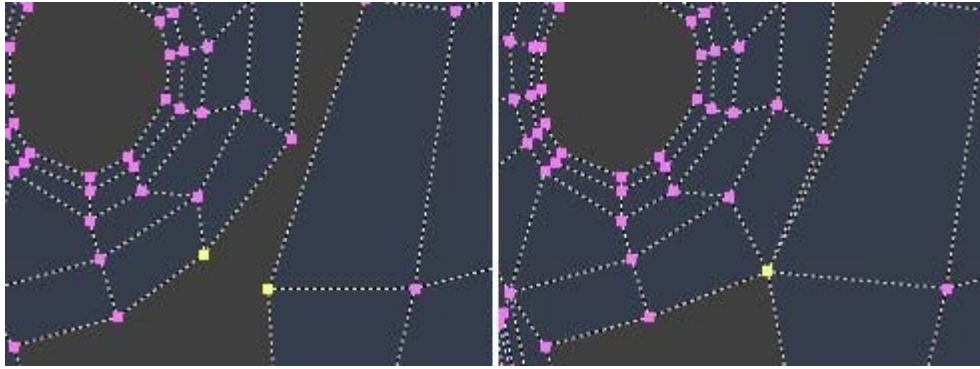
Si deseleccionamos las caras en la ventana 3D, las perderemos de vista en el editor UV, pero podemos activar la opción *Draw Shadow Mesh* para visualizarlas aun cuando no esten seleccionadas.

11.6.3. "Cosiendo el Mapa"

Tenemos que unir de alguna manera estas dos "islas". Activemos la opción *Stick UVs to Mesh Vertex* (**CTRL-C**) para poder seleccionar los UVs a coser de una vez. Selecciona un vertice UV del contorno de la cara, verás su homólogo en la otra isla seleccionarse también.

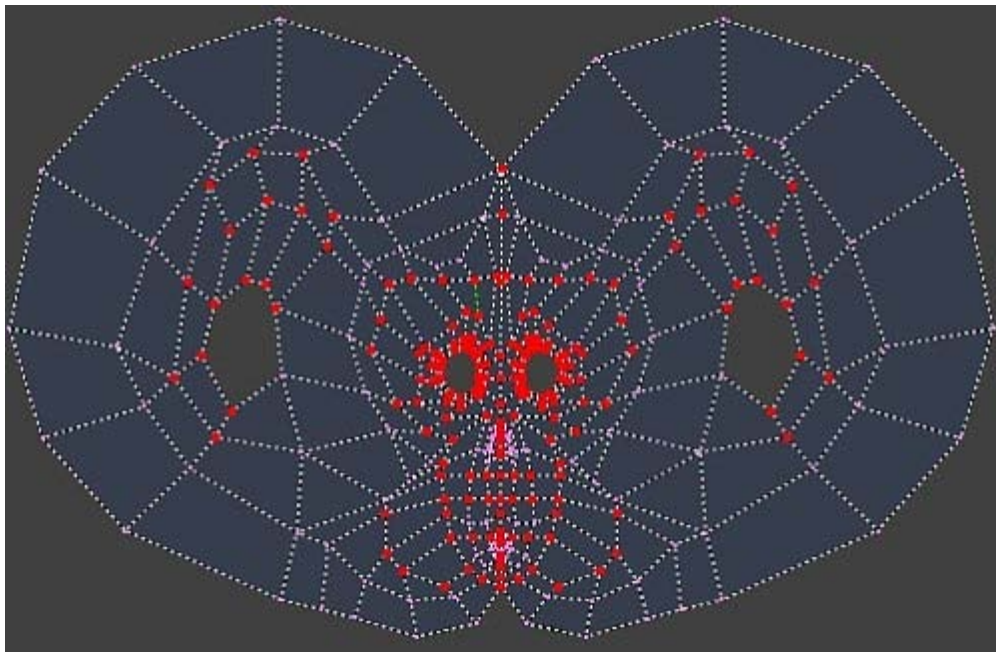
"Cosamos" estos vertices pulsando **VKEY**. Ambos vertices se fundirán en uno solo ubicado a mitad de camino entre ellos.

Figure 11-60. Cosiendo.



Ahora bien, quizás sea un poco tedioso andar cosiendo de a pares de vertices. Selecciona todos los vertices del contorno de la cara de Suzanne. Ahora cosamos. Fijemos los vertices seleccionados. Recuerda fijar solamente aquellos vertices que no pertenezcan a caras solapadas. Si despues de un Stitch, quedan caras solapadas, continua cosiendo el resto pero no fijas este vertice cosido.

Figure 11-61. Stitching.



Ahora tenemos las dos islas cosidas. Sin embargo hay zonas aun donde las caras estan encimadas o solapadas. Aseguremosnos de tener fijos los vertices bonitos y hagamos otro LSCM. Solo pulsa **EKEY**, no es necesario seleccionar nada. Si trabajas con cuidado, en cada paso de LSCM se irán arreglando las caras encimadas, recuerda fijar cada cara "resuelta" para no perderla en el siguiente cálculo.

Tenemos total libertad para escalar o mover los vertices antes de un cálculo LSCM, siempre que seleccionemos de a pares par amantener la simetría bajo control.

11.6.4. Otra vez por favor ;-)

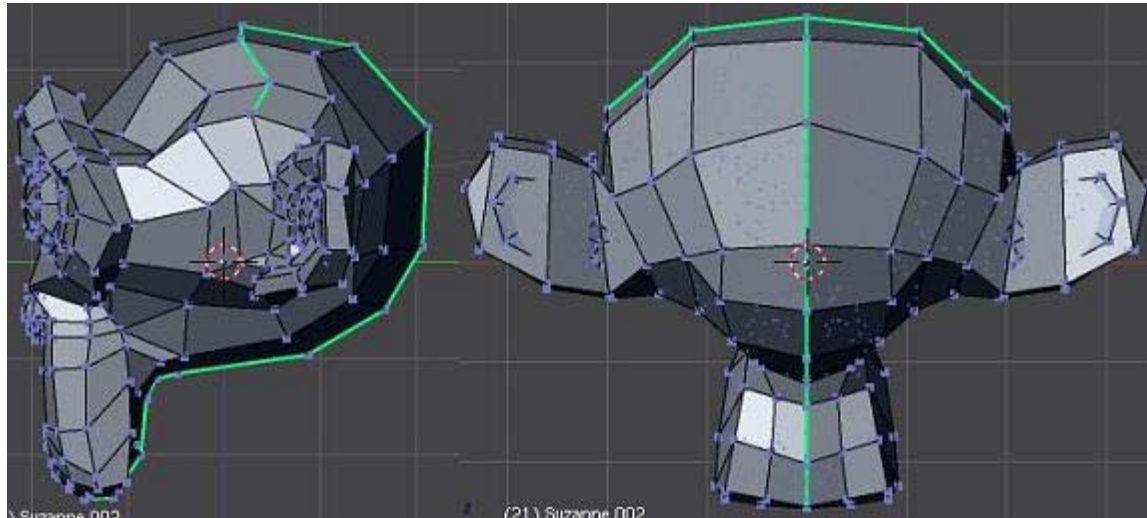
Muy bien. Hemos usado un grupo de herramientas y obtenido un mapa UV medianamente decente. No obstante, a decir la verdad, podriamos haberlo hecho mejor y más rápido.

Las costuras que hemos marcado, no tienen necesariamente que ser "cerradas". Como dije líneas arriba, estas costuras son solo cortes en la superficie, por los cuales Blender "doblará" la malla. Así que si hacemos estos cortes con un poco de astucia podremos hacer todo el volcado

UV en un par de pasos de LSCM.

Desmarquemos todas las costuras en Suzie. La idea principal es mantener las áreas estiradas tan ocultas como sea posible. Como un cirujano plástico, no queremos que las costuras queden demasiado visibles. Con esto en mente, vamos a marcar una costura desde algún lugar en el centro de la parte superior de Suzanne, hasta algún punto cerca del mentón, sin tocar la cara. Podemos hacerlo simplemente usando "Loop Select" para seleccionar el loop central y luego deseleccionando los vértices de la cara. Marquemoslo como una costura. Observa que esta costura está "abierta", es decir no forma un lazo cerrado alrededor de la cabeza como en nuestro primer intento.

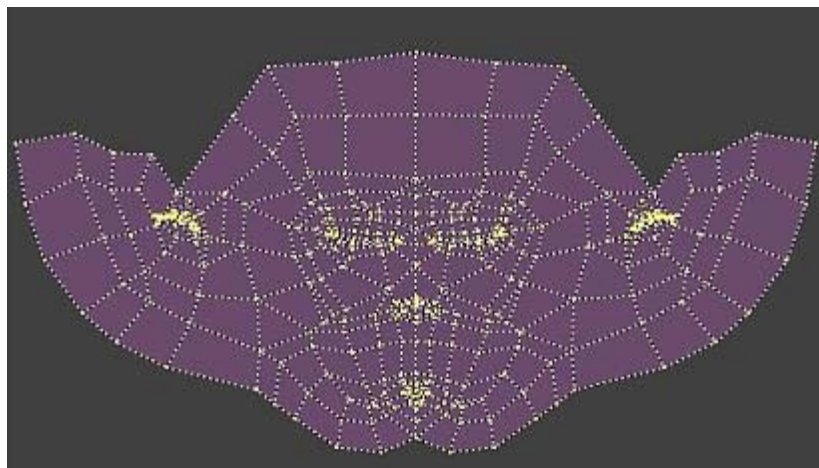
Figure 11-62. Costuras revisadas.



Ahora seleccionamos un loop corto, desde el vértice inicial de la costura anterior a algún lugar cerca de la oreja. Seleccionemos también el loop simétrico correspondiente y marquemos ambas costuras. Eso es todo.

Seleccionemos todas las caras excepto las de los ojos y hagamos un unwrap usando LSCM. Definitivamente lucirá mejor.

Figure 11-63. Y otro desenrollado LSCM ...



Ahora podemos coser y fijar algunos vértices de la misma manera que hicimos antes, pero con mucho menos trabajo.

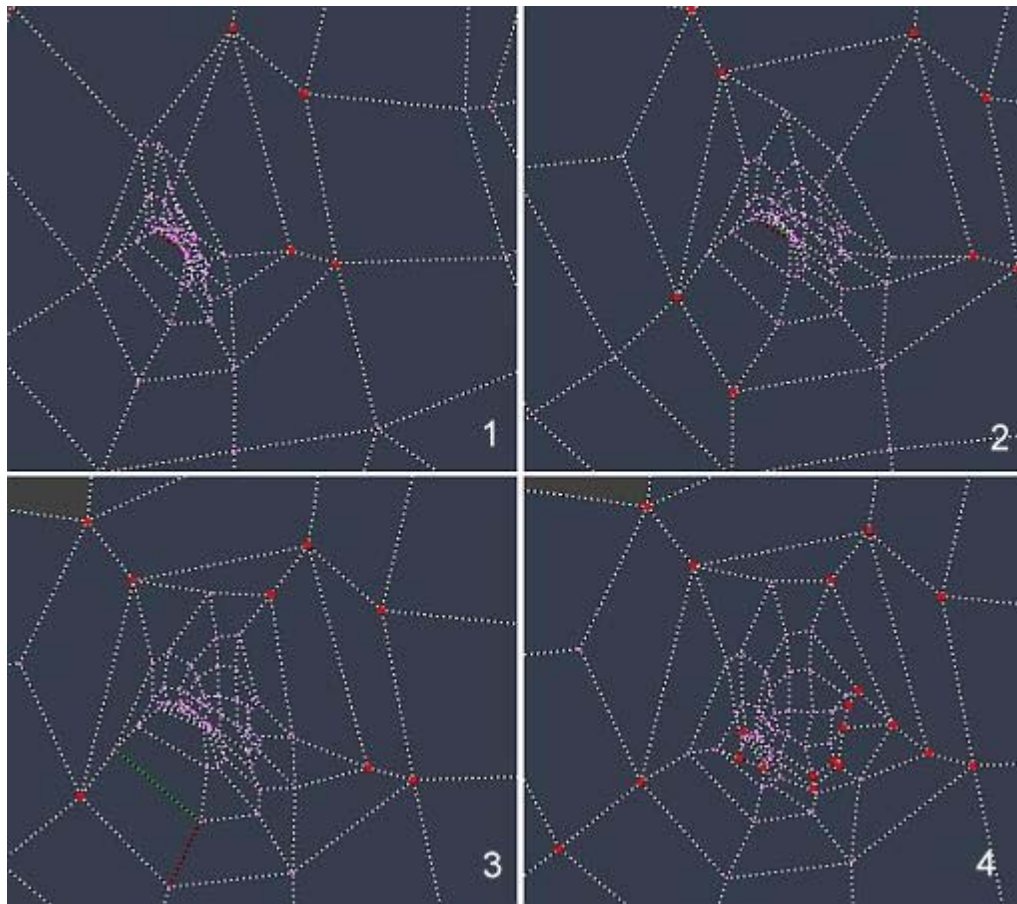
11.6.5. Finalizando

Vamos a usar ahora los métodos que he explicado para corregir todas las caras solapadas de las orejas de Suzanne.

Primero, intentaremos separar los UV exteriores de ambas orejas, seleccionándolos y escalándolos en X. Fijemos estos UV y hagamos un paso de LSCM. Verás que los UV de la oreja empeizan a desencimarse. Continúa con un par interior de UVs y escalalo y muevelo

apartandolo del centro de cada oreja. Fija los UV que acabamos de mover. Haz unpaos LSCM después de fijar UVs. Después de unos pasos más las cosas se empezarán a ver despejadas.

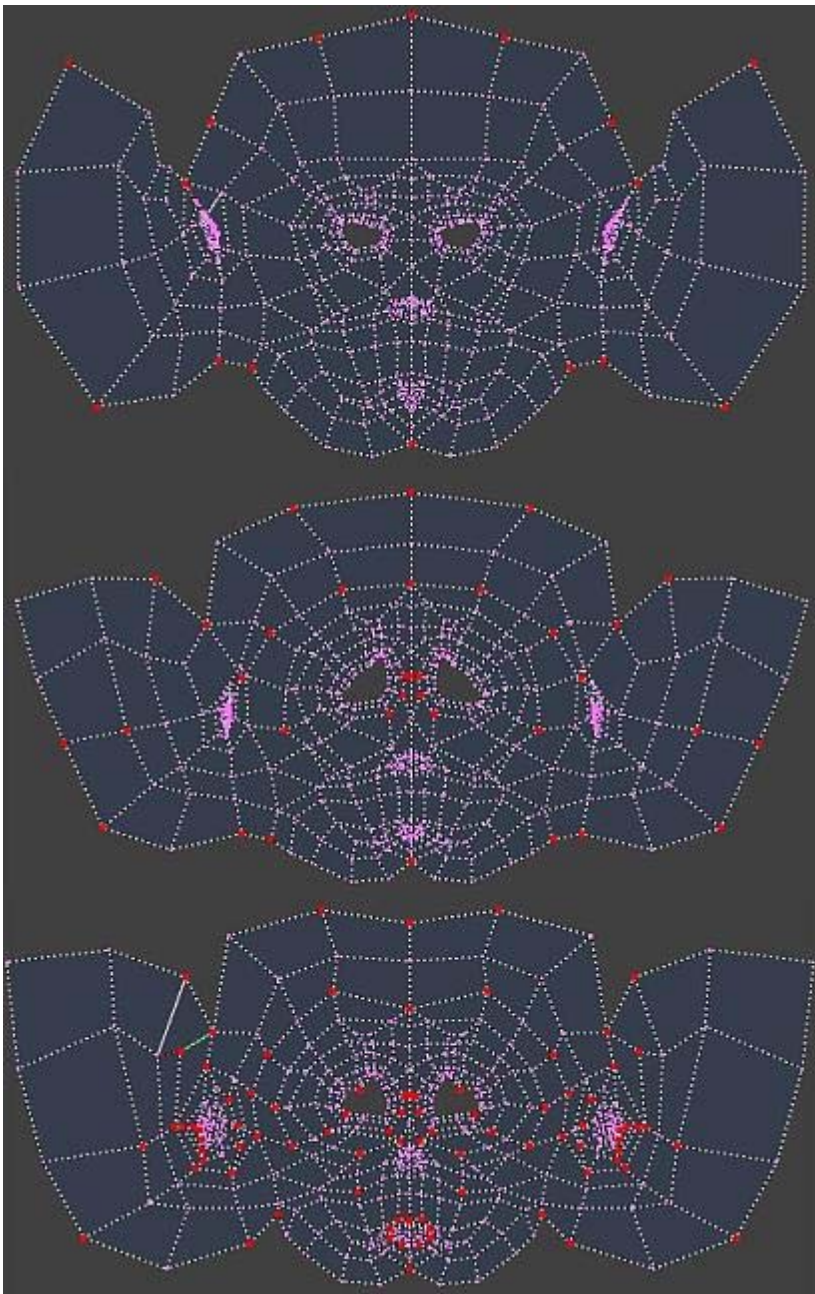
Figure 11-64. Desenrollando las orejas.



Te puedes ayudar seleccionando caras completas de la parte interior y escalandolas. Puedes incluso seleccionar toda la oreja y rotarla 5 o 10 grados para uno u otro lado (dependerá de cual oreja hayas seleccionado).

Con un poco de paciencia obtendrás una oreja completamente desenrollada.

Figure 11-65. Las etapas de desenrollado. Notar los UV que fueron fijados.



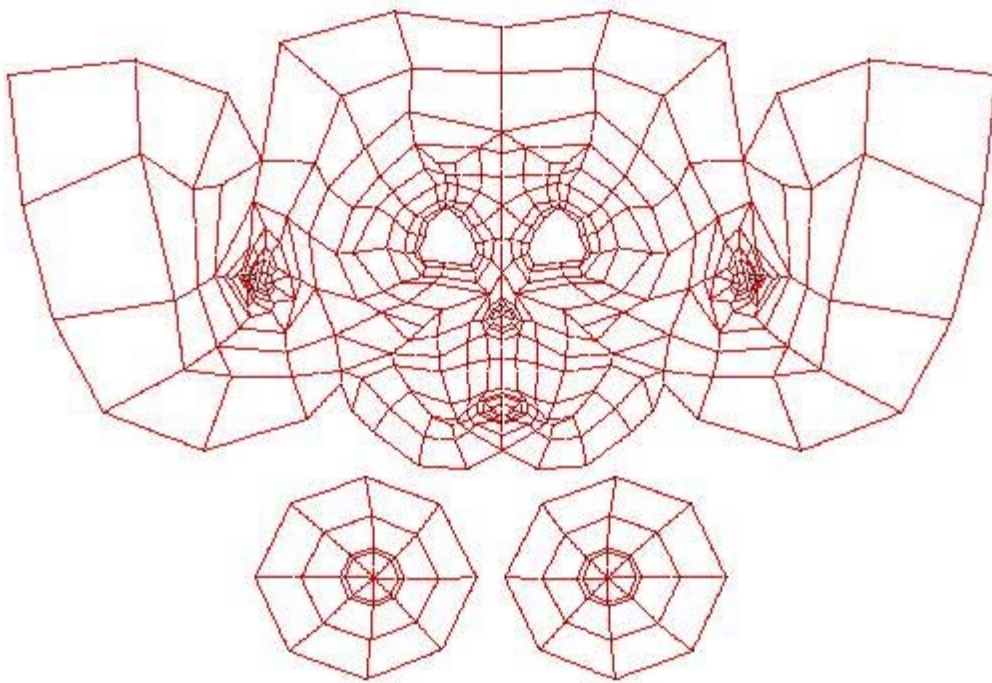
No queda mucho por hacer. podemos seleccionar los grupos de caras de los ojos y mapearlos usando From Window o LSCM. Como los vértices de los ojos no están conectados con el resto de Suzanne, el comando Stitch no funcionará. Si quieres fundir los UVs de los ojos con el resto del mapa UV, deberás usar el comando Weld (Soldar) (**WKEY**) en su lugar. Sin embargo, yo prefiero mantenerlos separados de manera de poder pintar una textura detallada para los ojos con más comodidad.

Si tu personaje tiene párpados, es aconsejable mapearlos "cerrados" para tener una buena superficie donde pintar texturas luego.

11.6.6. ¿ Y ahora qué ?

Una vez que hemos terminado el desenrollado UV, podemos exportarlo usando la opción *Save UV Layout* en el menú, lo que ejecuta un script python incluido en la distribución oficial de Blender. Entre los parámetros configurables tenemos el tamaño de imagen (Image Size), donde deberemos tener en mente que el mapa UV es cuadrado, por lo que solo será necesario ingresar un número para la resolución, y el nombre de archivo donde se guardará la imagen. El script salvará una imagen TGA del mapa UV, la que podremos cargar en Gimp, Photoshop o cualquier otro programa y usar como plantilla para pintar nuestras texturas.

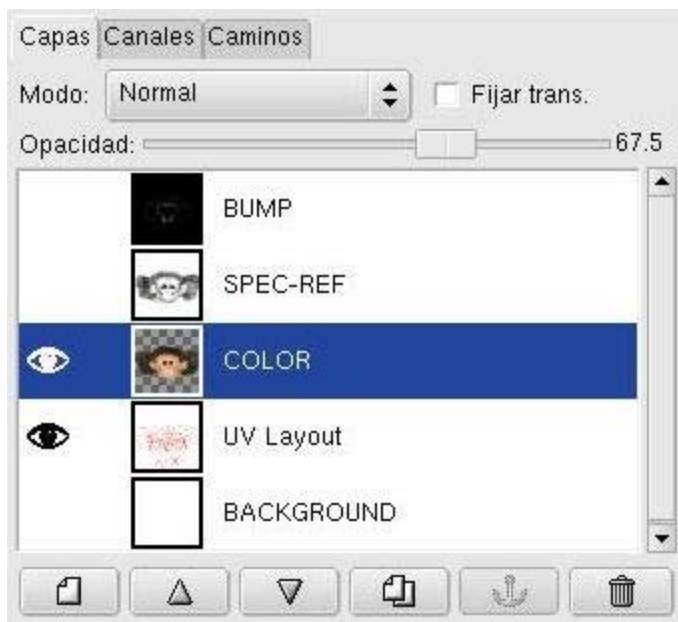
Figure 11-66. Mapa UV exportado final.



Para poder usar este mapa UV como plantilla o referencia, necesitamos que nuestro programa 2D sea capaz de manipular capas transparentes. En mi caso, trabajando en linux, uso Gimp, pero puedes usar cualquier otro que cumpla este requisito.

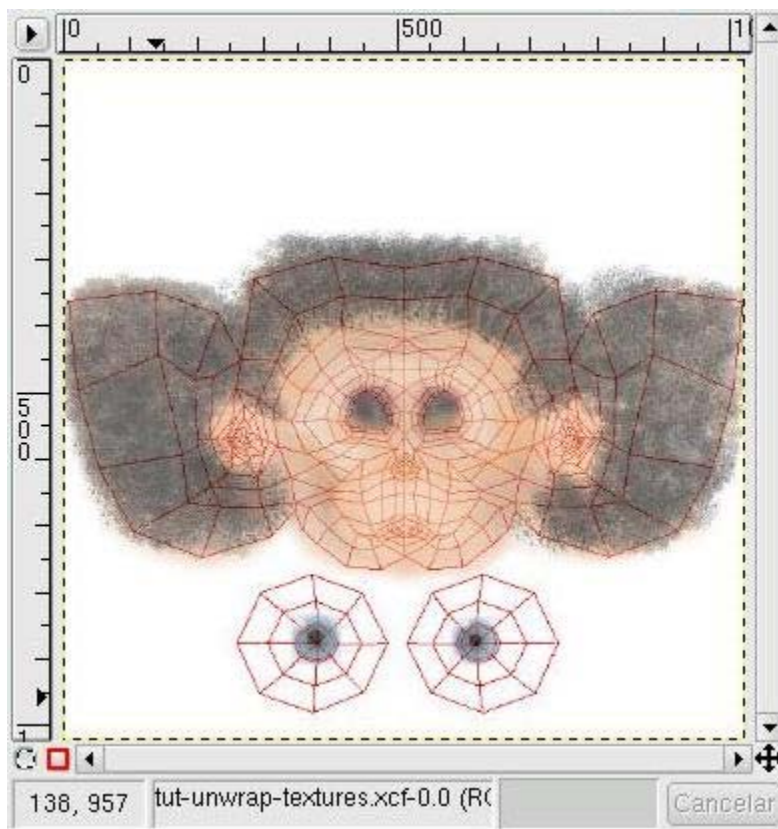
Crea una nueva imagen del mismo tamaño que el mapa UV y carga el mapa en una capa. Agrega un par de capas más encima. Yo he creado tres capas llamadas COLOR, BUMP y SPEC-REF. Estas generarán tres imagenes distintas que usaré como mapas de textura separados para los canales Col, Nor y Ref/Spec en mi material Blender. Sin embargo quizás prefieras usar un unico mapa de textura para todo. Lo bueno de usar varias texturas es que podremos retocar por ejemplo el Bump del material sin alterar el trabajo de color.

Figure 11-67. Gimp es un programa de retoque 2D que soporta capas transparentes.



He apagado todas las capas excepto en la que estoy trabajando y el mapa UV de referencia. Debemos reducir la opacidad de la capa actual para poder tener todo el tiempo a la vista el mapa UV.

Figure 11-68. Usando el mapa UV como referencia para mapas de textura.



El mapa de color no tiene muchos misterios. Estos colores se mapearán sobre el modelo exactamente como se ven. Las texturas SEPC-REF y BUMP son un poco más sofisticadas. Todo lo que sea blanco se verá "más reflectivo", o sea más iluminado en el primer caso, o "con relieve" en el segundo. Las áreas negras por el contrario, se verán oscuras o en bajorrelieve respectivamente.

Figure 11-69. Los tres mapas de textura.



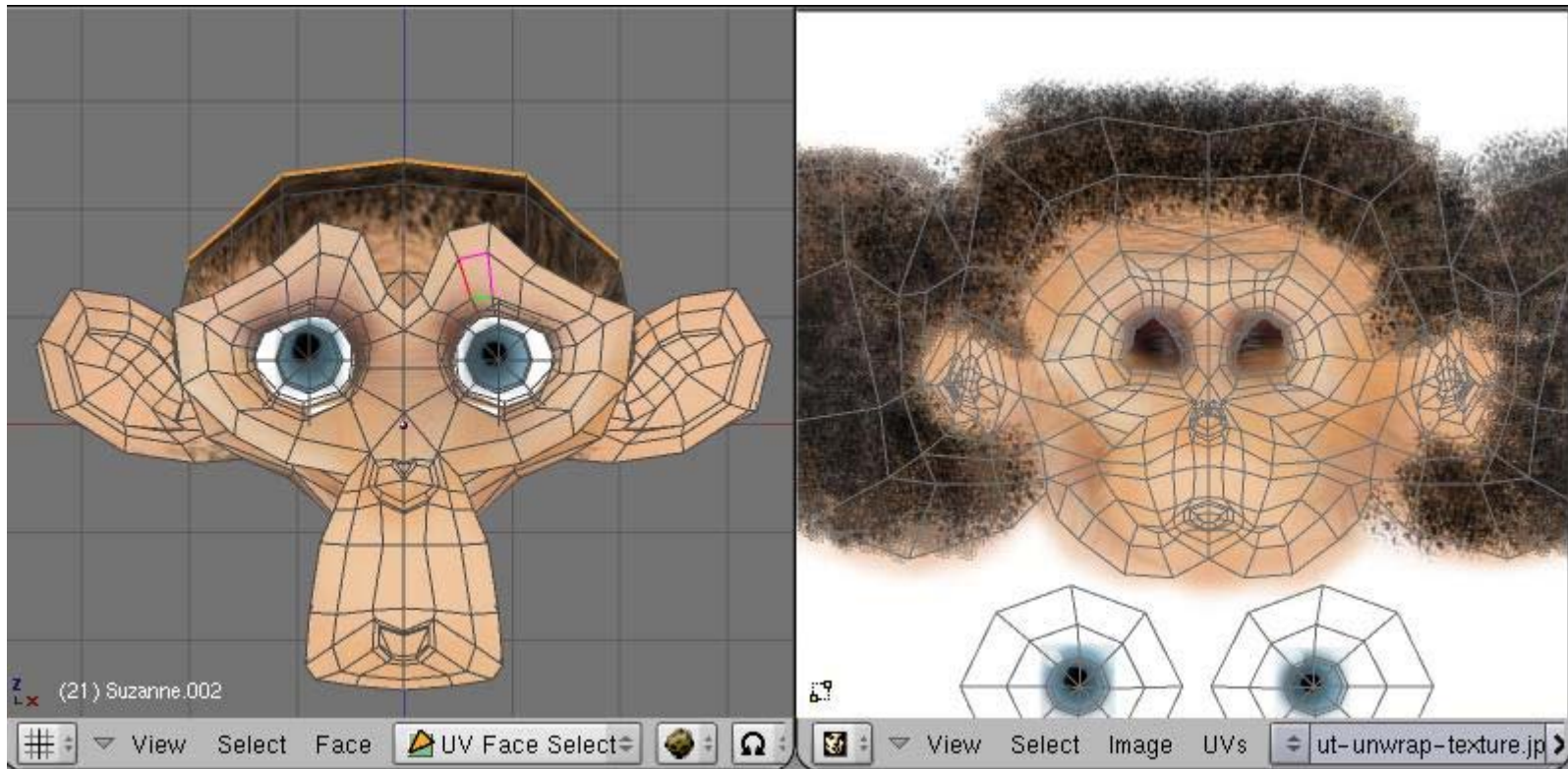
Podemos generar tantos mapas de textura como canales tengamos disponibles en Blender, por ejemplo Reflejos por raytracing, Translucidez, Emisividad, Especularidad, etc. Todos estos mapas de texturas compartirán el mismo mapa UV mientras hayamos activado el botón *UV* en el panel de Input de coordenadas de texturas en los botones de materiales.

Para usar estos mapas de texturas que hemos creado, necesitamos crear primero un material y luego sucesivas texturas de tipo Image. No olvides activar la opción *UV* para CADA textura y aplicarla en el canal correcto. Si usas el mapa de color para el relieve las cosas se verán raras. Puedes mezclar el color del material con el mapa de color si te así te apetece. Simplemente usa el deslizador Col para marcar la intensidad deseada.

Solo necesitas cargar el mapa de textura en el editor UV si necesitas ajustar el mapa basado en como se ve la textura, o si quieres hacer

agregar detalles a la textura usando las herramientas de pintura. De lo contrario realmente no es necesario para obtener el render.

Figure 11-70. Ajustando el mapa de color.



Bueno, no queda mucho para decir sobre texturado UV. En el ejemplo renderizado usé dos materiales con casi los mismos parámetros para la cabeza y los ojos. Solo usé un valor distinto de cero para Emit en los ojos pero todo lo demás es igual. Podemos ajustar el mapa UV para que el mapa de textura se ajuste lo mejor posible y evitar "estiramientos" si hiciera falta.

Nos vemos y sigue blendereando !

Figure 11-71. Render final. Qué texturas más feas... debería pintarlas de nuevo...



[Prev](#)

EditorUV y Selección de Caras

[Home](#)

[Up](#)

[Next](#)

Plugins de Textura

11.7. Plugins de Textura

Relativo a Blender v2.31

Como nota final sobre texturas, echemos un vistazo al cuarto botón de tipo de textura, *Plugin*.

Blender permite, al momento de la ejecución, el enlace dinámico de objetos compartidos, tanto de plugins de textura como de secuencia. En ambos casos, estos objetos son trozos de código C escrito de acuerdo a un estándar determinado ([Chapter 27](#)). En el caso de los plugins de textura, estos trozos de código definen funciones que aceptan coordenadas como entrada y proveen Color, Normal e Intensidad como salida, exactamente como lo hacen las texturas procedurales.

Para usar un plugin de Textura, seleccione esta opción, y luego haga click en el botón *Load Plugin* que aparece en Texture Buttons. Una ventana vecina se convertirá en una ventana de selección de ficheros donde podrá seleccionar un plugin. Estos plugins son archivos *.dll* en Windows y archivos *.so* en sistemas Unix.

Una vez que un plugin es cargado, convierte la ventana Texture Buttons en su propio set de botones, como se describe en las referencias individuales de plugins.

Chapter 12. Iluminación

12.1. Introducción

Relevante a Blender v2.31

La iluminación es uno tema muy importante para el renderizado, tanto como el modelado, los materiales y las texturas.

La escena mejor modelada y texturizada con total precisión podría estropearse si no tiene un buen esquema de iluminación, mientras que un modelo simple puede llegar a ser muy realista si esta habilmente iluminado.

La iluminación, por tristemente, se suele menospreciar por los artistas con menos experiencia, que normalmente creen que, en el mundo real las escenas que suelen estar iluminadas por una sola luz (una lámpara, el sol, etc.) producirán el mismo efecto con una sola luz en gráficos por ordenador.

Esto es falso, debido a que en el mundo real, hay una sola luz presente, ésta se ve modificada por saltos y se vuelve a irradiar en toda la escena por los objetos en los que rebotga, haciendo sombras ligeras y zonas no completamente oscuras, pero iluminadas parcialmente.

Las físicas de las ondas lumínicas son simuladas por los renders de trazado de rayos (Ray Tracing) y pueden ser simuladas con Blender activando el motor "Radiosity" ([Chapter 18](#)).

El trazado de rayos y la radiosidad son procesos lentos. Blender puede ofrecer un render mucho mas rápido con su renderizados interno de escaneo de líneas (Scanline). Un buen renderizador de escaneo de líneas, sin embargo. Este tipo de motor de render es mucho más rápido debido a que no intenta simular el comportamiento real de la luz, asumiendo muchas hipótesis que lo simplifican.

En este capítulo analizaremos los diferentes tipos de luz en Blender y su comportamiento, analizaremos sus puntos fuertes y débiles, y terminaremos describiendo un esquema básico "realista" de iluminación, conocido como el método de los tres puntos, como uno de los más avanzados, realistas, pero por supuesto, de intenso trabajo de CPU, esquema de iluminación.

[Prev](#)

Plugins de Textura

[Home](#)

[Up](#)

[Next](#)

Tipos de luces

12.2. Tipos de luces

Relativo a Blender v2.31

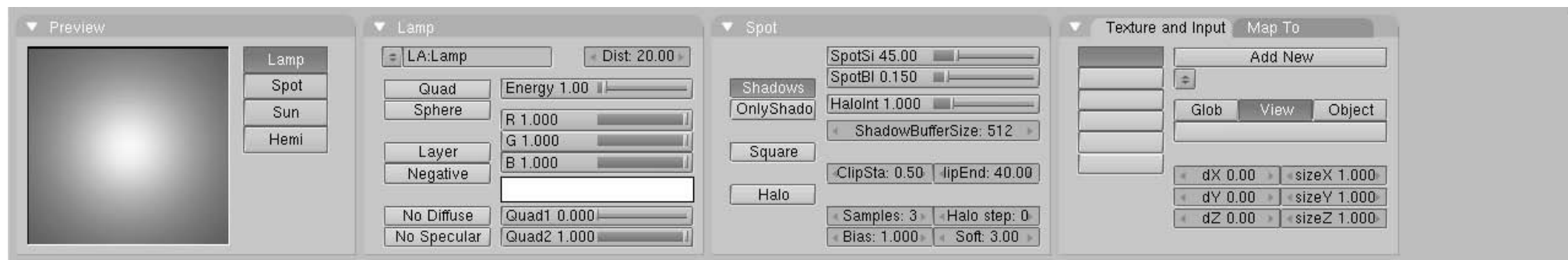
Blender proporciona cuatro tipos de luces:

- Sun Light
- Hemi Light
- Lamp Light
- Spot Light

Cualquiera de estas luces pueden añadirse a una escena pulsando **SPACE** y seleccionando la opción *Lamp* del menú. Esta acción añade una luz de tipo *Lamp Light*. Para seleccionar otro tipo diferente, o para modificar sus parámetros, necesitas cambiar a la ventana Shading Context (contexto de sombreado) [Figure 12-1](#) (F5) y al sub-contexto Lamp ().

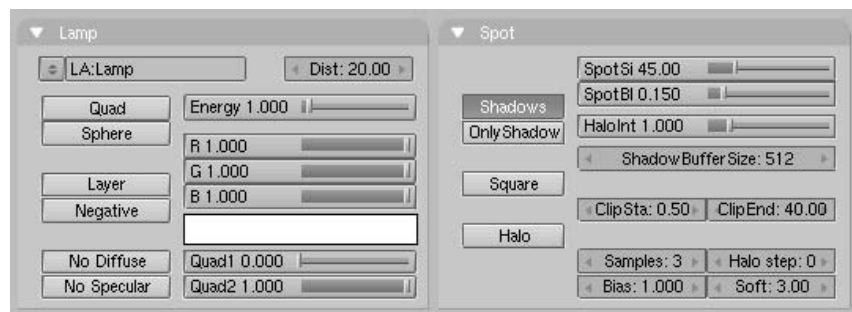
Una columna de botones, en el panel *Preview*, te permitirá escoger el tipo de luz.

Figure 12-1. Botones de luces.



Los botones de luces puede dividirse en dos categorías: los que afectan directamente a la luz, agrupadas en los paneles *Lamp* y *Spot*, y los que definen texturas para la luz, que están en el panel *Texture*, situado a la derecha, y que tiene dos pestañas. Las pestañas son muy similares a las relativas a los materiales. En las subsecciones siguientes nos centraremos en los primeros dos paneles ([Figure 12-2](#)), dejando una breve discusión sobre texturas a la sección **afinando la iluminación (Tweaking light)** section.

Figure 12-2. Botones generales de iluminación.



El panel *Lamp* contiene botones que son en su mayoría comunes a todos los tipos de luces, y merecen por tanto ser explicadas.

Negative - hace que la fuente emita luz 'negativa', es decir, la luz arrojada por el foco es sustraída, en lugar de añadida, a la proyectada por otras luces en la escena.

Layer - hace que la luz proyectada por el foco afecte únicamente a los objetos que están en la misma capa que éste.

No Diffuse - hace que la fuente emita luz que no afecta al modificador de material 'Diffuse', proporcionando únicamente reflejos 'especulares'.

No Specular - hace que la fuente emita una luz que no afecta al modificador de material 'Specular', proporcionando únicamente sombreado 'Diffuse'.

Energy - la energía irradiada por la luz.

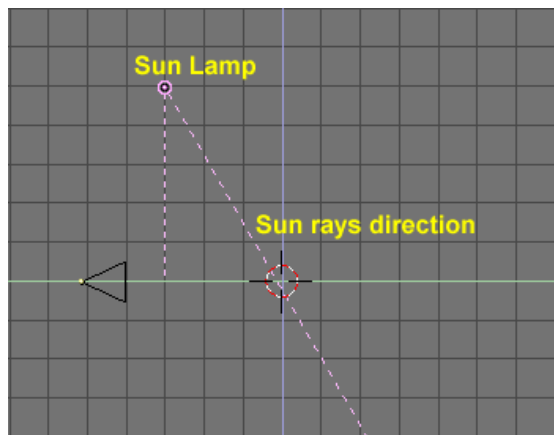
deslizadores *R*, *G*, *B* - las componentes roja, verde y azul de la luz proyectada por la fuente.

12.2.1. Luz Solar (Sun Light)

El tipo de luz más sencillo es probablemente la luz solar (Sun light)([Figure 12-3](#)). Una luz solar tiene una intensidad constante que viene de una dirección dada. En la vista 3D la luz solar se representa por un punto amarillo metido en un círculo, el cual se pone de color púrpura cuando se selecciona, además de una línea de puntos.

Esta línea indica la dirección de los rayos del sol. Por defecto es normal a la vista en la que se añadió la luz solar a la escena y puede rotarse seleccionando el sol y pulsando **RKEY**.

Figure 12-3. Luz solar.

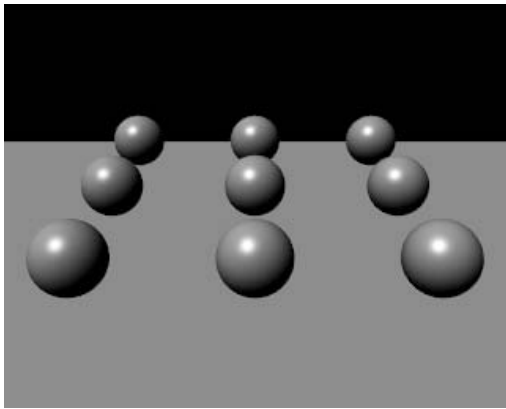


Los botones de luces que se usan con el sol son los descritos en la sección 'general'. Un ejemplo de iluminación solar se muestra en [Figure 12-4](#). Como es evidente, la luz viene de una dirección constante, tiene una intensidad uniforme y *no arroja sombras*.

Esto último es un punto muy importante de comprender en Blender: ninguna luz, excepto las del tipo "Spot", arrojan sombras. La razón para esto reside en la implementación de la luz en un renderizador por líneas y se discutirá brevemente en las subsecciones 'Puntual' (Spot) y 'Sombras' (shadows).

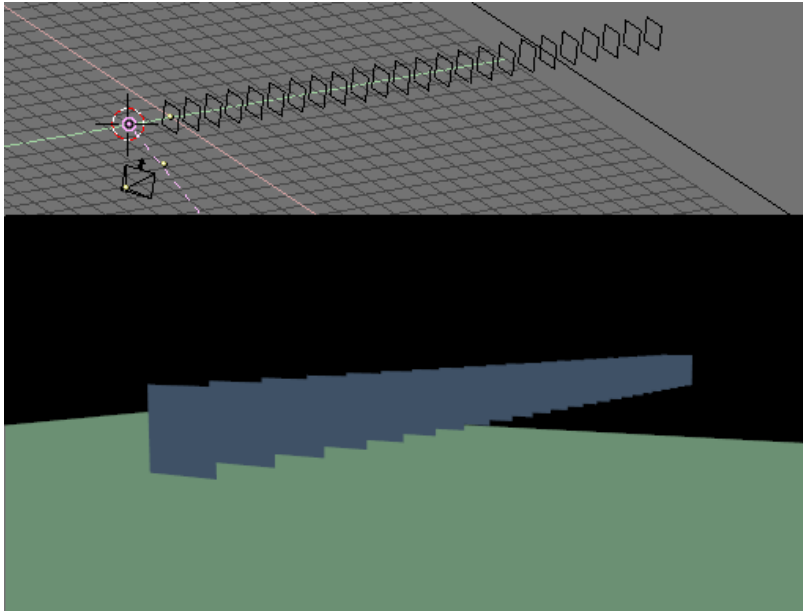
Por último, es importante darse cuenta que en tanto que la luz solar se define por su energía, color y *dirección*, la *ubicación* real en sí de la luz solar no es importante.

Figure 12-4. Ejemplo de luz solar.



[Figure 12-5](#) muestra una segunda configuración, compuesta por una serie de planos que distan 1 unidad de blender entre sí, iluminados con una luz solar. La uniformidad de la iluminación es aún más evidente. Esta imagen se usará como referencia para las comparaciones con otros tipos de luces.

Figure 12-5. Ejemplo de luz solar.



Trucos solares

Una luz solar puede ser muy útil para la iluminación de un espacio abierto a plena luz del día. El hecho de no arrojar sombras puede solucionarse añadiendo luces puntuales (spot) de tipo 'solo sombra' (shadow only). ¡Mira la sección Afinando la iluminación (Tweaking Light)! A Sun light can be very handy for a uniform clear day-light open-space illumination. The fact that it casts no shadows can be circumvented by adding some 'shadow only' spot lights. See the Tweaking Light section!

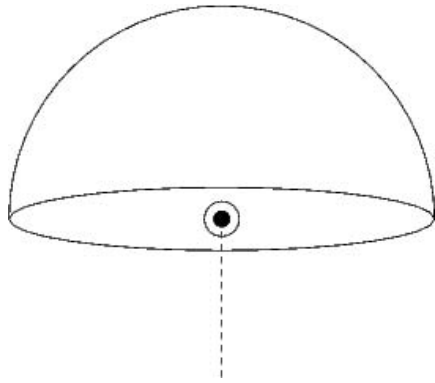
12.2.2. Luz Hemi

La luz Hemi es un tipo muy particular, diseñada para simular la luz proveniente de un cielo muy nublado o de un cielo uniforme. En otras palabras, es una luz que se proyecta, uniformemente, por un hemisferio que rodea la escena ([Figure 12-6](#)).

Es probablemente la menos usada, pero merece ser tratada antes de las otras dos iluminaciones de Blender debido a su simplicidad.

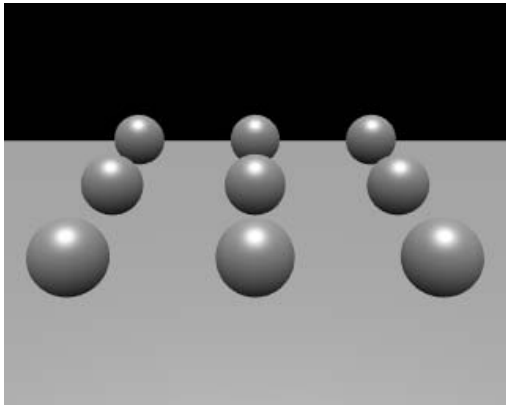
La configuración de esta luz básicamente recuerda a la de la luz solar. Su localización no es importante, mientras que su orientación sí que lo es. La línea de puntos representa la dirección en la que se radia la máxima energía, que es la normal al plano que define el corte del hemisferio, apuntando hacia el lado oscuro.

Figure 12-6. Esquema conceptual de la luz Hemi.



Los resultados de una luz Hemo para la configuración de 9 esferas se muestra es [Figure 12-7](#). Es evidente la superior suavidad de la luz Hemi en comparación con la solar.

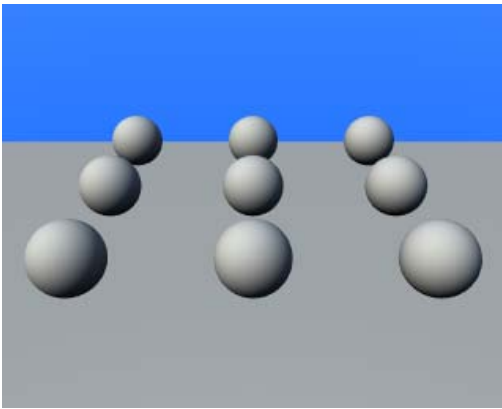
Figure 12-7. Ejemplo de luz hemi.



Truco de luz Hemi

Para conseguir una iluminación exterior más realista, sin el problema de la ausencia de sombras, puedes usar una luz solar, de energía 1.0 and color amarillo(naranja cálido, y una luz Hemi más débil simulando la luz que viene de un cielo despejado. [Figure 12-8](#) muestra un ejemplo con parámetros relativos. La figura también usa un Mundo. Revisa el capítulo correspondiente.

Figure 12-8. ejemplo de iluminación exterior. Energía de la luz solar=1 RGB=(1.,0.95,0.8) La dirección del sol en un sistema de referencia polar es (135°,135°). Energía de la luz Hemi=0.5 RGB=(0.64,0.78,1.) dirigida hacia abajo.



12.2.3. Luz lámpara

La luz de tipo lámpara es una luz de tipo omni-direccional, que es un punto sin dimensiones que irradia la misma cantidad de luz en todas las direcciones. En blender se representa por un punto amarillo encerrado en un círculo.

La dirección de los rayos de luz en la superficie de un objeto es dada por la línea que une el punto de la fuente de luz y el punto en la superficie del objeto. Es más, la intensidad de la luz decrece acorde a un radio dado de la distancia del foco.

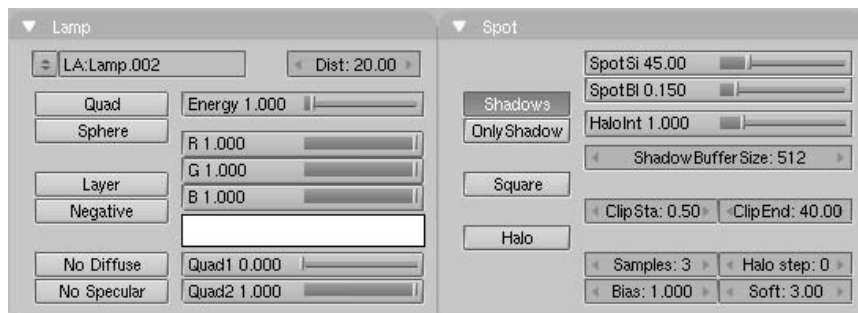
Además de los mencionados antes, se usan otros tres botones y dos deslizadores en el panel *Lamp* en una luz lámpara (Figure 12-9):

Distance - da la distancia a la que la intensidad de la luz es la mitad de la Energía. Los objetos que están más cerca que ese punto, reciben más luz, mientras que los que están más lejos reciben menos luz.

Quad - Si este botón está desactivado, se usa un ratio de atenuación lineal en distancia- en lugar del carente de física. Si está activado, se usa una atenuación más compleja, la cual puede ser definida con más precisión por el usuario desde una atenuación lineal, como la usada por defecto en Blender, a un ratio de atenuación cuadrático en distancia -físicamente correcto. Esto último es un poco más difícil de aprender. Se gobierna por los dos botones numéricos *Quad1* y *Quad2* y se explicará más adelante.

Sphere - Si se presiona este botón, la luz arrojada por la fuente resulta confinada en la esfera de radio *Distance* en lugar de dispersarse hacia el infinito con su radio de atenuación.

Figure 12-9. Botones de luz lámpara.

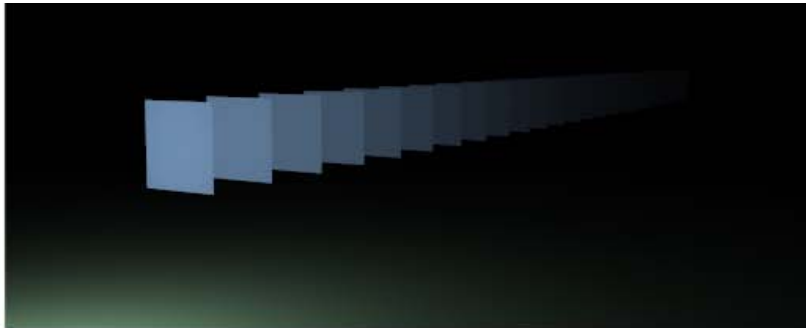
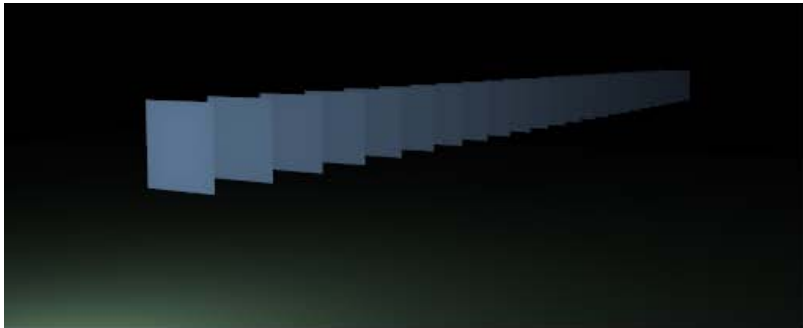


A continuación [Figure 12-10](#) se muestra la misma configuración que en el último ejemplo de luz solar, pero con una luz lámpara con diferentes valores *Distance* y con la atenuación *Quadratic* activada y desactivada.

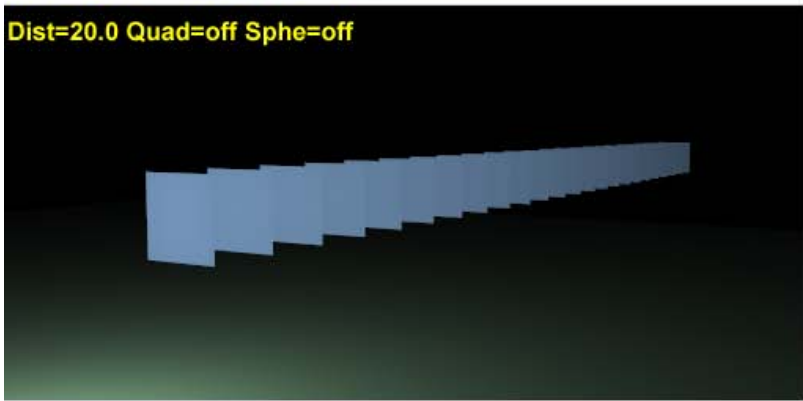
Figure 12-10. Ejemplo de luz lámpara. En los ejemplos Quad Quad1=0, Quad2=1.

Dist=5.0 Quad=off Sphe=off

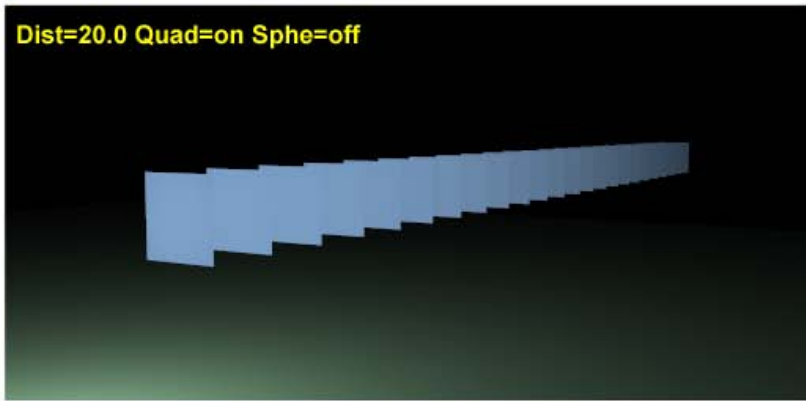
Dist=5.0 Quad=on Sphe=off



Dist=20.0 Quad=off Sphe=off



Dist=20.0 Quad=on Sphe=off



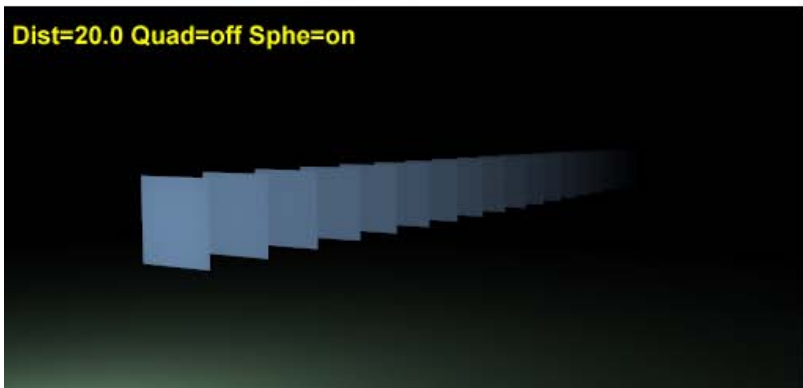
Dist=5.0 Quad=off Sphe=on



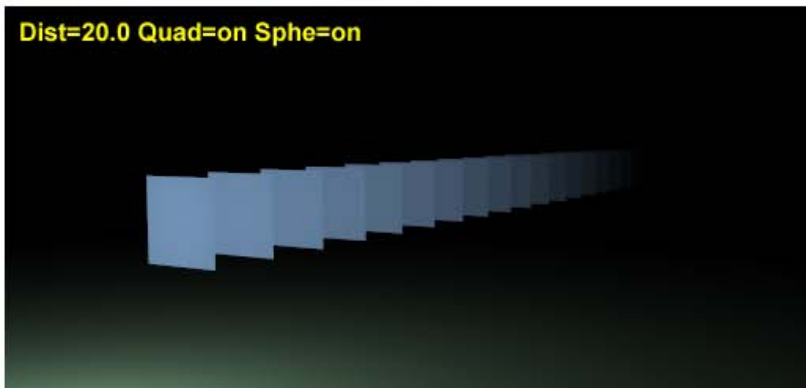
Dist=5.0 Quad=on Sphe=on



Dist=20.0 Quad=off Sphe=on



Dist=20.0 Quad=on Sphe=on



El efecto del parámetro Distance es evidente, mientras que el del botón Quad es más sutil. En cualquier caso la ausencia de sombras sigue siendo un problema considerable. En realidad, únicamente debería iluminarse el primer plano, porque todos los demás deberían caer bajo la sombra del primero.

Para los entusiastas de las matemáticas, y para los que deseen profundizar un poco más, las leyes que gobiernan la atenuación son las siguientes.

Haz que D sea el valor del botón numérico Distance, E el valor del deslizador Energy y r la distancia de la lámpara al punto donde la intensidad de la luz I va a ser calculada.

si los botones Quad y Sphere están desactivados:

$$I = E \frac{D}{D + r}$$

Es evidente lo que se afirmó antes: que la intensidad de la luz es igual a la mitad de la energía por $r = D$.

si el botón Quad está activado:

$$I = E \frac{D}{D + Q_1 r} \frac{D^2}{D^2 + Q_2 r^2}$$

Esta es un poco más compleja y depende de los valores de los deslizadores Quad1 (Q_1) y Quad2 (Q_2). No obstante se nota que la atenuación es lineal para Nevertheless it is apparent how the decay is fully linear for $Q_1=1$, $Q_2=0$ y totalmente cuadrática para $Q_1=0$, $Q_2=1$, siendo ésta última la usada por defecto. Es suficientemente interesante si $Q_1=Q_2=0$ entonces la luz no se atenúa

Si el botón Sphere está por encima de la intensidad de luz calculada I se modifica posteriormente multiplicándolo por el término que tiene una progresión lineal para r de 0 a D y es exactamente 0 en cualquier otro caso.

Si el botón Quad está desactivado y el Sphere está activado:

$$I = E \frac{D}{D + r} \cdot \begin{cases} \frac{D-r}{D} & \text{if } r < D \\ 0 & \text{otherwise} \end{cases} =$$

$$= \begin{cases} E \frac{D-r}{D+r} & \text{if } r < D \\ 0 & \text{otherwise} \end{cases}$$

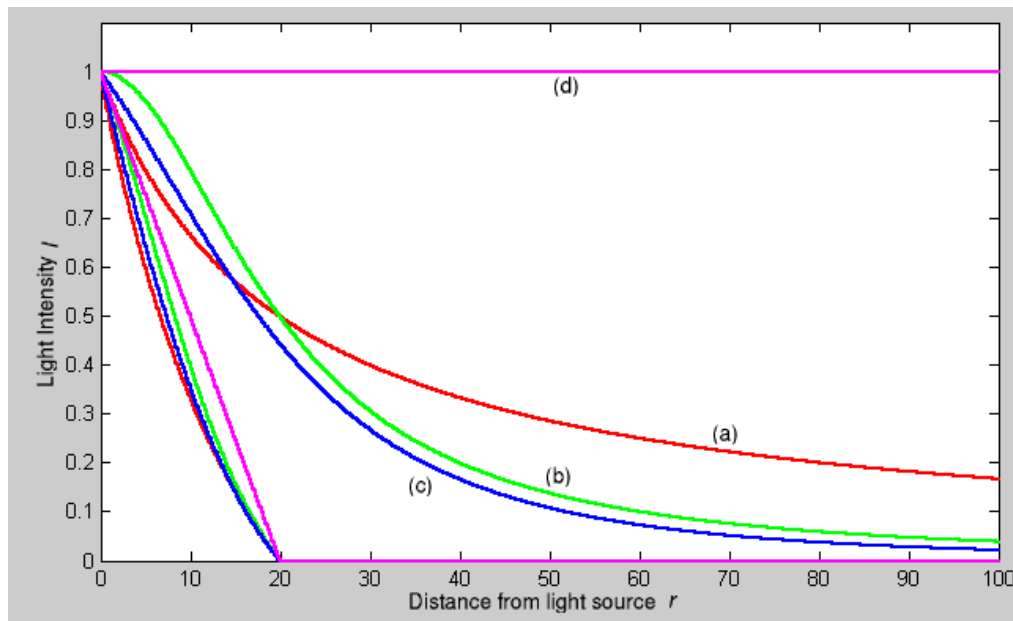
Si los botones Quad y Sphe están activados:

$$I = E \frac{D}{D + Q_1 r} \frac{D^2}{D^2 + Q_2 r^2} \cdot \begin{cases} \frac{D-r}{D} & \text{if } r < D \\ 0 & \text{otherwise} \end{cases} =$$

$$= \begin{cases} E \frac{D-r}{D + Q_1 r} \frac{D^2}{D^2 + Q_2 r^2} & \text{if } r < D \\ 0 & \text{otherwise} \end{cases}$$

[Figure 12-11](#) puede resultar útil a la hora de comprender estos comportamientos de manera gráfica.

Figure 12-11. Atenuación de la luz: a) Lineal, por defecto en Blender; b) Cuadrática, por defecto en Blender con Quad1=0, Quad2=1; c) Cuadrática, con Quad1=Quad2=0.5; d) cuadrática, con Quad1=Quad2=0. También se muestran en el gráfico las mismas curvas, en los mismos colores, pero con el botón esfera activado.



Tip Truco de luz lámpara

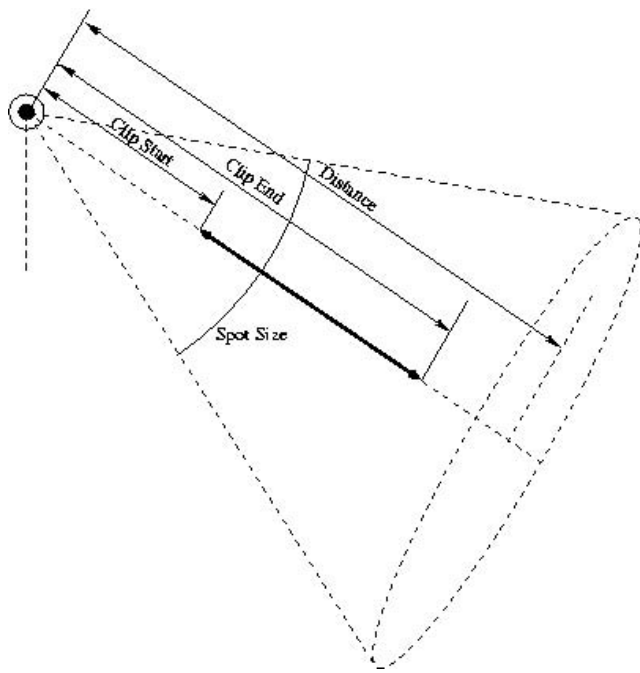
Como la luz lámpara no proyecta sombras, brilla a través de los muros y objetos similares. Si quieres lograr efectos como fuego, o una habitación iluminada por una vela vista desde fuera de una ventana, la opción a considerar es el uso de una Esfera. Trabajando con cuidado en el valor de Distance (distancia), puedes conseguir una tenue luz que sólo ilumine dentro de la habitación, mientras que ilumines el exterior con una suave luz de luna, obteniendo ésta última con una luz de tipo solar (sun), hemi o ambas.

12.2.4. Luz focal

La luz focal es la más compleja de las luces de Blender y es de las más usadas gracias al hecho de ser la única que puede arrojar sombras.

Una luz focal es un rayo con forma de cono que parte de la ubicación de la fuente luminosa, la cual es el vértice del cono, en una dirección determinada. [Figure 12-12](#) should clarify this.

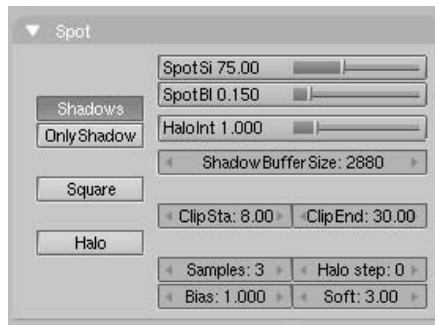
Figure 12-12. Esquema de luz focal.



La luz focal usa todos los botones de una luz de tipo lámpara, y con el mismo significado, pero es tan complejo que necesita un segundo panel de botones ([Figure 12-13](#)): *Spot*.

12.2.4.1. Opciones de luz focal

Figure 12-13. Los botones de opciones de lámpara



Shadows - activa/desactiva la proyección de sombras para este foco.

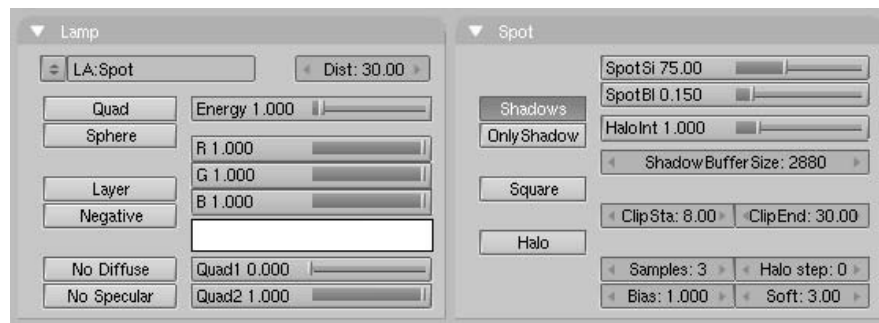
Only Shadow - hace que el foco sólo proyecte la sombra y no la luz. Esta opción se analizará posteriormente en la sección 'Afinando la iluminación'.

Square - Las luces focales habitualmente proyectan un cono de luz de sección circular. Hay casos en los que sería útil una sección rectangular, y tener una pirámide de luz en lugar de un cono. Este botón cambia esta opción.

Halo - permite al foco proyectar un halo como si los rayos de luz atravesaran un medio nublado. Esta opción se explica posteriormente en la sección 'Luz volumétrica'.

12.2.4.2. Botones de luz focal

Figure 12-14. Botones de luz focal.



La columna de botones situada más a la derecha en el Panel *Spot* controla la geometría del foco y las sombras([Figure 12-14](#)):

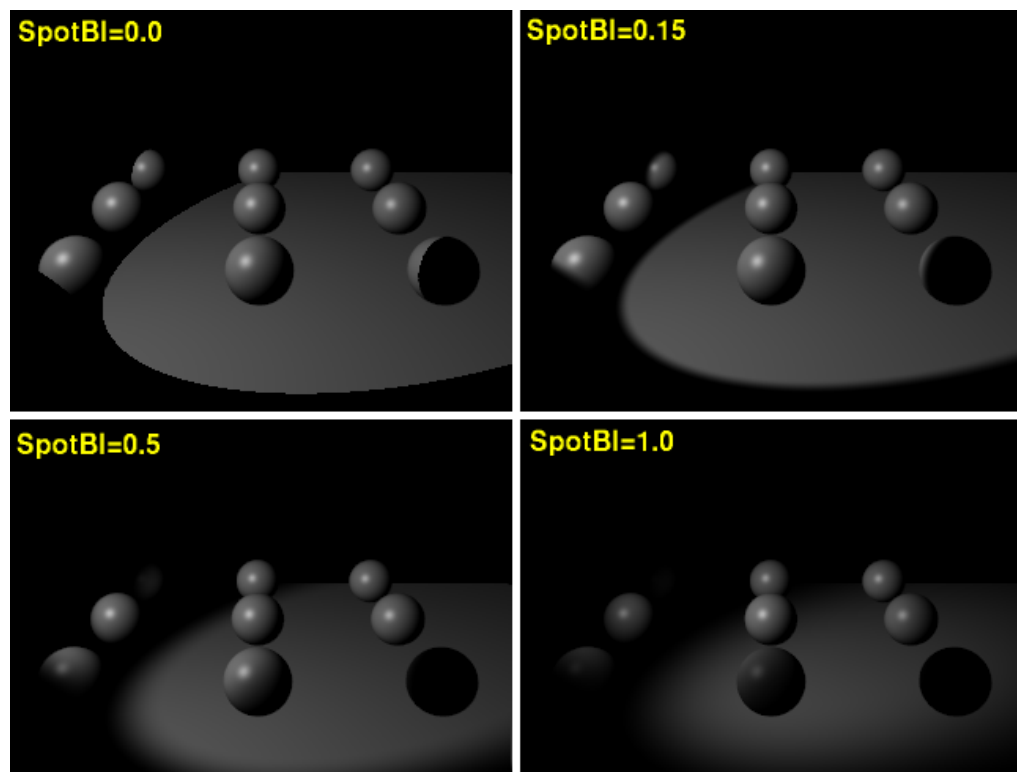
SpotSi - el ángulo en el vértice del cono, o la abertura del foco.

SpotBl - la unión entre el cono de luz y el área en penumbra que lo rodea. Cuanto más bajo sea el valor, más contrastado es el cambio, y cuanto más elevado, más suave. Fíjate que esto sólo se aplica a los bordes de los focos, no a la suavidad de los bordes de las sombras proyectadas por el foco. Estos últimos son gobernados por otro conjunto de botones descritos en la subsección 'Sombras'.

HaloInt - Si el botón Halo está activado, este deslizador define la intensidad del halo del foco. De nuevo, te remitimos a la sección 'Luz volumétrica'.

El grupo de botones inferior de la luz focal gobierna las sombras y es un tema lo suficientemente amplio como para merecerse una sección. Antes de cambiar a las sombras, [Figure 12-15](#) muestra el resultado de usar una luz focal para iluminar nuestro primer caso de prueba en diferentes configuraciones.

Figure 12-15. Ejemplos de luz focal para SpotSi=45°





Note ¡En [Figure 12-15](#) las sombras se desactivan! Las sombras se tratan en la siguiente sección.

[Prev](#)

Iluminación

[Home](#)

[Up](#)

[Next](#)

Sombras con el trazador de rayos

12.3. Sombras con el trazador de rayos

Aún sin escribir

12.4. Sombras con el buffer

Relativo a Blender v2.31

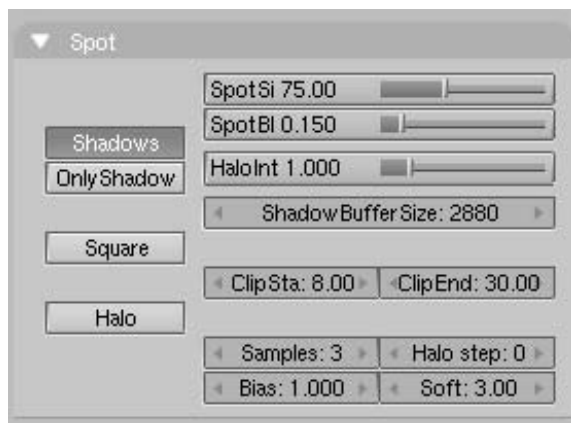
Los esquemas de iluminación analizados hasta ahora producen en los objetos solo áreas que son más o menos luminosas, pero no proyectan sombras, y una escena sin las sombras apropiadas pierde profundidad y realismo.

Por otro lado, un cálculo apropiado de sombras requiere un trazado de rayos completo (y lento). En un renderizador por líneas como Blender, las sombras son calculadas usando el *shadow buffer* en el caso de sombras producidas por las luces. Esto implica que una 'imagen', es vista desde la misma la luz Spot y renderizada y se almacena la distancia de cada punto al foco. Cualquier punto de la imagen renderizada que este mas lejos que cualquiera de estos puntos es entonces considerado como sombra.

El shadow buffer almacena estos datos. Para mantener el algoritmo compacto, eficiente y rápido el shadow buffer tiene un tamaño que es fijado desde el comienzo y que en en Blender puede ser desde 512x512 hasta 5120x5120. El valor mayor es el mas preciso.

El usuario puede controlar el algoritmo a traves de los botones inferiores en el panel *Spot* ([Figure 12-16](#)).

Figure 12-16. Botones de sombra de la luz Spot.



ShadowBuffSize - Botones Numericos, desde 512 hasta 5120, definen el tamaño del buffer de sombras.

ClipSta, *ClipEnd* - Para aumentar la eficiencia del cómputo de sombras, las mismas funcionan en realidad solo en un rango predefinido de distancias desde la posición del foco. Este rango va desde *ClipSta*, cercano a la luz Spot, hasta *ClipEnd*, mas lejos ([Figure 12-12](#)). Los objetos mas cercanos al *ClipSta* desde la luz Spot nunca son considerados para las sombras, y estan siempre iluminados. Los objetos mas lejanos al *ClipEnd* nunca son considerados para iluminarlos y estan siempre en la sombra. Para conseguir una sombra realista *ClipSta* debe estar mas cerca que la menor distancia entre cualquier objeto relevante de la escena y el foco, y *ClipEnd* mas grande que la mayor distancia. Para el mejor uso de la memoria asignada y una mejor calidad de sombras, *ClipSta* debe ser lo mas grande posible y *ClipEnd* lo menor posible. Esto

minimiza el volumen donde las sombras deben ser calculadas.

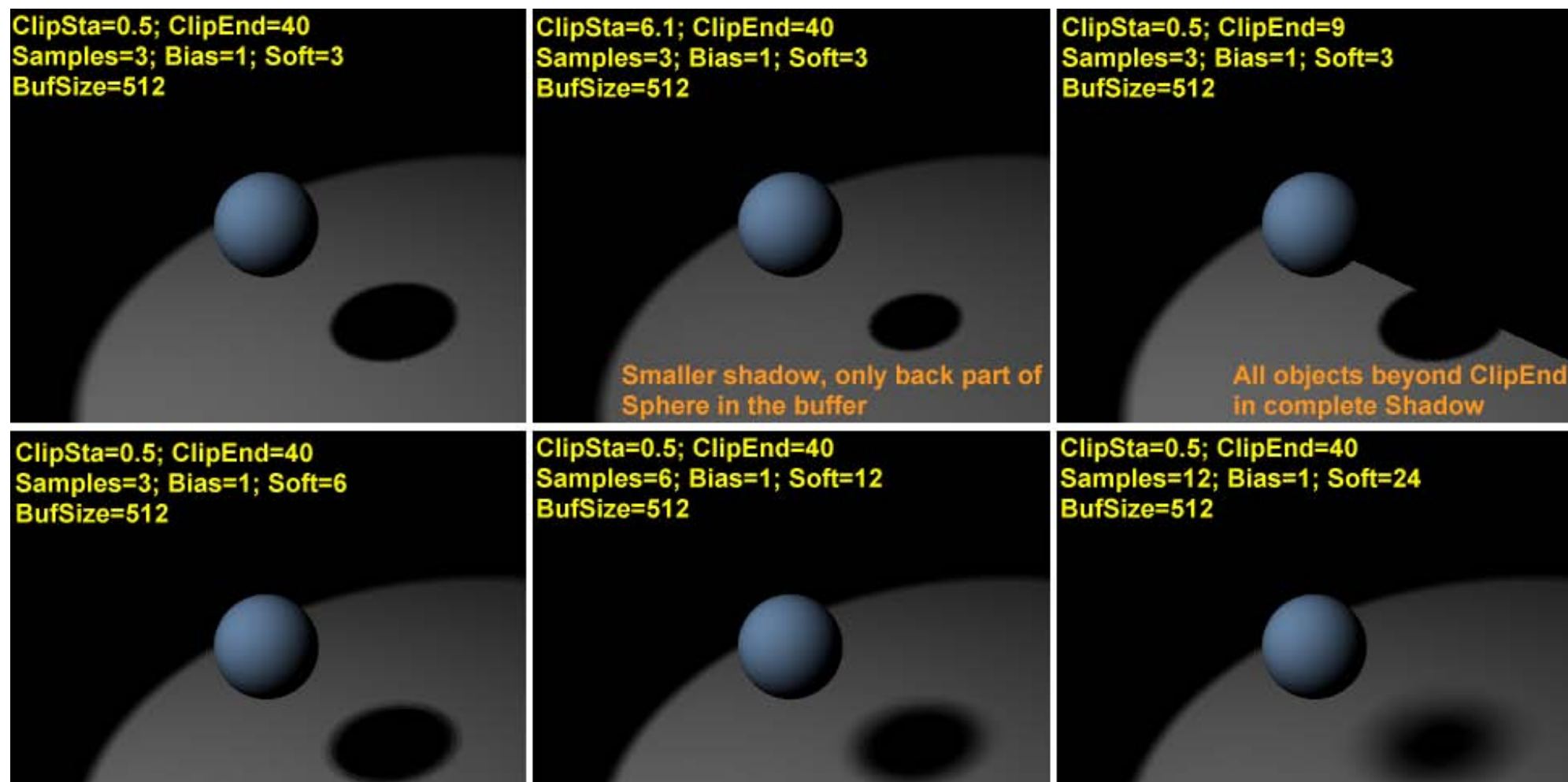
Samples - Para obtener sombras suaves el shadow buffer, una vez calculado, es renderizado por medio de su propio algoritmo de anti-aliasing que trabaja por promedio de valores de sombra sobre un cuadrado de un lado de un número dado de píxeles. Samples es el número de píxeles. Este es 3 por defecto, o sea un cuadrado de 3X3. Valores mayores dan un mejor anti-aliasing y un mayor tiempo de cálculo.

Bias - El bias es usado en el cálculo de las sombras, de nuevo mientras mayor es, mejor el resultado, y más lento.

Soft - Controla la suavidad del límite de la sombra. A mayor valor, más suave y extendida es el borde. Comúnmente este debe tener asignado un valor en el rango entre el valor del botón numérico *Sample* y el doble de este valor.

Halo step - Activa el paso del muestreo de halo para sombras volumétricas. Esto será explicado en [Section 12.5](#).

Figure 12-17. Ejemplos de luz Spot con sombra.





Note Para que las sombras sean renderizadas, deben ser habilitadas a nivel *global* . Esto significa que el boton *Shadow* en el Panel *Render* en el Contexto de botones de escena debe activarse.

[Prev](#)[Sombras con el trazador de rayos](#)[Home](#)[Up](#)[Next](#)[Luz Volumétrica](#)

12.5. Luz Volumétrica

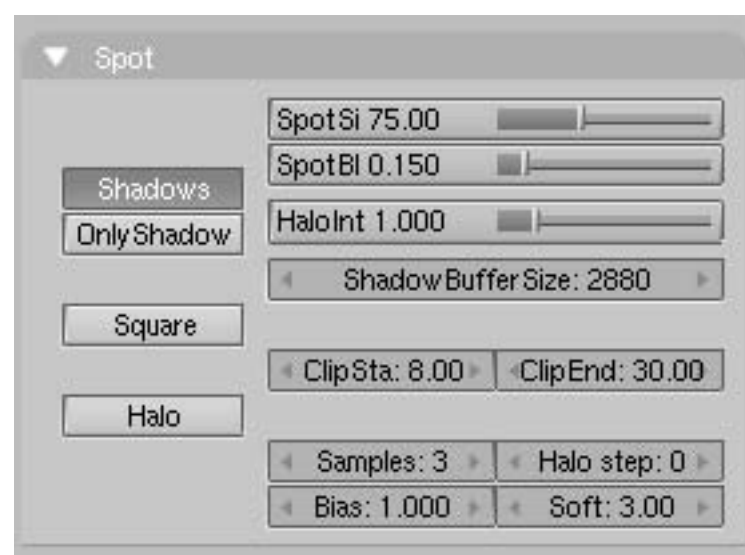
Aplicable a Blender v2.31

La luz volumétrica es el efecto que se puede ver en el aire brumoso, cuando los rayos de luz se hacen visibles debido a la dispersión que ocurre debido a niebla, bruma, partículas de polvo etc.

Si se usa cuidadosamente puede añadir mucho realismo a una escena... o echarla a perder.

En Blender la luz volumétrica sólo puede ser generada para lámparas tipo Spot, una vez que el botón *Halo* en el panel *Spot* es presionado ([Figure 12-18](#)).

Figure 12-18. Botón halo de la lámpara tipo Spot.



Si la escena de prueba mostrada en la [Figure 12-19](#) es creada y el botón halo presionado, la imagen renderizada será como la [Figure 12-20](#).

Figure 12-19. Ajustes de la lámpara tipo Spot

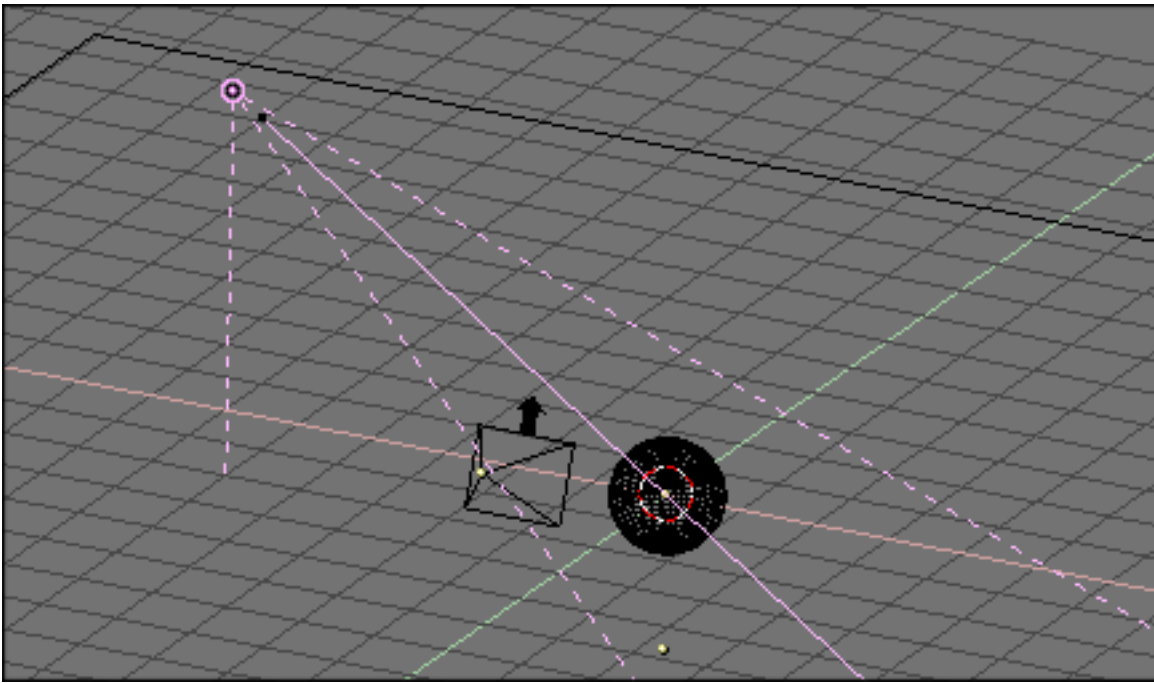
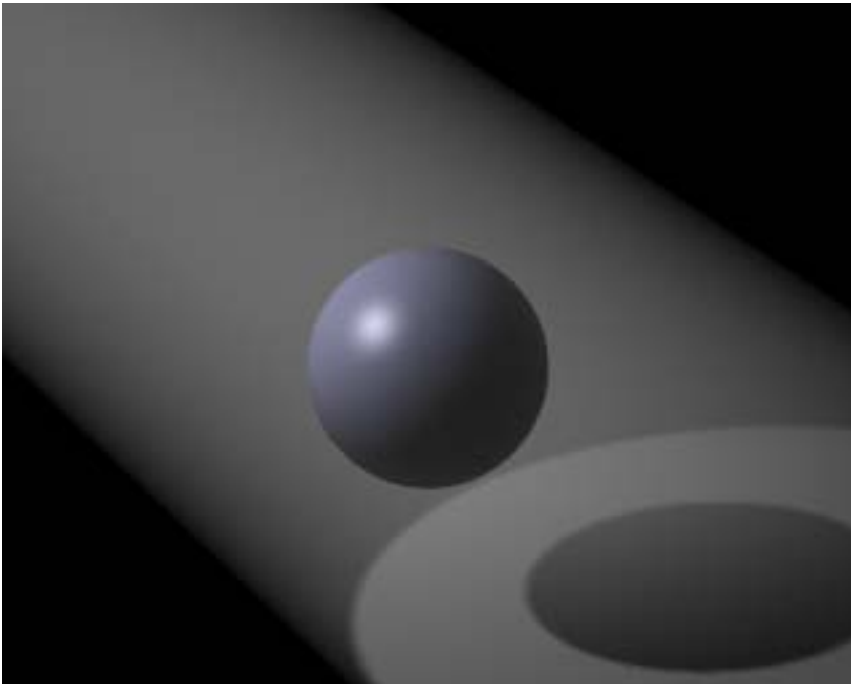


Figure 12-20. Renderizado con halo.



El efecto de luz volumétrica es más bien fuerte. La intensidad del Halo puede ser ajustada con el deslizador *HaloInt* ([Figure 12-21](#)). Valores bajos producen halos débiles.

Figure 12-21. Deslizador de intensidad de halo.



El resultado es interesante. Se obtiene luz volumétrica, pero carecemos de sombra volumétrica! El halo pasa a través de la esfera y, sin embargo, se proyecta sombra. Esto se debe a que el halo tiene lugar en la totalidad del cono de la lámpara Spot a menos que se le indique a Blender que haga otra cosa.

El cono debe ser muestreado para obtener sombra volumétrica, y el muestreo ocurre con una medida definida por el botón numérico *HaloStep* ([Figure 12-22](#)). El valor por defecto de 0 significa que no se hará muestreo alguno, por lo cual no hay sombra volumétrica. Un valor de 1 da un muestreo más detallado y por tanto mejores resultados, con el costo de hacer más lento el renderizado ([Figure 12-23](#)), aunque un valor mayor da resultados pobres con un renderizado más rápido ([Figure 12-24](#)).

Figure 12-22. Botón numérico Halo Step.



Figure 12-23. Halo con sombra volumétrica, Halo Step = 1

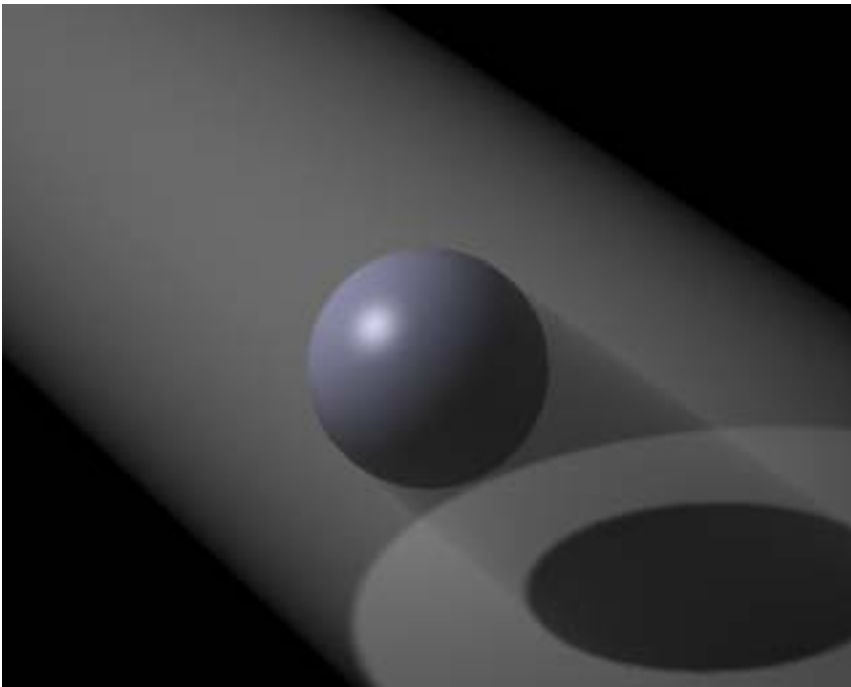
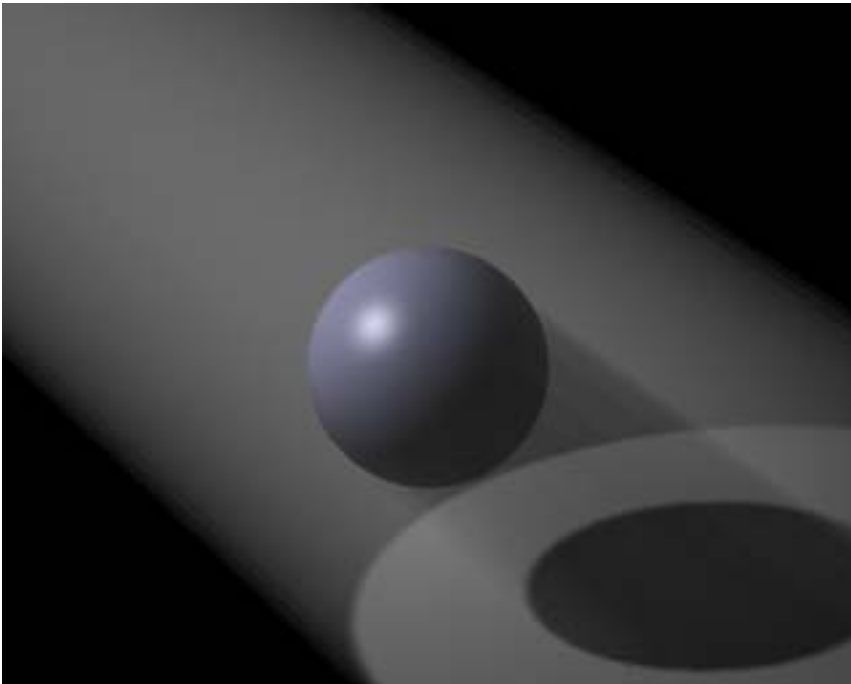


Figure 12-24. Halo con sombra volumétrica, Halo Step = 12



Tip Valores de HaloStep

Un valor de 8 corresponde, usualmente, a un buen balance entre rapidez y precisión.

[Prev](#)

Sombras con el buffer

[Home](#)

[Up](#)

[Next](#)

Ajustando la iluminación

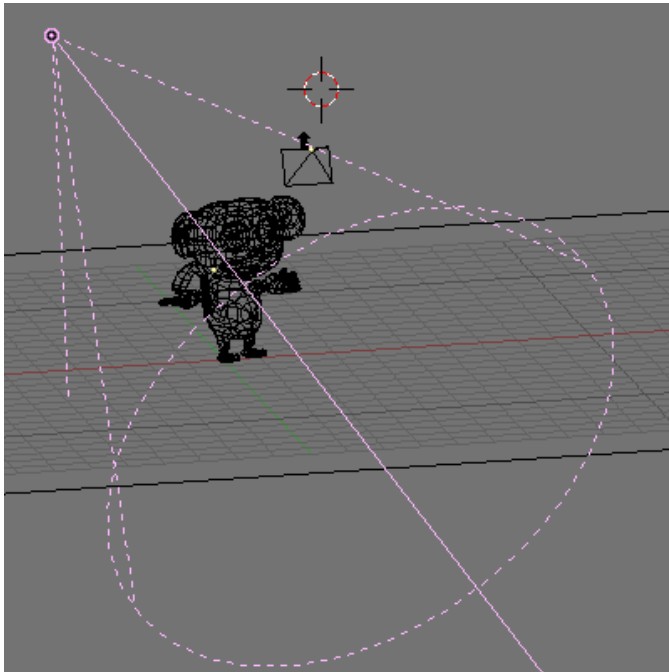
12.6. Ajustando la iluminación

Relativo a Blender v2.31

Ok, ahora que controlas lo básico, realmente podemos hablar de iluminación. Trabajaremos con un solo ejemplo, más complejo que una simple configuración 'esfera sobre plano', para ver lo que podemos conseguir en iluminación realista con Blender.

Recurriremos a la configuración en la [Figure 12-25](#). El simio de la figura es Cornelius, el hermanito de Suzanne. Tiene un material ligeramente pulido de color marrón. ($R=0.8$, $G=0.704$, $B=0.584$, $Ref=0.7$, $Spec=0.444$, $Hard=10$ - Sí, no muy mono, pero estamos hablando de luces, no de materiales!) y se apoya sobre un plano azul ($R=0.275$, $G=0.5$, $B=1.0$, $Ref=0.8$, $Spec=0.5$, $Hard=50$). Por ahora están iluminados por un simple Spot ($Energy=1.0$, $R=G=B=1.0$, $SpotSi=45.0$, $SpotBl=0.15$, $ClipSta=0.1$, $ClipEnd=100$, $Samples=3$, $Soft=3$, $Bias=1.0$, $BufSize=512$).

Figure 12-25. Ajustando la iluminación de Cornelius.



Un render de Cornelius con esta configuración, con $OSA=8$ y sombras activadas, da el resultado en [Figure 12-26](#). El resultado es malo. Obtienes una sombra muy negra y poco realista de Cornelius, y la sombra que crea Cornelius sobre si mismo es inaceptable.

Figure 12-26. Configuración de un simple spot.



El primer ajuste es sobre *ClipSta* y *ClipEnd*, Si se ajustan para abarcar la escena lo más justo posible (*ClipSta*=5, *ClipEnd*=21) el resultado mejora, al menos para la sombra proyectada. La sombra de Cornelius sigue demasiado oscura ([Figure 12-27](#)).

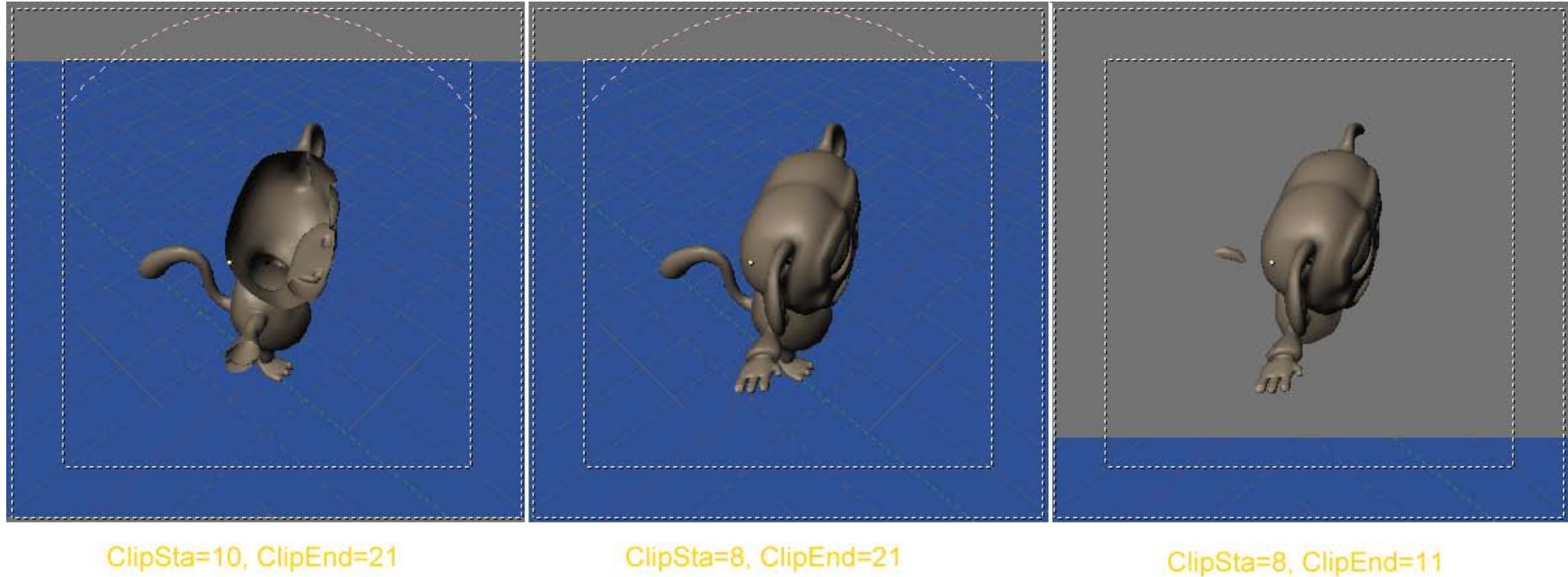
Figure 12-27. Configuración de un simple Spot con truncamiento apropiado.



Para poner buenos valores de truncamiento (límites), este es un buen truco: Cualquier objeto en Blender puede actuar como una cámara en la vista 3D. Por lo tanto puedes seleccionar la luz Spot y cambiar a una vista desde ella pulsando **CTRL-NUM0**. Lo que deberías ver en modo sombreado se muestra en [Figure 12-28](#).

Todo lo que haya más cerca de la Spot que *ClipSta* y mas lejos de la Spot que *ClipEnd* no es visible. Por lo tanto puedes ajustar estos valores con precisión verificando que todos los objetos que arrojan sombras sean visibles.

Figure 12-28. Ajustes de truncamiento de la luz Spot. Izquierda: ClipSta demasiado alto; Centro: Bien; Derecha: ClipEnd demasiado bajo.



De lo que todavía carece, es del fenómeno físico de la difusión. Un cuerpo iluminado emite luz (reflejada), por lo tanto las sombras no son totalmente negras, porque algo de luz escapa de las zonas cercanas iluminadas.

Esta difusión de la luz es tenida en cuenta correctamente por un trazador de rayos (Ray Tracer), y también por Blender, Atraves del motor de radiosidad. Pero hay configuraciones que pueden simular este fenómeno de forma aceptable.

Los analizaremos desde el más simple, hasta el más complejo.

12.6.1. Tres puntos de luz

La configuración de tres puntos de luz es un clásico y un modo muy simple de obtener una escena con iluminación suave. Nuestra luz Spot es la principal, o luz *Clave*, de la escena, la que arroja sombras. Añadiremos dos luces más para simular la difusión.

La siguiente luz que necesitamos se llama *Luz trasera*. está colocada detrás de Cornelius ([Figure 12-29](#)). Esta ilumina el lado oculto de nuestro personaje, y nos permite separar el primer plano de nuestra imagen, de el fondo, añadiendo en general una sensación de profundidad. Normalmente la luz trasera es igual de fuerte que la luz Clave, si no lo es más. Aquí hemos usado una Lamp con energía 1 ([Figure 12-30](#)).

Figure 12-29. Configuración de luz trasera.

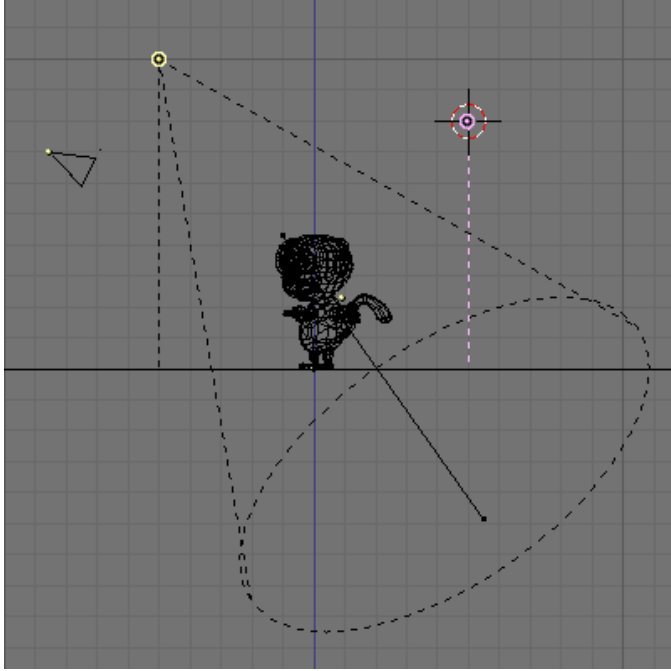


Figure 12-30. Solo luz clave (Izquierda), Solo luz trasera (centro), y ambas (derecha).



Key light



Back Light



Back + Key Light

El resultado es ya mucho mejor. Finalmente, la tercera luz es la de *Relleno*. La finalidad de la luz de relleno es iluminar las sombras en frente de Cornelius. Colocaremos la luz de relleno exactamente donde está la cámara, con una energía más baja que la más baja de las otras dos luces. ([Figure 12-31](#)). Para este ejemplo se eligió *Energy*=0.75 ([Figure 12-32](#)).

Figure 12-31. Configuración de la luz de relleno.

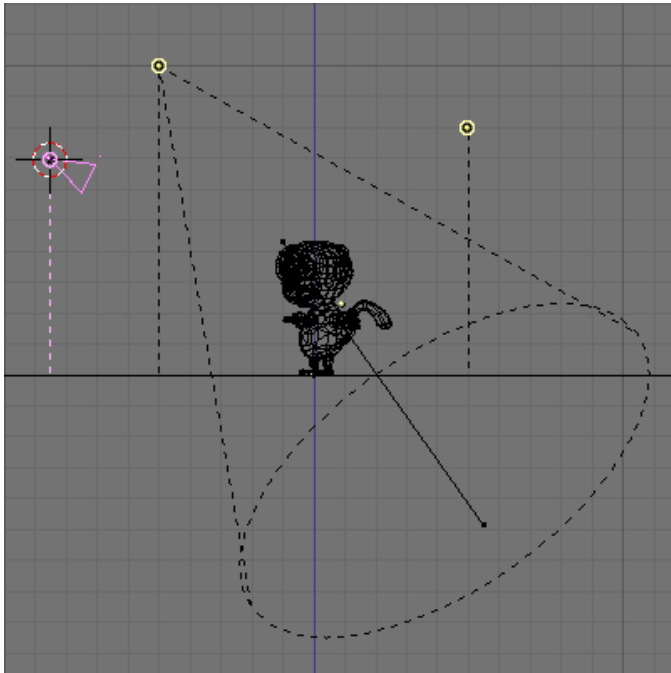


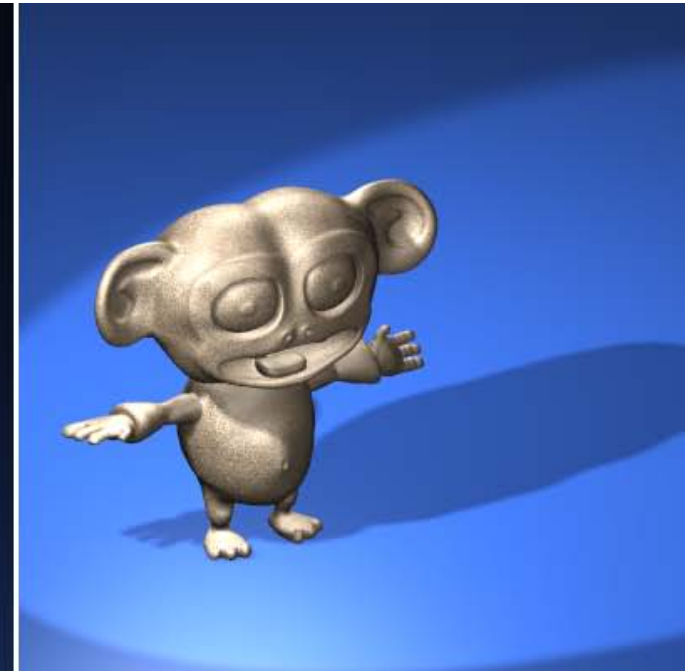
Figure 12-32. Solo luz Clave y Trasera (izquierda), Solo luz de Relleno (centro), y las tres (derecha).



Back + Key Light



Fill Light



Key + Back + Fill Light

La luz de Relleno hace visibles partes del modelo que estaban completamente en la oscuridad con la luz Clave y Trasera unicamente.

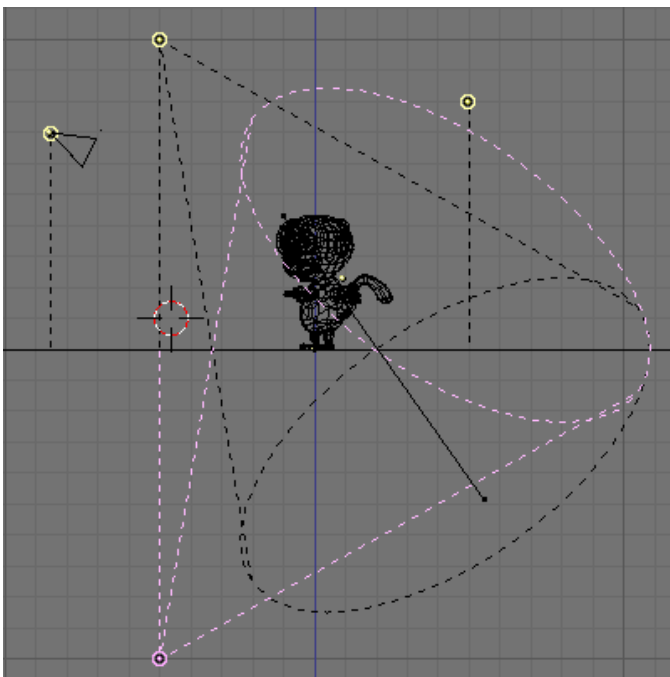
Tip Rebote de color

La configuración "Tres puntos de luz" puede ser mejorada aún más con la incorporación de una cuarta luz, especialmente cuando hay un suelo de color brillante, como en este caso.

Si hay un suelo de color brillante, nuestros ojos esperan que este reparta algo de la luz al rededor, y algo de esa luz incida en el modelo.

Para simular este efecto, colocaremos una Spot de forma exactamente especular con respecto al suelo. Esto significa que - Si el suelo está horizontal en $z=0$, como en nuestro ejemplo, y la luz Clave está en el punto $(x=-5, y=-5, z=10)$, entonces la luz difusa del suelo debe colocarse en $(x=-5, y=-5, z=-10)$, apuntando hacia arriba ([Figure 12-33](#)).

Figure 12-33. Configuración de la luz difusa del suelo.



La energía de esta luz debe ser más baja que la de la luz Clave (Aquí es de 0.8) y su color debe igualar el color del suelo (aquí $R=0.25$, $G=0.5$, $B=1.0$). El resultado se muestra en [Figure 12-34](#).

Figure 12-34. Configuración de cuatro puntos de luz.



Por favor date cuenta de que usamos una luz Spot y no una Lamp, así que sería completamente bloqueada por el suelo (a causa de la sombra) si no la hicieramos Shadeless (sin sombra) pulsando el botón apropiado.

De hecho podríamos haber usado una Lamp, pero si el suelo es pulido la luz que sale es más reflejada que difusa. La luz reflejada es físicamente un cono empezando en la fuente especular.

Puedes mejorar el efecto haciendo que el Spot arroje sombras y ajustando el valor de *ClipStart* suficientemente alto como para que el suelo no cree sombras, o haciendo que afecte solo a su Capa (Layer) y poniendo el suelo en otra Capa.

12.6.2. Tres puntos de luz - Exteriores

Al usar una luz Spot como luz Clave, el metodo anterior está tristemente relegado a escenas de interior o, como mucho, escenas exteriores nocturnas. Esto se debe a que la luz clave tiene una distancia finita, sus rayos se degradan y el suelo no es iluminado uniformemente.

Si estuviéramos al aire libre en un día soleado y despejado, todo el suelo estaría uniformemente iluminado y los objetos arrojarían sombras.

Para obtener una iluminación uniforme, sobre todo el suelo, una luz de tipo Sun es lo mejor. Y si añadimos una luz tipo Hemi para simular la luz que viene desde todo el cielo (como en [Figure 12-8](#)) podemos obtener una bonita iluminación exterior... pero no tendremos sombras!

La configuracion de la luz Clave (el sol, $R=1.0$, $G=0.95$, $B=0.9$, $Energy=1.0$) y las luces Relleno/Trasera (ambas representadas por la Hemi, $R=0.8$, $G=0.9$, $B=1.0$, $Energy=0.4$) se muestran en [Figure 12-35](#) y el correspondiente render en [Figure 12-36](#)

Figure 12-35. Configuración de luces Sun y Hemi para exteriores.

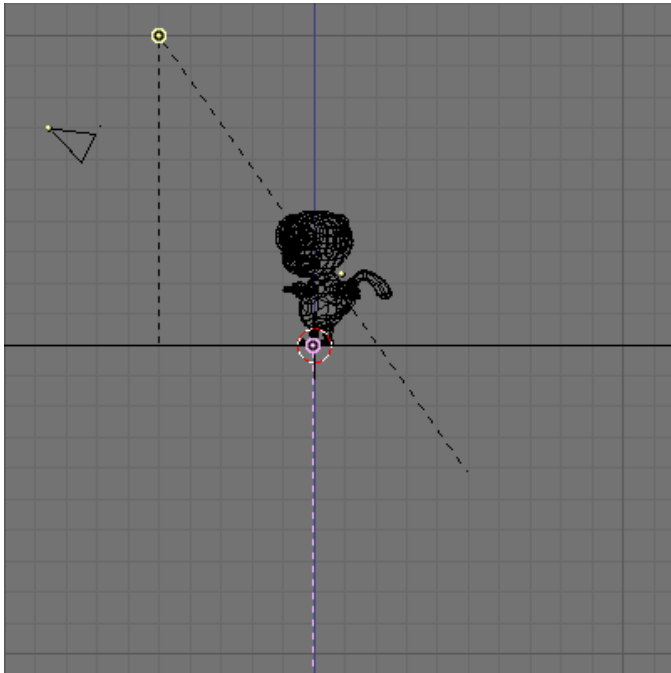
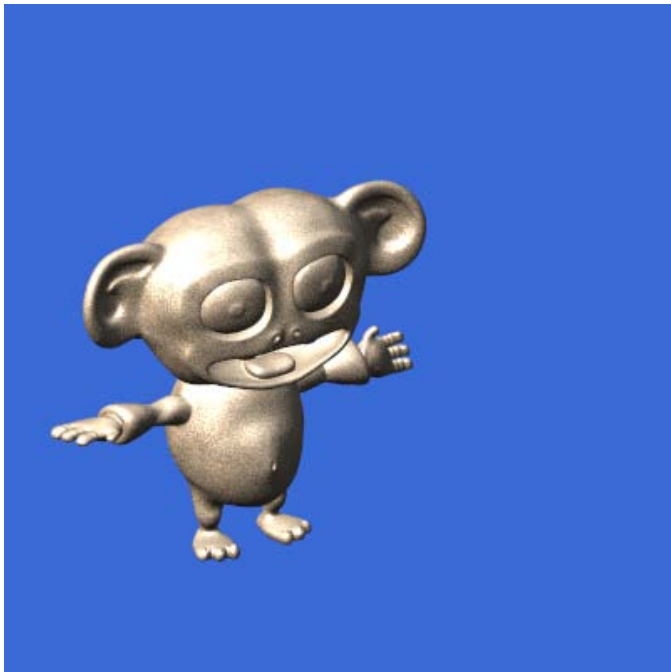


Figure 12-36. Render de luces Sun y Hemi para exteriores.



La falta de sombras hace que Cornelius parezca flotar en el espacio. Para obtener sombra, colocar una Spot en el mismo lugar que que el sol y con la misma orientación. Haciendo que el Spot sea de solo sombra (Shadow Only Spot) con el botón apropiado. Si la energía es disminuida a 0.9 y mantenemos el resto de parametros del anterior ejemplo (*BufSize=512, Samples=3 Soft=3 Bias=1 ClipSta=5, ClipEnd=21*) el

resultado es el de [Figure 12-37](#) (centro).

Figure 12-37. Render de exteriores.



La sombra es un poco cuadriculada porque Cornelius tiene muchos detalles finos, el *BufSize* es demasiado pequeño, y el valor de *Sample* es demasiado bajo para tenerlo en cuenta correctamente. Si *BufSize* es elevado a 2560, *Samples* a 6 y *Bias* a 3.0 el resultado es [Figure 12-37](#) (derecha). Mucho más liso.

12.6.3. Luz tipo Area

El concepto de luz proveniente de un punto es una aproximación. ninguna luz en el mundo real está carente de dimensión. Toda luz proviene de superficies, no de puntos.

Esto posee un par de implicaciones interesantes, principalmente sobre las sombras:

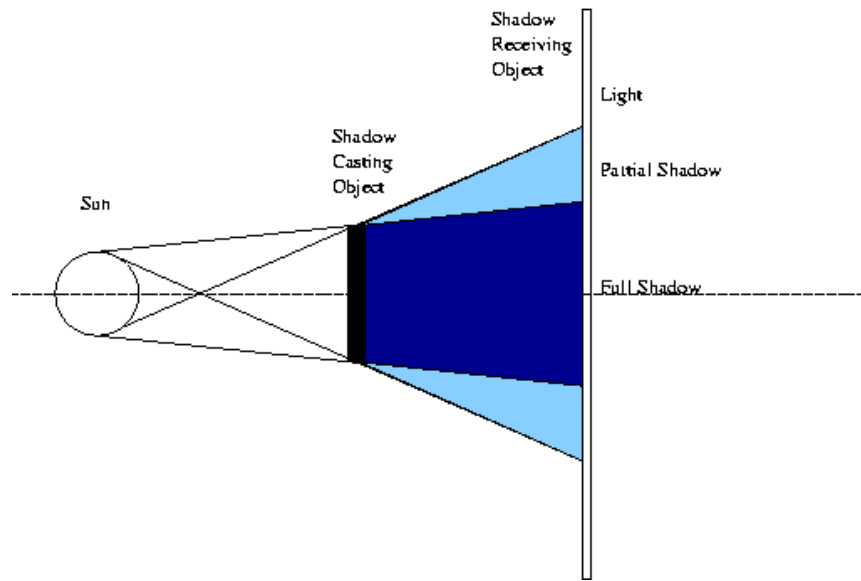
- Las sombras nítidas no existen: tienen bordes borrosos.
- lo difuminado del borde de la sombra depende de la posición y tamaño relativo de la luz, del objeto que produce las sombras y del objeto que las recibe.

La primera de estas situaciones es aproximada con el ajuste 'Soft' de la luz Spot, pero la segunda no. Para entender claramente este punto imagina un poste fino y alto en el centro de un plano llano iluminado por el sol.

El sol no es un punto, tiene una dimensión y, para nosotros los terrestres, esta es medio grado de ancho. Si miras la sombra, te darás cuenta de que es muy nítida en la base del poste y se va haciendo más difusa conforme vas hacia la sombra de la punta. Si el poste es suficientemente alto y delgado, la sombra se desvanecerá.

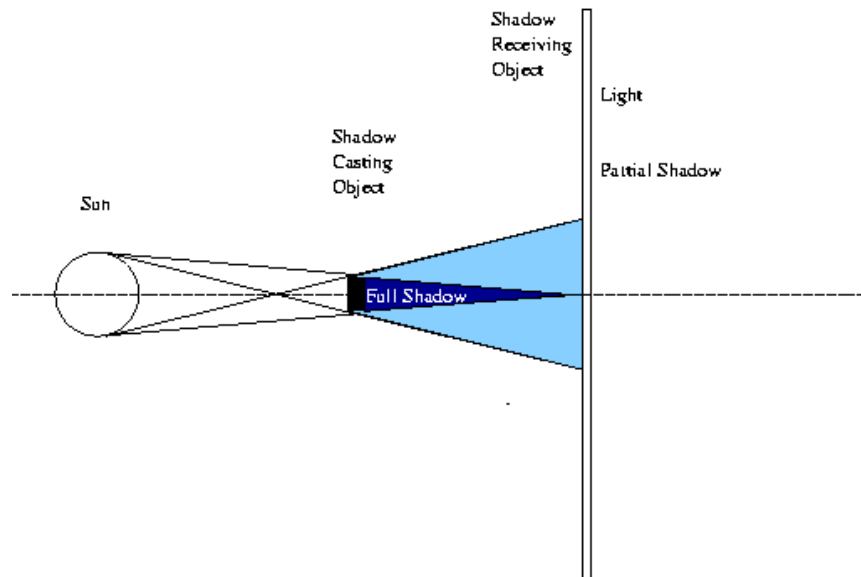
Para entender mejor este concepto, echale un ojo a [Figure 12-38](#). El sol emite luz, en el medio el objeto obstruye completamente los rayos del sol solo en la zona azul oscuro. para cada punto en la zona azul claro el sol es parcialmente visible, por lo tanto cada una de estas zonas es parcialmente iluminada.

Figure 12-38. La luz de Area y su sombra.



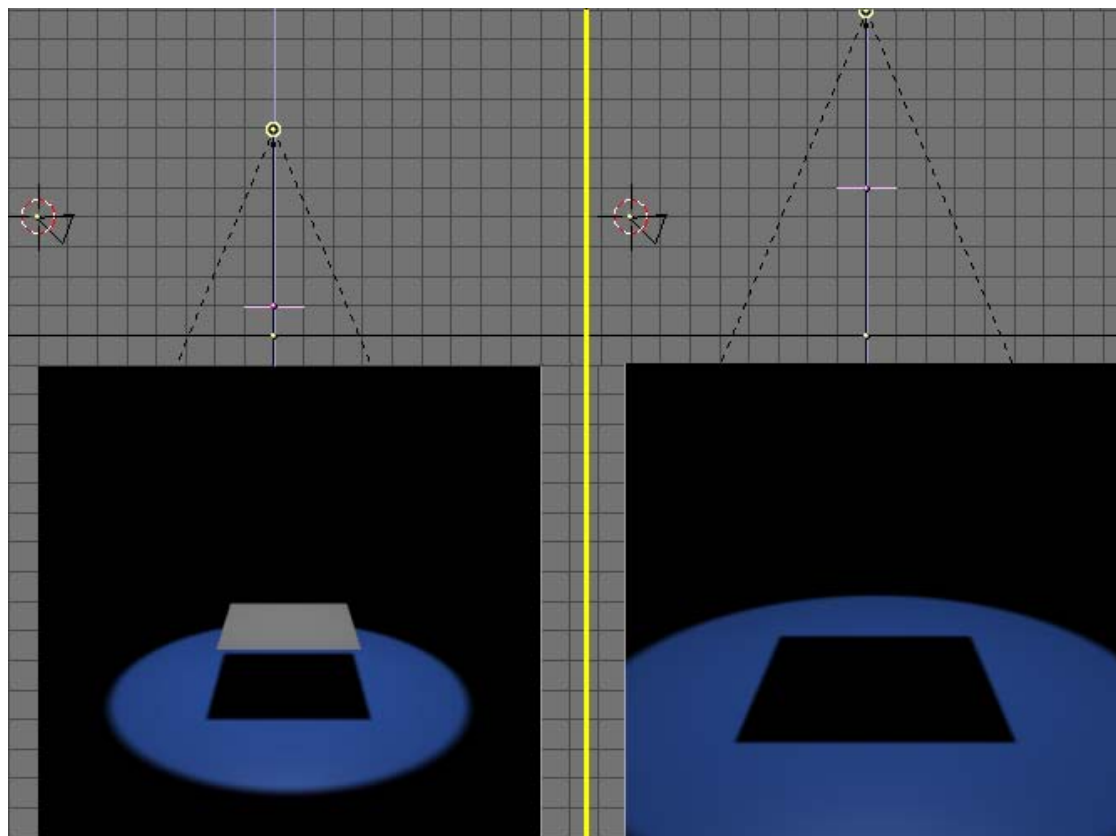
La zona azul claro es una zona parcialmente sombreada, donde la iluminación va desde luz total hasta sombra total. Es además evidente, en [Figure 12-38](#) que esta zona de transición es más pequeña cerca del objeto que produce la sombra y se hace más grande conforme se aleja de este. Además, si el objeto que produce la sombra es más pequeño que el objeto que emite la luz (y si el objeto que emite la luz es el sol como es el caso) hay una distancia más allá de la cual solamente permanece la sombra parcial [Figure 12-39](#).

Figure 12-39. La luz de Area y su sombra 2



En Blender, si colocamos un solo Spot a una distancia fija desde un primer plano, y miramos la sombra producida en un segundo plano, nos daremos cuenta de que la sombra se hace más grande conforme se aleja, pero no más difusa ([Figure 12-40](#))

Figure 12-40. La luz Spot y su sombra



Para simular una luz de tipo area en Blender, podemos usar varios Spots, como si muestreáramos el area luminosa con un numero discreto de puntos de luz.

Esto se puede conseguir colocando varios Spots a mano, o usando la función DupliVert de Blender ([Section 22.2](#)), la cual es más eficiente.

Añade una malla tipo rejilla (Grid) 4x4, donde está el Spot, asegurate de que las normales estén apuntando hacia abajo haciendo que Blender las muestre e invirtiéndolas (si fuese necesario), como se explica en [Section 6.2.2](#) ([Figure 12-41](#)). Emparenta el Spot a la rejilla, selecciona la rejilla y en el panel *Anim Settings* (**F7**) pulsa *DupliVert* y *Rot*. *Rot* no es estrictamente necesario pero te ayudará a colocar la luz de area más tarde. Tendrás un bloque de Spots como en [Figure 12-42](#).

Figure 12-41. Configuración de la rejilla

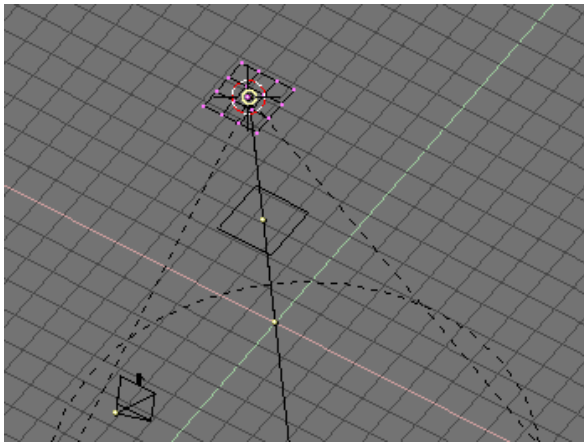
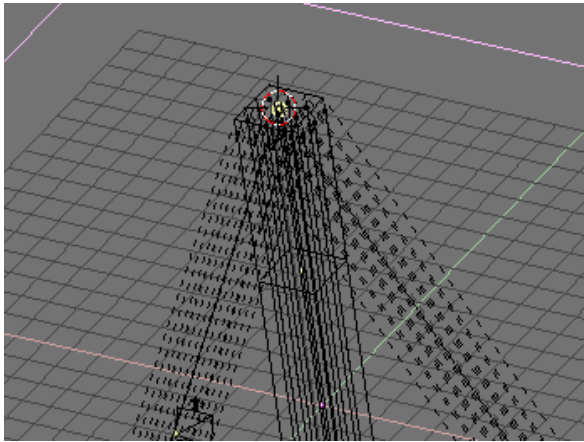


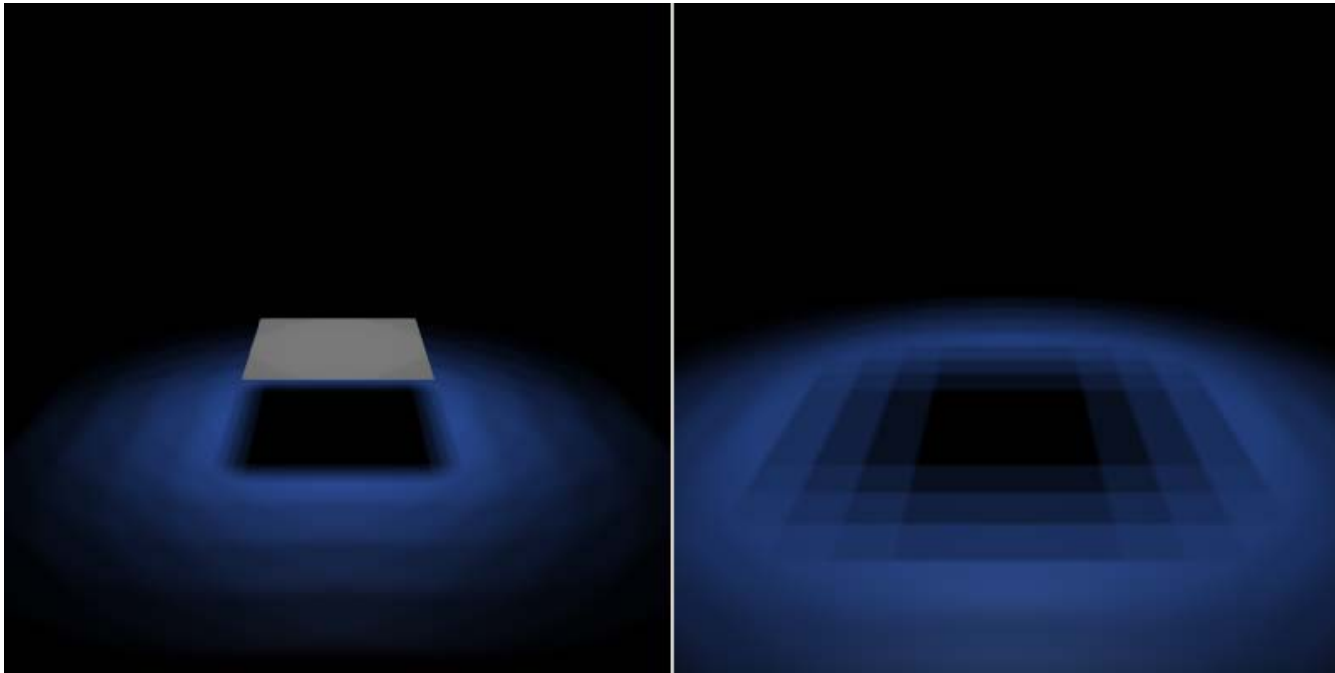
Figure 12-42. Luces Spot y sus DupliVerts



Entonces disminuye la energía del Spot. Si para un solo Spot usabas una cierta cantidad de energía, ahora debes dividir esa energía entre todos los duplicados. Aquí tenemos 16 Spots, así que a cada uno se le debe asignar 1/16 de esa energía (esto es *Energy*=0.0625).

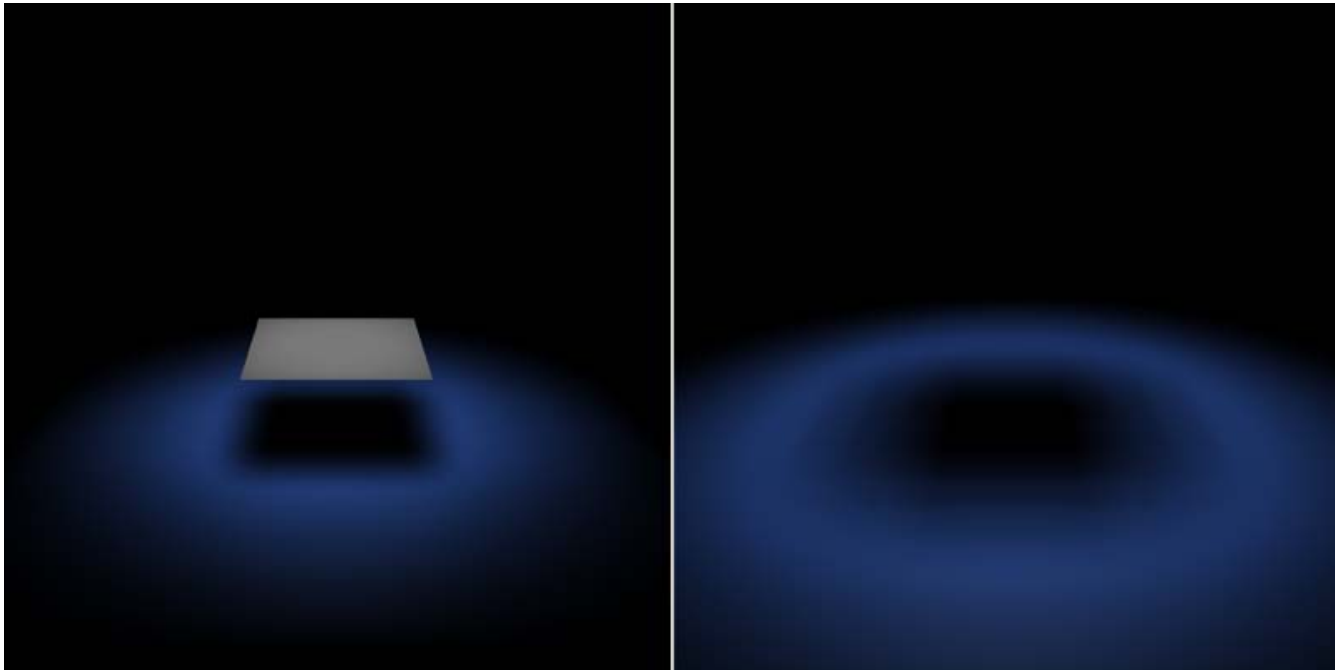
Los dos mismos renders de arriba, con esta nueva luz de área trucada dará este resultado [Figure 12-43](#). El resultado está lejos de lo que se esperaba, porque el sampleo de las luces Spot para ese área es demasiado grueso. Por otro lado, un mejor sampleo implicaría un mayor número de Spots y por ello unos tiempos de render inaceptablemente altos.

Figure 12-43. Luz de Área trucada con multiples Spots.



Pueden obtenerse resultados mucho mejores, suavizando los Spots, con valores como estos *SpotBl*=0.45, *Sample*=12, *Soft*=24 y *Bias*=1.5 ([Figure 12-44](#))

Figure 12-44. Luz de Area trucada con multiples Spots suavizados.



Finalmente, [Figure 12-45](#) muestra que ha pasado con Cornelius una vez la luz Clave ha sido sustituida con 65 Spots duplicados de Energía=0.0154 en un patrón circular. Fíjate como la sombra va suavemente desde nítida en el pie, hasta suave y difusa conforme se aleja de él. Este es el comportamiento físico correcto.

Figure 12-45. Cornelius bajo una luz de Area.



12.6.4. Iluminación Global (y Sombreado Global)

Las técnicas antedichas trabajan bien cuando hay una sola, o por lo menos un número finito de luces, arrojando sombras distintas.

Las únicas excepciones son: la configuración para exteriores donde la luz Hemi simula la luz emitida por el cielo, y la luz de Area donde multiples Spots simulan una fuente de luz de dimensiones finitas.

la primera de estas dos está muy cerca de una buena iluminacion exterior, excepto porque la luz Hemi no arroja sombras y por lo tanto no obtiene resultados realistas.

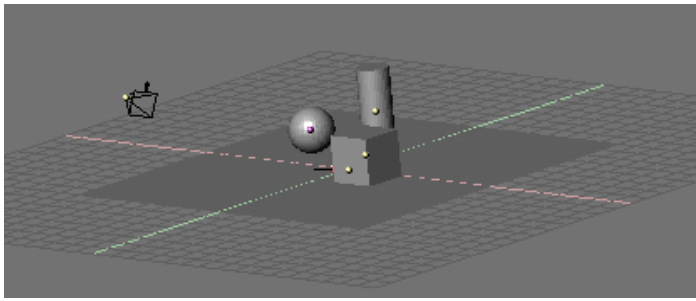
Para obtener una configuración de exteriores realmente buena, especialmente para días nublados, necesitas que la luz provenga de todo el cielo y que cree sombras!

Esto se puede obtener aplicando una técnica muy similar a la usada en la configuración de luz tipo Area, pero usando media esfera como malla padre. A esto se le llama normalmente "Iluminación Global".

Puedes usar una UVsphere o una IcoSphere, la última tiene los vértices unifomemente distribuidos, mientras la primera tiene una gran concentración de vértices en los polos. Por lo tanto una IcoSphere consigue una iluminación más 'uniforme', todos los puntos del cielo irradian a la misma intensidad; en cambio, una UVsphere irradia mucha más luz desde su(s) polo(s). Personalmente recomiendo la IcoSphere.

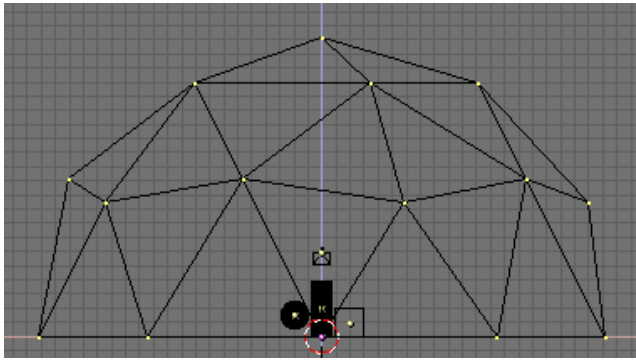
Preparemos una configuracion, compuesta de un plano y algunos solidos, como en [Figure 12-46](#). Usaremos figuras simples para apreciar mejor los resultados.

Figure 12-46. Escena de Iluminación Global.



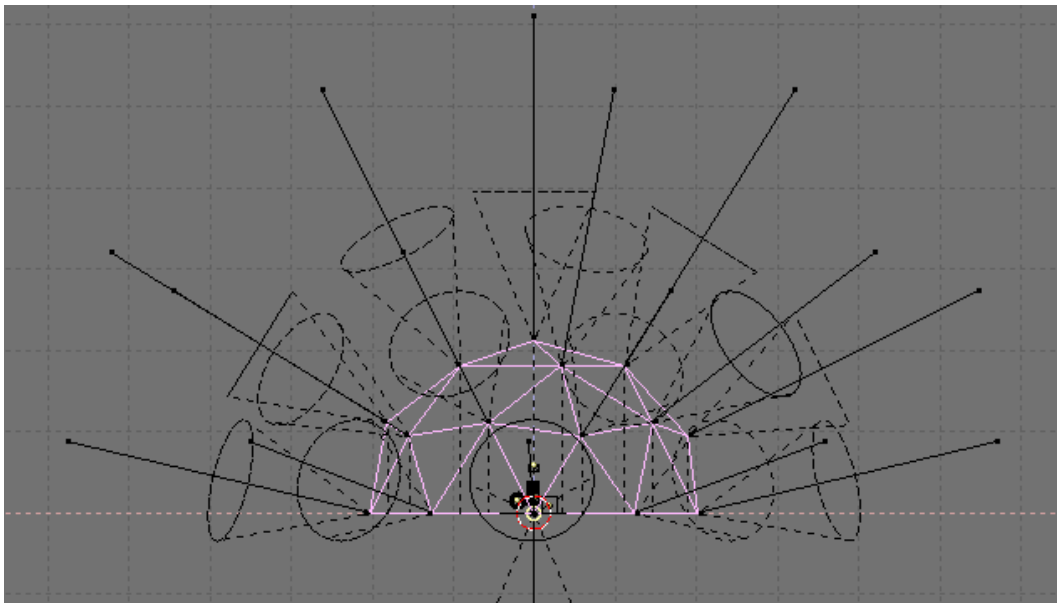
Cambia a la vista superior y añade una IcoSphere, con un nivel 2 de subdivisión es suficiente normalmente, un nivel 3 consigue resultados aún más suaves. Escala la IcoSphere de modo que contenga la escena entera. Cambia a vista frontal y, en modo edición, borra la mitad inferior de la IcoSphere ([Figure 12-47](#)). Esta será nuestra "bóveda celeste" donde los spots serán emparentados y DupliVerteados.

Figure 12-47. Bóveda celeste.



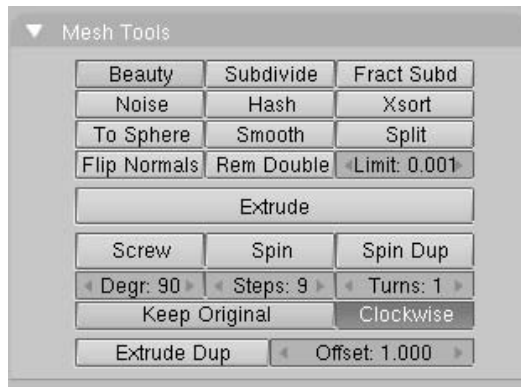
De nuevo en la vista superior añade una luz Spot, emparentala a la media IcoSphere (**CTRL-P**) y pulsa los botones *DupliVert* y *Rot* exactamente como en el ejemplo anterior. El resultado, en la vista frontal, es el de [Figure 12-48](#).

Figure 12-48. Bóveda celeste con Spots duplicados.



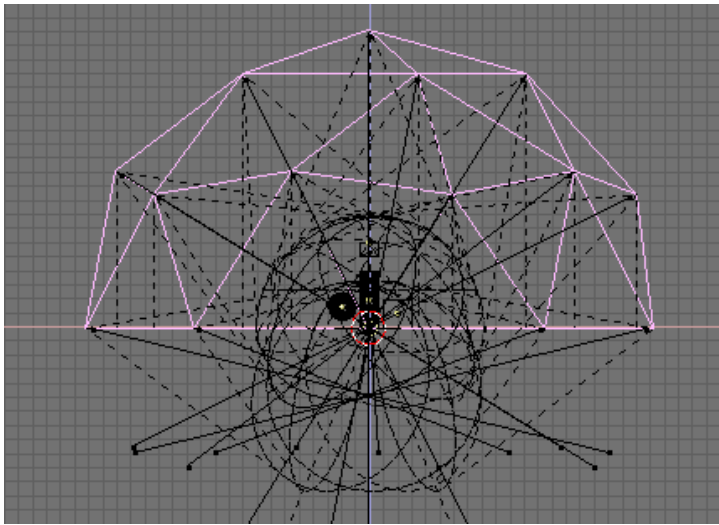
Esto no es lo que queremos, ya que todos los Spots apuntan hacia afuera y la escena no es iluminada. Esto ocurre porque las normales de la IcoSphere apuntan hacia afuera. Es posible invertir la dirección seleccionando todos los vértices en modo edición y pulsando el botón *Flip Normals* en el panel *Mesh Tools* que esta en la ventana de edición (**F9**) ([Figure 12-49](#)).

Figure 12-49. Invirtiendo las normales.



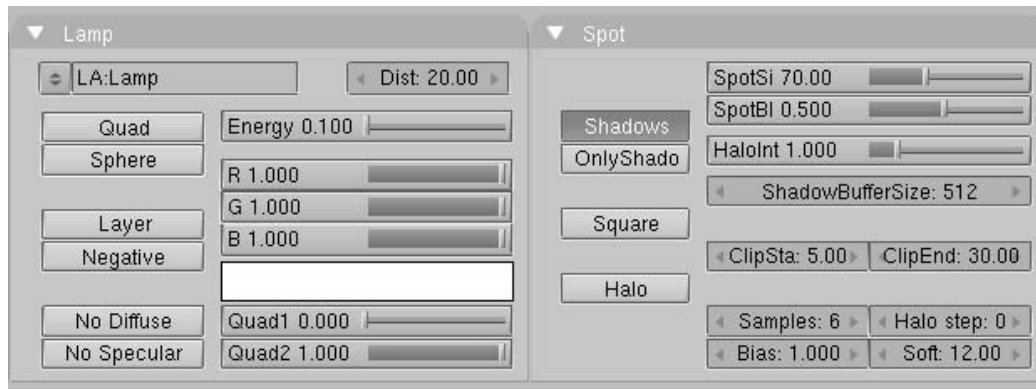
Esto conduce a la nueva configuración en [Figure 12-50](#).

Figure 12-50. Bóveda celeste correcta y luces Spot DupliVerteadas.



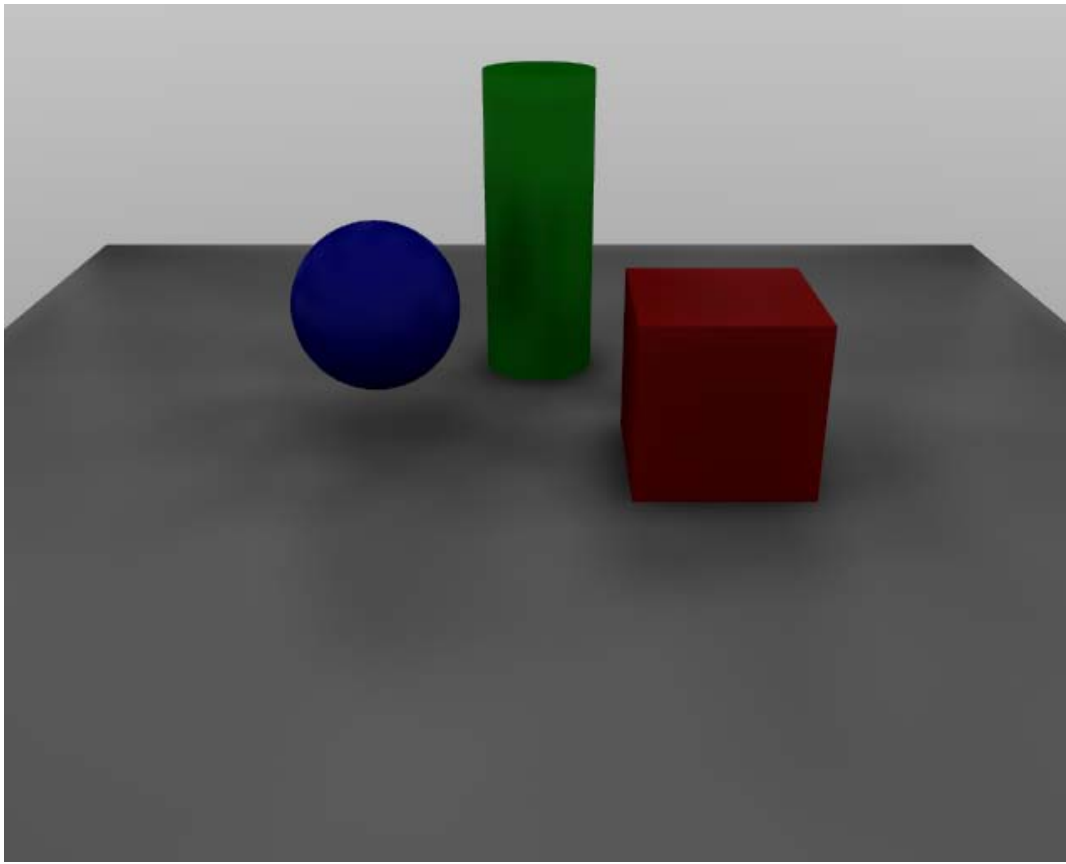
Para obtener buenos resultados selecciona la luz Spot original y cambia sus parametros a un ángulo ancho con contorno difuso (*SpotSi*=70.0; *SpotBl*=0.5); con los valores convenientes de *ClipSta* y *ClipEnd* ; en este caso 5 y 30, respectivamente, en cualquier caso usa los valores necesarios para envolver toda la escena; incrementa Samples a 6 y Softness a 12. Decrementa la energía a 0.1; recuerda que estás usando muchos spots, así que cada uno de ellos debe ser débil. ([Figure 12-51](#)).

Figure 12-51. Configuración de luz Spot.



Ahora puedes renderizar. Si se le dan algunos materiales y un entorno, el resultado debe ser algo como [Figure 12-52](#). fijate en las sombras suaves y en la iluminación 'omnidireccional'. Todavía se pueden conseguir mejores resultados con una IcoSphere con nivel 3 de subdivisión.

Figure 12-52. Configuración de luz Spot.

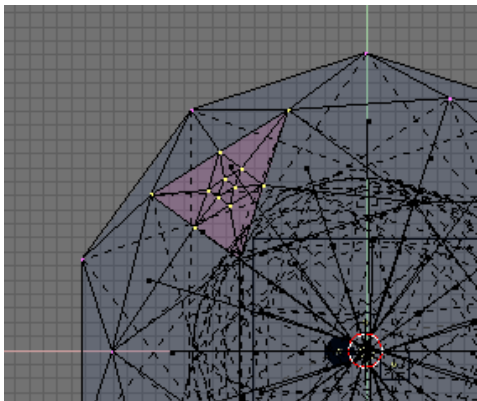


Esta técnica de iluminación global sustituye efectivamente, a muy alto costo de computación, la Hemi de la anterior configuración para exteriores.

Es posible añadir un componente de luz direccional simulando el sol mediante un solo Spot o una luz de Area.

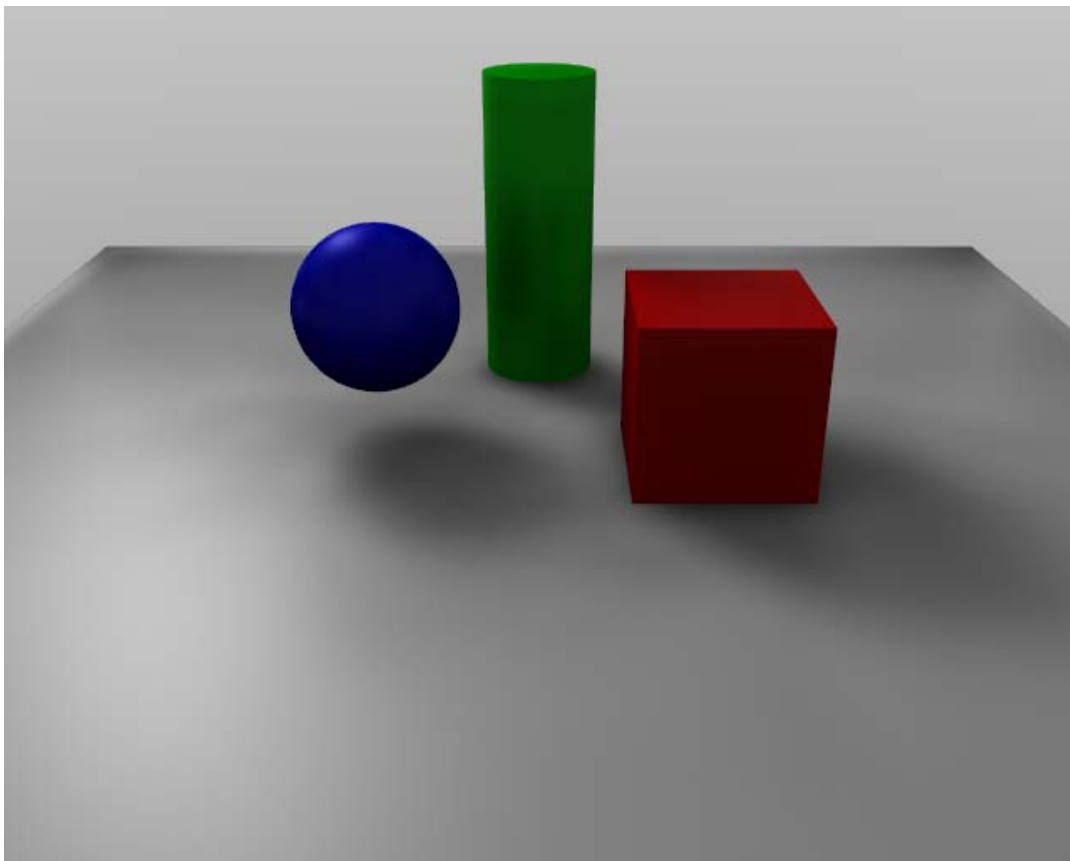
Una posible alternativa es hacer la IcoSphere menos uniforme subdividiendo una de sus caras un número de veces, como se aprecia en una de las caras posteriores en [Figure 12-53](#). Esto se hace seleccionando una cara y pulsando el botón *Subdivide*, de nuevo en el panel *Mesh Tools* de la ventana de Edición (**F9**). Entonces deseleccionamos todo , re-seleccionamos únicamente la pequeña cara interior y subdividimos de nuevo, etc...

Figure 12-53. Haciendo más densos los Spots en un area.



El resultado es una luz direccional muy suave junto con una bóveda celeste de iluminación global o, más sencillo, una bóveda celeste anisotrópica ([Figure 12-54](#)). Es muy bueno para condiciones nubladas, pero no tan bueno para días soleados y despejados. Para días realmente despejados, es mejor dejar la bóveda celeste separada de la luz del sol, permitiendonos usar diferentes colores para cada una.

Figure 12-54. Render de la bóveda celeste Anisotrópica.



Chapter 13. El Mundo y el Universo

Blender proporciona una gran cantidad de opciones muy interesantes para completar las generaciones de imágenes (renderings) añadiendo un bello fondo y unos cuantos efectos 'de profundidad' interesantes. Todo esto está disponible mediante el Shading Context (Contexto de sombreado)(F5) y el subcontexto World Buttons (Botones de Mundo) () mostrados en [Figure 13-1](#). Por defecto se presenta un mundo uniforme y muy sencillo, aunque se puede editar este Mundo o añadir uno nuevo.

Figure 13-1. Botones del mundo

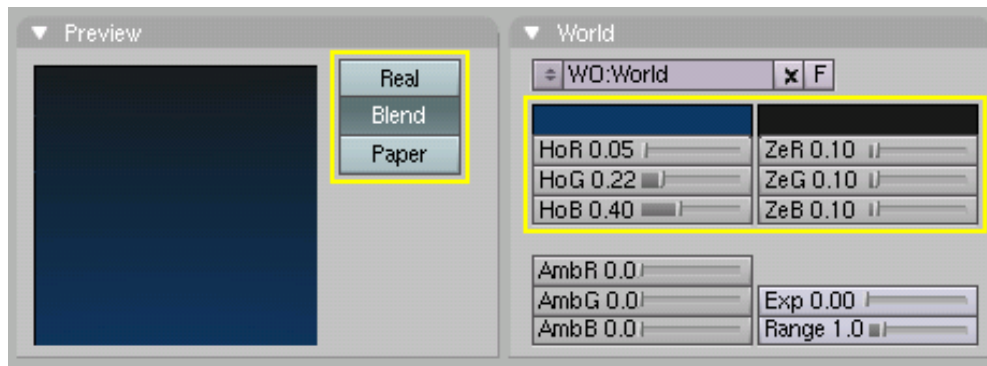


13.1. El Fondo del Mundo

Importante en Blender v2.31

La forma más fácil para usar los "World Buttons" es dar a las imágenes un fondo bonito hecho con gradientes para imágenes. Los botones en el *World* Panel ([Figure 13-2](#)) permiten definir un color en el horizonte (botones *HoR*, *HoG*, *HoB*) y en el cenit (botones *ZeR*, *ZeG*, *ZeB*).

Figure 13-2. Background colors



Estos colores se interpretan de forma distinta, según se usen los Botones en el *Preview Panel* ([Figure 13-2](#)):

- *Blend* - El color de fondo se degrada desde el horizonte hacia el cenit. Si sólo está pulsado botón, el gradiente va desde la parte inferior a la superior de la imagen generada independientemente de la orientación de la cámara
- *Real* - Si este botón también está pulsado el degradado depende de la orientación de la cámara. El color del horizonte será exactamente ése en el horizonte (en el plano x-y), y el color del cenit se usará para los puntos que estén justo en la vertical de la cámara, tanto arriba como abajo.
- *Paper* - Si se pulsa este botón el gradiente se produce en los colores cenit-horizonte-cenit. De esta forma, habrá dos transiciones en la imagen, lo que refleja la rotación de la cámara pero mantiene el color del horizonte hacia el centro y el color del cenit en los extremos.

El menú *World Buttons* también tiene un Panel de *Textura* con dos Pestañas. Se usan prácticamente igual que las texturas de *Material*, excepto por un par de diferencias ([Figure 13-3](#)):

- Sólo hay seis canales de textura.
- *Texture mapping* - Tiene sólo las opciones *Object* y *View*, siendo *View* (Vista) la orientación por defecto.
- *Affect* - La Textura afecta sólo al color, pero de cuatro maneras diferentes: Puede afectar al canal de degradado *Blend*, haciendo que el color del Horizonte aparezca donde la textura es distinta de cero; el color del horizonte *Horizonta*; y el color del cenit, arriba o abajo (*Zen Up*, *Zen Down*)

Figure 13-3. Botones de textura



[Prev](#)

Ajustando la iluminación

[Home](#)

[Up](#)

[Next](#)

Exposición

13.2. Exposición

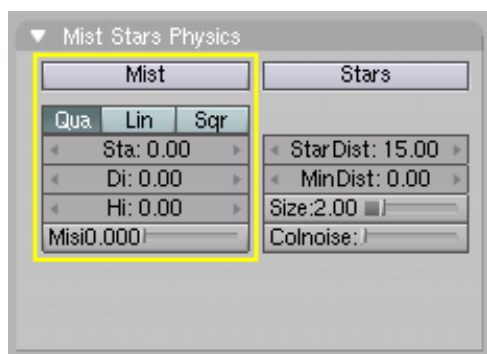
Aún por escribir

13.3. Niebla (Mist)

Importante en Blender v2.31

La niebla puede mejorar enormemente la sensación de profundidad en la imagen generada. Para crear niebla, Blender mezcla simplemente el color de fondo con el objeto del color y aumenta la intensidad del primero conforme el objeto esté más lejos de la cámara. Las opciones de Niebla están en "Mist settings" del Panel *Mist Stars Physics* y se muestran en [Figure 13-4](#).

Figure 13-4. Botones de niebla



El botón *Mist* activa o desactiva la niebla. La fila de tres botones "Toggle" (conmutar) debajo de este botón fija el grado de degradación de la niebla según las funciones cuadráticas, lineal, y raíz cuadrada (*Quadratic*, *Linear*, and *Square Root*). Estas opciones controlan la regla que gobierna la intensidad de la niebla a medida que el objeto se aleja de la cámara.

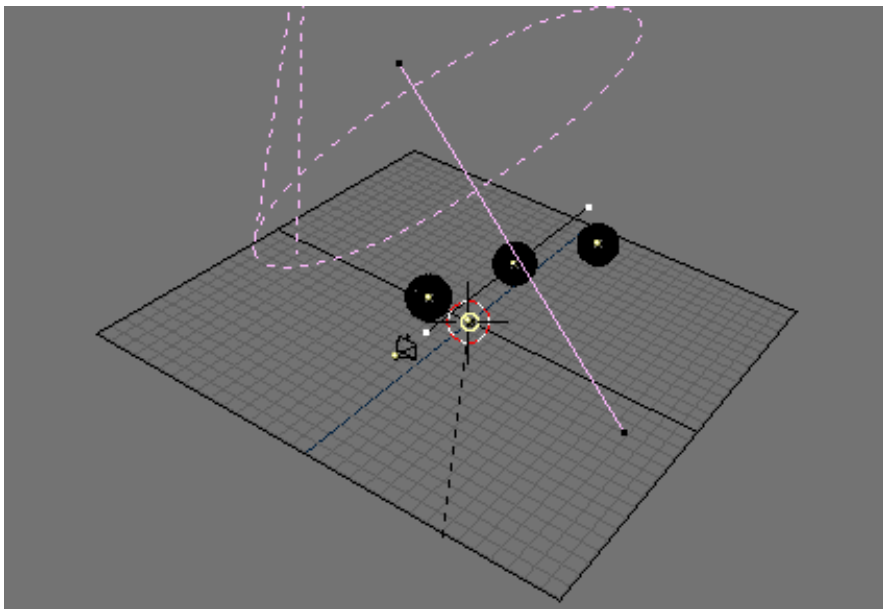
La niebla comienza a una distancia de la cámara definida por el botón *Sta:*, y se computa según la distancia definida por el botón *Di:*. Los objetos que estén alejados de la cámara más allá de $Sta + Di$ estarán totalmente ocultos por la niebla.

Por defecto, la niebla cubre toda la imagen uniformemente. Para producir un efecto más realista se podría hacer que la niebla disminuyera con la altura (eje z) usando el NumButton *Hi:*. Si el valor de este botón es distinto de cero, ajusta, en unidades de Blender, un intervalo alrededor de $z=0$ en el cual la niebla cambia de intensidad máxima (abajo) a cero (arriba)

Finalmente, el Numbutton *Misi:* define la intensidad o densidad de niebla

[Figure 13-5](#) muestra un ejemplo de prueba.

Figure 13-5. Configuración de la niebla



[Figure 13-6](#) muestra los resultados con y sin niebla. Los valores de control se muestran en [Figure 13-7](#); la textura es la de una nube sencilla procedural con nivel de ruido *Hard*.

Figure 13-6. Generación sin niebla (izquierda) y con niebla (derecha).

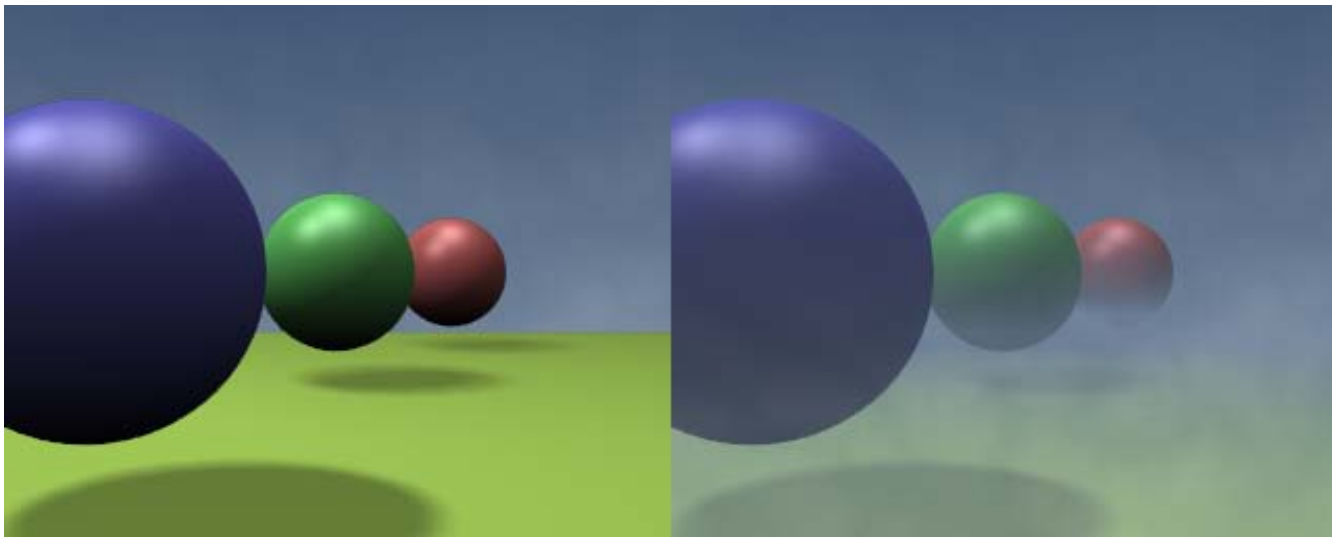
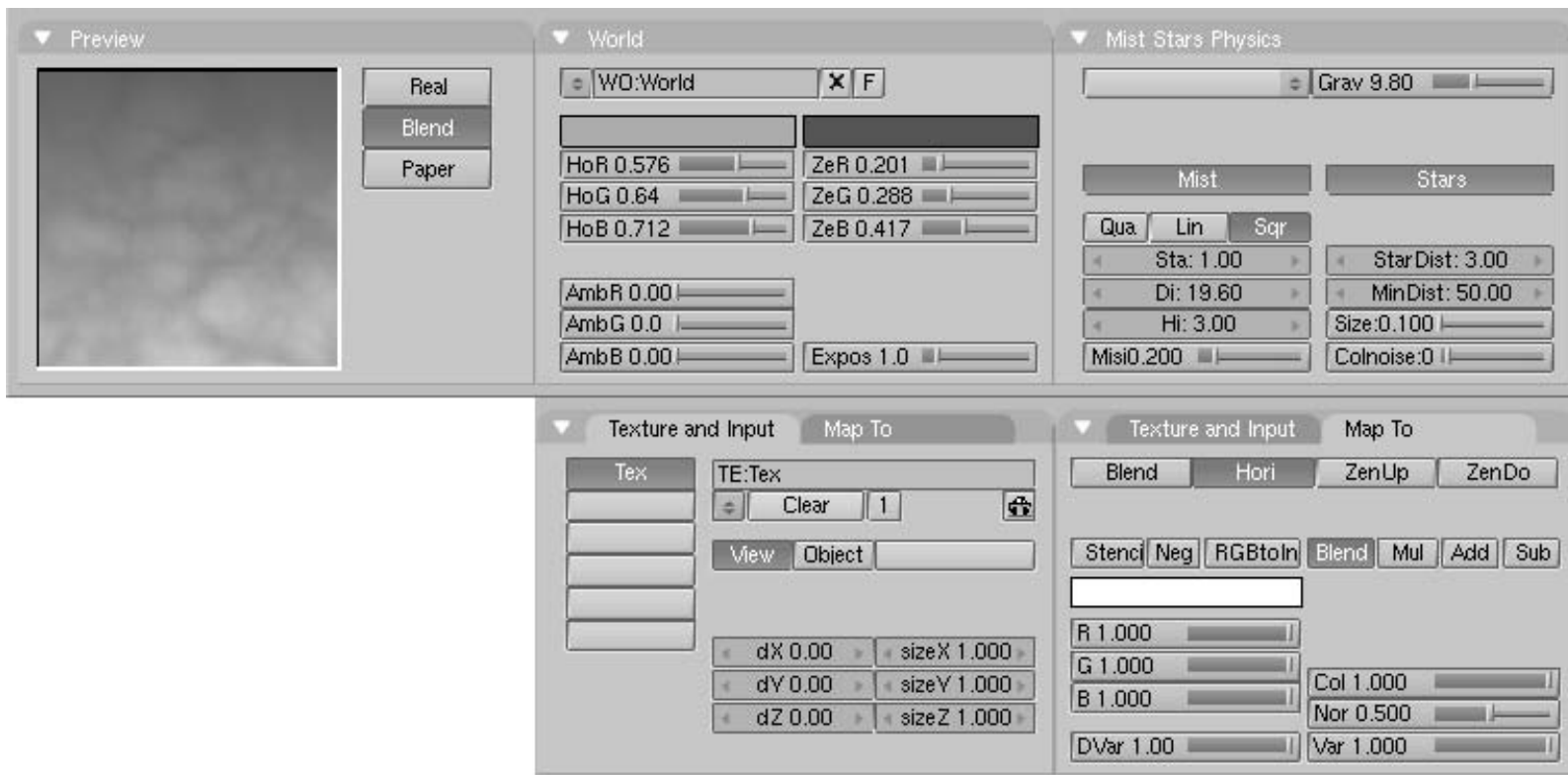


Figure 13-7. Configuración de Mundo.



Tip Distancias de niebla

Para ver a lo que afectará realmente la niebla seleccione su cámara, vaya a Contexto de Edición (Editing Context o **F9**) y pulse el botón conmutador *Show Mist* en el Panel *Camera*. La cámara mostrará los límites de la niebla como un segmento que se proyecta desde la cámara desde *Sta* y de distancia *Di*.

[Prev](#)

Exposición

[Home](#)
[Up](#)
[Next](#)

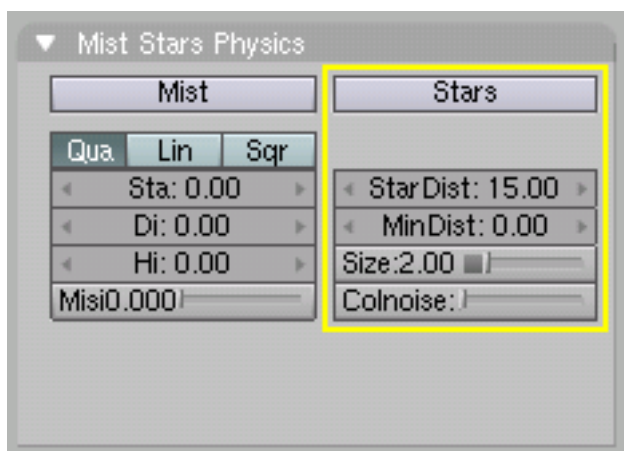
Estrellas

13.4. Estrellas

Importante en Blender v2.31

Las estrellas se colocan aleatoriamente como objetos con halo que aparecen en el fondo. Las opciones para estrellas también se encuentran en el Panel *Mist Stars Physics* ([Figure 13-8](#)).

Figure 13-8. Botones de Estrella



Cuando se quieren crear estrellas, deben entenderse previamente algunos conceptos importantes: *StarDist*: es la distancia *media* entre estrellas. Las estrellas son características intrínsecamente 3D ubicadas en el espacio, no en una imagen.

Min Dist: es la distancia *mínima* desde la cámara a la que se colocan las estrellas. Ésta debería ser mayor que la distancia desde la cámara al objeto *más alejado* de la escena, a menos que se quiera correr el riesgo de que las estrellas aparezcan *delante* de los objetos.

El botón *Size*: define el tamaño real del halo de la estrella. Debe mantenerse este valor mucho más pequeño que el propuesto por defecto para mantener el material más pequeño que el tamaño del pixel y crear así estrellas hechas de un sólo punto, que son mucho más realistas.

El botón *Colnoise*: añade un tono de color aleatorio a las estrellas que de otro modo son blancas. Normalmente es una buena idea añadir un poco de *ColNoise*.

[Figure 13-9](#) muestra la misma imagen neblinosa que [Figure 13-7](#), pero con estrellas añadidas. Se

muestra la configuración para las estrellas en [Figure 13-10](#).

Figure 13-9. Generación de estrellas.

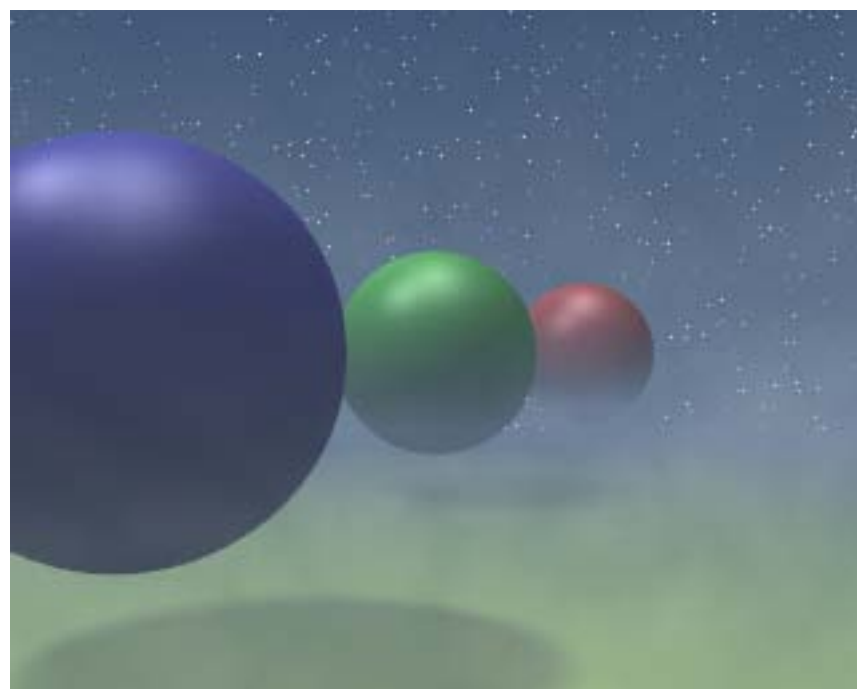
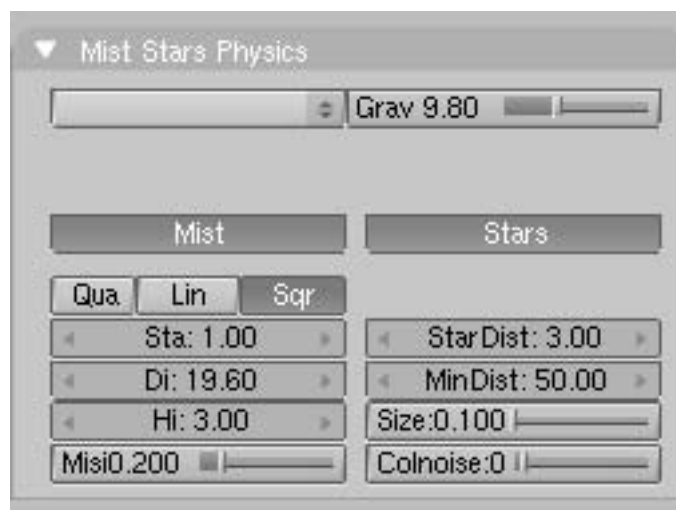


Figure 13-10. Configuración de las estrellas.



[Prev](#)

Niebla (Mist)

[Home](#)

[Up](#)

[Next](#)

Oclusión Ambiental

13.5. Oclusión Ambiental

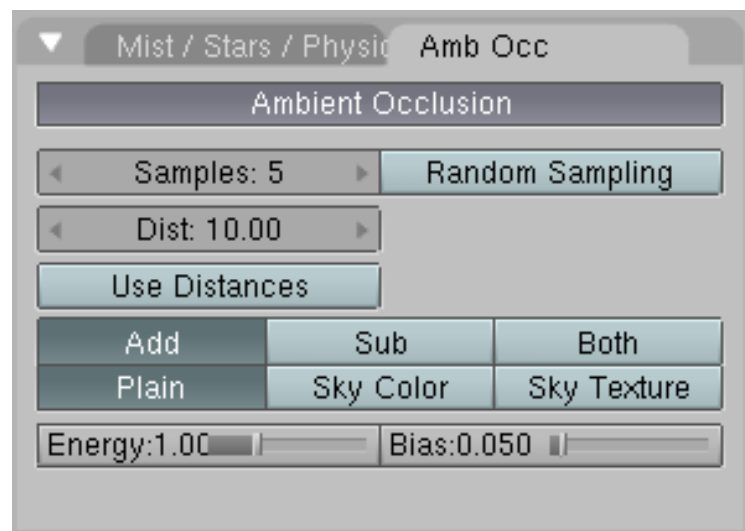
La Oclusión Ambiental, es un sofisticado truco ambiental que simula la iluminación global "blanda", teniendo en cuenta la cantidad de cielo (que se asume como fuente de luz) visible desde un punto dado.

Esto se logra de hecho, emitiendo rayos desde cada punto visible, y contando cuantos de ellos logran alcanzar el cielo, y cuántos por el contrario, son bloqueados por objetos. La cantidad de luz sobre el punto es entonces proporcional al número de rayos que han "escapado" y han alcanzado el cielo.

Se dispara una hemi-esfera de rayos-sombra. Si un rayo pega contra una cara (es ocluído) entonces el rayo es considerado "sombra", de otro modo se lo considera "luz". La relación entre la cantidad de rayos "sombra" y "luz", define cuán brillante será el pixel.

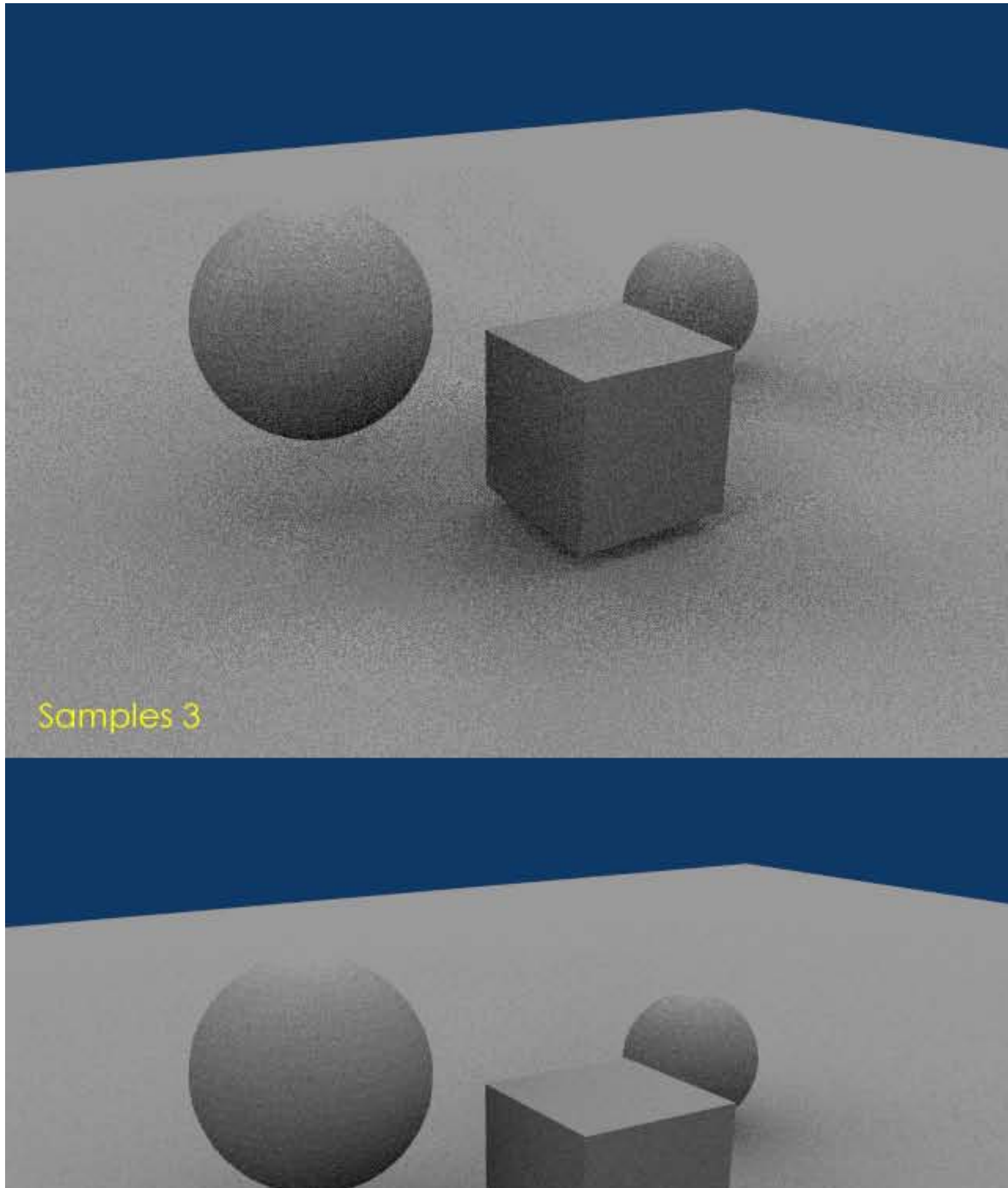
Los parámetros de Oclusión Ambiental ("Ambient Occlusion" - AO) están en la pestaña *Amb Occ* del Sub-contexto Botones de World, en el Contexto de Sombreado. Por omisión, la AO está desactivada, si se la activa la pestaña se llena de botones ([Figure 13-11](#)).

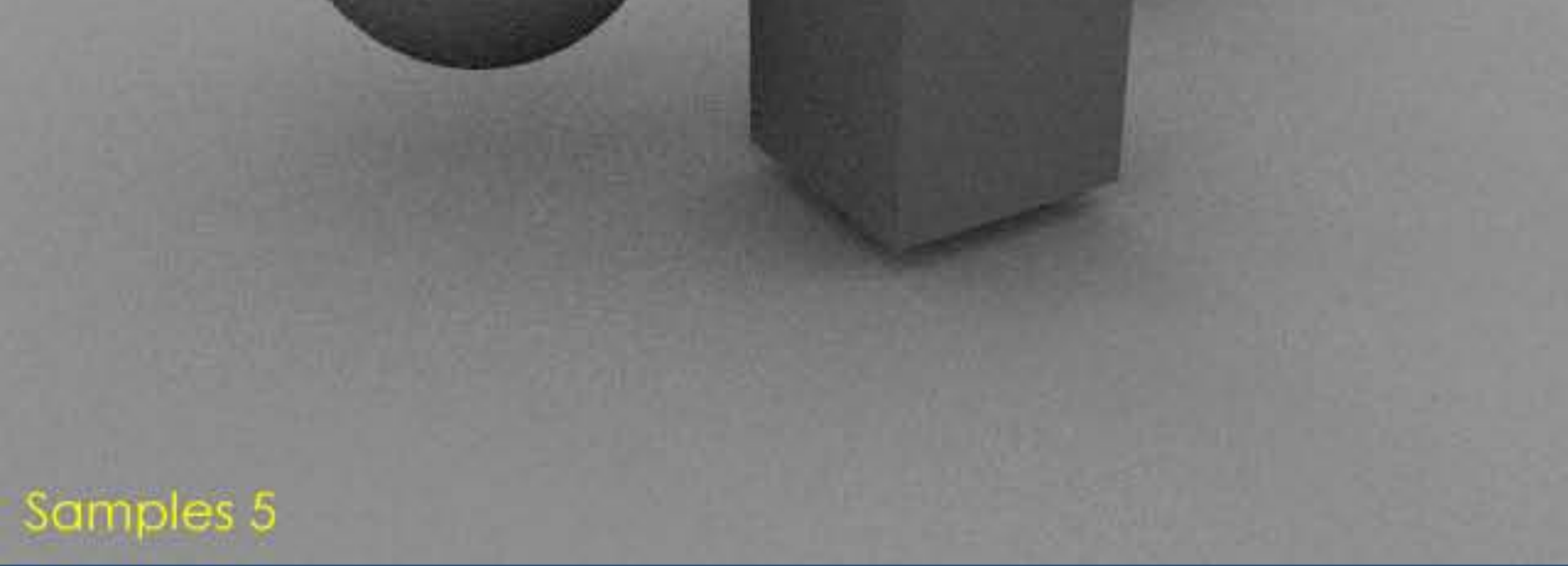
Figure 13-11. Panel de Oclusión Ambiental.



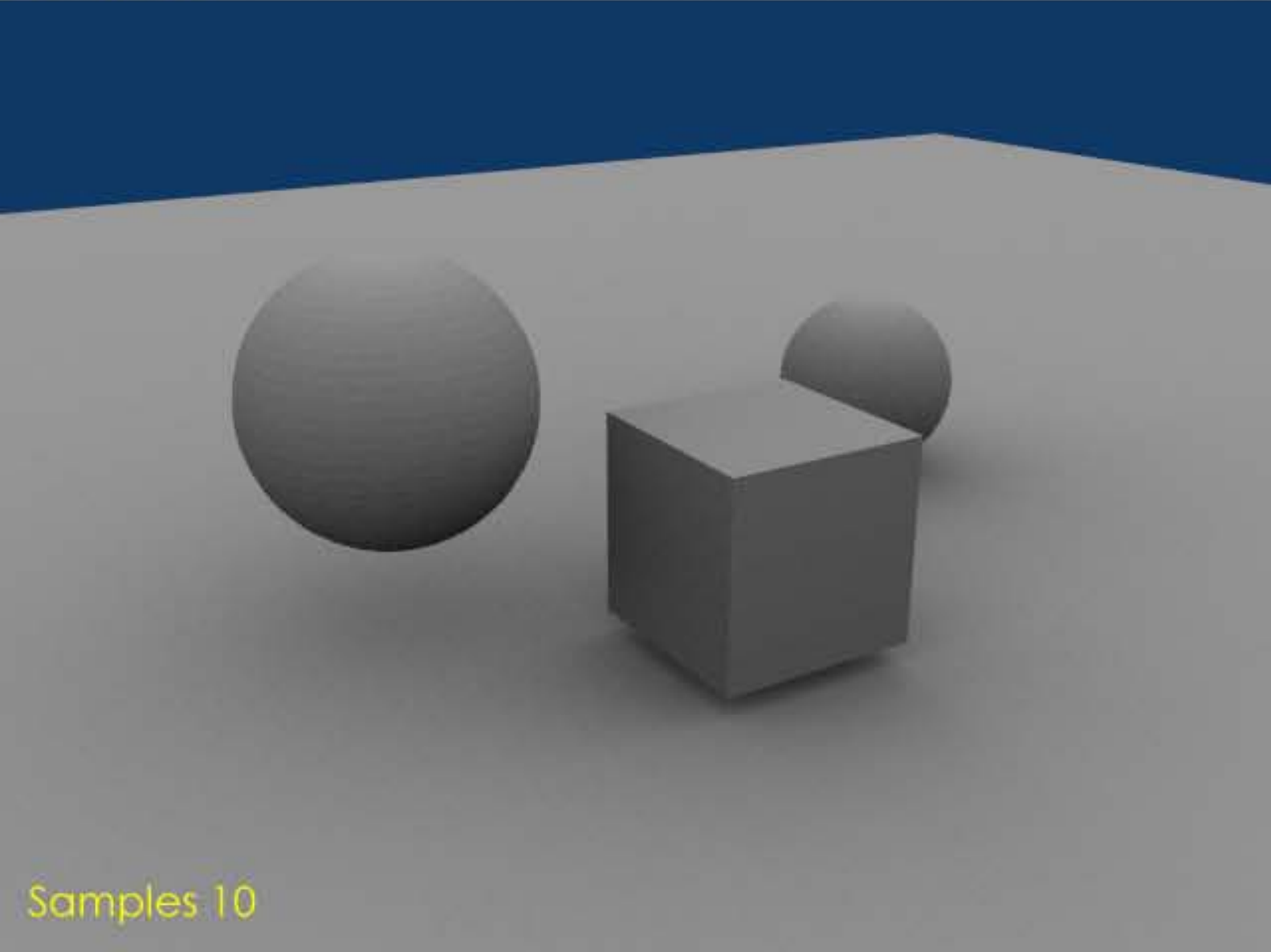
Los rayos se disparan a la hemiesfera siguiendo un patrón aleatorio, esto causa diferencias apreciables en el patrón de oclusión de pixeles vecinos a menos que el número de rayos disparados sea lo suficientemente grande como para producir buenos datos estadísticos. Esta es la razón por la que la AO genera un patrón granuloso que parece suciedad, si no hubiera rayos suficientes. La cantidad de rayos disparados se controlan por medio del botón numérico *Samples*. El valor por omisión de 5 es en general bueno para previsualizaciones. La cantidad efectiva de rayos disparados es el cuadrado de este número. (por ejemplo *Samples (Muestreos)*=5 significa 25 rayos). La [Figure 13-12](#) muestra una escena simple con número creciente de muestreos. Por supuesto, el tiempo de render ; aumenta con el número de muestreos !

Figure 13-12. Efecto de distintas cantidades de muestreos.





Samples 5



Samples 10

Los botones *Dist* y *Use Distances* permiten un control sutil del sombreado estableciendo una conducta dependiente de la distancia y amortiguamiento de la oclusión.

La fila de botones *Add*, *Sub* y *Both* controlan como se comporta la oclusión:

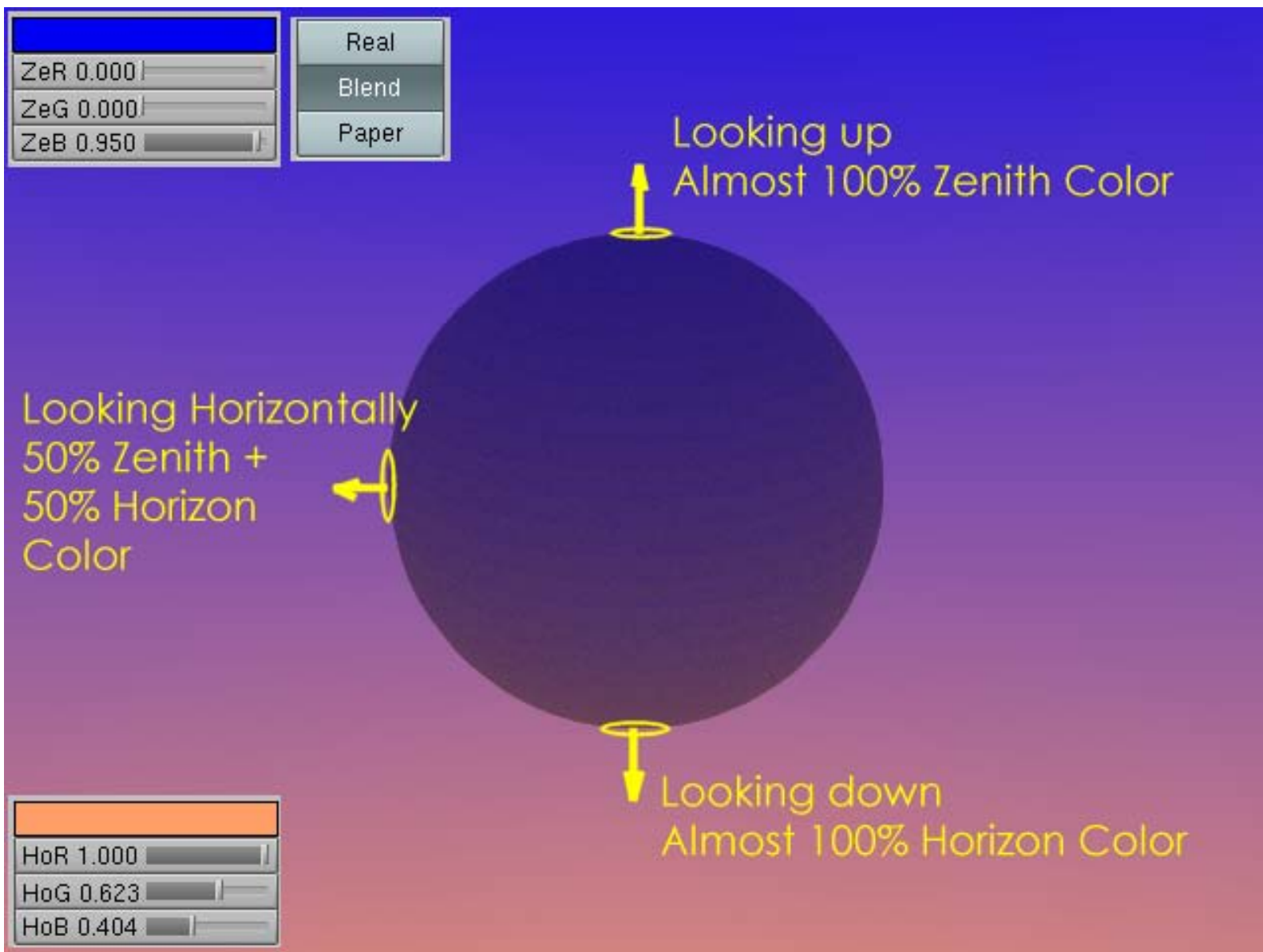
- *Add* - El pixel recibe luz de acuerdo al número de rayos no-obstruidos. La escena es clareada.
- *Sub* - El pixel recibe sombra (luz negativa) de acuerdo al número de rayos obstruidos. La escena se oscurece.
- *Both* - Ambos efectos tienen lugar, La escena tiene más o menos el mismo brillo.

Note Si se elige *Sub* entonces *debe* haber alguna fuente de luz en alguna parte, de otro modo la escena tenderá al negro . En los otros dos casos, la escena es iluminada aún cuando no exista ninguna luz explícita.

La fila de botones *Plain*, *Sky Color* y *Sky Texture* controla el color de la luz:

- *Plain* - El pixel recibe luz blanca pura de acuerdo al número de rayos no-obstruidos.
- *Sky Color* - El pixel recibe luz coloreada, el color se calcula basandose en la porción del cielo tocada por los rayos no-obstruidos ([Figure 13-13](#)).
- *Sky Texture* - Debe haber una textura de Imagen para el Cielo, preferentemente un *Mapa Angular* o un *Mapa Esférico*. Funciona como *Sky Color* pero el color del rayo depende del color del pixel de la textura Cielo alcanzado por el mismo.

Figure 13-13. Oclusión Ambiental con Color Cielo. Zenith es azul, Horizon es anaranjado, y el tipo es *Blend* para que el cielo sea un anaranjado total en el Nadir.

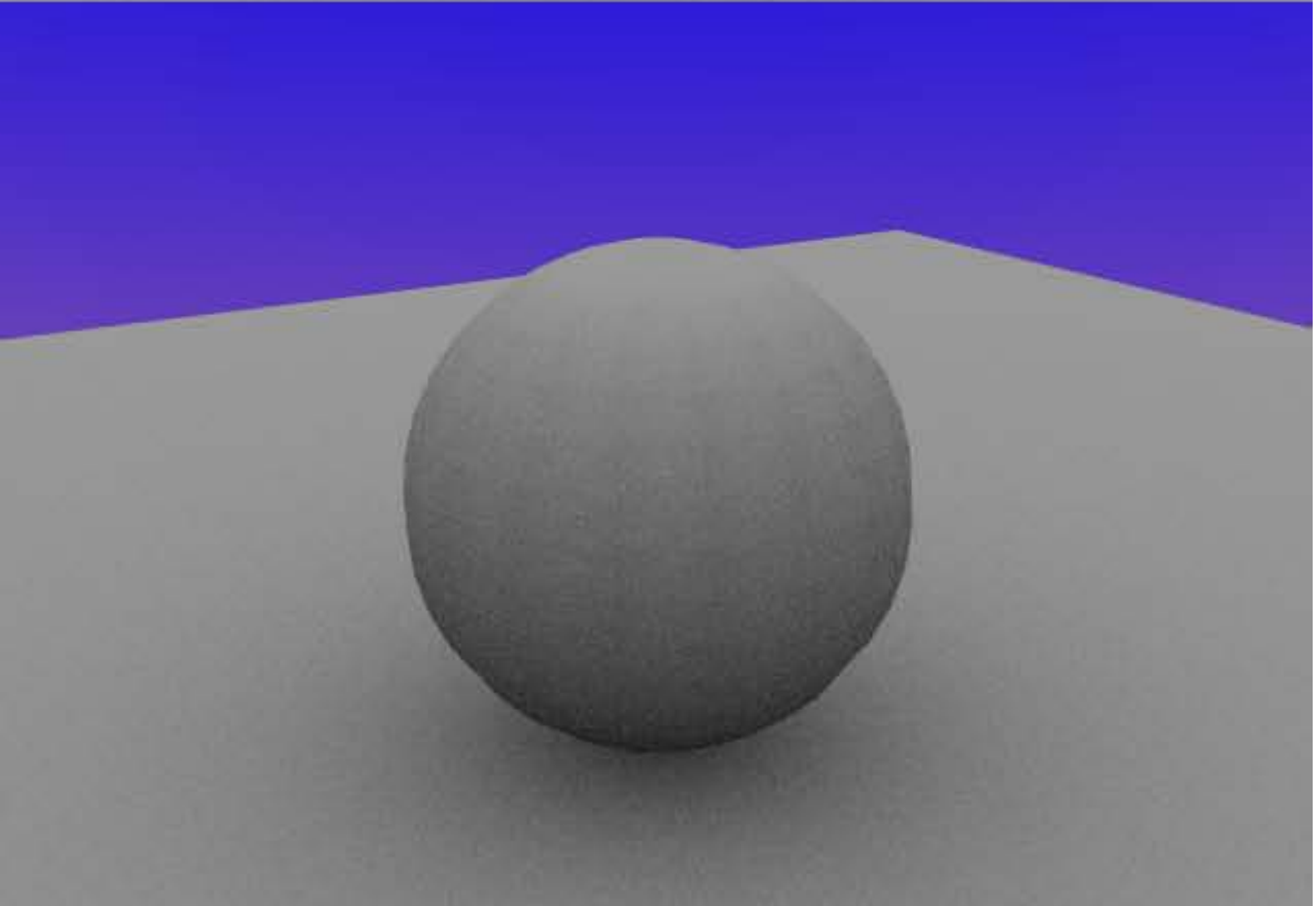
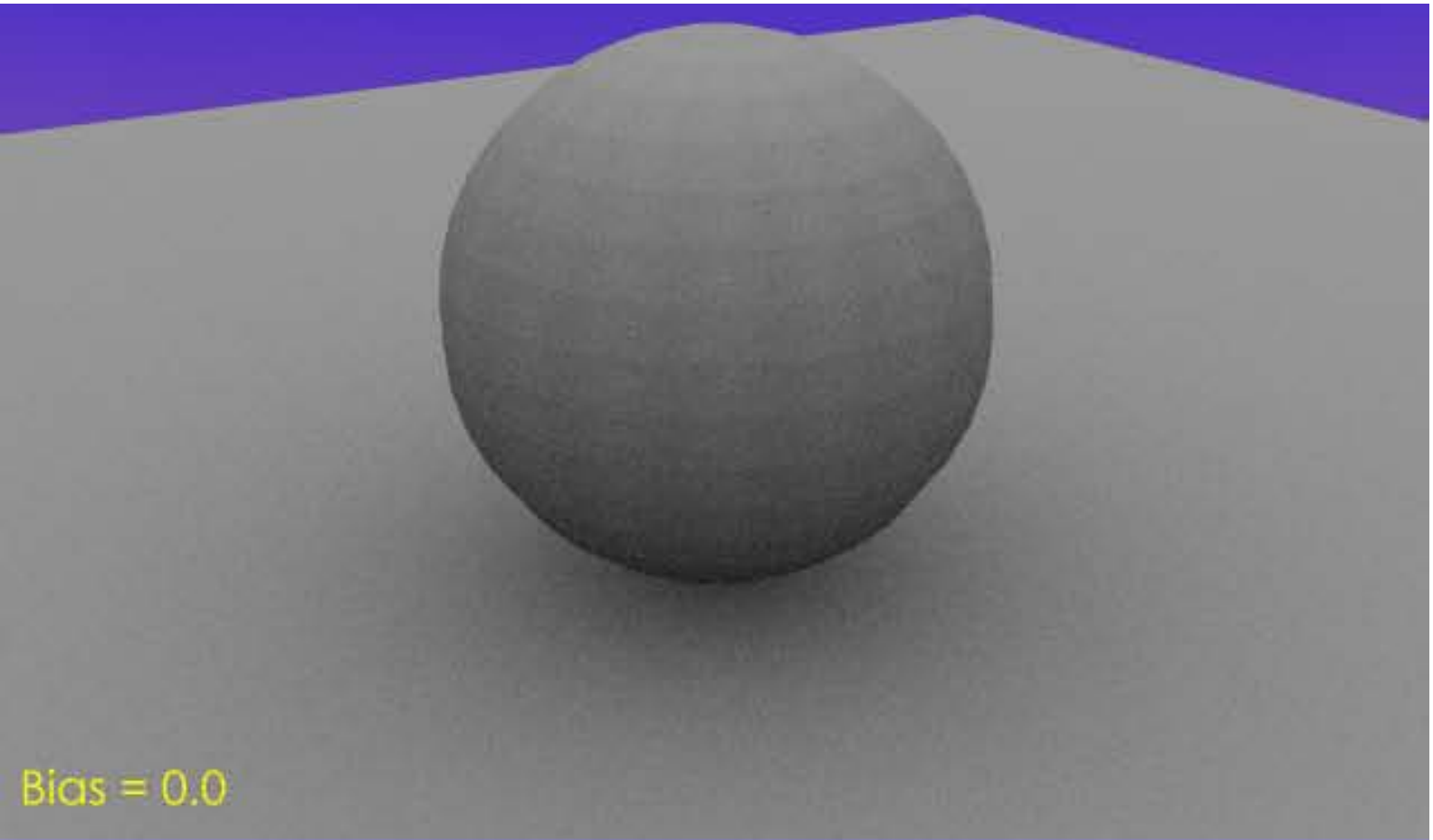


El deslizador *Energy* controla la cantidad real de luz/sombras que crea el procedimiento AO.

Debido a que la AO tiene lugar sobre la malla facetada original, es posible que la luz AO haga visibles las caras sobre objetos con 'smooth' activado. Esto se debe a la manera en que se disparan los rayos AO, y puede controlarse con el deslizador *Bias*. El valor Bias permite controlar la suavidad con la que las caras "suavizadas" aparecerán en el renderizado con AO. El Bias ("desviación") representa el angulo (en radianes) que se estrechará la hemiesfera. Típicamente valores de 0.05 a 0.1 funcionan bien ([Figure 13-14](#)).

Figure 13-14. Valores de Bias de Oclusión Ambiental.





Bias = 0.05 (default)

Bias = 0.1

Por favor, tenga presente que esto es sólo trazado de rayos, por lo que tiende a ser lento. Más aún, la performance depende del tamaño del Octree, ver el Capítulo de Renderizado para detalles adicionales.

[Prev](#)
Estrellas

[Home](#)
[Up](#)

[Next](#)
Animación

III. Animación

Si ya estás preparado para adentrarte en la animación de gráficos por computador entonces esta es tu parte. Aquí analizaremos las características de Blender que conciernen al movimiento de objetos en tres dimensiones y su deformación a lo largo del tiempo.

El capítulo más relevante de esta parte es el dedicado a la Animación de Personajes, donde las increíbles capacidades de Blender son ilimitadas.

Table of Contents

- 14. [Animación de Objetos No Deformables](#)
 - 14.1. [El bloque IPO](#)
 - 14.2. [Cuadros clave](#)
 - 14.3. [Las Curvas IPO](#)
 - 14.4. [Curvas IPO y claves IPO](#)
 - 14.5. [Otras aplicaciones de las Curvas IPO](#)
 - 14.6. [La Ipo del tiempo](#)
 - 14.7. [Trayectoria de la animación](#)
- 15. [Animación de Deformaciones](#)
 - 15.1. [Claves de Vértices Absolutas](#)
 - 15.1.1. [Claves de Curva y Superficie](#)
 - 15.1.2. [Claves de Retícula](#)
 - 15.2. [Claves de Vértices Relativas](#)
 - 15.3. [Animación de Retículas](#)
- 16. [Animación de Personajes](#)
 - 16.1. [Introducción: Luces, Cámaras y... ACCION !](#)
 - 16.2. [Herramientas Generales](#)
 - 16.3. [El Objeto Armature](#)
 - 16.3.1. [Nombrando Huesos](#)
 - 16.3.2. [Emparentado y cadenas IK](#)
 - 16.3.3. [El Panel Armature](#)
 - 16.4. [Skinning](#)
 - 16.4.1. [Grupos de Vértices](#)
 - 16.4.2. [Pintura de Pesos \(Weight Painting\)](#)

- 16.5. [Modo Pose](#)
- 16.6. [Ventana de Acciones](#)
- 16.7. [Animación No Lineal](#)
 - 16.7.1. [Trabajando con Bandas de Acción](#)
- 16.8. [Restricciones \(Constraints\)](#)
 - 16.8.1. [Tipos de Constraints](#)
 - 16.8.2. [Reglas de Evaluación de Restricciones y Precedencia](#)
 - 16.8.3. [Influencia](#)
- 16.9. [Rigging de Mano y Pie](#)
 - 16.9.1. [La Mano](#)
 - 16.9.2. [El Pie](#)
- 16.10. [Rigging de elementos mecánicos](#)
 - 16.10.1. [Ejes pivote](#)
 - 16.10.2. [La armadura](#)
 - 16.10.3. [Mecanismos hidraulicos](#)
- 16.11. [Cómo hacer un ciclo de andar usando NLA](#)
 - 16.11.1. [El camino del éxito](#)
 - 16.11.2. [Moviendo las manos mientras caminamos](#)

[Prev](#)

[Home](#)

[Next](#)

Oclusión Ambiental

Animación de Objetos No
Deformables

Chapter 14. Animación de Objetos No Deformables

Los objetos se pueden animar de varias formas. Pueden animarse como Objetos, cambiando su posición, orientación o tamaño en el tiempo; pueden animarse deformándolos; esto es, animando sus vértices o puntos de control; o pueden animarse mediante una compleja y flexible interacción con una clase especial de objeto: el Esqueleto.

En este capítulo cubriremos el primer caso, pero los conceptos básicos dados aquí son igualmente vitales para entender también los siguientes capítulos.

Hay tres métodos usados normalmente en programas de animación para hacer moverse a un objeto en 3D:

- *Key frames (fotogramas clave)* La posición completa se guarda por unidad de tiempo (frames o fotogramas). Una animación se crea interpolando fluidamente el objeto a lo largo de los fotogramas. La ventaja del método es que te permite trabajar con unidades claramente visibles. El animador puede trabajar desde una posición a la siguiente, puede cambiar posiciones creadas anteriormente, o moverlas en el tiempo.
- *Motion Curves (curvas de movimiento)* Pueden dibujarse curvas para cada componente XYZ de la posición, rotación y tamaño. Esto forma el gráfico del movimiento, con el tiempo dispuesto horizontalmente y el valor en vertical. La ventaja de este método es que te da control preciso sobre el resultado del movimiento.
- *Path (Camino)* Se dibuja una curva en un espacio 3D y se le asigna al objeto para que la recorra de acuerdo a una función de tiempo dada para la posición a lo largo del camino.

Los primeros dos sistemas en Blender se integran completamente en uno único, el *IPO* (sistema de InterPOLación). Fundamentalmente, el sistema IPO consiste en curvas de movimiento estándar. Una simple pulsación de un botón cambia la IPO a un sistema de claves, sin conversión ni cambios en los resultados. El usuario puede trabajar como quiera con las claves, cambiar a curvas de movimiento y volver de nuevo, de la forma que mejores resultados le produzca o satisfaga sus preferencias personales.

El sistema IPO también tiene implicaciones relevantes en las animaciones por Caminos.

14.1. El bloque IPO

Relevante para Blender v2.31

El bloque IPO en Blender es universal. No diferencia si se controla el movimiento de un objeto o la configuración del material. Una vez has aprendido a trabajar con IPOs para objetos, será obvio cómo trabajar con otros IPOs. De todas formas, Blender hace distinción entre diferentes *tipos* de IPOs y la interfaz lo tiene en cuenta automáticamente.

Cada tipo de bloque IPO tiene un número fijo de *canales* disponibles. Cada uno tiene un nombre (*LocX*, *SizeZ*, etc.) que indica cómo se aplican. Cuando añades una curva IPO a un canal, la animación empieza inmediatamente. A tu albedrío (y hay canales separados para ello), una curva puede ser enlazada directamente a un valor (*LocX...*), o puede aplicarse a un incremento del mismo (*dLocX...*). Éste último te permite mover un objeto tal y como se haría normalmente arrastrándolo con la tecla G, sin interrumpir la IPO. La posición actual se determina entonces con curvas IPO *relativas* a esa posición.

El interfaz de Blender ofrece varias opciones para copiar IPOs, enlazar IPOs a más de un objeto (una IPO puede animar a varios objetos), o borrar enlaces IPO. En la sección "Ventana de Referencia de IPO" se da una descripción de tallada de esto. Este capítulo se limita a las principales opciones para su aplicación.

[Prev](#)

Animación

[Home](#)[Up](#)[Next](#)

Cuadros clave

14.2. Cuadros clave

Relevante para Blender v2.31

Figure 14-1. Menú Insertar Clave.



La forma más fácil para crear un objeto IPO es con el comando "Insertar clave" (**Tecla "i" (IKEY)**) en la ventana 3D, con un Objeto seleccionado. Un menú emergente ofrece una amplia selección de opciones ([Figure 14-1](#)). Seleccionaremos la primera opción: *Loc*. Ahora se ha guardado la posición actual X-Y-Z y todo se coloca automáticamente:

- Si no hay un bloque IPO, se crea uno nuevo y se asigna al objeto.
- Si no hay Curvas IPO en los canales *LocX*, *LocY* and *LocZ*, se crearán.
- Entonces se añadirán los vértices a las Curvas IPO con los valores exactos de la posición del objeto.

Nos movemos 30 frames más lejos (pulsando tres veces la **Flecha Arriba (UPARROW)**) y movemos el objeto. De nuevo usamos **Tecla "i" (IKEY)**. Ahora podemos pulsar inmediatamente **ENTER** para que Blender recuerde nuestra última opción y la seleccione. La nueva posición se inserta en la Curva IPO. Podemos verlo lentamente volviendo hacia atrás a través de los cuadros (**Flecha Izquierda (LEFTARROW)**). El objeto se mueve entre las dos posiciones.

En este caso, puedes crear la animación moviéndote a lo largo de los cuadros, posición por posición. Observa que la posición del objeto es *directamente* enlazada a las curvas. Cuando cambias cuadros, las IPOs son siempre reevaluadas y reaplicadas. Puedes mover libremente el objeto en el mismo cuadro,

pero tan pronto como cambies de cuadro, el objeto 'salta' a la posición que había sido determinada por el IPO.

La rotación y el tamaño del objeto son completamente independientes en este ejemplo. Pueden ser cambiadas o animadas con el procedimiento de Insertar Clave seleccionando del menú que aparece al pulsar la **Tecla "i" (IKEY)** las otras opciones como Rotación (Rotation), Tamaño (Size) y cualquiera de sus combinaciones.

[Prev](#)

Animación de Objetos No
Deformables

[Home](#)
[Up](#)

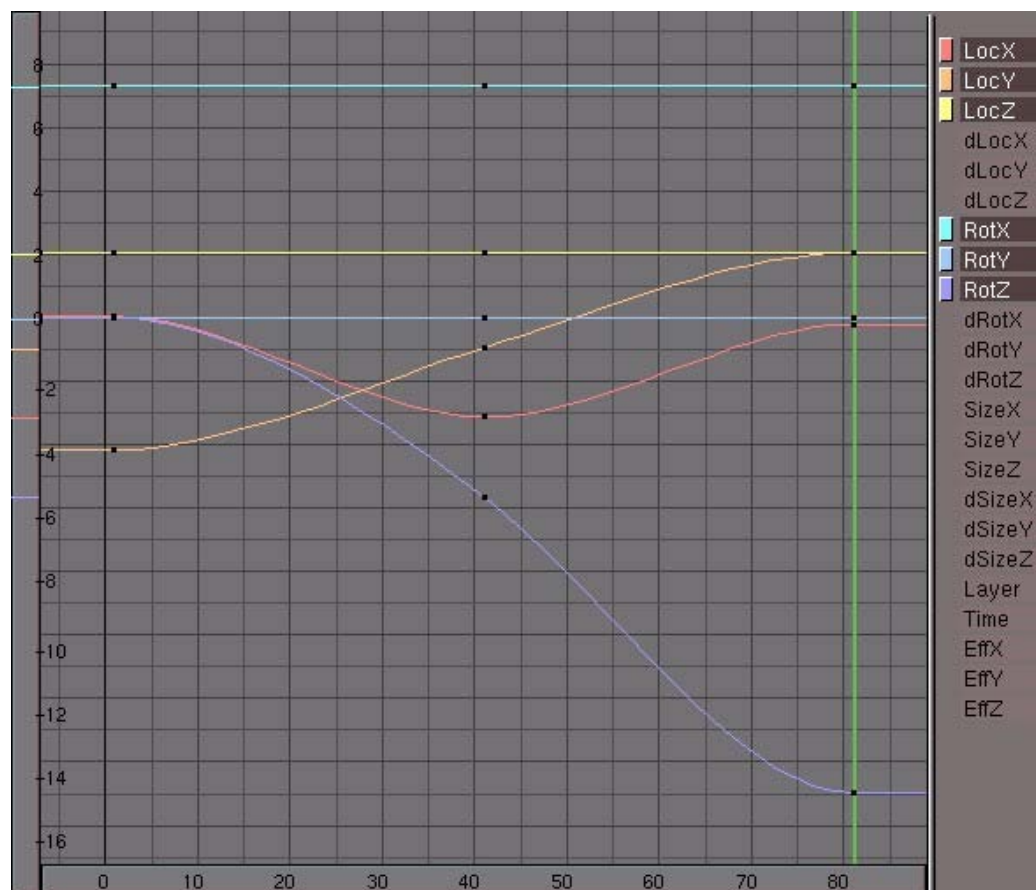
[Next](#)

Las Curvas IPO

14.3. Las Curvas IPO

Relativo a Blender v2.31

Figure 14-2. La ventana IPO.



Ahora queremos ver exactamente qué ha pasado. La primera ventana que aparece al cargar el archivo estándar de inicio de Blender es estupenda para esto. Actívala con **CTRL-FLECHA_IZQUIERDA**. A la derecha podemos ver la Ventana IPO representada ([Figure 14-2](#)). Desde luego, puedes convertir cualquier ventana en una ventana IPO eligiendo este tipo en el menú de Tipo de Ventana, pero es más manejable tener la vista 3D y la ventana IPO a la vez. Ésta muestra todas las Curvas IPO, los canales usados y los disponibles. Puedes hacer zoom en la Ventana IPO y moverla tal y como se haría en cualquier otra ventana de Blender.

Además de los canales estándar, que pueden ser inicializados pulsando la tecla "i" (**IKEY**), tienes las opciones *delta*, como la *dLocX*. Estos canales te permiten asignar un cambio *relativo*. Esta opción se usa principalmente para controlar múltiples objetos con la misma IPO. Además, es posible trabajar con "capas" de animación. Puedes conseguir efectos sutiles de esta forma sin tener que dibujar curvas complicadas.

Cada curva puede seleccionarse individualmente con el botón derecho del ratón (**RMB**). Además, los modos de Mover y Escalar funcionan exactamente como en una vista 3D. Puedes seleccionar IPOs también seleccionando el botón coloreado en la columna de nombres de canales de la derecha. Seleccionando el nombre de canal IPO muestras u ocultas la curva relativa. Seleccionando todas las curvas (con la tecla "a", **AKEY**) y moviéndolas a la derecha (con la tecla "g" **GKEY**), puedes mover y completar la animación a lo largo del tiempo.

Cada curva puede ser posicionada individualmente en modo edición, o puede hacerse colectivamente. Selecciona las curvas y pulsa **TAB**. Ahora se muestran los vértices individuales y los *manejos* de la curva. Los manejos de Bézier están programados tal y como aparecen en el Objeto Curva:

- Free Handle (Manejo Libre, en negro). Puede usarse para lo que quieras. Atajo de teclado: tecla H (**HKEY**) (cambia entre Libre y Alineado).
- Aligned Handle (Manejo Alineado, en rosa). Esto coloca todos los manejos en línea recta. Atajo: tecla H (**HKEY**) (cambia entre Libre y Alineado).
- Vector Handle (Manejo por Vector, en verde). Las dos partes del manejo siempre apuntan hacia el manejo anterior o siguiente. Atajo: Tecla V (**VKEY**).
- Auto Handle (Auto Manejo, en amarillo). Este manejo tiene una dirección y longitud totalmente automática. Atajo: Mayúsculas+Tecla H (**SHIFT-HKEY**).

Los manejos pueden moverse seleccionando primero el vértice central con el botón derecho del ratón (**RMB**). Esto selecciona también los otros dos vértices. Inmediatamente después puedes moverlos manteniendo pulsado el **RMB** y moviendo el ratón. Los manejos pueden *rotarse* seleccionando primero el final de uno de los vértices y después moviéndolo manteniendo pulsado el **RMB** y moviendo el ratón.

En cuanto los manejos son rotados, su tipo cambia automáticamente:

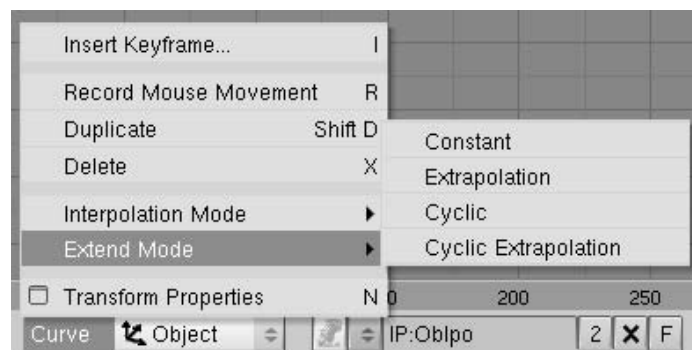
- Auto Handle se vuelven Aligned.
- Vector Handle se vuelven Free.

Los "Auto" manejos se colocan por defecto en la curva. El primer y último Auto manejo siempre se mueve horizontalmente, lo que crea una interpolación fluida.

Las Curvas IPO tienen una característica importante que las distingue de las curvas normales: es imposible poner más de un segmento de curva horizontalmente. Ciclos y círculos en una IPO son ambiguos y no tienen sentido. Una IPO sólo puede tener un valor en un determinado momento. Esto se detecta automáticamente en la Ventana IPO. Al mover parte de la Curva IPO horizontalmente, puedes ver que los vértices seleccionados se mueven "a lo largo" de la curva. Esto te permite duplicar partes de una curva (**SHIFT-D**) y moverla hasta otro cuadro de tiempo.

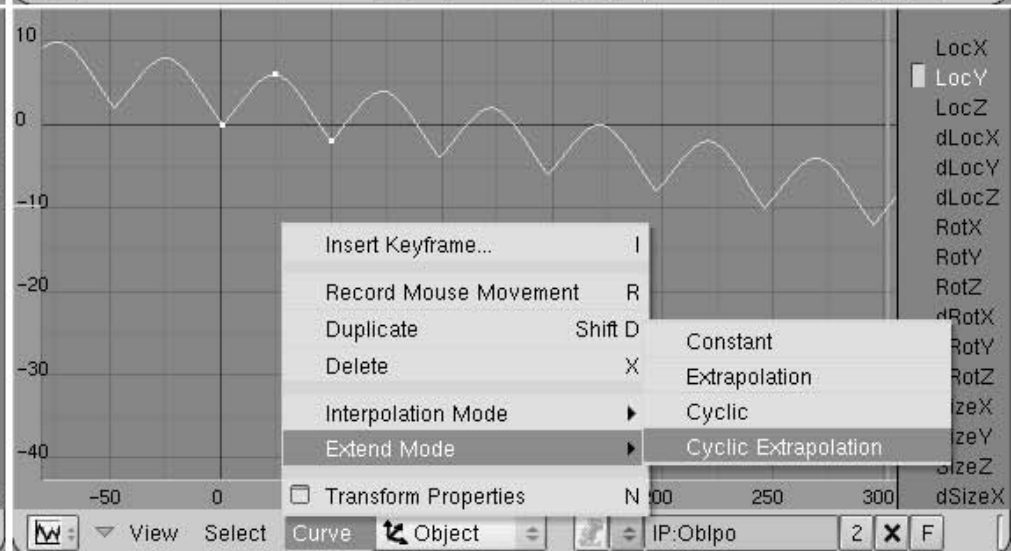
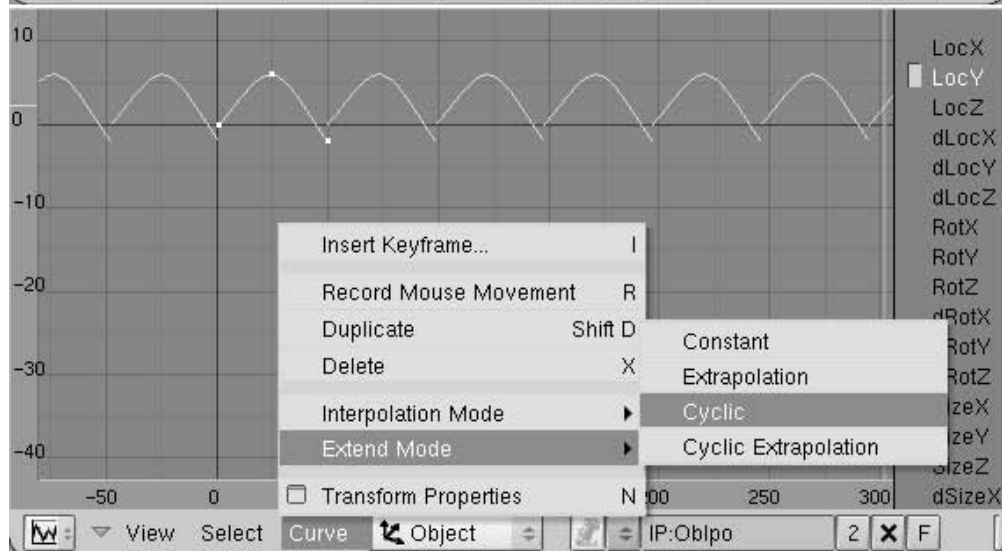
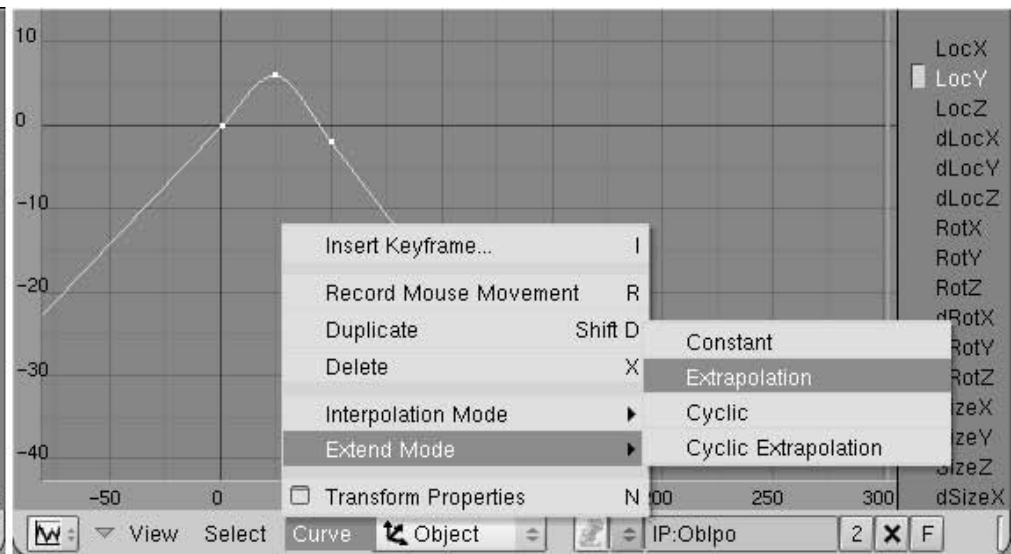
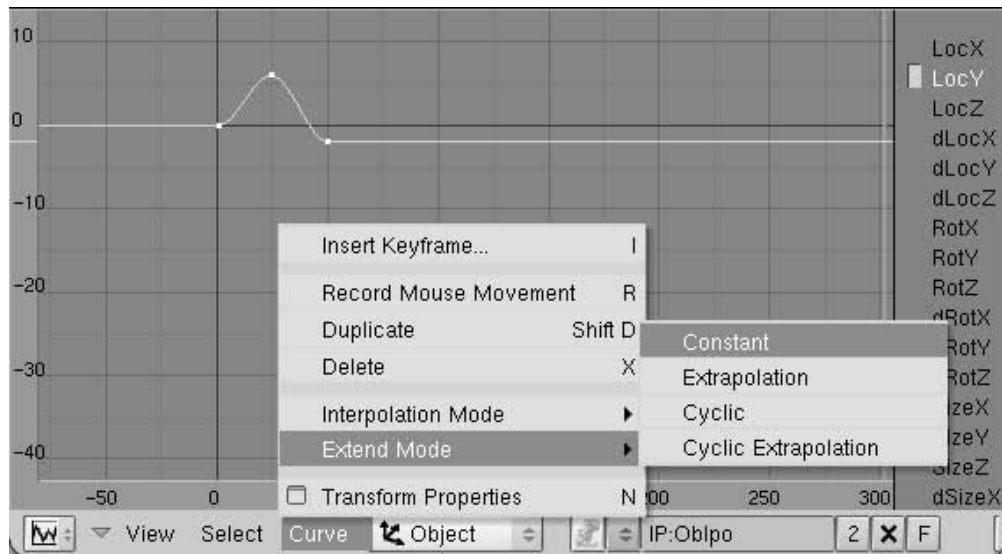
También es importante especificar cómo una Curva IPO debe ser interpretada *desde fuera* de la misma curva. Hay cuatro opciones para esto en el Submenú *Curve* > *Extend Mode* de la cabecera de la ventana IPO ([Figure 14-3](#)).

Figure 14-3. Opciones de extensión de IPO.



El efecto de cada una puede apreciarse en ([Figure 14-4](#)).

Figure 14-4. IPOs extendidos.



De izquierda a derecha:

Modo de extensión Constante:

Los finales de la Curva IPO seleccionada son extrapolados de forma continua (horizontalmente). Es el comportamiento por defecto.

Modo de extensión Extrapolado:

Los finales de la Curva IPO seleccionada continúan en la dirección en la que fueron terminados.

Modo de extensión Cíclico:

La anchura completa de la Curva IPO se repite cíclicamente.

Modo de extensión Extrapolación Cíclica:

La anchura completa de la Curva IPO se extrapola cíclicamente.

Además de las Béziers, hay otros dos posibles tipos de Curvas IPO. Usa el comando de la tecla "T" (**TKEY**), y el diálogo emergente que surge, o la entrada del submenú *Curve>>Interpolation Mode* para elegirla. La interpolación de las Curvas IPO seleccionadas pueden ser:

- *Constante* - tras cada vértice de la curva, su valor permanece constante. No hay interpolación..
- *Lineal* - se interpola linealmente entre los vértices.
- *Bezier* - la interpolación suave estándar.

No hay por qué crear la Curva IPO sólo mediante Cuadros Clave. También se pueden dibujar "a mano". Usa el comando **CTRL-LMB**. Las reglas son las siguientes:

Todavía no hay un IPO block (en esta ventana) y un *canal* está seleccionado:

se crea un nuevo Bloque IPO a lo largo de la primera Curva IPO, cuyo primer vértice se coloca donde se pulsó el ratón.

Ya hay un Bloque IPO, y un *canal* sin una Curva IPO se ha seleccionado:

se añade una nueva Curva IPO con un vértice.

Ya hay un Bloque IPO, y un *canal* con una Curva IPO se ha seleccionado:

Se añade un nuevo punto a la Curva IPO seleccionada.

Esto *no* es posible si hay varias Curvas IPO seleccionadas o en Modo Edición.



Haz rotar un objeto

Éste es el mejor método para especificar ejes de rotación rápidamente: Selecciona el objeto; en la Ventana IPO, pulsa uno de los canales "Rot" y utiliza **CTRL-LMB** para insertar dos puntos. Si el eje de rotación ha de ser continuo, debes usar la entrada del menú *Curve>>Extend Mode>>Extrapolation*.

Una desventaja de trabajar con curvas de movimiento es que la *libertad* de las transformaciones está limitada. Puedes trabajar bastante intuitivamente con curvas de movimiento, pero sólo si ésta puede ser procesada en una base XYZ. Para una posición esto es trivial, pero para un tamaño o rotación hay disponibles mejores descriptores: matrices (3x3 números) para el tamaño y cuaterniones (4 números) para rotaciones. Éstas también podrían procesarse en los canales, pero conllevaría fácilmente a situaciones confusas y matemáticamente complicadas.

Limitar el *tamaño* a los tres números XYZ es obvio, pero esto limita a una distorsión rectangular. Un escalado diagonal como un "esquilado" (shearing) es imposible. Simplemente trabajando con jerarquías puede resolverse esto. Un objeto padre escalado *no*-uniformemente influenciaría la rotación de un objeto hijo como en un "esquilado".

La limitación a las rotaciones de tres números XYZ es menos intuitiva. La llamada "rotación euleriana" no es uniforme (la misma rotación puede expresarse con diferentes números) y tiene el fastidioso efecto de que *no* es posible rotar desde cualquier posición a cualquier otra por el famoso "*gimbal lock*". Mientras está trabajando con distintas claves de rotación, el usuario puede enfrentarse de repente con interpolaciones inesperadas, o puede volvérselo imposible efectuar una particular rotación sobre un eje cuando hace cambios manuales. Aquí, de nuevo, una solución mejor es trabajar con jerarquías. Un objeto padre *siempre* asignará el eje de rotación especificado al hijo. (Es conveniente saber que las rotaciones en X, Y y Z se calculan una *detrás* de la otra. La curva que afecta al canal *RotX siempre* determina la rotación en el eje X).

Afortunadamente, Blender calcula todo internamente con matrices y cuaterniones. Las jerarquías trabajan así normalmente, y el modo Rotación hace lo que tendría que hacer. Sólo las IPOs son una limitación aquí, pero en este caso la facilidad de uso prevalece sobre una no demasiado intuitiva pureza matemática.

[Prev](#)

Cuadros clave

[Home](#)

[Up](#)

[Next](#)

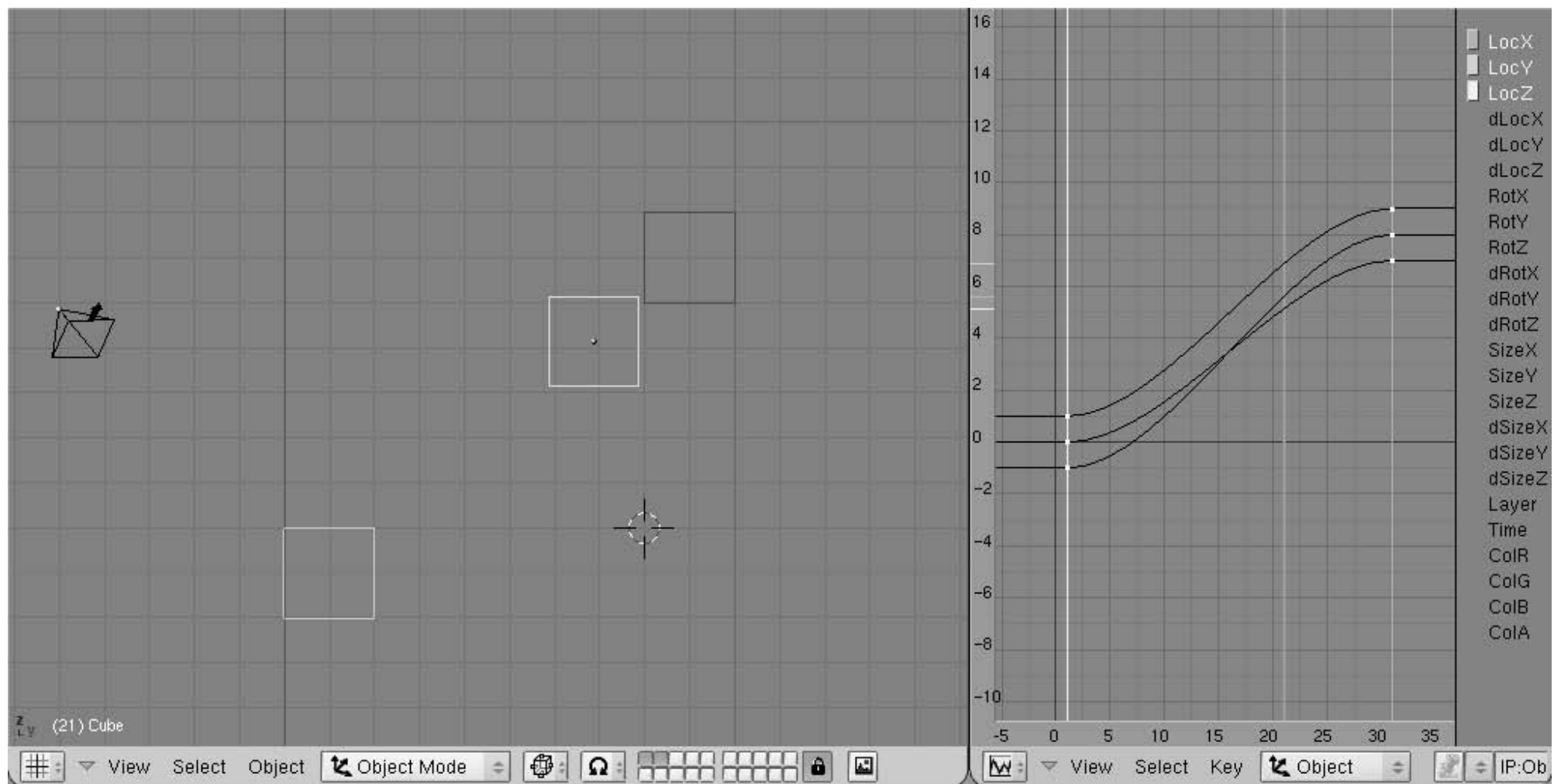
Curvas IPO y claves IPO

14.4. Curvas IPO y claves IPO

Relevante para Blender v2.31

La forma más fácil de trabajar con curvas de movimiento es convertirlas a claves IPO. Volvemos a la situación del ejemplo anterior: hemos especificado dos posiciones en un objeto IPO en el cuadro 1 y el 31 con la tecla "i" (**IKEY**). A la derecha de la pantalla, puedes ver una ventana IPO. Vamos a situarnos en el frame 21 ([Figure 14-5](#)).

Figure 14-5. El modo clave de IPO (IPOKey).



Pulsa la tecla "k" (**KKEY**) mientras el cursor del ratón está en la ventana 3D. Pasarán dos cosas:

- La Ventana IPO cambia al modo clave de IPO.
- Al objeto seleccionado se le asigna la opción de "dibujado de clave" (DrawKey).

Las dos acciones tienen significados distintos.

- La ventana IPO ahora dibuja líneas verticales a lo largo de todos los vértices de todas las curvas IPO visibles (las IPOs ahora son negras). Los vértices con el mismo valor de "cuadro" están relacionadas a las líneas verticales. Las líneas verticales (las "claves IPO") pueden seleccionarse, moverse o duplicarse, al igual que los vértices en Modo de Edición. Puedes mover las claves IPO únicamente en horizontal.
- El objeto no sólo se muestra en su posición actual, sino que también se muestran objetos "fantasma" en todas las posiciones Clave. Además de poder ver las posiciones clave del objeto, también puedes modificarlas *en* la ventana 3D. En este ejemplo, usa el modo de movimiento (Grab) en el objeto para cambiar la clave IPO *seleccionada*.

Abajo hay unas cuantas instrucciones para utilizar mejor la potencia del sistema:

- Sólo puedes usar el **RMB** para seleccionar claves IPOKeys en la ventana de IPO. La selección por borde y la selección *extendida* también funcionan aquí. Selecciona todas las claves IPO para transformar el sistema de animación completa en la ventana 3D.
- La "Inserción de Clave" (Insert Key) siempre afecta a *todos* los objetos seleccionados. La Clave IPO para múltiples objetos también puede transformarse simultáneamente en la ventana 3D. Usa el comando **SHIFT-K: Muestra y selecciona todas las claves** para transformar animaciones completas de un grupo de objetos todos a la vez.
- Usa los comandos **PAGEUP** y **PAGEDOWN** para seleccionar claves subsiguientes en la ventana 3D.
- Puedes crear claves IPO en cada conjunto de canales. *Excluyendo* a consciencia ciertos canales, puedes forzar una situación en la cual los cambios en las posiciones clave de la ventana 3D sólo afectarán a los valores especificados por los canales visibles. Por ejemplo, con sólo el canal *LocX* seleccionado, las claves sólo se moverán en la dirección X.
- Cada Clave Ipo está formada por los vértices que tienen *exactamente* el mismo valor de cuadro. Si los vértices se mueven manualmente, puede pasar que tengamos un enorme número de claves y cada una teniendo sólo una curva. En este caso, usa el comando "Unir" (Join) pulsando la tecla "j" (**JKEY**) para combinar las Claves IPO seleccionadas. También es posible asignar vértices seleccionados de Claves Ipo a *todas* las curvas visibles: usa la tecla "i" (**IKEY**) en la ventana de IPO y elige "Selected keys".
- La opción de Dibujado de Claves (DrawKey) y el modo Clave IPO pueden ser activados y desactivados independientemente. Usa los botones EditButtons->DrawKey para desactivar esta opción u objeto. Puedes activar y desactivar el modo Clave IPO tú mismo con la tecla "k" (**KKEY**) en la ventana de IPO. Sólo pulsando **KKEY** en la ventana 3D activa o desactiva a la vez los modos Dibujado de Clave y Clave IPO.

[Prev](#)[Las Curvas IPO](#)[Home](#)[Up](#)[Next](#)[Otras aplicaciones de las Curvas IPO](#)

14.5. Otras aplicaciones de las Curvas IPO

Relevante para Blender v2.31

Hay muchas otras aplicaciones para las IPOs aparte de la simple animación del movimiento de objetos.

Los botones del menú "IPO Type" de la cabecera [Figure 14-6](#) permiten la selección del tipo del Bloque IPO, la activa ahora es el Objeto IPO descrito hasta ahora, pero hay Material IPO, Mundo (World) IPO, Claves de Vértices (Vertex Keys) IPO, Restricciones (Constraints) IPO y Secuencia (Sequence) IPO. No todas las opciones están siempre presentes, dependiendo del contexto. El bloque Curva IPO aparece si el objeto seleccionado es una curva y no una Malla; la Luz (Lamp) IPO aparece sólo si el objeto seleccionado es una luz.

Figure 14-6. La ventana IPO.



El Material IPO es una forma de animar un Material. Como con los objetos, las Curvas IPO pueden usarse para especificar 'posiciones clave' para Materiales. Con el ratón en la ventana de botones (ButtonsWindow), pulsando la tecla "i" (**IKEY**) abre un menú emergente con opciones para las distintas variables del Material. Si estás en el bloque IPO de Material, Luz (Lamp) o Mundo (World) entonces un pequeño botón numérico (Num Button) aparece al lado del Menú de tipo de IPO en la barra de herramientas de la ventana IPO. Esto indica qué canal de textura está activo. ¡El mapeado para los 8 canales puede ser controlado con Curvas IPO!

Estrictamente hablando, se puede animar de otras dos formas usando texturas. Dado que hay objetos que pueden asignar las coordenadas de texturas a otros objetos (cada objeto en Blender puede ser usado

como una fuente de coordenadas de textura. Para hacer esto, debe seleccionarse la opción objeto (Object) en el botón verde titulado Origen de Coordenadas (Coordinates input) y escribir el nombre del objeto en él. Se realizará una transformación inversa en las coordenadas de renderización globales para obtener las coordenadas *locales* del objeto), es posible animar la textura simplemente animando la posición, tamaño y rotación del objeto.

Además, en cada cuadro (frame), Blender puede cargar otra imagen (numerada) como mapa de texturas en lugar de tener una sola imagen fija. También es posible usar archivos de películas en formato SGI o AVI para hacer esto.

[Prev](#)

Curvas IPO y claves IPO

[Home](#)

[Up](#)

[Next](#)

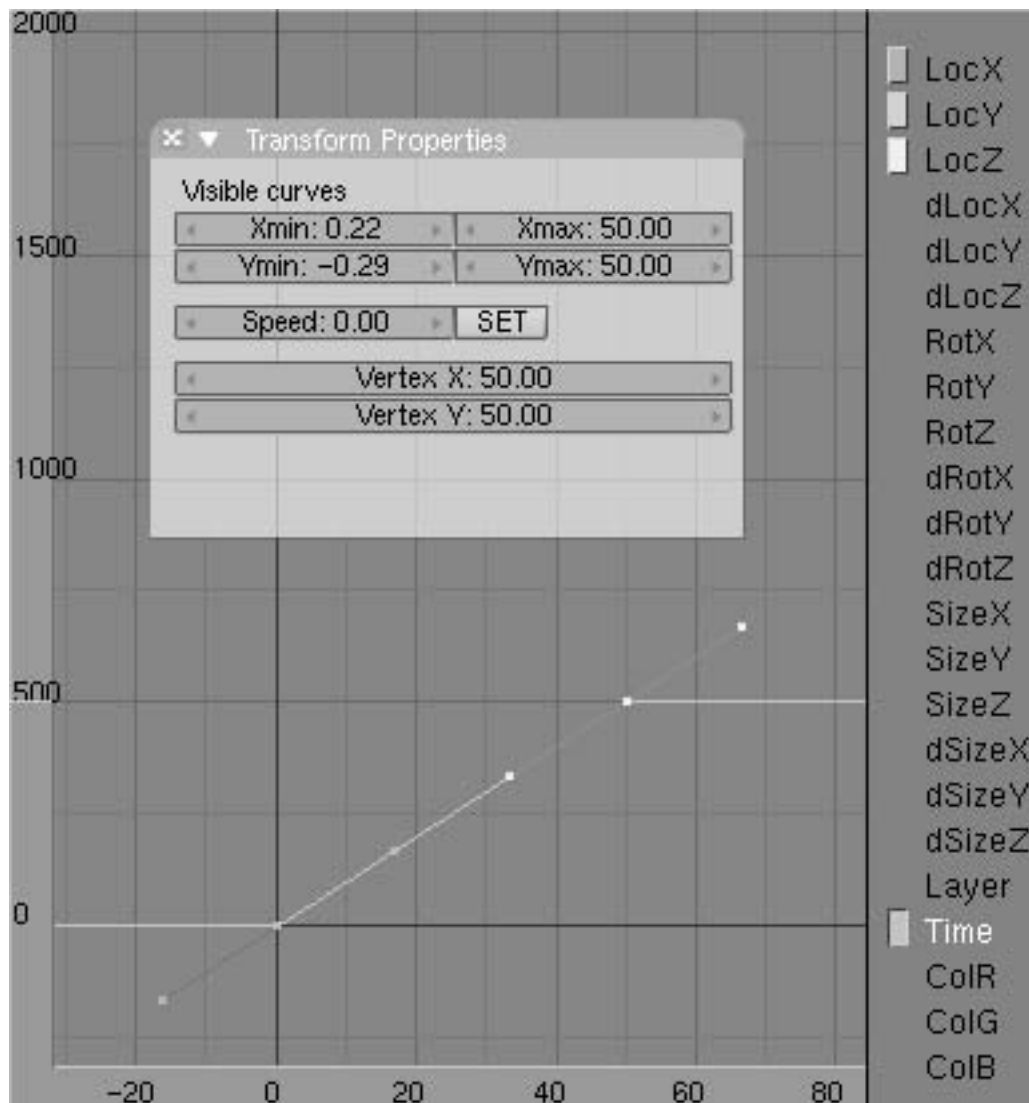
La Ipo del tiempo

14.6. La Ipo del tiempo

Relevante a Blender v2.31

Con la curva Ipo del tiempo (llamada en adelante Ipo-tiempo) puede manipular el tiempo de los objetos en la animación sin variar la animación u otras curvas Ipos. De hecho, dicha curva cambia el mapeado del tiempo de la animación a tiempo global ([Figure 14-7](#)).

Figure 14-7. Curva Ipo-tiempo lineal



Para asimilar este concepto, haga una sencilla animación consistente en mover un objeto, desde una

posición a otra en, digamos, 50 frames (imagenes). Entonces seleccione el canal *Time (Tiempo)* y cree una Ipo-Tiempo en la Ventana Ipo que vaya desde el punto (1,1) al punto (50,50). Es sencillo configurar el comienzo y el final de una IPO usando la **Tecla N** e introduciendo los valores numericamente.

En los frames donde la pendiente de la Ipo-Tiempo sea positiva, su objeto avanzará en la animación. La velocidad depende del valor de la pendiente. Una pendiente más grande que 1 provocará una animación más rápida que la animación base. Una pendiente inferior a 1 provocará que la animación sea más lenta. Una pendiente igual a 1 no producirá cambios en la velocidad de la animación, y una pendiente negativa hará que la animación vaya hacia atrás.

La Ipo-Tiempo es especialmente interesante para sistemas de partículas, permitiendo "congelar" las partículas o animar partículas para que sean absorbidas en lugar de emitidas. Otras posibilidades son crear un lapso de tiempo o una animación a cámara lenta.

Tip Multiples curvas IPO-tiempo

Necesita copiar la curva Ipo-Tiempo para todo el sistema de animación para obtener un movimiento a cámara lenta. Pero deteniendo unicamente algunas animaciones, y continuando la animación, por ejemplo, de la cámara puede conseguir algunos efectos muy interesantes (como esos usados para simular que el tiempo va más lento en la película "The Matrix")

[Prev](#)

Otras aplicaciones de las Curvas
IPO

[Home](#)

[Up](#)

[Next](#)

Trayectoria de la animación

14.7. Trayectoria de la animación

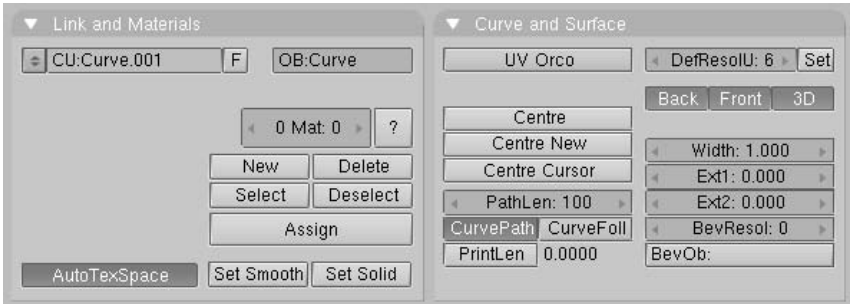
Relevante a Blender v2.31

Una manera diferente para tener objetos moviendose en el espacio consiste en restringir su movimiento para que siga una trayectoria dada.

Cuando los objetos necesitan seguir una trayectoria, o es demasiado difícil realizar alguna clase de movimiento especial con el método del keyframe (Piense en un planeta siguiendo su movimiento alrededor del Sol. Animar esto con keyframes es virtualmente imposible) los objetos curva pueden usarse para indicar la trayectoria de la animación.

Si el objeto Curva contiene más de una curva solo la primera de ellas, en el objeto, será usada.

Figure 14-8. La ventana de acciones con los botones de trayectorias.



Hay dos métodos de animación con trayectorias, el antiguo, anterior a Blender 2.30 que es descrito en esta sección y el método nuevo, el cual actualmente define una restricción, y será descrito [Section 16.8](#). Cuando emparente un Objeto a una Curva tendrá la opción de elegir entre *Normal Parent (emparentado normal)* o *Follow Path (seguir una trayectoria)*. El método antiguo es el que necesita para realizar una tradicional animación que sigue una trayectoria, pero otras acciones necesitarán llevarse a cabo después. La segunda opción crea una restricción del tipo "sigue la trayectoria", y es todo lo que necesita hacer.

Cualquier clase de curva puede convertirse en una trayectoria activando o poniendo a "ON" la opción *CurvePath* en la ventana de botones de animación (**F7**) [Figure 14-8](#).

Cuando una curva tiene hijos, esta puede convertirse a una trayectoria seleccionandola, yendo al contexto de Edición (**F9**) y activando el botón *CurvePath* en el panel "Curve and Surface". Ahora los objetos hijos de la curva se moverán a lo largo de la trayectoria especificada. Es una buena idea convertir la curva en tridimensional a través del botón *3D* de los botones de edición de curvas para que de esta forma la curva pueda ser libremente modelada.

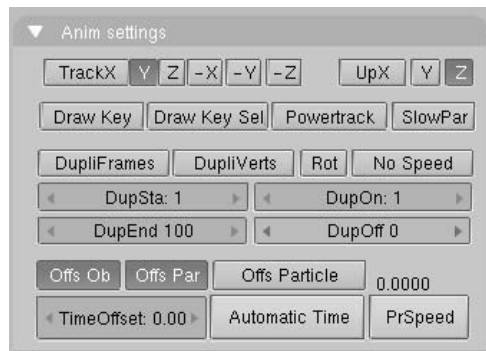
De otra manera, en el menu 'ADD' bajo Curve->Path, existe ya una primitiva con la configuración correcta. Es una curva NURBS de quinto orden, que puede ser usada para crear un muy fluido y continuo movimiento.

Normalmente una trayectoria tiene una longitud de 100 frames y es seguida por el hijo en esos 100 frames. Puede hacerla más larga o más corta variando el botón *PathLength*:

La *velocidad* a lo largo de una trayectoria es determinada con una curva en la ventana IPO. Para visualizarla, en la ventana IPO debe seleccionar el tipo *Curve (curva)* para el bloque IPO. Un único canal, es la *velocidad*. La trayectoria completa recorre en la ventana IPO los valores verticales de cero a uno. Trazar una curva entre esos valores le permitirá enlazar el tiempo a la posición de la trayectoria. Para la mayoría de las trayectorias, una curva IPO debe moverse *exactamente* entre los valores de 0.0 a 1.0 en Y. Para realizar esto, use el menú numérico (**Tecla N**) en la ventana IPO. Si la curva IPO es eliminada, el valor de *PathLen* determina la duración de la trayectoria. En este caso se definirá un movimiento lineal. La velocidad de la IPO es una manera delicada de controlar la longitud de la trayectoria. La trayectoria tiene una longitud de 1 para la IPO-tiempo, y si la IPO-tiempo va de 0 a 1 en 200 frames entonces la trayectoria tiene 200 frames de longitud.

Usando la opción *CurveFollow (Seguir curva)*, en el panel *Curve and Surface*, se proporcionará una rotación a los objetos hijos de la trayectoria, de tal manera que apunten permanentemente en la dirección de la trayectoria. Use los botones "tracking" en el panel *Anim settings* del contexto Objeto (**F7**) para especificar el efecto de la rotación ([Figure 14-9](#)) que vaya hacer para el Tracking:

Figure 14-9. Botones de Tracking



TrackX, *Y*, *Z*, *-X*, *-Y*, *-Z* Esto especifica la dirección del eje, en el ejemplo el eje está situado en la trayectoria.

UpX, *UpY*, *UpZ* Especifica que eje debe apuntar "hacia arriba", en la dirección del eje local positivo Z. Si los ejes *Track* (*pista*) y *Up* (*hacia arriba*) coinciden se desactiva.

Note Las trayectorias tienen el mismo problema que las curvas 'Bevel' en lo concerniente a la definición de la dirección 'hacia arriba'.

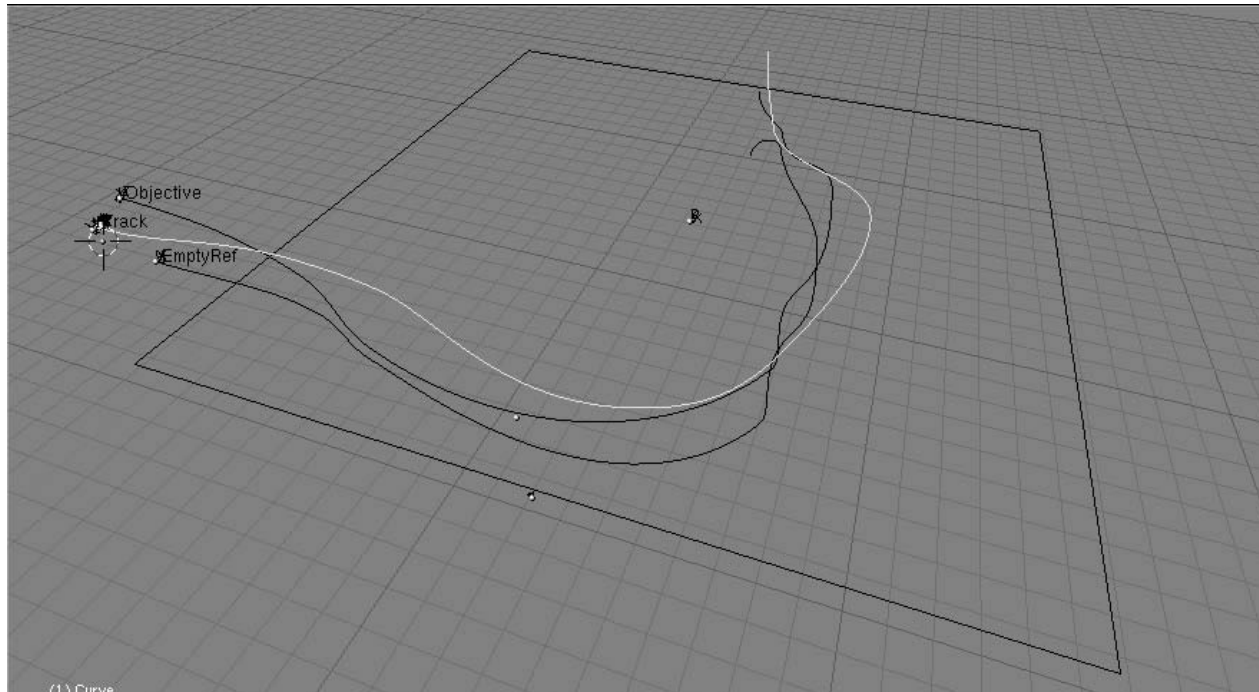
Para visualizar precisamente esas rotaciones, debemos posibilitar que un hijo tenga sus propias rotaciones. Elimine la rotación del hijo con **ALT-R**. También elimine la "Parent Inverse": **ALT-P**. El mejor método es emparentar un hijo sin rotación a la trayectoria con el comando **SHIFT-CTRL-Tecla P**: "Make parent without inverse (Hacer padre sin inversa)". Ahora el hijo salta directamente a la trayectoria y los puntos del hijo están en la dirección correcta .

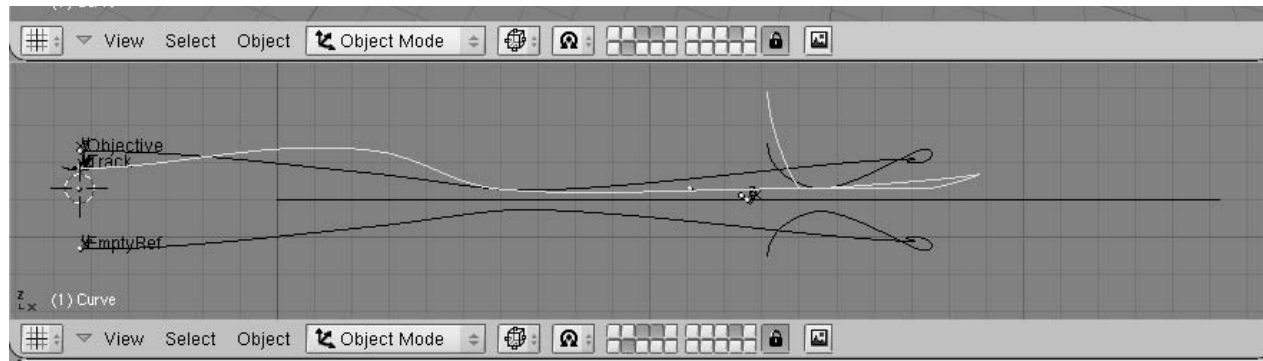
Las trayectorias tridimensionales tienen un valor extra por cada vértice: el 'tilt' (inclinación). Esto puede ser usado específicamente en la rotación de un eje. Use la **Tecla T** en modo edición para cambiar el 'tilt' de los vértices seleccionados en modo edición, por ejemplo para tener un hijo moviéndose alrededor como si fuera una peonza.

La [Figure 14-10](#) muestra una aplicación compleja. Queremos hacer que un avión vaya a través de un cañón, volando próximo al agua y entonces levante el vuelo, todo esto siguiendolo con nuestra cámara y, posiblemente, teniendo reflejos en el agua!

Para conseguir esto necesitaremos tres trayectorias. La trayectoria 1 tiene un avión emparentada a ella, es decir, el avión volará siguiendola.

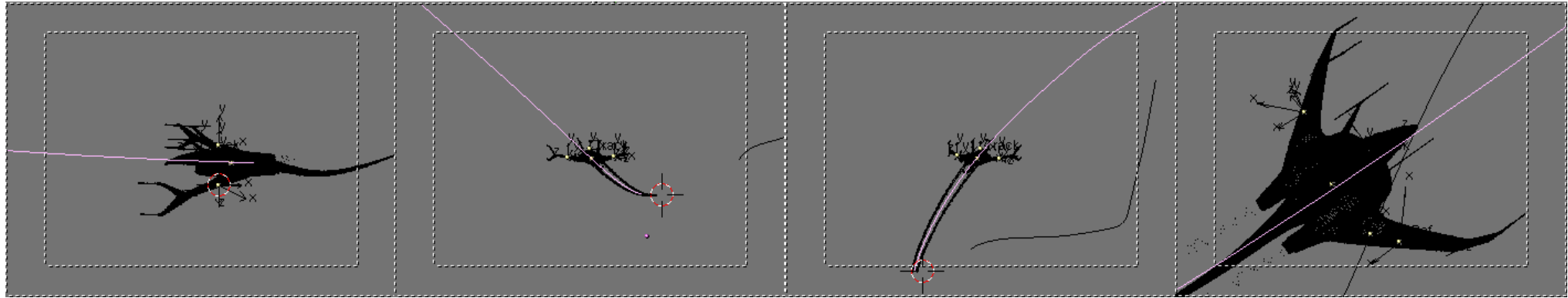
Figure 14-10. Trayectoria compleja de animación





El avión tiene un Empty llamado 'Track (pista)' emparentado a él en una posición estratégica. Una cámara es entonces emparentada a otra curva, la trayectoria 2, y la sigue, siguiendo el Empty 'Track'. El avión tiene la *velocidad* IPO constante, mientras que la cámara no. Primero va rápido, luego despacio y siempre siguiendo el empty, y por lo tanto el avión, así nosotros tendremos unos movimientos muy fluidos de la cámara desde el lateral del avión, al frontal del avión, al otro lado, desde atrás, etcétera ([Figure 14-11](#)).

Figure 14-11. Algunas imágenes de la cámara siguiendo al avión.



Desde que nosotros queremos que nuestro avión sobrevuele un río, necesitaremos configurar un mapa entorno para obtener reflejos de la superficie del agua. Pero el Empty usado para los cálculos siempre tiene que estar en una posición especular con respecto a la cámara... Y la cámara se está moviendo a lo largo de una trayectoria!.

La trayectoria 3 es por lo tanto creada como un espejo de la trayectoria 2 con respecto al plano el agua, duplicándola, y usando la **Tecla M**, en modo edición con respecto a cursor, una vez que cursor esté en el plano.

El Empty para el cálculo del mapa de entorno es entonces emparentado a su nueva trayectoria, y la IPO-tiempo de la trayectoria 2 es copiada a la trayectoria 3. La [Figure 14-12](#) nos muestra una imagen renderizada. Algunos sistemas de partículas se usaron para crear la estela.

La escena presenta muchos trucos pequeños, como las partículas para los motores del avión, la niebla, un cielo añadido a una esfera rodeando la escena y así.

Figure 14-12. Una imagen de la animación final.



[Prev](#)

La Ipo del tiempo

[Home](#)

[Up](#)

[Next](#)

Animación de Deformaciones

Chapter 15. Animación de Deformaciones

Animar un Objeto/Material, no es lo único que se puede hacer en Blender. ¡También se puede cambiar, remodelar, deformar tus objetos según transcurre el tiempo!

Hay muchos modos de hacer esto, una de estas técnicas es tan potente y general que hay un capítulo entero para ella: Animación de Personajes. Las otras técnicas se mostrarán aquí.

15.1. Claves de Vértices Absolutas

Relativo a Blender v2.31

Además de las Claves de Objeto (Objects Keys) en las que se especifican las posiciones de los objetos también se pueden crear en Blender las Claves de Vértice (Vertex Keys) en las que se especifican las posiciones de los vértices *dentro* de un Objeto. Ya que esto podría involucrar miles de vértices, se usa el sistema tradicional de la Clave de Posición. Una Curva IPO sencilla se usa para determinar como se realiza la interpolación y los momentos en los que se puede ver una Clave de Vértice.

Las Claves de Vértice son parte de los datos del objeto no del objeto. Cuando se duplican los datos del objeto, el bloque asociado de Claves de Vértices también se copia. En Blender no se permite que múltiples objetos compartan la misma Clave de Vértices, ya que esto no sería muy práctico.

El bloque de Clave de Vértice es universal y comprende la distinción entre una malla, una curva, una superficie o una retícula. La interfaz y uso está por lo tanto unificada. En esta sección se explica con detalle como trabajar con Claves de Vértice de una malla, que contiene cierto número de comentarios breves sobre los otros Datos del Objeto.

La primera posición de clave de vértice que se crea es siempre la clave de *referencia*. Esta clave define las coordenadas de la textura. Solo si esta Clave está *activa* se pueden cambiar las caras y las curvas, o el *numero* de vértices. Se permite asignar otras claves a un diferente número de vértices. El sistema de Claves las interpola automáticamente.

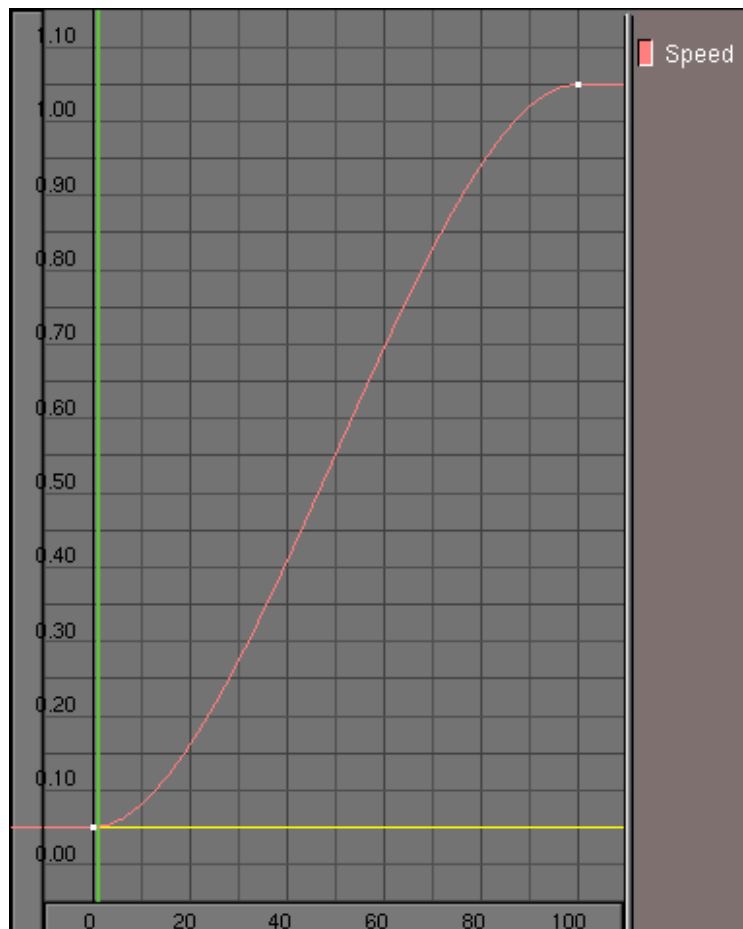
A continuación se da un ejemplo práctico. Cuando se trabaja con Claves de Vértices, es muy cómodo tener una ventana IPO abierta. Usa la primera Pantalla del fichero estándar de Blender, por ejemplo. En la ventana IPO, tenemos que especificar que queremos ver las Claves de Vértices. Para hacer esto usa el Botón de Menú IPO y selecciona *Vertex*. Ve a la Ventana3D con el cursor del ratón y aprieta **IKEY**. Con un objeto Mesh seleccionado y activo. El menú de la "Tecla Insert" tiene varias opciones, y la última es *Mesh*. Tan pronto como esto haya sido seleccionado, aparece un nuevo cuadro de diálogo ([Figure 15-1](#)) preguntando por Claves de Vértices Absolutas o Relativas.

Figure 15-1. Menú de Insertar Clave.



Elegiremos *Absolute Keys*; se dibujará una línea horizontal amarilla en la ventana IPO. Esta es la primera clave y por lo tanto la Clave de *referencia*. También se crea una curva IPO para la "Velocidad" (Speed) ([Figure 15-2](#)).

Figure 15-2. Clave de Referencia y Velocidad IPO.



Tip Creación de Claves de Vértices

Crear Claves de Vértices en Blender es muy simple, pero el hecho de que el sistema sea muy sensible en términos de su configuración puede hacer que ocurran un número de cosas "no visibles". Por lo tanto, la siguiente regla se debe tener en cuenta.

Tan pronto como se inserta una posición de Clave de Vértice, se activa *inmediatamente*. Todos los cambios siguientes en la Malla se enganchan a *esta* posición Clave. Por lo tanto, es importante que la posición de la clave se añada *antes* de comience la edición.

Adelanta unos pocos cuadros y vuelve a seleccionar: **IKEY**, *Mesh* (en la Ventana 3D). La segunda Clave se dibuja como una línea azul claro. Esta es una clave Normal; esta clave y todas las siguientes Claves afectan sólo a la información de Vértices. Aprieta **TAB** para EditMode y traslada uno de los vértices de la malla. Entonces vuelve atrás unos cuadro: ¡no pasa nada! tanto en cuanto estamos en EditMode, *no* se aplican otras Claves de Vértices. Lo que se ve en el ModoEdición es *siempre* la Clave de Vértice *activa*.

Sal del ModoEdición y navega por los fotogramas de nuevo. Ahora se puede ver el efecto del sistema de Claves de Vértice. Las Claves de Vértice sólo se pueden seleccionar en la Ventana IPO. Siempre se debe *salir* del ModoEdición: el 'contenido' de las claves de Vértices se muestran de modo temporal en la malla. Se puede editar la clave especificada entrando en el ModoEdición.

Hay tres métodos para trabajar con Claves de Vértices:

- El método 'animación cronológica'. Este método trabaja enteramente en ModoEdición, cronológicamente de posición en posición:
 - Inserta Clave. Se especifica la referencia.
 - Unos fotogramas después: Inserta Clave. Edita la Malla para la segunda posición.
 - Otros pocos fotogramas después: Inserta Clave. Edita la Malla para la tercera posición.
 - Continúa con el proceso anterior...
- El método 'edición'.
 - Primero insertamos todas las claves que se necesiten, a menos que ya hayamos creado las claves usando el método descrito anteriormente.
 - Blender *no* está en ModoEdición.
 - Selecciona una Clave. Ahora entra en ModoEdición, cambia la Malla y sal del ModoEdición
 - Selecciona una Clave. Entra en ModoEdición, cambia la Malla y sal del ModoEdición.
 - Continúa el proceso anterior...
- El método 'inserción'
 - En este método no importa si ya se han creado Claves o si estamos en ModoEdición.
 - Se va hasta el fotograma en el que se quiere insertar la nueva Clave.
 - Se inserta la Clave.
 - Se va a un nuevo fotograma, se inserta la Clave.
 - Continúa con el proceso anterior...

Mientras se está en el ModoEdición, las Claves no se pueden cambiar . Si el usuario intenta hacerlo, aparecerá un aviso.

Cada Clave se representa por una línea que se dibuja a cierta altura. La altura será de modo que la clave intersecte la "Velocidad" IPO en el fotograma en el que se encuentra la Clave.

Tanto la Curva IPO y la Clave de Vértice pueden seleccionarse separadas con **RMB**. Ya que sería muy difícil de otro modo trabajar con ellas, la selección de las líneas de las Claves está desactivada cuando la curva está en Modo Edición. El botón *channel* se puede usar para ocultar temporalmente la curva (**SHIFT-LMB** on "Speed") para hacer más fácil la selección de Claves.

Las líneas de Claves en la Ventana IPO, una vez se toman, pueden situarse en cualquier posición vertical. Selecciona la línea y usa el Modo Grab para hacerlo. La Curva IPO también puede procesarse aquí del mismo modo que se describe en el capítulo anterior. En vez de un 'valor', la curva determina la interpolación entre las Claves, e.g. una curva sinodal se puede usar para crear una animación cíclica.

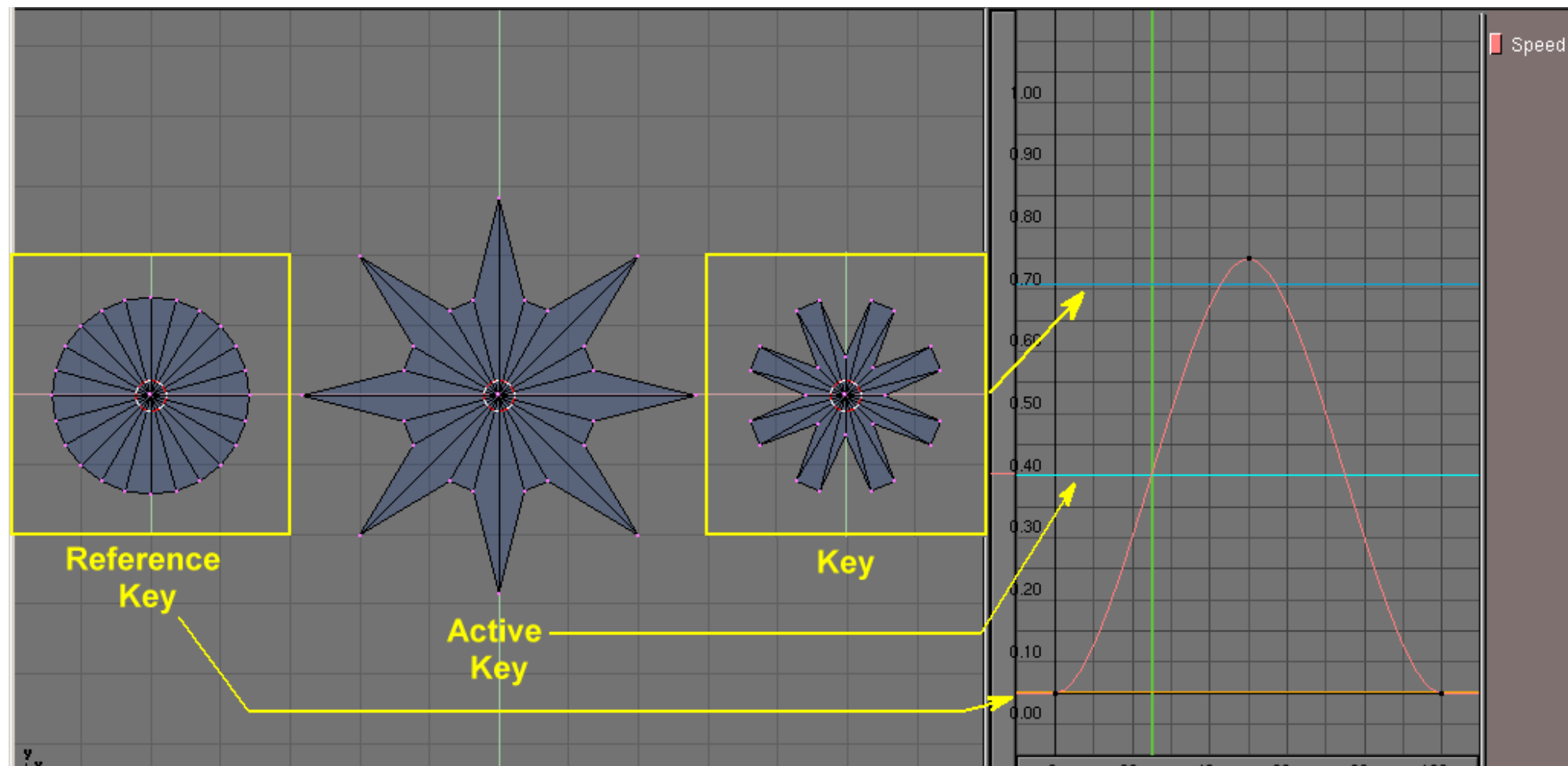
Durante la animación el contador de fotogramas muestra un cierto valor de la velocidad IPO, que se usará para elegir la(s) Clave(s) que se va(n) a usar, posiblemente con interpolación, para producir la malla deformada.

La Velocidad IPO tiene el comportamiento estándar de un IPO, también para interpolar. La línea de Clave tiene tres tipos distintos de interpolación. Presiona **TKEY** con una línea de Clave seleccionada para abrir un menú con las opciones:

- *Lineal*: La interpolación entre las Claves es lineal. La línea de Clave se muestra como una línea punteada.
- *Cardinal*: La interpolación entre las Claves es fluida, la asignación estándar.
- *BSpline*: La interpolación entre las Claves es extrafluida e incluye cuatro Claves en el cálculo de la interpolación. Las posiciones no se muestran de modo preciso. La línea de Clave se dibuja como una línea de guiones.

Figure 15-3 muestra una animación simple de Vértices de un cilindro. Cuando se ejecuta el cilindro se convierte en una gran estrella, entonces se deforma a una pequeña estrella, entonces ya que la Velocidad IPO retrocede a 0 la deformación se repite en orden inverso.

Figure 15-3. Claves Absolutas.



Algunas pistas útiles:

- Las posiciones de las Claves se añaden *siempre* con IKEY, incluso si no se encontrasen en la misma posición. Esto se usa para copiar posiciones mientras se inserta. Se pueden usar dos líneas en la misma posición para cambiar el efecto de la interpolación.
- Si *no* se seleccionan Claves, el ModoEdición se puede invocar como es normal. En cualquier caso, cuando se deja el ModoEdición, se deshacen todos los cambios. Inserta la Clave *en* el ModoEdición en este caso.
- Para las Claves, *no* hay diferencia entre seleccionada y *activa*. Por lo tanto, no es posible seleccionar múltiples Claves.
- Cuando se trabaja con Claves con diferente número de vértices, las caras se pueden desordenar. No hay herramientas para especificar de un modo *preciso* una secuencia de vértices. Actualmente esta opción está sólo asequible para Mallas que tienen vértices como Halos.

15.1.1. Claves de Curva y Superficie

Como se mencionó anteriormente, las Claves de Curva y Superficie trabajan exactamente igual que las Claves de Malla. Para las Curvas, es particularmente interesante localizar las Claves de Curva en el objeto bevel. Aunque esta animación *no* se mostrará en tiempo real en la ventana 3D, si será renderizada.

15.1.2. Claves de Retícula

Tan pronto como una Clave esté presente en una Retícula, los botones que se usan para determinar la resolución se bloquean.

[Prev](#)

Trayectoria de la animación

[Home](#)

[Up](#)

[Next](#)

Claves de Vértices Relativas

15.2. Claves de Vértices Relativas

Para Blender v2.31

Las Claves de Vértices Relativas (RVK) trabajan de modo diferente tanto en cuanto que solo se almacena la diferencia entre la malla de referencia y la malla deformada. Esto permite doblar varias claves juntas para conseguir animaciones complejas.

Pasearemos por RVK por medio de un ejemplo.

Crearemos una animación facial con RVK. Mientras que las Claves de Vértice absolutas se controlan con sólo *una* curva IPO, las Claves Relativas de Vértices se controlan con una curva de interpolación por cada posición de clave, lo que pone de manifiesto 'cuanta' de esa deformación relativa se usa para producir la malla deformada. Es por esto que las claves relativas se pueden mezclar (sumar, restar, etc.).

Para la animación facial, la base de partida podría ser una posición relajada con una boca ligeramente abierta y los párpados medio abiertos. Entonces las claves podrían definirse para los ojos izquierdo/derecho cerrado, felicidad, tristeza, sonriendo, frunciendo el ceño, etc.

El truco con las claves de vértice relativas es que sólo los vértices que han cambiado entre la base y la clave afectan la salida final de la deformación. Esto significa que es posible tener varias claves que afecten al objeto en diferentes sitios todos al mismo tiempo.

Por ejemplo, una cara con tres claves: sonreír, el ojo derecho cerrado y el ojo izquierdo cerrado podría ser animado para sonreír, entonces cerrar el ojo izquierdo, después cerrar el derecho, entonces abrir los dos ojos y finalmente dejar de sonreír - todo deformando tres claves. Sin las claves relativas de vértice se hubiesen necesitado seis claves de vértice, una para cada posición intermedia.

Consideremos la cabeza femenina de [Figure 15-4](#)

Figure 15-4. La cabeza femenina que queremos animar.

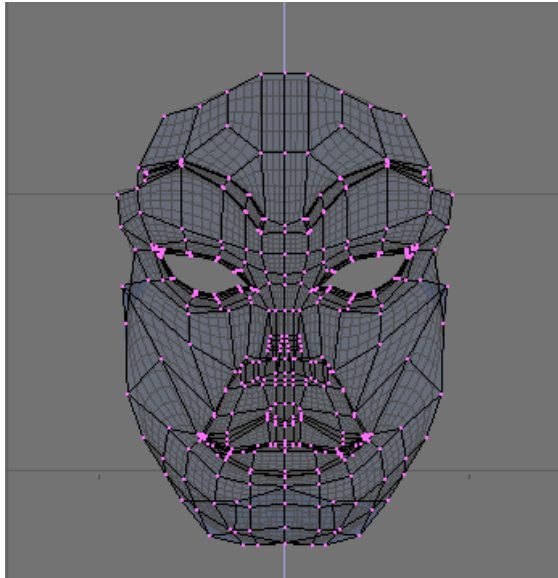


Para añadir RVK se aprieta **IKEY** y se selecciona *Mesh(Malla)* igual que para las claves absolutas, pero, desde el menú pop up selecciona *Relative Vertex Keys*. Esto almacena la clave de referencia y aparecerá como una raya horizontal amarilla en la ventana IPO.

La Claves Relativas se definen insertando otras claves de vértice. Cada vez que se apriete **IKEY** y que se seleccione *Mesh* aparecerá una nueva línea horizontal en la ventana IPO. Si el número de fotograma se incrementa

cada vez la línea horizontal se sitúa por encima de la anterior. Para simplificar el modelado vamos a ocultar todos los vértices excepto los de la cara [Figure 15-5](#).

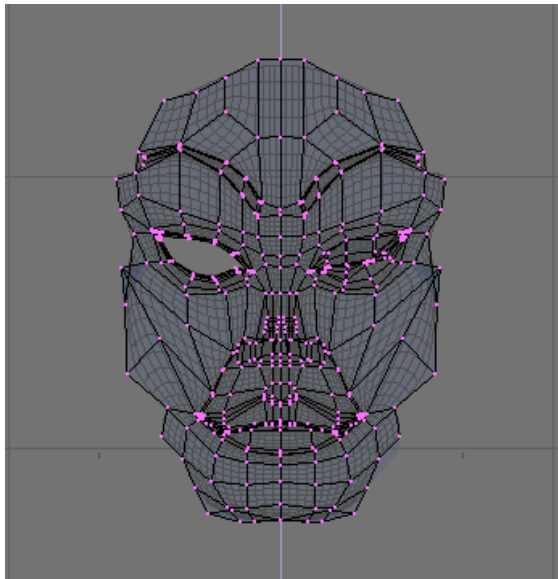
Figure 15-5. Todos los vértices ocultos salvo los de la cara.



Ahora nos movemos a otro fotograma, digamos que el número 5, y añadimos una nueva Clave. Una línea cian aparecerá por encima de la amarilla, que ahora estará naranjada. Cambiamos a modo Edición y cerramos el párpado izquierdo.

Cuando salgamos del modo Edición. Si se selecciona la Clave de Referencia se verá la malla original. Si se selecciona la primera Clave de Vértice Relativa se verá la malla deformada ([Figure 15-6](#)).

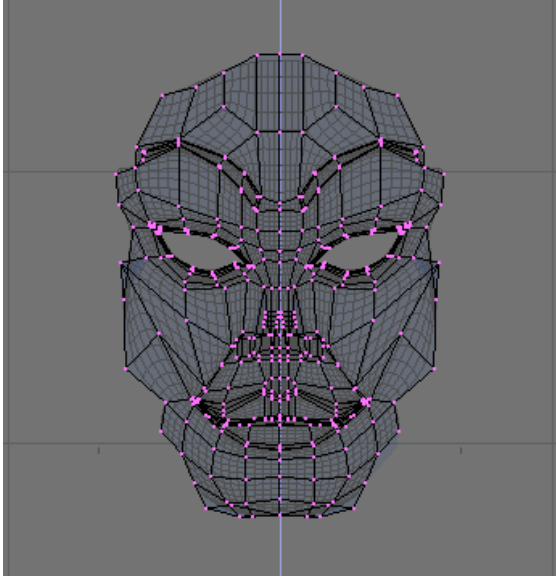
Figure 15-6. Ojo izquierdo cerrado.



Repitamos el paso anterior para el ojo derecho. Hay que tener cuidado de que la clave nueva que se inserta se basa en la malla de la clave que está actualmente como activa, así que en general es buena idea seleccionar la clave de referencia antes de apretar **IKEY**.

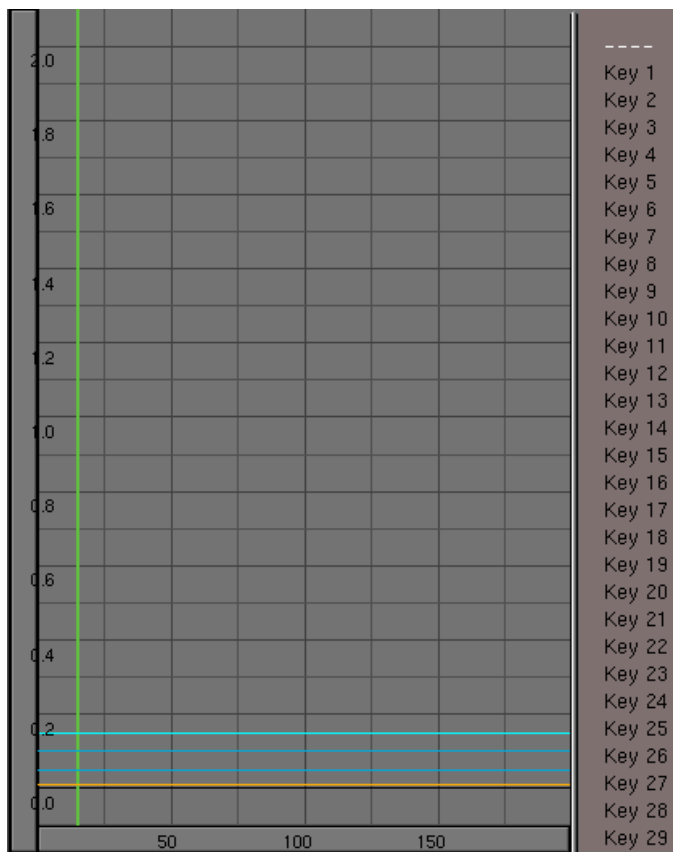
Ahora añadimos una sonrisa ([Figure 15-7](#)).

Figure 15-7. Sonriendo.



La ventana IPO debe ser algo como [Figure 15-8](#).

Figure 15-8. Claves en la Ventana IPO.

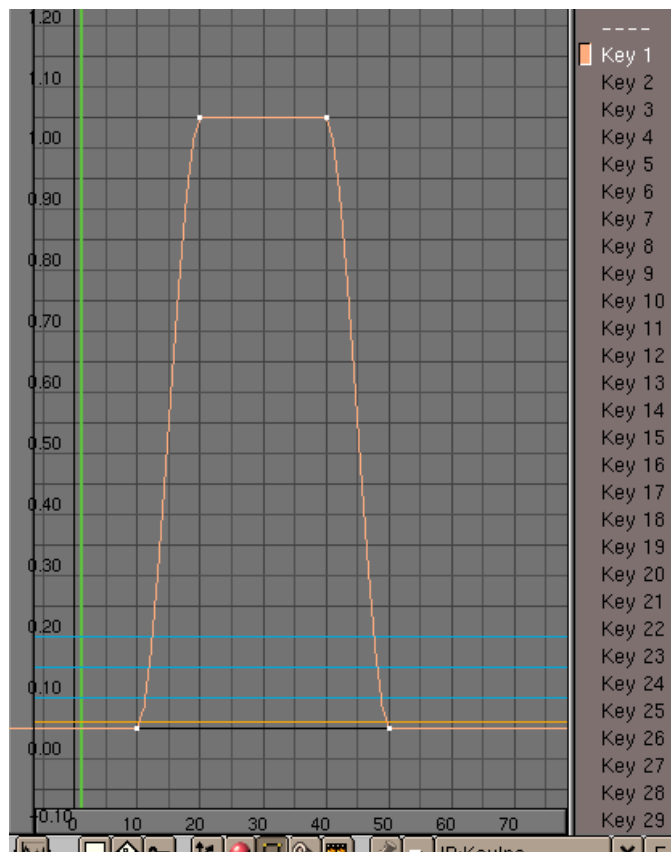


El orden vertical de las Claves de Vértice (Las líneas azules) de abajo a arriba determina su curva IPO correspondiente, i.e. la línea de clave azul más baja será controlada por la curva *Key1*, la segunda más baja será controlada por la curva *Key2*, y así sucesivamente.

No hay IPO para la malla de referencia ya que es la malla que se utiliza cuando cualquier otra clave tiene un IPO con valor cero en determinado fotograma.

Selecciona *Key1* y añade un IPO con tu método favorito. Haz que se parezca algo a [Figure 15-9](#).

Figure 15-9. La curva IPO de la clave 1.



Esto hará que nuestra malla no se deforme hasta el fotograma 10, entonces desde el 10 hasta el 20 la Clave 1 (Key 1) empieza a afectar a la deformación. Del fotograma 20 al fotograma 40 la Clave 1 sustituye completamente a la malla de referencia (el valor de IPO es 1), y el ojo estará completamente cerrado. El efecto irá desapareciendo del fotograma 40 al 50.

Se puede comprobar con **ALT-A**, o asignado los números de fotograma a mano. La segunda opción es mejor, a menos que el ordenador se muy potente.

Copia este IPO usando el botón de la flecha hacia abajo en la barra de herramientas de la Ventana IPO ([Figure 15-10](#)). Selecciona la Key 2 (Clave 2) y pega la curva con la flecha hacia arriba. Ahora ambas claves tendrán la misma influencia en la cara y ambos ojos se cerrarán al mismo tiempo.

Figure 15-10. Botones de copy/paste.

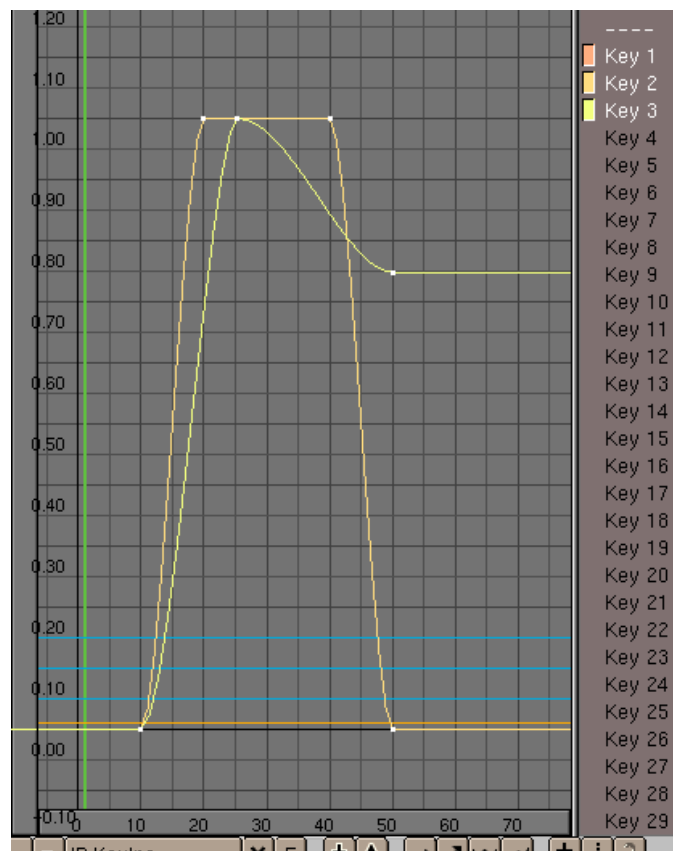


Tip Desplazando la Barra de Herramientas

Puede pasar que la barra de herramientas sea más larga que la ventana y algunos de los botones no se vean. Se puede desplazar cualquier barra de herramientas horizontalmente It may happen that the toolbar is longer than the window and some buttons are not shown. You can pan horizontally pulsando el botón del medio y arrastrándola con el ratón.

Añade también un IPO para la Key 3. Pero hagamos esta diferencia ([Figure 15-11](#)).

Figure 15-11. Todos los IPOs.



De esta forma los ojos se cierran y ella comienza a sonreír, la sonrisa está al máximo con los ojos cerrados, entonces ella sonríe 'menos' mientras los ojos se vuelve a abrir y se queda sonriendo ([Figure 15-12](#)).

Figure 15-12. Sequence.

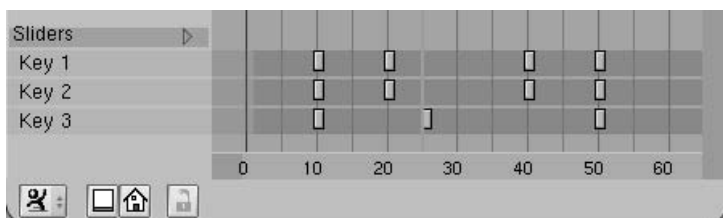


La curva IPO de cada Clave controla la deformación entre las claves relativas. Estas curvas deben crearse del modo típico. La posición final se determina añadiendo todos los efectos de cada curva IPO.

Tip RVK en la Ventana de Acción

También se puede operar con RVK en la Ventana de Acción (**SHIFT-F12**), no IPO ([Figure 15-13](#)). La influencia de cualquier Clave se da por medio de un desplazador(slider). Hay marcas en los puntos de las Claves (i.e. donde la IPO tendría un punto de control).

Figure 15-13. RVK en la Ventana de Acción.



Tip Valores fuera del rango [0,1]

Una parte importante de las Claves Relativas es el uso de posiciones aditivas o extrapoladas. Por ejemplo, si la posición base para una cara es con la boca recta, y una clave se define para sonreír, entonces es posible que la aplicación negativa de la clave resulte en curvar la boca al contrario de una sonrisa. En el caso de extendamos la Curva IPO por encima de 1.0 se "extrapolará" esa clave, construyendo una sonrisa extrema.

[Prev](#)

Animación de Deformaciones

[Home](#)

[Up](#)

[Next](#)

Animación de Retículas

15.3. Animación de Retículas

Relativo a Blender v2.31

Emparentar una malla a una retícula es una buena forma de aplicar deformaciones a la malla mientras se modela, pero ¡también es la forma de hacer las deformaciones en el tiempo!.

Se pueden usar Retículas en animaciones de dos maneras:

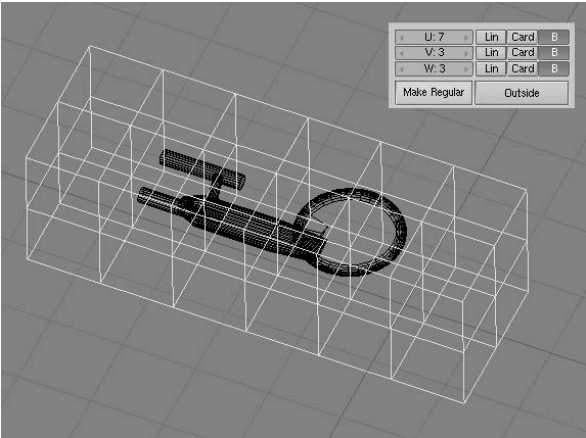
- Animar los vértices con claves de vértice (o claves de vértice relativas);
- Mover la retícula o el objeto hijo de la retícula.

La primera técnica es básicamente lo contenido en las dos secciones previas pero aplicado a la retícula que tiene un objeto emparentado.

Con la segunda clase se pueden crear animaciones que aplaste cosas entre rodillos, o para obtener el bien conocido efecto de la aceleración de una nave espacial a la velocidad de doblado.

Hacemos una nave espacial y añadimos una retícula alrededor de la nave. Hacemos la retícula con los parámetros de [Figure 15-14](#).

Figure 15-14. Configuración de la Retícula



Selecciona la nave, extiende la selección a la retícula (manteniendo apretada la tecla **SHIFT** mientras seleccionamos), y presionando **CTRL-P** para hacer la retícula el objeto padre de la nave. No se debería ver ninguna deformación de la nave por que la retícula todavía es regular.

Para los próximos pasos es importante que los hagamos en Modo Edición. Así que ahora se selecciona la retícula, se entra en Modo Edición, se seleccionan todos los vértices (**AKEY**), y se escala la retícula a lo largo de su eje x (presiona el botón del medio **MMB** mientras se inicia el escalado) para conseguir el estrechamiento que se quiere. La malla de la nave muestra inmediatamente la deformación causada por la retícula ([Figure 15-15](#)).

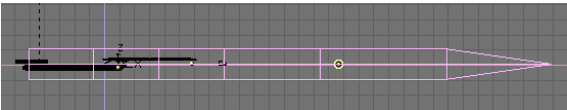
Figure 15-15. Estrechando



Ahora edita la retícula en Modo Edición de forma que los vértices de la derecha tengan una distancia mayor entre ellos. Esto incrementará el estrechamiento así como la nave va dentro de la retícula. Los vértices del extremo de la derecha se han escalado al mínimo de manera que prácticamente están en un punto; esto causará el desvanecido de la nave al final ([Figure 15-16](#)).

Selecciona la nave de nuevo y muevela a través de la retícula para ver un previo de la animación. Ahora ya se puede hacer una animación normal con claves de fotograma para conseguir que la nave vuele a través de la retícula.

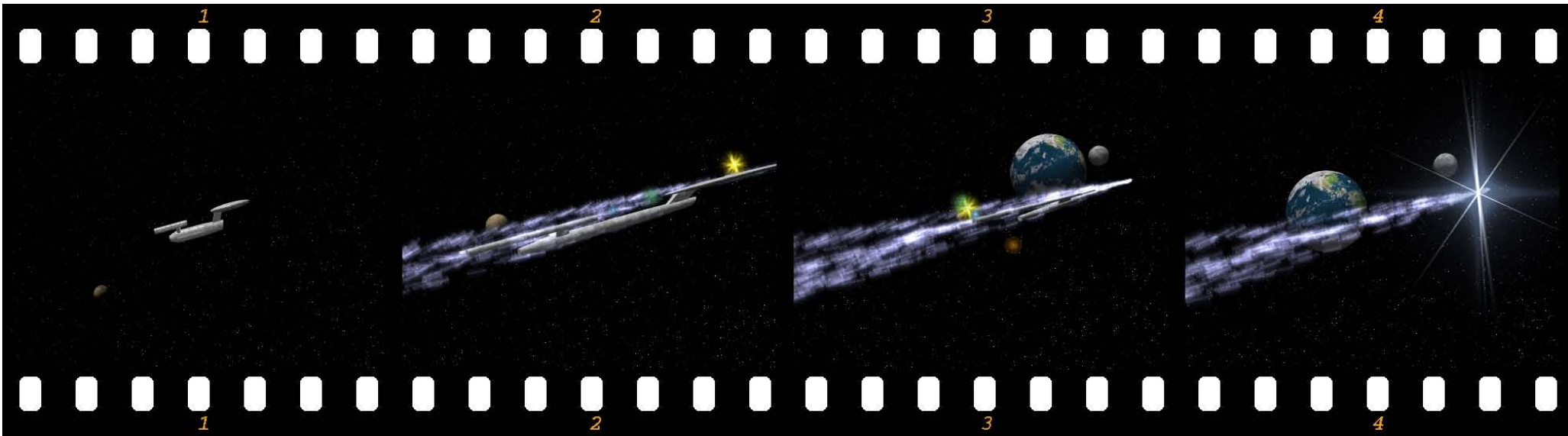
Figure 15-16. Deformación Final de la Retícula



Tip Seguimiento de la Cámara

Con esta animación de retícula, no se puede usar el punto de pivote del objeto para seguimiento (tracking) o emparentamiento (parenting). Se movería fuera del objeto. Para eso se necesita emparentar los vértices de un Vacío(Empty) a la malla. Para hacerlo, selección el Vacío(Empty), después la malla, entra en Modo Edición y selecciona un vértice, entonces presiona **CTRL-P**.

Figure 15-17. Algunos fotogramas de la animación resultante.



[Prev](#)

Claves de Vértices Relativas

[Home](#)

[Up](#)

[Next](#)

Animación de Personajes

Chapter 16. Animación de Personajes

traducción por Claudio "malefico" Andaur

16.1. Introducción: Luces, Cámaras y... ACCION !

Introducción por Claudio "malefico" Andaur

Como hemos visto en [Section 4.5](#) Blender utiliza *Armatures* para la animación de personajes . Un armature es solo un esqueleto el cual una vez que hemos emparentado nuestra malla de persona al mismo, nos permite definir una serie de *poses* a lo largo de la línea de tiempo de nuestra animación.

Un armature está compuesto de un número arbitrario de *huesos*. El tamaño, la posición y orientación de cada hueso en el armature depende de ti, y a lo largo de este capítulo encontrarás distintas situaciones que requerirán un arreglo particular de huesos para que tu personaje funcione correctamente.

A medida que animes tu armature verás que es mucho mejor organizar las distintas poses en algo llamado una *action*, lo que es más o menos lo mismo que sucede en el mudo real. Cuando caminamos podemos imaginarnos atravesando distintas poses instantáneas como si fueran los cuadros de una película, el proceso entero de la caminata es una acción a fin de cuentas. Pero hay acciones y acciones. Como animador necesitarás desarrollar la habilidad de reconocer cómo dividir u movimiento natural o acción en acciones más simples con las que sea más sencillo trabajar. Usar acciones más simples normalmente ahorra tiempo y trabajo (y por qué no: dinero !) ya que es común que estas acciones sean reutilizables en varias animaciones.

Una vez que hayas creado tus primeras acciones, podrás combinarlas utilizando el poderoso editor de *Animación No Lineal* (o *NLA*), de Blender, otorgándole a tu personaje un aspecto vivo y natural.

En este capítulo cubriremos cada detalle de las funcionalidades de Blender relativas a Armatures, Acciones y el Editor NLA. Incluso veremos varias configuraciones de Armatures que te darán un punto de partida para aplicar a tus propias creaciones. Relajate y disfruta.

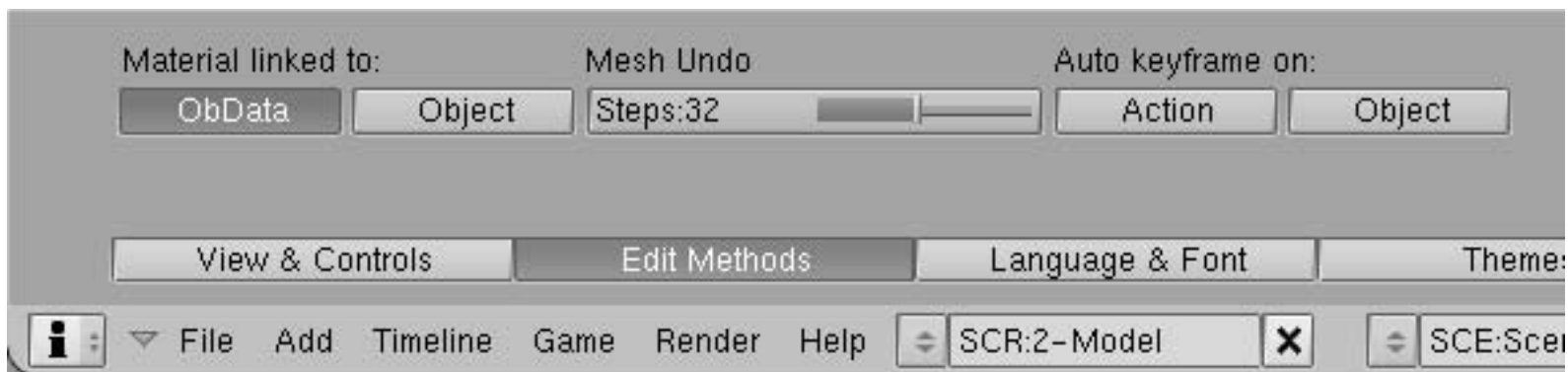
16.2. Herramientas Generales

Relativo a Blender v2.31

Hay algunas características de Blender que pueden facilitarnos la vida cuando animamos un personaje. Veámoslas antes de profundizar en los detalles.

En la Ventana de Información podemos encontrar la opción auto-key. Cuando está activa, Blender insertará keyframes automáticamente cuando movamos un objeto. Esto es útil sobre todo para aquellos que no están acostumbrados a insertar explícitamente keyframes con **IKEY**. Hay dos modos separados para el auto-key: uno para Modo Objeto y otro para Modo Pose. Estas dos opciones pueden activarse independientemente una de otra desde el grupo de botones *Edit Method* en la ventana de Preferencias del Usuario ([Figure 16-1](#)).

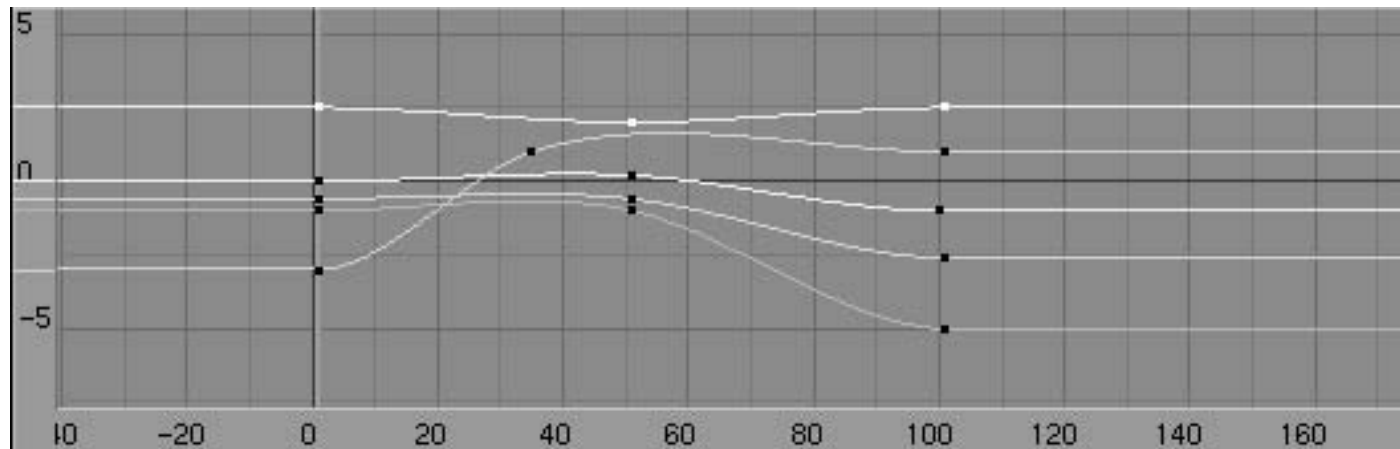
Figure 16-1. Opciones de Auto key



Auto Keyframe sobre un *Objeto* insertará KeyFrames para todos los objetos que sean movidos en Modo Objeto . Aquellos usuarios familiarizados con la interfaz de Blender probablemente querrán mantener esta opción desactivada.

Auto Keyframe sobre una *Acción* insertará KeyFrames para transformaciones hechas en Modo Pose. Esto nos garantizará que no perdamos una pose por olvidar insertar un Keyframe. Incluso aquellos usuarios familiarizados con la interfaz de Blender encontrarán de utilidad esta característica.

Es posible mostrar diferentes IPOs en ventanas deferentes. Esto es especialmente valioso cuando editamos Acciones, que tienen una IPO diferente para cada hueso.

Figure 16-2. VentanaIPO de Acción Clavada

Puedes "clavar" una IPO o acción (fijarla en la ventana actual) presionando el icono de la tachuela en la cabecera de la ventana ([Figure 16-2](#)). Los contenidos de la ventana permanecerán allí aún cuando el objeto haya sido deseleccionado u otro objeto sea seleccionado. Observar que el color del menú de bloque IPO cambiará, junto con el color del fondo de la ventana IPO. Esto sirve como recordatorio de que esa ventana no está necesariamente mostrando la IPO del objeto seleccionado actualmente.

El menú browse sigue disponible cuando una ventana está "clavada". En este caso sin embargo, cambiar los datos actuales no afectará al objeto actual; meramente cambia los datos mostrados.

[Prev](#)

Animación de Personajes

[Home](#)
[Up](#)
[Next](#)

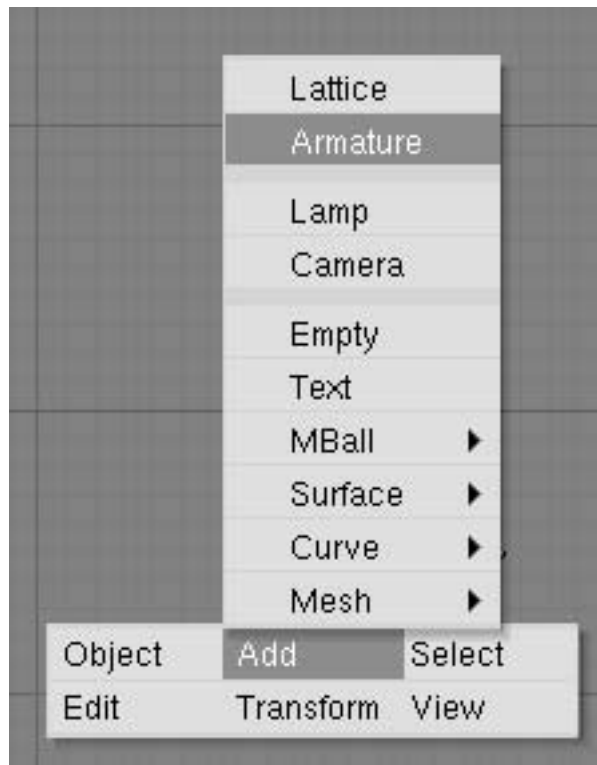
El Objeto Armature

16.3. El Objeto Armature

Relevante a Blender v2.31

El objeto Armature es el objeto básico para animación de personajes. Es un objeto que consiste en una serie de "huesos" que pueden o no estar interconectados. Una serie de huesos interconectados puede ser una "Cadena de Cinemática Inversa (IK)" o simplemente una "Cadena" de huesos. Una Cadena IK es algo más complejo que una relación Parent estándar en el sentido que no solo los movimientos del hueso "Padre" son transmitidos a los huesos "hijos", sino que los movimientos del último "hijo" en la cadena son transmitidos a lo largo de la cadena hasta el hueso "padre" si se utilizara una Solución de Cinemática Inversa. Los huesos pueden moverse como si fuesen un grupo de Objetos rígidos indomables con uniones articuladas perfectas. Piensa en un armature como si se tratara de un esqueleto de una criatura viva. Los brazos, piernas, columna y cabeza son todas partes del mismo objeto esqueleto.

Figure 16-3. Agregando un Armature



Para crear un nuevo armature, selecciona **SPACE>>Add>>Armature** de la barra de herramientas ([Figure 16-3](#)). Aparecerá un nuevo hueso con su raíz en la posición del cursor 3D. A medida que muevas

el ratón, el hueso cambiará de tamaño correspondientemente. **LMB** finalizará el hueso y comenzará otro nuevo como hijo del anterior. De esta manera puedes crear una cadena completa. Pulsando **ESC** cancelarás la creación de un nuevo hueso.

Puedes agregar otro hueso al armature mientras estés en Modo Edición con **SPACE>>Add>>Armature** otra vez en la caja de herramientas. Esto empezará otra vez el modo de agregar huesos y los nuevos huesos serán parte de la armature actual pero formarán una cadena separada

También puedes "extrudar" huesos de los huesos existentes seleccionando el extremo de un hueso y presionando **EKEY**. El hueso recién creado será hijo del hueso de donde fue extrudado pero *no* de su cadena IK.

Estando en Modo Edición puedes realizar las siguientes operaciones sobre los huesos de un armature.

- *Ajuste* - Selecciona uno o mas extremos de huesos y utiliza cualquiera de las operaciones de transformación estándar para ajustar la posición y orientación de cualquier hueso en el armature. Notar que las cadenas IK no pueden tener ninguna separación entre los huesos que la componen por lo que mover el extremo final de un hueso moverá también el extremo inicial, o *raíz* de sus hijos.

Se puede seleccionar una cadena IK completa en un solo paso moviendo el puntero del ratón sobre un extremo de hueso en la cadena y pulsando **LKEY**. También se puede usar la herramienta de selección por límites (**BKEY**).

- *Borrado* - Se pueden borrar uno o más huesos seleccionando el comienzo y final de los mismos. Al hacer esto observarás que el hueso entero se resaltará con un color. Pulsando **XKEY** se borrarán los huesos resaltados. Observar que seleccionando solo uno de los extremos no es suficiente para borrar el hueso..
- *Ajuste a punto* - Es posible ajustar los extremos de un hueso a la cuadrícula o al cursor usando el menú Snap accesible con **SHIFT-S**.
- *Modo Numérico* - Para una edición más precisa, pulsando **NKEY** aparecerá la caja de diálogo numérica. Aquí se puede ajustar la posición de los extremos iniciales y finales así como la rotación del hueso sobre su propio eje.

Una forma sencilla de orientar automáticamente el eje Z de todos los huesos seleccionados (necesario para un correcto funcionamiento de la opción "pose invertida") es pulsar **CTRL-N**. No olvides hacer esto antes de empezar cualquier animación del armature.

- *Deshacer* - Estando en Modo Edición puedes cancelar los cambios realizados en la sesión de edición actual presionando **UKEY**. El armature se revertirá al estado anterior al comienzo de la edición.

También se pueden unir dos armatures juntos en un único Objeto. Para hacerlo, asegurate de estar en Modo Objeto, selecciona ambos armatures y pulsa **CTRL-J**.

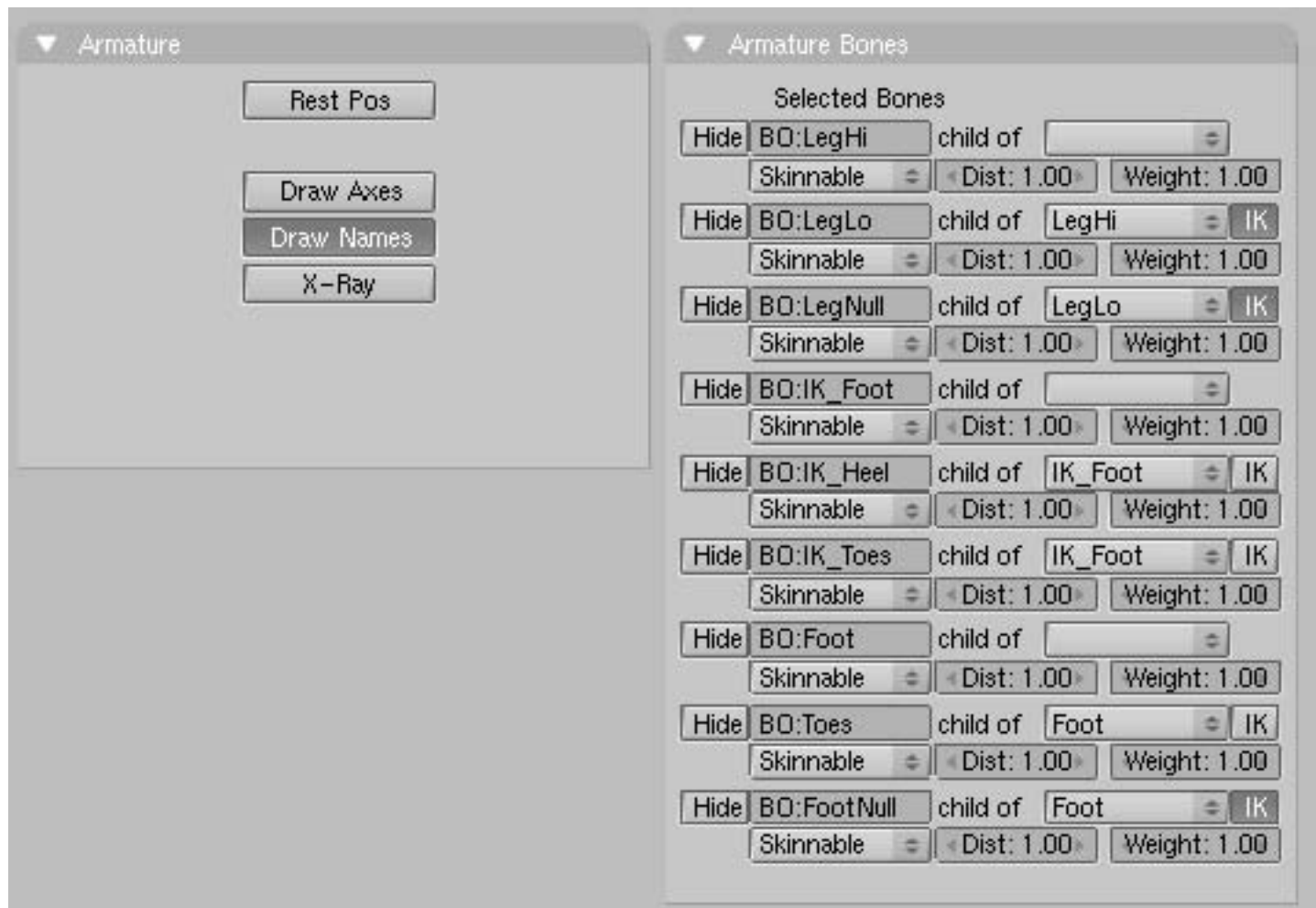
16.3.1. Nombrando Huesos

Es importante asignar nombres significativos a los huesos por varias razones. Primero que nada, hará las cosas más fáciles cuando edites acciones en el Editor de Acciones. Segundo, los nombres de los huesos se utilizan para asociar canales de Acción a los huesos cuando intentes reutilizar Acciones, y tercero los nombres se usan para aprovechar la ventaja de la opción de pose inversa automática..

Fíjate que los nombres de los huesos solo necesitan ser únicos dentro de una unidad armature. Puedes tener varios huesos llamados "Cabeza" siempre y cuando se encuentren en distintas armatures.

Para cambiar los nombres de uno o más huesos, selecciona los huesos en Modo Edición y cambia a los botones de Contexto Edición con **F9**. Una lista de todos los huesos seleccionados debería aparecer en el Panel *Armature Bones* ([Figure 16-4](#)). Cambia el nombre de un hueso haciendo **SHIFT-LMB** en la caja con el nombre del hueso y escribe un nuevo nombre.

Figure 16-4. Botones de Edición para un Armature



Es más sencillo nombrar los huesos editandolos de a uno a la vez o asegurándose de activar la opción

DrawNames en los botones de edición **F9** ([Figure 16-5](#)).

Tip Convenciones de Pose Invertida

Típicamente, las armatures de personajes son axialmente simétricas. Esto significa que muchos elementos se encuentran de a pares, unos sobre la izquierda y otros sobre la derecha. Si nombras correctamente Blender puede invertir una pose dada alrededor del eje de simetría facilitando la tarea de animar ciclos de andar, por ejemplo.

Para cada hueso que tenga par, añádele un sufijo a los nombres para distinguir el par izquierdo y el derecho. Puede ser tanto ".L" y ".R" como ".Left" y ".Right". Nota: No funcionará escribirlo en español, y dará lo mismo si ".L" es realmente la izquierda o la derecha. Los huesos que estén sobre el eje de simetría o que no tengan un homólogo no lo necesitan. La parte del nombre que precede al sufijo debe ser idéntica para ambos lados . Así por ejemplo si tenemos dos manos, deberían llamarse "Mano.R" y "Mano.L".

16.3.2. Emparentado y cadenas IK

Para modificar relaciones de emparentado dentro del armature, selecciona el hueso que debería ser *hijo* y entra al Panel *Armature Bones* de la ventana de Botones de Edición. Al lado del nombre del hueso verás una lista desplegable con la etiqueta *Child Of*. Para convertir este hueso en hijo de otro, elige de la lista el hueso que será padre. Observa que esto es mucho más sencillo si los huesos han sido nombrados correctamente. Para eliminar una relación de emparentado, elige la opción en blanco de la lista.

Observa que el menú de emparentado solo contiene los nombres de padres válidos. Aquellos huesos que no pueden ser padres (por ejemplo los hijos del hueso actual) no serán mostrados.

El botón IK junto a cada hueso que tiene padre, se usa para determinar si la solución IK deberá propagar sus efectos a través de esta articulación. Si el botón IK está activo, el extremo final del padre se moverá para coincidir con el extremo inicial del hijo. Esto ocurre para cumplir con el requerimiento de que no debe haber brechas o espacios en una cadena IK. Desactivando el botón IK no se restaurará la posición del extremo inicial del hijo a su posición original, pero mover esa articulación no afectará ya al extremo final del padre.

Note Solo puede haber una relación IK entre un hueso y su hijo por lo que sólo uno de los botones *IK* de los hijos de un hueso dado puede activarse a la vez.

Tip Configurando los Ejes Locales

Para lograr los mejores resultados cuando se anima, es necesario asegurarse que los ejes locales de cada hueso sean coherentes a lo largo de todo el armature. Esto es algo que debemos hacer antes de animar nada.

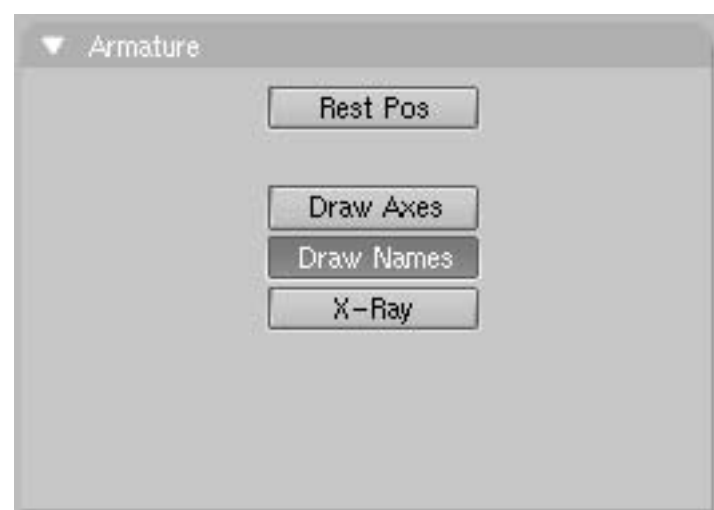
También es necesario que cuando el objeto armature se encuentre en su posición sin transformaciones en Modo Objeto, el frente del armature sea visible en la Vista Frontal, el lado izquierdo sea visible en la Vista Lateral Izquierda y así sucesivamente. Puedes asegurar esto orientando el armature para alinearle correctamente con las vistas y luego presionando **CTRL-A** para aplicar la escala y rotación actual al objeto. Esto también debe hacerse antes de empezar a animar al personaje.

La orientación de los *roll handles* de los huesos es importante para obtener buenos resultados del sistema de animación. Puedes ajustar el ángulo de revolución de la articulación del hueso, seleccionado la articulación y pulsando **NKEY**. El número exacto que debe ser ingresado aquí depende de la orientación del hueso.

El eje Z de cada hueso de orientarse en una dirección coherente en los huesos homólogos. Una buena solución es hacer que los ejes Z apunten hacia arriba (o hacia adelante si el hueso es vertical). Esta operación se facilita mucho si se activa la opción "Draw Axes" en el Panel *Armature* Panel en la Ventana de Botones de Edición.

16.3.3. El Panel Armature

Figure 16-5. Opciones de Visualización para Armatures



Este panel contiene solo unos pocos botones. Cuando se activa el botón *Rest Pos* ([Figure 16-5](#)), el armature se mostrará en su posición de "descanso". Esto es muy útil si se necesita editar la malla asociada con el armature después de haber animado o cambiado la pose del armature. Nota que las Acciones y poses aún están allí pero han sido temporalmente desactivadas mientras el botón esté

activado.

Los botones *Draw Axes* y *Draw Names* permiten visualizar los ejes locales y los nombres de cada hueso en la Vista 3D.

El botón *X-Ray* evita que los huesos del armature sean ocultados por el modelo cuando nos encontramos en Modo Sólido o Sombreado.

[Prev](#)

Herramientas Generales

[Home](#)

[Up](#)

[Next](#)

Skinning

16.4. Skinning

Relativo a Blender v2.31

Una vez que el Armature - el 'esqueleto del personaje' - está listo, es necesario emparentarle la "piel" del personaje. El "Skinning" es una técnica para crear deformaciones de la malla suaves usando un armature. Esencialmente el skinning es la relación entre los vértices de una malla y los huesos de un armature, y cómo afectarán las transformaciones de cada hueso a la posición de los vértices de la malla.

Cuando se emparenta una malla a un armature se presentarán varias opciones:

Parent to Bone

En este caso, aparecerá un menú que te permitirá elegir a qué hueso se debe emparentar el o los objetos hijos. Esto funciona muy bien para robots, que tienen cuerpos compuestos de mallas *separadas* que no se supone que deban doblarse o deformarse cuando se mueven.

Parent to Armature

Eligiendo esta opción se deformarán la o las mallas hijas de acuerdo con sus grupos de vértices. Si las mallas hijas no tienen grupos de vértices definidos, serán sometidas a skinning automático. En efecto, un segundo menú aparecerá con las opciones:

- *Don't create groups* - No hace nada más, se usa skinning automatico;
- *Name Groups* - crea grupos de vértices vacíos cuyos nombres concuerdan con los nombres de los huesos, pero sin asignarles vértices;
- *Create from closest bone* - en el caso que quieras crear y llenar automáticamente los grupos de vértices.

Parent to Armature Object

Eligiendo esta opción hará que los hijos vean al armature como un Empty para todo propósito.

Si estás preparando un animación de personajes, la mayoría de las veces emparentarás tu personaje al armature usando la opción "Armature". Se aconseja enérgicamente usar la opción *Name Groups*. Esto te proveerá de los grupos de vértices, ahorrándote el tedioso trabajo de crearlos y nombrarlos, evitándote errores de tipo.

La opción *Create from closest bone* se encuentra actualmente en desarrollo. Utilizará los "Tipos de huesos" que se pueden definir por medio del menú a la derecha de los botones *IK* ([Figure 16-4](#)) para resultados óptimos.

Actualmente solo las opciones *Skinnable* y *Unskinnable* son funcionales. La primera opción hace que los Grupos de Vértices sean creados (y asignados si así lo has indicado) para el hueso dado, la segunda opción hace que el hueso sea ignorado en el proceso de skinning.

Note El algoritmo de asignación de vértices actual, crea grupos de vértices no óptimos, por ello se recomienda verificarlos uno a uno.

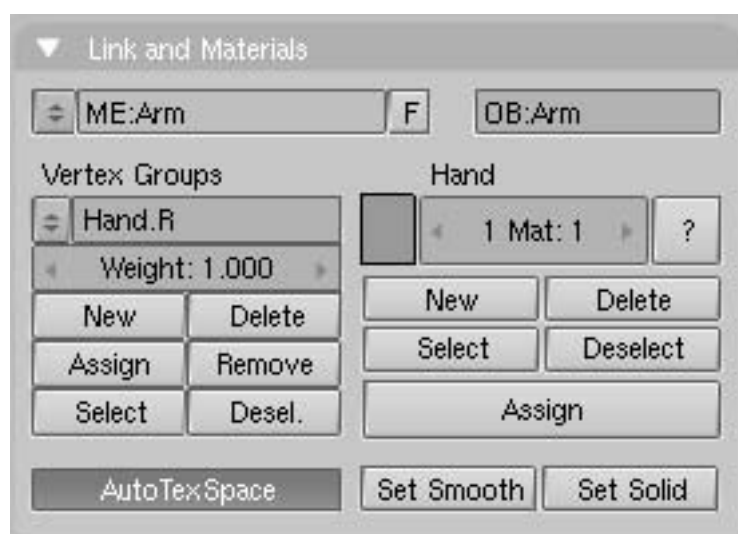
Si una malla no posee ningún grupo de vértices y se convierte en hija de un *armature*, Blender intentará calcular la información de deformación al vuelo. Esto es muy lento y no se recomienda. Es aconsejable crear y usar grupos de vértices en su lugar.

Tip **Weight (Peso) y Dist**

Los valores *Weight* y *Dist* junto a *IK* solo son utilizados por el skinning automático que es una característica discontinuada porque requiere mucho consumo de tiempo de CPU, produce molestas ralentizaciones del sistema y da peores resultados que otros métodos..

16.4.1. Grupos de Vértices

Figure 16-6. Grupos de Vértices



Para definir qué huesos deforman a qué vértices, se necesitan los Grupos de Vértices. Un vértice puede ser miembro de varios grupos, en cuyo caso la deformación será un promedio ponderado de las deformaciones producidas por los huesos a que está asignado. De esta manera es posible crear

articulaciones suaves.

Para agregar un nuevo grupo de vértices a una malla debes estar en Modo Edición. Crea un grupo nuevo de vértices haciendo click en el botón *New* en los botones de edición de la malla en el Panel *Mesh Tools 1* ([Figure 16-6](#)).

Un grupo de vértices puede ser eliminado pulsando sobre el botón *Delete*.

Cambia el grupo activo eligiendo uno de la lista desplegable de grupos.

Los grupos de vértices tienen que tener el *mismo* nombre que los huesos por los que serán manipulados. Tanto la ortografía como las mayúsculas importan aquí. ¡ Por ello es que es tan cómoda la creación de nombres automática !. Renombra un grupo de vértices haciendo **SHIFT-LMB** sobre el botón con el nombre y escribiendo un nuevo nombre. Observa que los grupos de vértices deben ser únicos en una malla dada.

Se pueden asignar los vértices al grupo activo, seleccionándolos y pulsando en el botón *Assign*. Dependiendo del valor indicado en el botón *Weight*, los vértices recibirán más o menos influencia de ese hueso. Este peso, es importante solamente en los vértices que son miembros de más de un grupo. El valor de *Weight* no es un valor absoluto sino uno relativo. Para cada vértice, el sistema calcula la suma de los pesos de todos los huesos que afectan al vértice. Las transformaciones de cada hueso se dividen por esta cantidad o sea que cada vértice siempre recibe una deformación del 100%.

Asignando peso 0 a un vértice lo eliminará del grupo activo en la práctica.

Para eliminar vértices del grupo actual, seleccionalo y pulsa en el botón *Remove*.

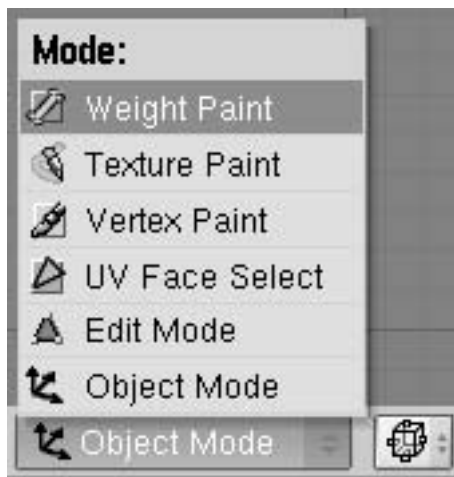
Presionando el botón *Select* añadirá a la selección actual, los vértices del grupo elegido. Pulsando el botón *Deselect* deseleccionará los vértices del grupo actual. Esto es práctico para determinar qué vértices están en un grupo dado.

16.4.2. Pintura de Pesos (Weight Painting)

La pintura de pesos es una técnica alternativa para asignar pesos a los vértices en los grupos de vértices. El usuario puede "pintar" pesos sobre el modelo y ver los resultados en tiempo real. Esto permite lograr articulaciones suaves en forma más fácil.

Para activar el modo de pintura de pesos, selecciona una malla con grupos de vértices y haz click sobre el icono de Weight Paint ([Figure 16-7](#)).

Figure 16-7. El botón Pintura de Pesos (Weight Paint).



La malla activa será dibujada en modo Color-de-Pesos. En este modo, el azul oscuro representa áreas con peso nulo para el grupo de vértices actual, mientras que el rojo representa áreas con peso lleno. Solo puede visualizarse un grupo a la vez. Cambiando el grupo de vértices activo en los botones de edición, cambiarás la representación de la pintura de pesos.

Los pesos se pintan sobre la malla usando una técnica similar a la utilizada en la Pintura de Vértices (vertex-painting), con algunas diferencias. El "color" es el valor de peso especificado en los botones de edición. El desplazador de *opacidad (opacity)* en los botones de vertex paint se usa para modular el peso. Para borrar el peso de los vértices, pon el peso a "0" y comienza a pintar.

Note Es bastante sencillo cambiar el peso, ya que **TAB** te llevará del Modo Pintura de Pesos al Modo Edición y los paneles cambiarán automáticamente según el contexto.

[Prev](#)[El Objeto Armature](#)[Home](#)[Up](#)[Next](#)[Modo Pose](#)

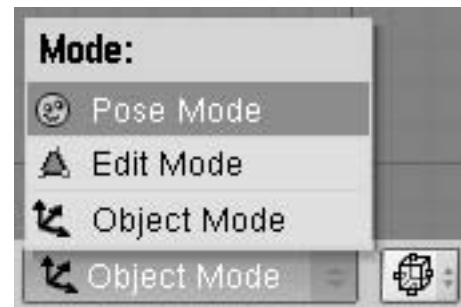
16.5. Modo Pose

Relativo a Blender v2.31

Para manipular los huesos en un armature debes entrar al Modo Pose. En el Modo Pose solo pueden seleccionarse huesos del armature activo. A diferencia del Modo Edición no se pueden ni agregar ni borrar huesos en este modo.

Para ingresar al Modo Pose, selecciona el armature y pulsa **CTRL-TAB**. Alternativamente puedes activar el Modo Pose seleccionando el armature y pulsando en la entrada *Pose Mode* en el menú *Modo* de la cabecera de la Ventana 3D ([Figure 16-8](#)). Puedes salir del Modo Pose con el mismo método o entrando a Modo Edición.

Figure 16-8. La opción de menú Modo Pose.



Dentro del Modo Pose puedes manipular los huesos del armature seleccionandolos con **RMB** y usando las teclas de transformaciones estándar: **RKEY**, **SKEY** y **GKEY**. No puedes "mover" huesos que son hijos IK de otro hueso ya que la cadena IK debe mantenerse continua.

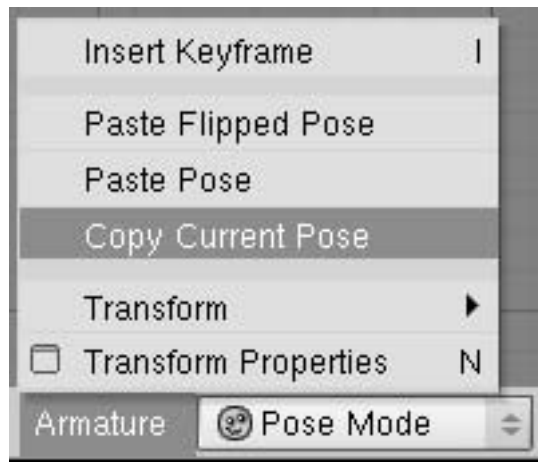
Para insertar Keyframes en los huesos seleccionados pulsa **IKEY**.

Si quieres borrar la pose de uno o más huesos, seleccionalos y pulsa **ALT-R** para borrar las rotaciones, **ALT-S** para borrar el escalado y **ALT-G** para borrar los desplazamientos. Ejecutando estos tres comandos sobre los huesos seleccionados volverán el armature a su posición inicial de reposo.

Frecuentemente resulta conveniente copiar poses de un armature a otro. o de una acción a un punto diferente de la misma acción. Aquí es donde intervienen las herramientas de copiado de poses en el menú *Armature*.

Para obtener mejores resultados asegurate de seleccionar todos los huesos en Modo Edición y pulsar **CTRL-N** para auto-orientar los huesos antes de comenzar a animar.

Figure 16-9. El botón Modo Pose.



Para copiar una pose, selecciona uno o más huesos en Modo Pose, luego selecciona la opción *Armature>>Copy Current Pose* del menú en la cabecera del a Ventana 3D ([Figure 16-9](#)). Las transformaciones de los huesos seleccionados serán almacenadas en un buffer hasta que sea necesario o hasta que se efectúe otra operación de copia.

Para "pegar" una pose, simplemente selecciona la opción *Armature>>Paste Pose* del menú ([Figure 16-9](#)). Si se encuentra activado el autoinsertado de keyframes para *Action*, los KeyFrames serán generados automáticamente.

Para pegar la "pose invertida" (si el personaje estaba inclinado hacia la izquierda en la pose copiada, se pegará inclinado hacia la derecha en la pose invertida), usa la opción *Armature>>Paste Flipped Pose* del menú ([Figure 16-9](#)). Observar que si el armature no fue configurado correctamente, la técnica de pegar la pose invertida no funcionará según lo esperado.

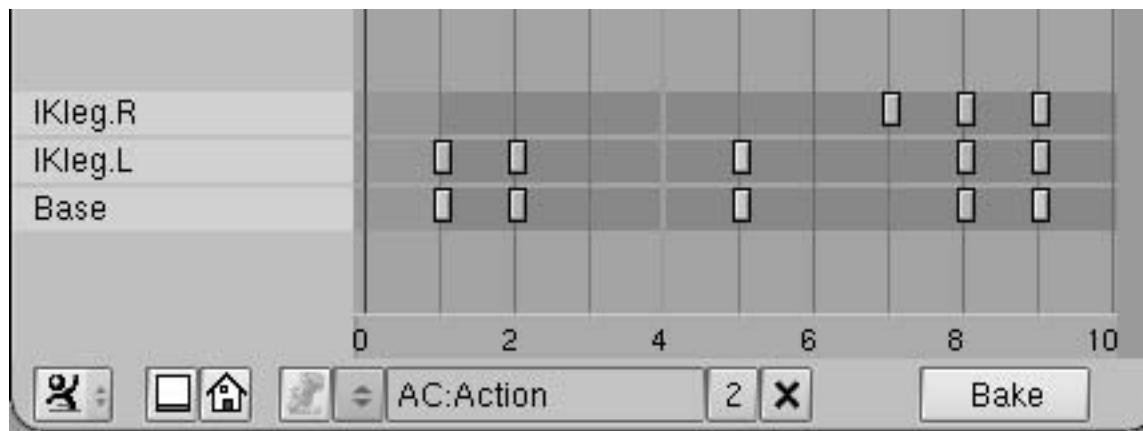
16.6. Ventana de Acciones

Relativo a Blender v2.31

Una Acción está hecha de uno o más canales de Acción. Cada canal corresponde a uno de los huesos del armature, y además, cada canal tiene una curva IPO de Acción asociada al mismo. La Ventana de Acción provee una forma de visualizar y editar juntas todas las curvas IPO asociadas con la Acción.

Tip Puedes activar la ventana de Acción con **SHIFT-F12** ([Figure 16-10](#)).

Figure 16-10. Ventana de Acción



Para cada grupo de keys en una IPO de Acción, en la Ventana de Acción se mostrará un marcador en el frame que corresponda. Esto es parecido al modo "Key" en la Ventana IPO. Para los canales con IPOs de Restricciones, habrá uno o más canales de restricción adicionales debajo de cada canal de Acción. Estos canales se pueden seleccionar independientemente de los canales de sus dueños. ([Figure 16-11](#)).

Figure 16-11. Ventana de Acción con una Restricción



Un bloque de keys de Acción se puede seleccionar tanto con **RMB** sobre él como utilizando la

herramienta de selección por áreas (**BKEY**). Los keys son resaltados en amarillo. Una vez seleccionados, los keys pueden moverse pulsando **GKEY** y moviendo el ratón. Manteniendo presionada **CTRL** el movimiento se hará en intervalos de un cuadro. Haciendo **LMB** se terminará el desplazamiento de keys, mientras que con **ESC** se cancela y se vuelve al estado inicial.

Un bloque de keys de Acción también puede escalar horizontalmente (lo que acelerará o ralentizará la Acción), para ello basta con seleccionar un número de keys y presionar **SKEY**. Moviendo el ratón horizontalmente se escalará el bloque. Con **LMB** se terminará la operación.

Una o más keys de Acción pueden borrarse pulsando **XKEY** cuando el puntero del ratón se halle sobre el área de KeyFrames en la Ventana de Acción.

Un bloque de Acción puede duplicarse y moverse dentro de la misma Acción, seleccionando los keys que se deseen y pulsando **SHIFT-D**. Esto entrarás inmediatamente en modo Mover con lo que se podrán trasladar el nuevo grupo de keys. El movimiento lo finalizamos con **LMB**. Con **ESC** se abortará el desplazamiento pero no eliminará los duplicados generados.

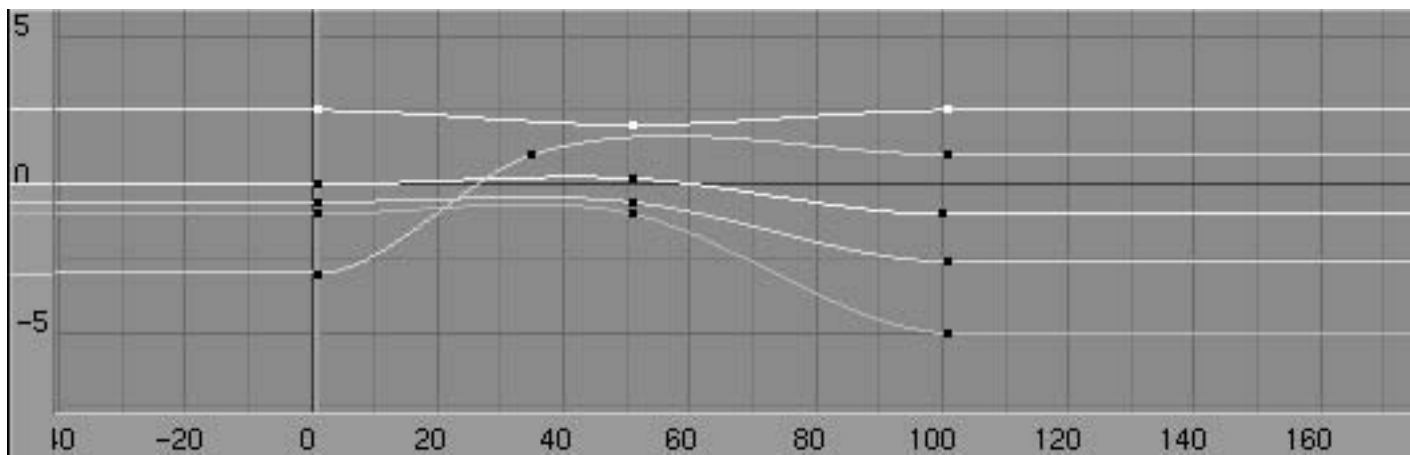
También puedes borrar una o más Acciones o canales de restricción completos (con todas las keys asociadas) seleccionando los canales en la parte más a la izquierda de la Ventana de Acción (los canales seleccionados se resaltarán en azul). Con el puntero del ratón sobre esta zona izquierda de la ventana presiona **XKEY** y confirma el borrado. Observa que no hay deshacer así que realiza esta operación cuidadosamente. Asimismo observa que si borras un canal de Acción que contiene canales de restricción, borrarás esos canales de restricción también.

Tip Horneando Acciones

Si tienes una animación que involucra restricciones y deseas usarla en el motor de juego (que no evalúa restricciones y cuya descripción escapa al alcance de este libro), puedes "hornear" la Acción pulsando el botón **BAKE** en la barra de herramientas de la Ventana de Acción. Esto creará una nueva Acción en la que cada cuadro es un KeyFrame. La Acción puede ser reproducida en el motor de juego y debería ejecutarse perfectamente una vez eliminadas todas las restricciones. Para obtener resultados óptimos, asegurate que todos los blancos de las restricciones estén ubicados dentro de la misma armature.

Puedes ver la IPO de Acción asociada a un hueso en la Ventana IPO en lugar de la Ventana de Acción simplemente de hecho cambiando a una ventana IPO ([Figure 16-12](#)). La IPO de Acción es un tipo de IPO especial que solo es aplicable a huesos. En lugar de ángulos de Euler para codificar la rotación, las IPO de Acción utilizan cuaterniones, los que dan una mejor interpolación entre Poses.

Figure 16-12. IPO de Acción



Los Quaterniones usan vectores de cuatro componentes. En general es difícil y poco intuitivo describir las relaciones entre estos quaterniones y la orientación resultante pero a menudo esto no es necesario. Es mucho mejor generar los Keyframes de quaterniones manipulando directamente los huesos y editando estas curvas solo para ajustar transiciones de ataque y decaimiento.

[Prev](#)

Modo Pose

[Home](#)[Up](#)[Next](#)

Animación No Lineal

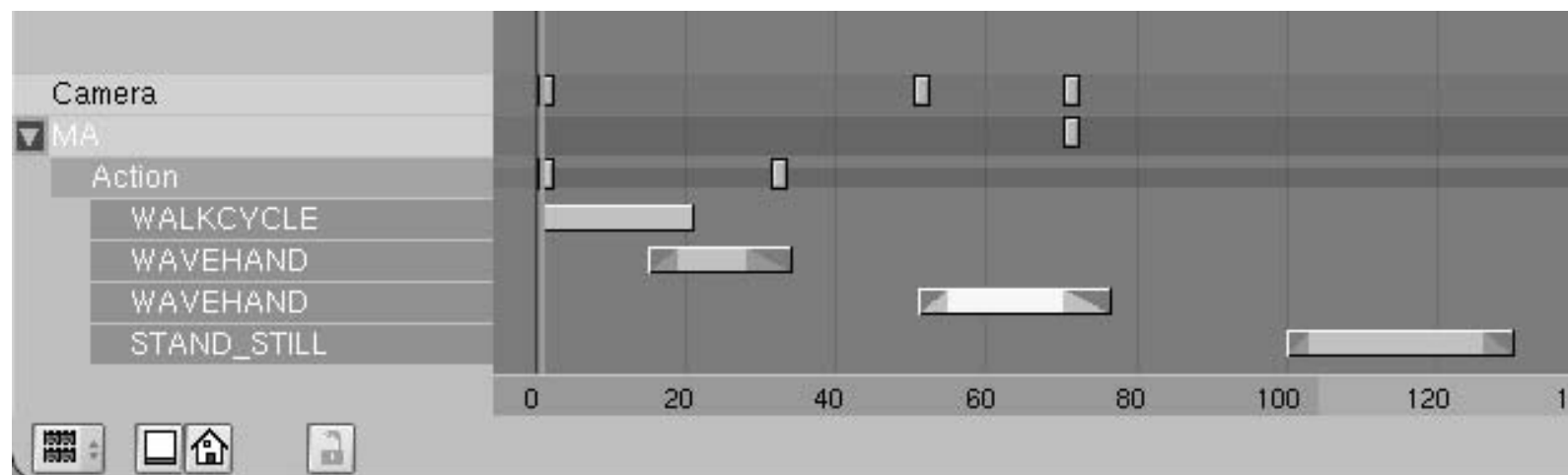
16.7. Animación No Lineal

Relativo a Blender v2.31

La Animación No Lineal (NLA) es una técnica de alguna manera similar a RVK, utilizada para mezclar Acciones simples individuales para generar Acciones complejas y fluidas. La ventana NLA ofrece un panorama general de toda la animación que transcurre en la escena. Desde aquí puedes editar el timing de *todas* las IPOs, como si estuvieran en la Ventana de Acción. La mayor parte de la funcionalidad es idéntica a la de la Ventana de Acción.

La Ventana NLA puede verse con **CTRL-SHIFT-F12** ([Figure 16-13](#)).

Figure 16-13. Ventana NLA



También puedes usar esta ventana para realizar mezclas de Acciones y otras tareas de Animación No Lineal. Puedes agregar y mover Bandas de Acción en una manera semejante a la del Editor de Secuencias y generar transiciones de mezcla en las mismas.

En la ventana NLA, las Acciones se muestran como bandas individuales debajo de la banda del objeto; todos los Keyframes de la Acción (incluyendo canales de restricción) se representan en una línea ([Figure 16-14](#)). Para ver una vista expandida de la Acción, utiliza la Ventana de Acción.

Figure 16-14. Acción Expandida en la Ventana NLA



Los Objetos con canales de restricción mostrarán una o más bandas adicionales debajo de la banda del objeto. La banda de restricción se puede seleccionar independientemente del objeto al que pertenece. ([Figure 16-15](#)).

Figure 16-15. Restricción Expandida en la Ventana NLA



Haciendo click con **RMB** sobre los nombres de los objetos en la Ventana NLA seleccionará los objetos correspondientes en la Ventana 3D. Las bandas seleccionadas se representan en azul y las deseleccionadas en rojo.

Puedes eliminar los canales de restricción de los objetos haciendo click con **RMB** sobre el nombre del canal de restricción y luego presionando **XKEY**.

Note Observa que solo armatures u objetos con IPOs aparecerán en la ventana NLA.

16.7.1. Trabajando con Bandas de Acción

Se pueden agregar Bandas de Acción solamente a objetos Armature. Los objetos no tienen que tener necesariamente una Acción asociada a ellos inicialmente.

Para agregar una Banda de acción a un objeto, mueve el puntero del ratón sobre el nombre del objeto en la Ventana NLA, pulsa **SHIFT-A** y elige la Acción deseada del menú emergente. Observa que solo puedes tener una banda de Acción por línea.

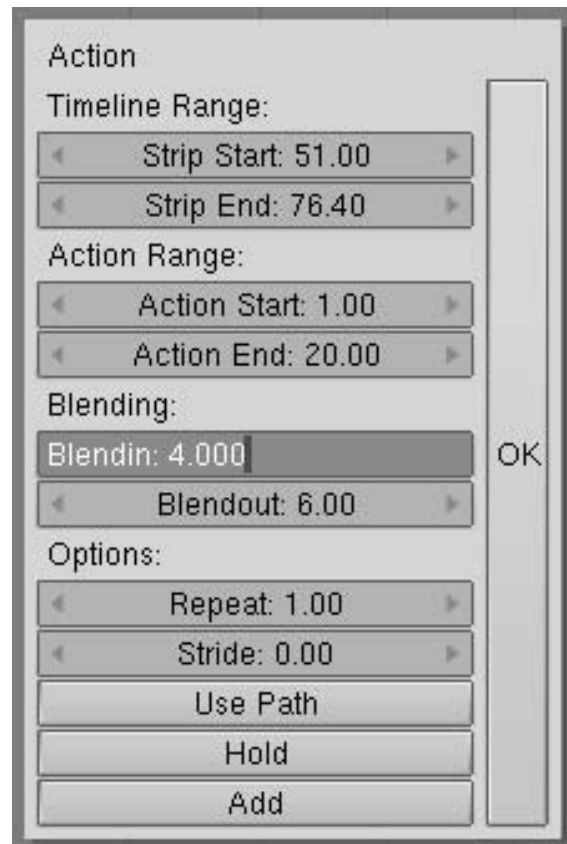
En la Ventana NLA puedes seleccionar, mover y borrar bandas de Acción junto con otros Keyframes.

Las bandas se evalúan de arriba hacia abajo. Los canales especificados en bandas sucesivas reemplazarán a los canales especificados en las bandas precedentes.

Puedes también crear una animación sobre el armature propiamente dicho. Los canales en la Acción local del armature, reemplazarán a los canales de las bandas. Observa que una vez creado un canal en la Ventana de Acción, siempre reemplazará todas las acciones. Si deseas reemplazar solo una parte de la línea de tiempo, puedes convertir la Acción local en una Banda de Acción presionando **CKEY** con el ratón sobre el nombre del armature en la ventana NLA. Esto quita la Acción del armature y la pone al final de la lista de bandas de acción.

Cada banda posee varias opciones a las que puedes acceder seleccionando la banda y pulsando **NKEY** ([Figure 16-16](#)). Las opciones disponibles son las siguientes:

Figure 16-16. Opciones de la Banda de Acción NLA



- *StripStart/StripEnd* - El primero y último cuadros de la banda de Acción en la línea de tiempo.
- *ActionStart/ActionEnd* - El rango de keys de la Acción que se leerán. El final puede ser menor que el comienzo lo que provocará que la Acción sea reproducida en sentido inverso (de atrás hacia adelante).
- *Blendin/Blendout* - La cantidad de cuadros de transición que se generarán entre esta Acción y la anterior en la lista de bandas de Acción.
- *Repeat* - El número de veces que el rango de la Acción debe repetirse. No es compatible con la opción *Use Path*.
- *Stride* - La distancia (en unidades de Blender) que el personaje se desplaza en un único ciclo de Acción (normalmente un ciclo de andar). Este campo solo es necesario si se activa la opción *Use Path*.
- *Use Path* - Si un armature es hijo de un path o curva y tiene un valor de *Stride*, este botón elige qué cuadro de la animación se mostrará de acuerdo a la posición del objeto sobre el path. Muy útil para ciclos de andar.
- *Hold* - Si esta está activada, el último cuadro de la Acción se mostrará eternamente a menos que sea reemplazada por otra Acción. De otra forma el armature volverá a su posición de descanso.
- *Add* - Especifica que las transformaciones de esta banda deberán *sumarse* a los datos de animación existente en lugar de reemplazarlos.

Ventana de Acciones

[Up](#)

Restricciones (Constraints)

16.8. Restricciones (Constraints)

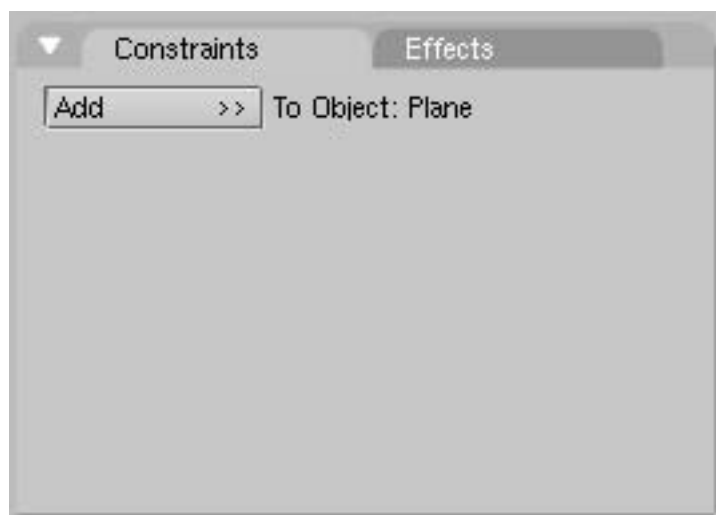
Relativo a Blender v2.31

Las Restricciones o Constraints son filtros que se aplican a las transformaciones de los huesos y objetos. Esta sección es en realidad bastante general y no se aplica exclusivamente a la animación de personajes ya que muchos otros tipos de animación pueden beneficiarse con el uso de restricciones.

Las restricciones en Blender pueden ofrecer una variedad de servicios incluyendo el seguimiento o tracking y soluciones IK.

Para añadir una restricción a un objeto, asegurate de estar en Modo Objeto y en el Contexto Objeto (F7) y con el objeto en cuestión seleccionado. Si estás añadiendo una restricción a un hueso, asegurate de encontrarte en Modo Pose en vez de en Modo Objeto, y con el hueso en particular seleccionado. Los botones de Contexto Objeto mostrarán el Panel *Constraints* ([Figure 16-17](#)). Pulsa sobre el botón *Add*. Aparecerá un menú con todas las posibles restricciones.

Figure 16-17. Panel de Restricciones.



Una vez elegida la restricción deseada, aparecerán sus correspondientes botones. Una restricción se puede borrar haciendo click sobre el icono con la "X" que aparece al lado de la misma. Una restricción se la puede colapsar haciendo click sobre el icono con el triángulo anaranjado. Cuando se encuentra colapsada, se la puede mover arriba o abajo en la lista de restricciones pulsando sobre ella y eligiendo *Moved Up* o *Moved Down* del menú emergente.

En la mayoría de las restricciones es necesario especificar un blanco (target) en el campo correspondiente. En este campo se debe poner el nombre del objeto que se desea como blanco. Si es un hueso, primero debe escribirse el nombre del armature al que pertenece el hueso. Entonces aparecerá otra caja de texto donde podrás ingresar el nombre del hueso en sí.

16.8.1. Tipos de Constraints

Es posible aplicar varios tipos de Restricciones. Todas pueden aplicarse a huesos pero algunas solo se aplican a otros objetos:

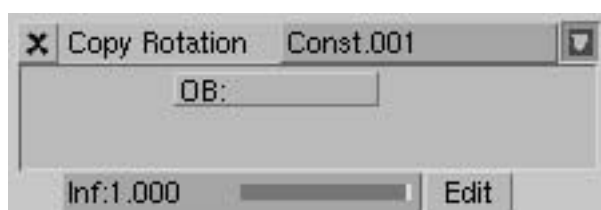
- *Copy Location* - Esta restricción fuerza a un objeto a tener una o más coordenadas (elegidas a través de los tres botones) de su posición iguales a las del objeto blanco ([Figure 16-18](#)).

Figure 16-18. Restricción Copiar Posición (Copy Location).



- *Copy Rotation* - Esta restricción copia la rotación global del blanco y la aplica al poseedor de la restricción ([Figure 16-19](#)).

Figure 16-19. Restricción Copiar Rotación (Copy Rotation).



- *Track To* - Esta restricción hace que el portador de la restricción apunte uno de sus ejes (por omisión el eje Y) hacia el blanco tanto en la dirección positiva como la negativa, dependiendo de la elección en los botones Radio Buttons. La rotación del Objeto será calculada de forma que uno de sus ejes (por omisión el eje Z) apunte hacia arriba, otra vez, esto puede cambiarse por medio de los botones Radio Buttons correspondientes ([Figure 16-20](#)).

Figure 16-20. Restricción Seguimiento (Track To).



- *Locked Track* - Esta restricción provoca que el poseedor de la restricción, apunte uno de sus ejes (por omisión el eje Y) hacia el blanco, tanto en su dirección positiva como negativa, dependiendo de los botones Radio Buttons. La rotación del Objeto se calcula de manera que la dirección de otro de sus ejes (por defecto el eje Z) permanezca *fija*, también puede modificarse vía los botones pertinentes.

En realidad esto significa que el Objeto es girado alrededor de un eje fijo, de modo que el blanco yace sobre un plano definido por el eje trabado y el eje que apunta a él. ([Figure 16-21](#)).

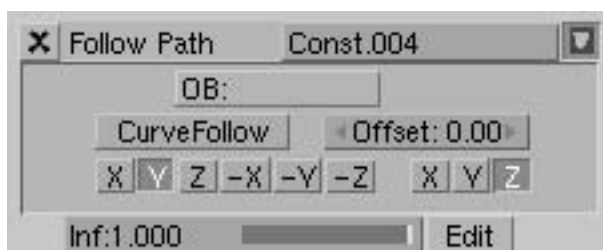
Figure 16-21. Seguimiento Trabado (Lock Track).



- *Follow Path* - Esta restricción necesita que el blanco sea una curva o un camino. Hace que el portador de la restricción siga un camino en el tiempo.

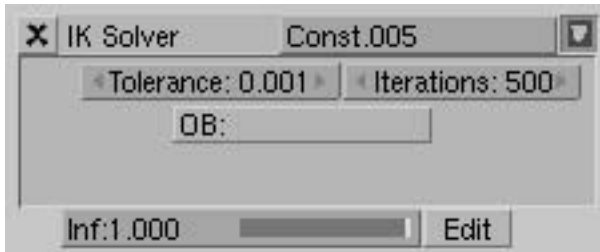
Por omisión, el objeto se desplaza a lo largo de la curva en 100 cuadros. Puedes hacer que la orientación del objeto cambie según la curva con el botón *CurveFollow* y configurando los botones Radio Buttons debajo para definir cual eje debería ser tangente a la curva y cual debería apuntar hacia arriba. Para cambiar el número de cuadros en que se sigue el camino, necesitas editar la IPO de Velocidad de la Curva (Speed IPO). ([Figure 16-22](#)).

Figure 16-22. Seguir Camino (Follow Path).



- **IK Solver** (Solamente para huesos) - Para simplificar la animación de extremidades multi-segmentadas (tales como brazos o piernas) puedes añadir una restricción de solución IK. Las restricciones IK solo se pueden añadir a huesos. Una vez que el blanco ha sido especificado, la solución IK intentará mover la *raíz* del hueso que posee la restricción reorientando a los padres del hueso (pero no moverá a la raíz de la cadena). Si la solución no es posible, la solución IK intentará aproximarse lo más cerca posible. Observa que esta restricción solapará las orientaciones de cualquiera de los padres del hueso IK ([Figure 16-23](#)).

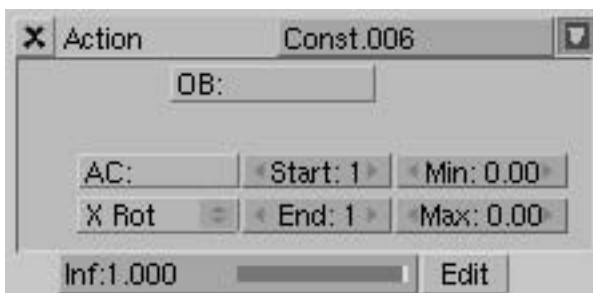
Figure 16-23. IK Solver Constraint.



Note Si el blanco de la restricción IK es otro hueso de la *misma* Armature, se recomienda enérgicamente que te asegures que este hueso, usualmente nombrado IK_Handle, *no* sea hijo de ningún otro hueso de la cadena IK. Caso contrario pueden surgir resultados extraños.

- **Action** (Solamente para huesos) - Una restricción de Acción se puede utilizar para aplicar un canal de acción de una acción diferente a un hueso, basado en la rotación de otro hueso u objeto. La típica forma de utilizar esto es hacer que un músculo se contraiga cuando gira la articulación. La restricción debería aplicarse al hueso que va a hacer la contracción muscular; el blanco debería apuntar a la articulación que está siendo girada ([Figure 16-24](#)).

Figure 16-24. Action Constraint.



El campo AC contiene el nombre de la Acción que contiene la animación de flexión. El único canal que se necesita en la Acción es el que contiene la animación de contracción para el hueso que posee la restricción.

Los campos Start y End especifican el rango del movimiento de la Acción.

Los campos Min y Max especifican el rango de rotación del hueso blanco. La Acción entre los campos Start y End se mapea a esta rotación (de modo que si la rotación del hueso se encuentra en el punto Min, la Pose indicada en Start será la que se aplique al hueso). Observa que el campo Min podría ser mayor que el campo Max.

El menú emergente, indicará qué componente de la rotación habrá de considerarse.

- *Null* - Esta es una restricción que no hace nada; no afecta directamente la transformación de un objeto. El propósito de la restricción nula, es usarla como separador y la razón de esto se aclarará en la sección siguiente ([Figure 16-25](#)).

Figure 16-25. Null Constraint.



16.8.2. Reglas de Evaluación de Restricciones y Precedencia

Las restricciones se pueden aplicar a objetos o huesos. En el caso de las restricciones aplicadas a huesos, cualquier restricción aplicada al *objeto* armature será evaluado antes que sean consideradas las restricciones sobre los huesos.

Cuando se evalúa una restricción específica, todas sus dependencias habrán sido ya evaluadas y estarán en sus orientaciones y posiciones finales. Ejemplos de dependencias son los padres de un objeto, los padres de sus padres (si los hubiere) y las jerarquías de los blancos indicados en la restricción.

Dentro de un objeto dado, las restricciones son ejecutadas de arriba hacia abajo. Las restricciones que se encuentran más abajo en la lista podrían solapar los efectos de restricciones más altas en la lista. Cada restricción recibe como entrada, los resultados de la precedente. La entrada de la primera restricción en la lista es la salida de las IPOs asociadas al objeto.

Si varias restricciones del mismo tipo se especifican en bloques contiguos, la restricción será evaluada *una vez* para el bloque entero utilizando un promedio de todos los blancos. De esta manera puedes restringir un objeto a hacer un seguimiento del punto medio entre dos objetos por ejemplo. Puede utilizar una restricción *Null* para insertar un corte en el bloque de restricciones si prefieres que cada restricción sea evaluada individualmente.

No se permiten restricciones en bucles. Si un bucle se detecta, todas las restricciones involucradas serán deshabilitadas temporariamente (y serán resaltadas en rojo). Una vez resuelto el conflicto, las restricciones se reactivarán automáticamente.

16.8.3. Influencia

El deslizador de Influencia junto a cada restricción se utiliza para determinar cuanto efecto tiene la restricción sobre la transformación del objeto portador.

Si solo hay una única restricción en un bloque (un bloque se define como una serie de restricciones del mismo tipo, una debajo de la otra), un valor de influencia de 0.0 significará que la restricción no tiene ningún efecto sobre el objeto, mientras que si la influencia es de 1.0, la restricción tiene un efecto total.

Si hubiera varias restricciones en un bloque, los valores de influencias se utilizarán como relaciones o tasas. Así en este caso, si hubiese dos restricciones, A y B, cada una con una influencia de 0.1, el blanco resultante se ubicará en el centro de los dos objetos blanco (una relación de 0.1:0.1 o 1:1 o 50% para cada blanco).

La Influencia se puede controlar por medio de una curva IPO. Para añadir una IPO de restricción a una restricción, abre una ventana IPO y cambia su tipo a "constraint" haciendo click sobre el icono de la cadena. Luego pulsa en el botón "Edit IPO" junto a la restricción que quieres modificar. Si no hubiera aún una IPO de restricción asociada con la restricción, se creará una. De otra manera, se mostrará la IPO asignada previamente. Por el momento, solos se pueden crear Keyframes para las IPO de restricción en la Ventana IPO, seleccionando el canal Inf y haciendo **CTRL-LMB** en el espacio IPO.

Cuando se mezclen Acciones con IPOs de restricción, observa que se considerarán solamente las IPOs sobre las IPOs de la Acción local. Las IPOs de restricción sobre las Acciones en bandas de animación serán ignoradas.

Important

En el caso de armatures, las IPOs de restricción son almacenadas en la Acción actual. Esto significa que cambiando la Acción, se cambiará también las IPOs.

[Prev](#)

Animación No Lineal

[Home](#)[Up](#)[Next](#)

Rigging de Mano y Pie

16.9. Rigging de Mano y Pie

por Lyubomir Kovachev

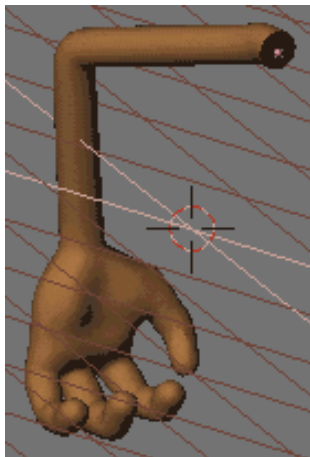
traducción por Claudio "malefico" Andaur

Relativo a Blender v2.31

16.9.1. La Mano

Configurar una mano para animación requiere algo de práctica. Los gestos, los movimientos de la muñeca y los dedos son muy importantes, expresan estados emocionales del personaje e interactúan con otros personajes y objetos. Por ello es que es muy importante tener una configuración eficiente, capaz de realizar todos los movimientos de la muñeca y dedos en forma sencilla. Aquí veremos cómo hacerlo:

Figure 16-26. El modelo del Brazo

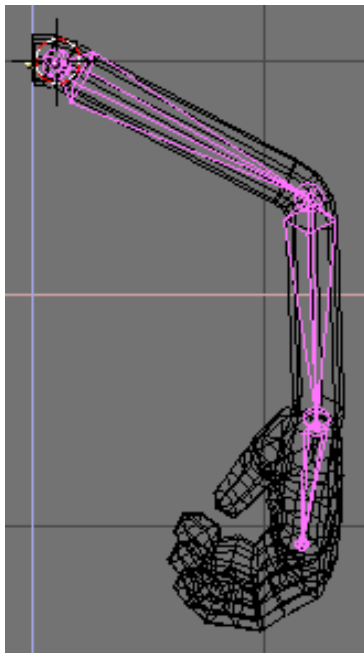


En este tutorial usaremos una malla de brazo simple y caricaturesca ([Figure 16-26](#)).

La siguiente configuración utiliza una solución IK para el movimiento del brazo completo y otras cuatro soluciones IK para cada uno de los dedos. La rotación de la muñeca se logra con un único hueso FK.

Muy bien. Echale una mirada a la malla del brazo y empecemos a construir el armature.

Figure 16-27. Diseñando el armature



Ubica el cursor 3D en el hombro, ve a la Vista Frontal y agrega un armature. Haz una cadena de tres huesos - uno en la parte superior del brazo, uno para la parte inferior, y un tercero que se ubicaría en la palma terminando donde empieza el dedo medio. Esto se llama una cadena de huesos. ([Figure 16-27](#)).

Figure 16-28. El armature en Vista Lateral.

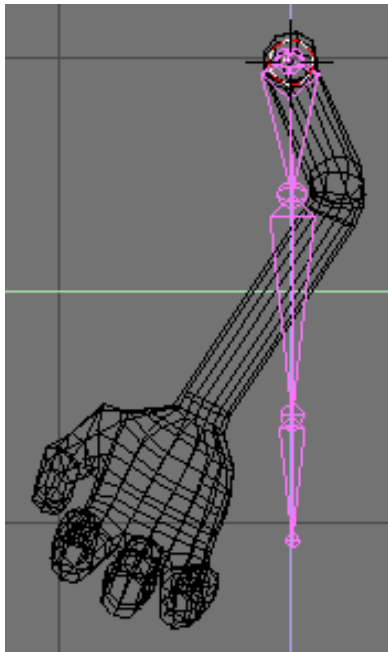
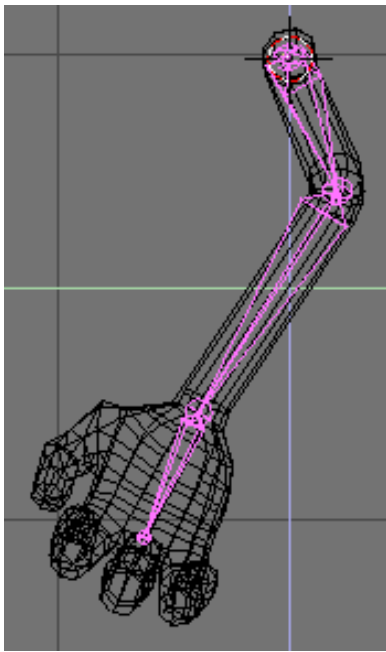
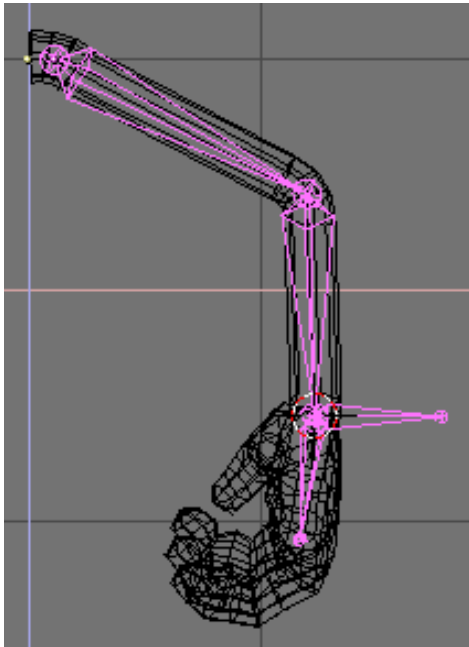


Figure 16-29. Ubicando el armature en la Vista Lateral.



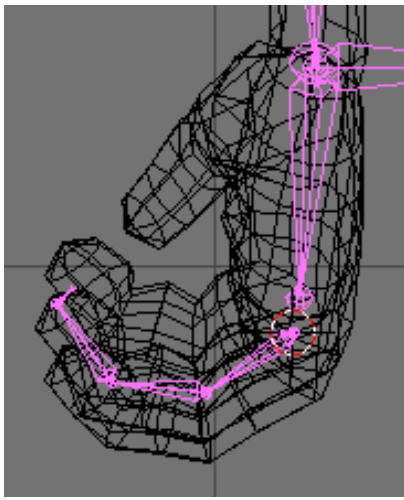
Ahora cambia a la Vista Lateral y desplaza los huesos de modo que se ajuste adecuadamente al brazo y a la palma ([Figure 16-28](#) y [Figure 16-29](#)).

Figure 16-30. Solución IK de la muñeca.



Amplía la vista en la mano y ubica el cursor en la raíz del hueso posicionado en la palma. Agrega un nuevo hueso apuntando a la derecha, con la misma longitud que el hueso de la palma. Este será la solución IK del brazo. ([Figure 16-30](#)).

Figure 16-31. Configurando los dedos.



Pon el cursor 3D al comienzo del dedo medio y en la Vista Frontal comienza a construir una nueva cadena de cuatro huesos ([Figure 16-31](#)). Tres de ellos corresponderán a los huesos propiamente dichos del dedo, y el cuarto será un hueso nulo - es decir, un hueso pequeño que apunte a la palma y que luego ayudará a convertir toda la cadena en una cadena IK.

Nuevamente cambia a la Vista Lateral y acomoda los huesos para que se ajusten bien al dedo. Esto puede ser algo complicado y quizás requiera que veas la escena usando el trackball mientras ajustas los huesos ([Figure 16-32](#)).

Figure 16-32. Configurando el dedo.

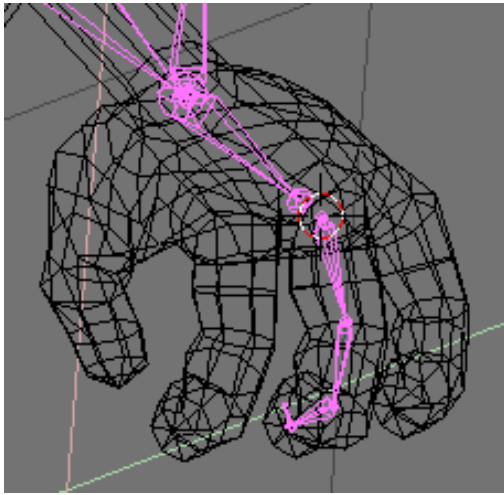
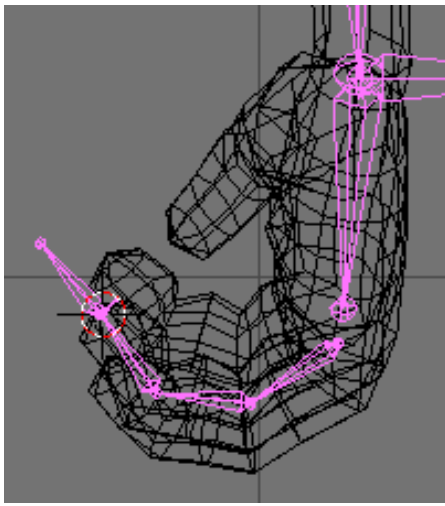


Figure 16-33. Añadiendo la solución IK del dedo.



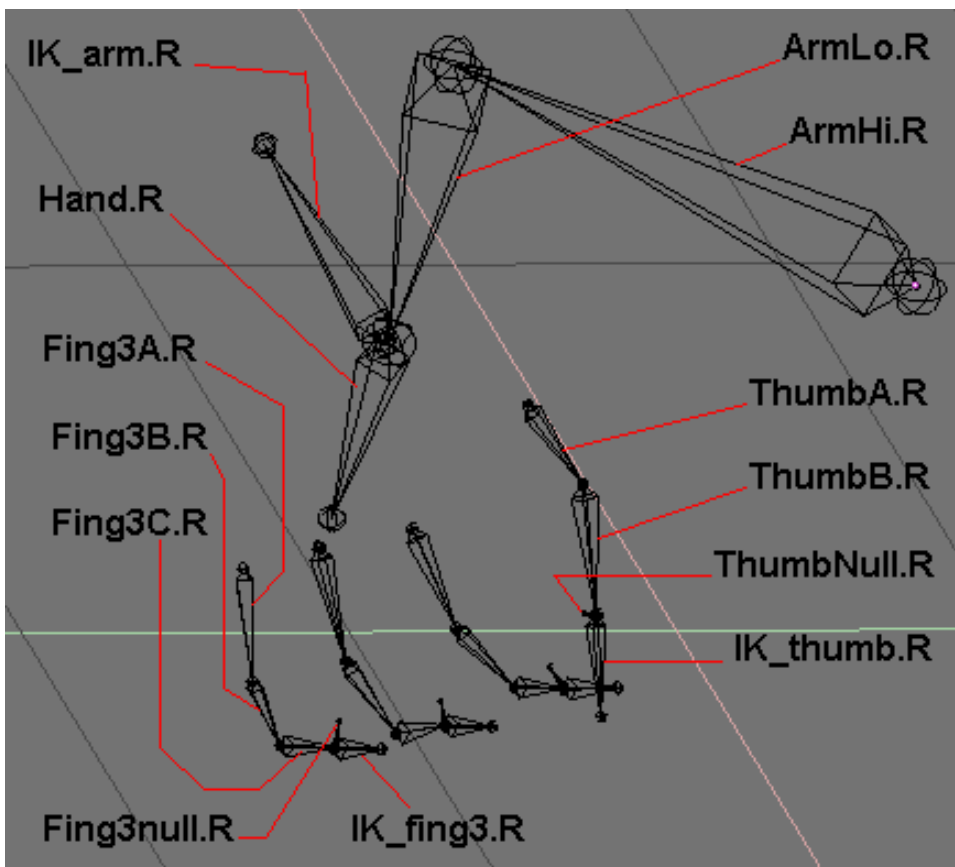
Ahora añade la solución IK para esta cadena de dedos. Posiciona el cursor 3D en la raíz del hueso nulo y agrega un hueso del mismo largo que los otros tres huesos del dedo ([Figure 16-33](#)).

Figure 16-34. Configurando los otros dedos.



Repite lo mismo para crear las cadenas IK de los otros tres dedos. La única diferencia con el pulgar es que este tiene de hecho dos huesos en lugar de tres. Puedes simplemente copiar y pegar la cadena y solo reajustar la forma lo que sea necesario... ([Figure 16-34](#)).

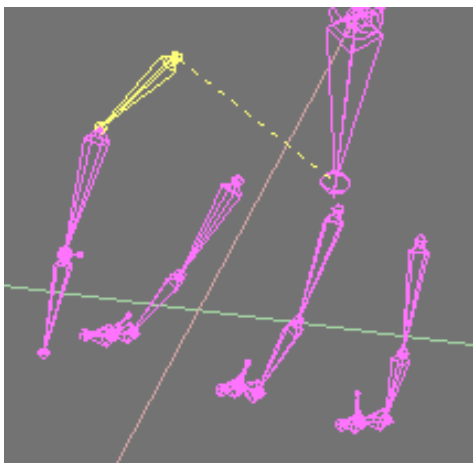
Figure 16-35. Vista general de los nombres.



Ahora ha llegado la hora de la parte aburrida - nombrar los huesos. No puedes saltarte este paso porque necesitarás los nombres de los huesos luego para la parte de Skinning. Los huesos se nombran como en [Figure 16-35](#).

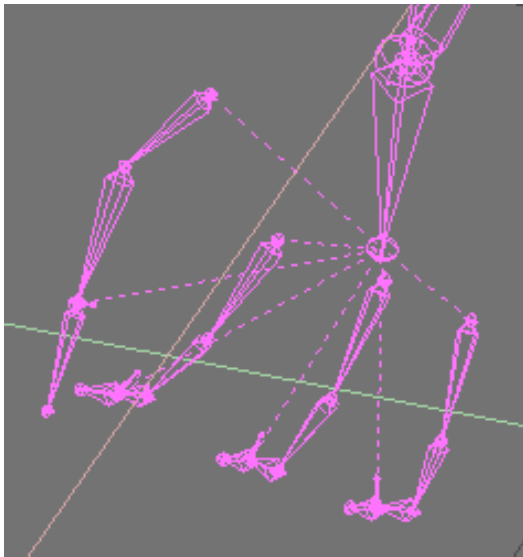
Note Los nombres de los huesos de los dedos 1 y 2 no se muestran aquí. Son idénticos a los nombres del hueso 3, solo cambia el número.

Figure 16-36. Emparentando el pulgar.



Ahora hagamos algunos emparentamientos.

Selecciona la raíz del hueso pulgar "ThumbA.R" ([Figure 16-36](#)) y en el menú de edición, pulsa sobre el campo "child of" y elige "Hand.R". Haz emparentado la cadena de huesos del pulgar al hueso de la mano.

Figure 16-37. Emparentando los otros dedos.

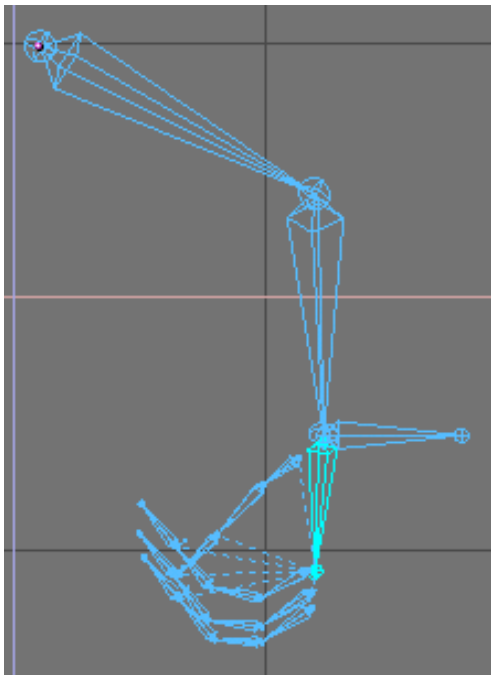
Repitiendo el mismo proceso emparenta los siguientes huesos ([Figure 16-37](#)):

- "Fing1A.R" a "Hand.R"
- "Fing2A.R" a "Hand.R"
- "Fing3A.R" a "Hand.R"
- "IK_thumb.R" a "Hand.R"
- "IK_fing1.R" a "Hand.R"
- "IK_fing2.R" a "Hand.R"
- "IK_fing3.R" a "Hand.R"

¿ Por qué hicimos todo esto ? ¿Por qué emparentamos tantos dedos a "Hand.R"? Porque cuando rotes la mano (por ej: "Hand.R") todos los dedos seguirán a la mano. De otra manera, los dedos permanecerán en su sitio y sólo se moverá la palma obteniéndose resultados extrañísimos.

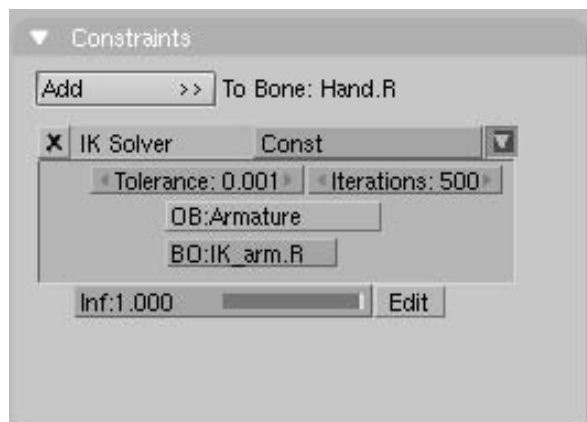
Note Ningún hueso IK es hijo de ninguno de los huesos que controla. Todos son hijos de "Hand.R".

Figure 16-38. Configurando el IK para la muñeca. Seleccionando el hueso.



Hora de agregar restricciones. Entra a Modo Pose ([Figure 16-38](#)) y ve al contexto Objeto (**F7**). Elige "Hand.R" y agrega una restricción de Solución IK al mismo en el Panel *Constraints*. En el campo *OB* escribe el nombre del objeto: "Armature". El hueso se fue al centro del armature pero lo arreglaremos en un instante. En el campo *BO* que aparece ahora, escribe el nombre del hueso "IK_arm.R". Este será el hueso IK que controlará el movimiento del brazo ([Figure 16-39](#)).

Figure 16-39. Configurando el IK para la muñeca. Completando la restricción.



Ahora repitiendo el mismo procedimiento:

- selecciona "ThumbNull.R" y agrega como solución IK a "IK_thumb.R",
- selecciona "Fing1null.R" y agrega como solución IK a "IK_fing1.R",
- selecciona "Fing2null.R" y agrega como solución IK a "IK_fing2.R",
- selecciona "Fing3null.R" y agrega como solución IK a "IK_fing3.R".

Haz terminado con la parte de los huesos. En Modo Pose selecciona diferentes soluciones IK y muevelas para probar las cadenas IK. Ahora puedes mover los dedos, el pulgar, y el brazo entero, y girando el hueso "Hand.R" puedes girar la mano entera.

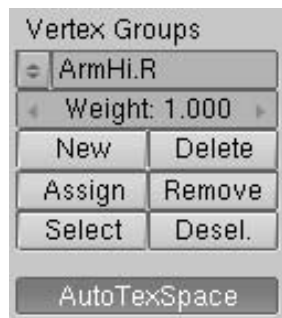
Hagamos ahora el skinning. Esta es la parte donde le decimos a la malla como deformarse. Agregarás grupos de vértices a la malla. Cada grupo de vértices debe ser nombrado con el hueso que lo deforma. Si no asignas grupos de vértices el proceso d

deformación requerirá mucho mayor uso de CPU, y el proceso de animación no solo se ralentizará dramáticamente sino que obtendrás resultados indeseados. Es altamente recomendable (casi obligatorio) que utilices mallas para tus personajes con subdivisión de superficies y de baja cantidad de vértices. Caso contrario, si utilizas mallas con muchos vértices, el skinning será mucho más difícil. No sacrifiques detalle, pero modela económicamente, usa la menor cantidad de vértices posible y siempre utiliza SubSurf.

Emparenta la malla al Armature, en el menú emergente selecciona *Armature* y en el siguiente menú, selecciona *Name Groups*. Tu malla será equipada con Grupos de Vértices vacíos.

Selecciona la malla del brazo, ingresa a Modo Edición y cambia al Contexto Edición (**F9**). En el panel *Mesh Tools 1* de la Ventana de Botones de Edición notarás un pequeño grupo de botones con la palabra *Group* encima. Gracias a la característica de nombrado automático, haz creado ya todos los grupos que necesitas. ([Figure 16-40](#)).

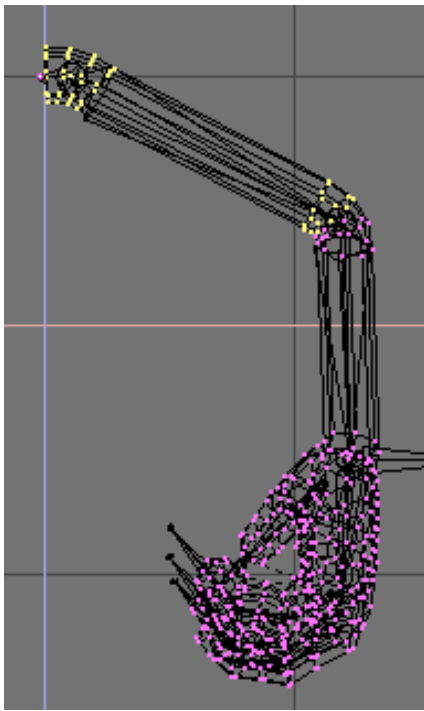
Figure 16-40. Nombres de grupos de vértices.



De hecho, el esquema de Agrupado Automático ha creado grupos de vértices también para los huesos "IK" y los huesos "nulos" *a menos que* los hayas configurado antes como *Unskinnable*. Estos grupos son innecesarios y puedes borrarlos sin preocupación.

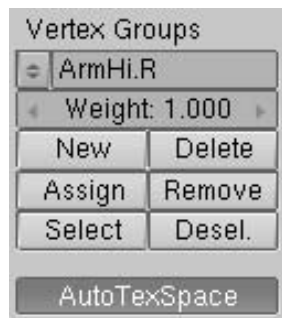
Vayamos ahora a la parte complicada: Selecciona el grupo de vértices "ArmHi.R" desde los botones de edición haciendo click sobre el botoncito con el signo negativo color blanco. Fíjate en la Ventana 3D. Selecciona todos los vértices que quieras que sean deformados por el hueso "ArmHi.R". ([Figure 16-41](#)).

Figure 16-41. Grupo de vértices ArmHi.R.



Ahora pulsa el botón *Assign* en la ventana de botones de edición ([Figure 16-42](#)). Acabas de agregar los vértices seleccionados al grupo de vértices "ArmHi.R". Estos vértices serán deformados por el hueso "ArmHi.R".

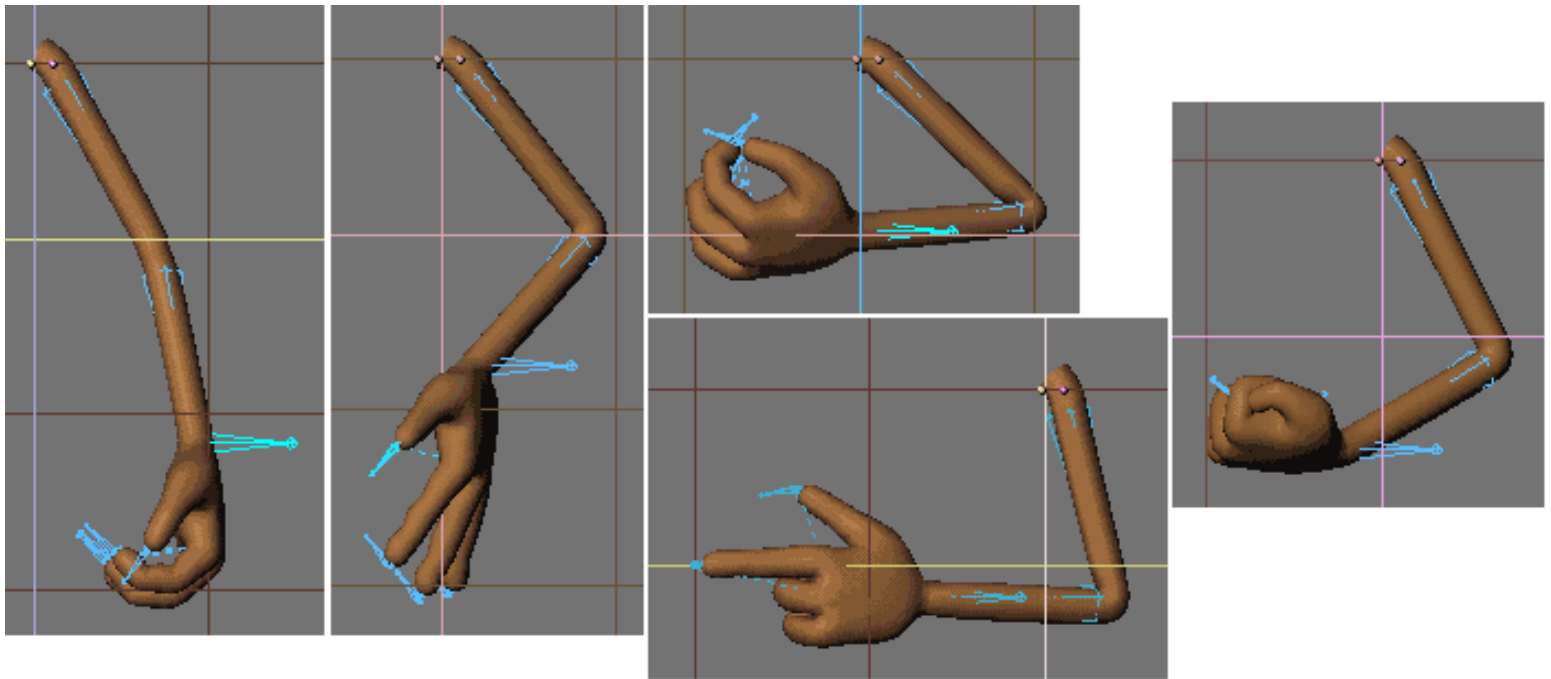
Figure 16-42. Asignando vértices a un grupo.



Repite los mismos pasos para los otros grupos de vértices: selecciona vértices y asignalos al grupo correspondiente. Es un proceso algo engañoso, hazlo con cuidado. Si por error hubieras asignado vértices a un grupo, no te preocupes. Solo selecciona los vértices indeseados y pulsa el botón *Remove*. Puedes agregar un vértice a más de un grupo de vértices. Por ejemplo los vértices que forman las articulaciones (de los dedos, la muñeca, el codo, etc) pueden asignarse a los dos grupos de vértices contiguos. También puedes asignar vértices que deformen con distinta intensidad. El peso por omisión es 1.000, pero puedes agregar vértices con peso 0.5 o menor. Mientras menor sea el peso, menor será la deformación obtenida para ese vértice. Puedes hacer que un vértice sea deformado un 75% por un hueso y un 25% por otro, o que se deforme 50% para cada uno. Todo es cuestión de probar la deformación hasta obtener el resultado deseado. En general si tu modelo de brazo tiene articulaciones semi-flexionadas (como el modelo en este tutorial) obtendrás buenos resultados sin utilizar valores distintos a 1.00. Mi propia regla práctica para modelar personajes es: Modela siempre los brazos, dedos y piernas semi-flexionados, nunca derechos. Esto es una garantía de buena deformación.

Cuando hayas terminado de agregar vértices a los grupos de vértices, si no has cometido ningún error, tendrás un brazo y mano bien configurados. Selecciona el armature, entra en Modo Pose, selecciona los distintos huesos IK y prueba el brazo y los dedos ([Figure 16-43](#)).

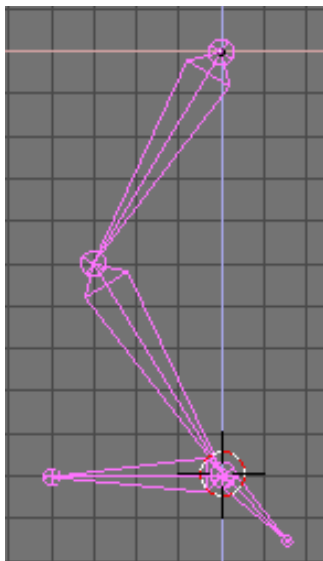
Figure 16-43. Diferentes Poses.



16.9.2. El Pie

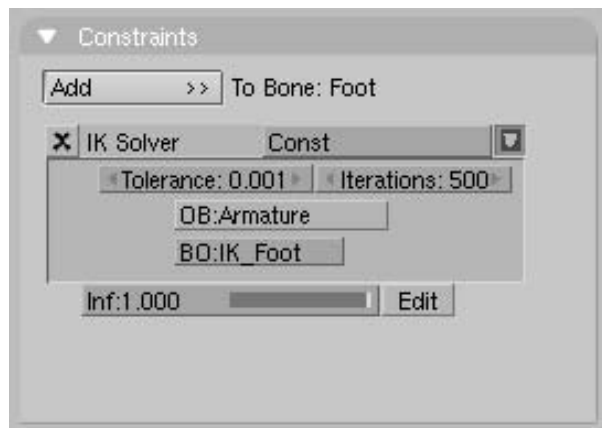
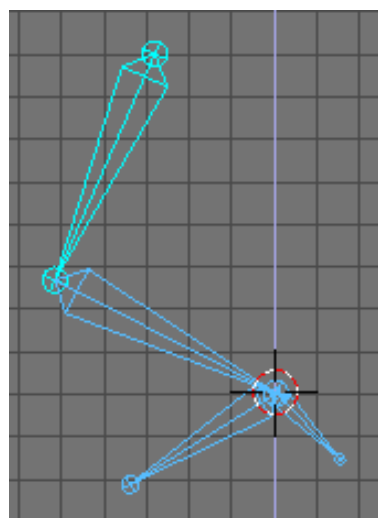
La configuración de piernas y pies es quizás lo más importante en todo el proceso de rigging. Una configuración errónea puede llevar al bien conocido efecto de "pies deslizantes", el cual es muy molesto y normalmente arruina por completo la animación. Una configuración compleja y bien hecha de pies debe ser capaz de mantener al personaje parado en el suelo mientras se mueve el cuerpo y permitir hacer otras cosas más complicadas como pararse en puntas de pies, mover los dedos, etc. Ahora veremos distintas configuraciones de pies que podrán utilizarse para las distintas situaciones.

Figure 16-44. Una configuración de pierna (equivocada).



Primero, analicemos cómo se ve una configuración errónea de pies, como ([Figure 16-44](#)).

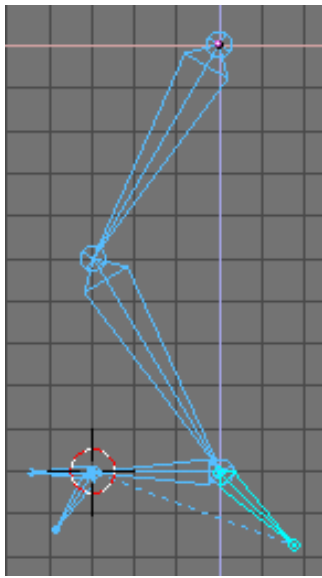
Empieza construyendo una cadena de tres huesos, uno para el muslo, otro para la tibia y otro para el pie. Ahora coloca el cursor 3D en la articulación del talón y agrega otro hueso - que será la solución IK. Ahora configura este hueso como una restricción IK para el hueso del pie ([Figure 16-45](#)).

Figure 16-45. Asignando la restricción IK.**Figure 16-46. El esqueleto en modo pose.**

Prueba el armature: En modo Pose, mueve el hueso IK - se mueve bien. Ahora mueve el primer hueso de la cadena (el hueso del muslo) y muevelo. ¡ El pie se mueve también y no queremos que esto suceda ! ([Figure 16-46](#)).

Normalmente durante la animación moverás mucho el cuerpo. El muslo estará emparentado al cuerpo y será afectado por él. Por lo que cada vez que hagas mover o girar el cuerpo de tu personaje, los pies se deslizarán o patinarán sobre el suelo y a veces se enterrarán bajo el mismo. Esto nos llevará a un resultado espantoso en la animación, especialmente en un ciclo de andar.

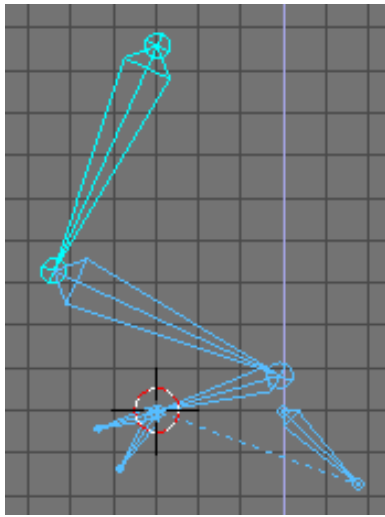
Figure 16-47. Agregando los dedos y un poco mas de animación IK.



Quizás creas que esto podría evitarse añadiendo una segunda solución IK en los dedos de los pies ([Figure 16-47](#)). Hagámoslo. Crea una nueva armature. Agrega una cadena de cuatro huesos: muslo, tibia, pie y dedos. Agrega dos soluciones IK - una para el pie y otra para los dedos. Emparenta el hueso IK de los dedos al hueso IK del pie.

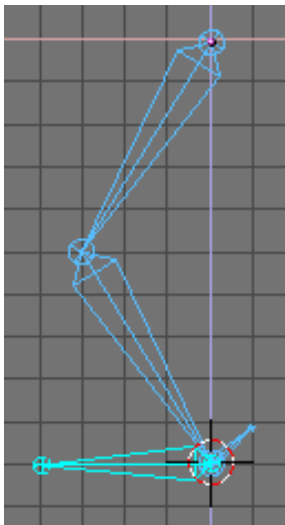
Note El IK de los dedos está emparentado al hueso IK del pie. Este último *no debe* ser hijo de ningún otro hueso del armature. Asegurate de esto y borra cualquier relación de parentesco, recuerda que puedes hacerlo seleccionado la opción vacía en el menú *Child of*. No olvides verificar esto en los ejemplos siguientes.

Figure 16-48. Moviendo la pierna.



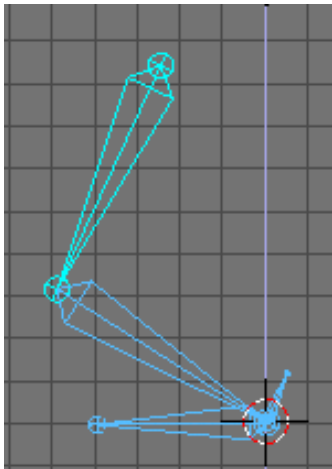
Prueba esta configuración - toma el hueso del muslo y muevelo ([Figure 16-48](#)). Bueno, ahora el deslizamiento no es tan notable como en la configuración anterior, pero es suficiente para arruinar la animación.

Figure 16-49. Rigging con un hueso nulo.



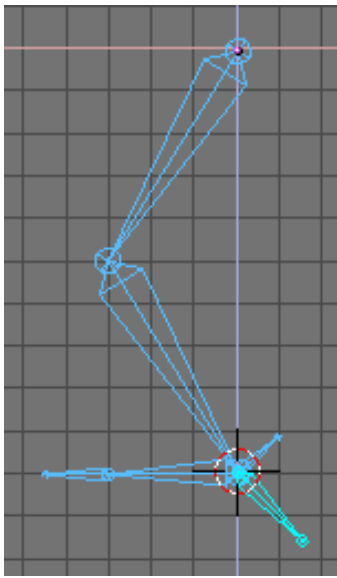
Crea un nuevo armature. Haz una cadena de tres huesos - muslo, tibia y un hueso nulo. El hueso nulo es un hueso pequeño al que le añadiremos una restricción IK. Ahora ubica el cursor 3D en el talón y añade un hueso para el pie. Ahora configura el hueso del pie como una restricción de solución IK al hueso nulo ([Figure 16-49](#)). (También puedes agregar otro hueso como solución IK y agregarle una restricción de "copy location" al hueso del pie, que apunte al hueso IK.)

Figure 16-50. Rigging con un hueso nulo.



Prueba esto - ahora funciona. Cuando mueves el muslo, el pie permanece quieto ([Figure 16-50](#)). Eso está bien . Pero no es suficiente. Mueve el muslo un poco más. La cadena de la pierna sube pero el pie permanece en el suelo. Bueno, esa es una desventaja de esta configuración, pero se supone que no subirás tanto el cuerpo sin levantar los pies durante una animación...

Figure 16-51. Agregando los dedos.



Crea nuevamente una cadena de tres huesos - muslo, tibia y un hueso nulo. Coloca el cursor 3D en el talón y añade una cadena de dos huesos - un hueso para el pie y otro para los dedos. Ahora agrega una solución IK al hueso del pie ([Figure 16-51](#)).

Pruebalo. Esta es una buena configuración con pies estables y aislados y movimiento de los dedos. Pero aún no puedes pararte en puntitas de pie con ella.

Figure 16-52. Rig de la pierna completa.

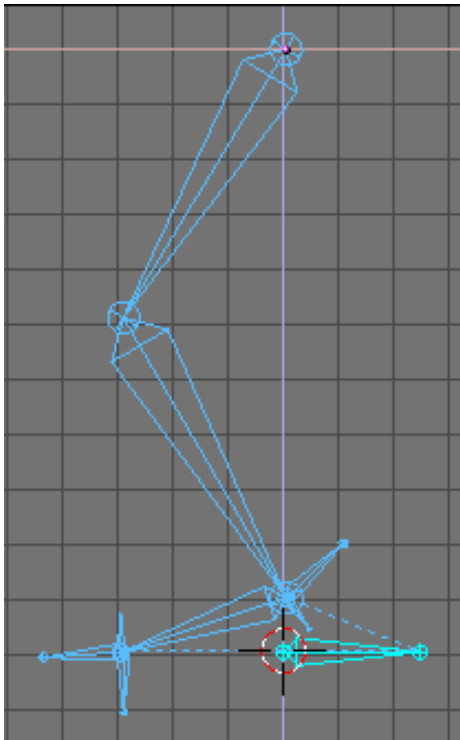
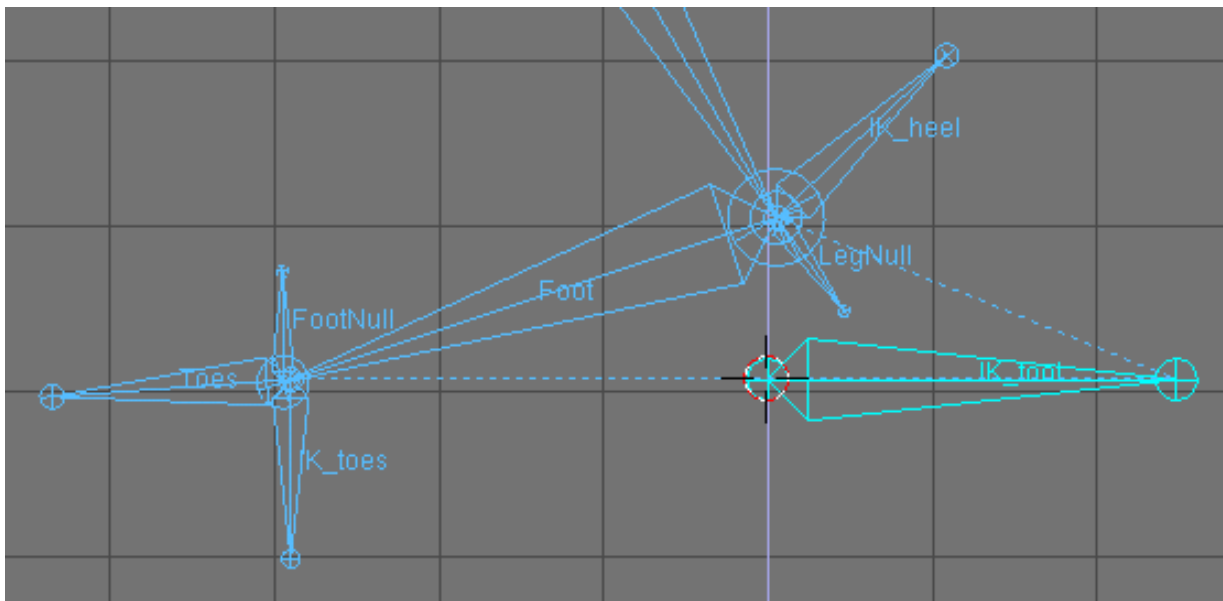


Figure 16-53. Ampliación del pie.



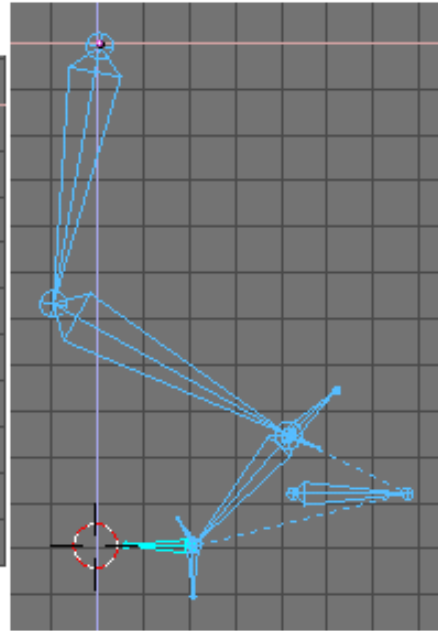
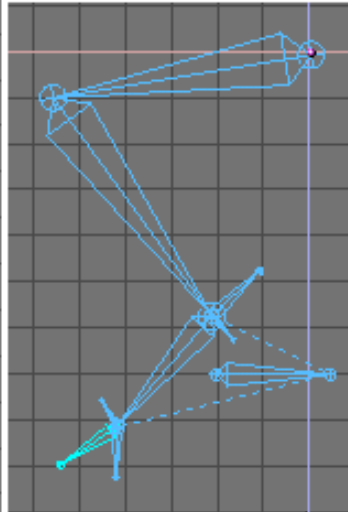
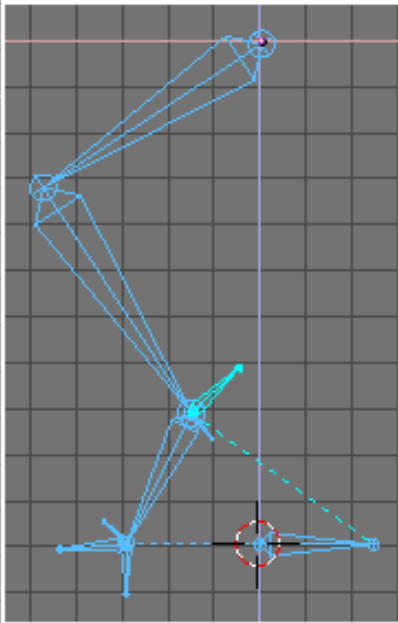
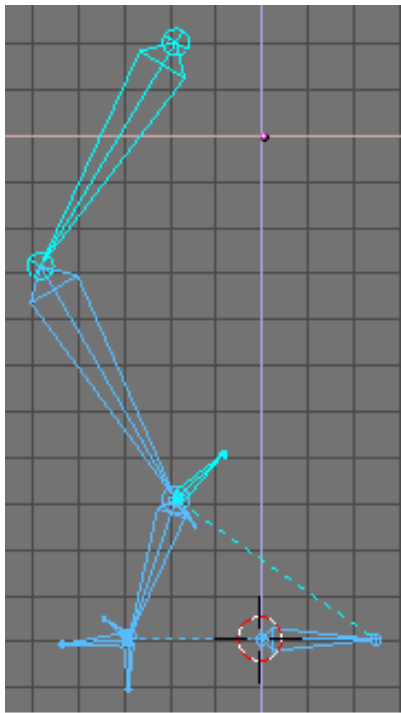
Construye una cadena de tres huesos - muslo, tibia y un hueso nulo (llamalo LegNull) ([Figure 16-52](#)). Comenzando en el talón, crea una segunda cadena de solo dos huesos - pie (Foot) y un pequeño hueso nulo (FootNull). Ponel cursor 3D al final del hueso del pie y crea una hueso para los dedos (Toes). Desde el mismo lugar crea un hueso solución IK (IK_toes). Ahora pon el cursor 3D en el talón y añade otra solución IK allí (IK_heel). Por último, comenzando en algún lugar cerca del talón, añade un hueso solución IK más grande (IK_foot) ([Figure 16-53](#)).

Ahora, agreguemos las restricciones. Haz lo siguiente:

- Al hueso "Toes" ponle una restricción "copy location" con "IK_toes" como blanco.
- A "FootNull" - una restricción IK (blanco - "IK_toes").
- A "Foot" - copy location (blanco - "LegNull").
- A "LegNull" - una restricción IK (blanco - "IK_heel").

Muybien, eso es todo. Prueba el armature. Toma "IK_foot" y subelo. Ahora toma "IK_toes" y bájalo. El pie cambia su rotación pero pareciera que los dedos están desconectados del mismo. Pero si animas cuidadosamente, siempre podrás evitar que los dedos se separen del pie. Ahora vuelve el armature a su pose inicial. Toma "IK_heel" y "LegHi" y levántalos. Ahora el personaje se para de puntas de pie. El pie puede aparecer desconectado de los dedos otra vez, pero puedes remediar la pose seleccionando "IK_heel" solamente y moviéndolo un poquito hacia adelante o hacia atrás. Esta configuración puede no ser la más sencilla de animar, pero da mayores posibilidades que las anteriores. Normalmente cuando no necesitas que tu personaje se pare de puntas de pie, será mejor utilizar alguna de las configuraciones más sencillas. Nunca lograrás una configuración perfecta para todas las situaciones. Puedes mejorarlas pero siempre tendrán alguna desventaja.

Figure 16-54. Probando la configuración.



[Prev](#)

Restricciones (Constraints)

[Home](#)

[Up](#)

[Next](#)

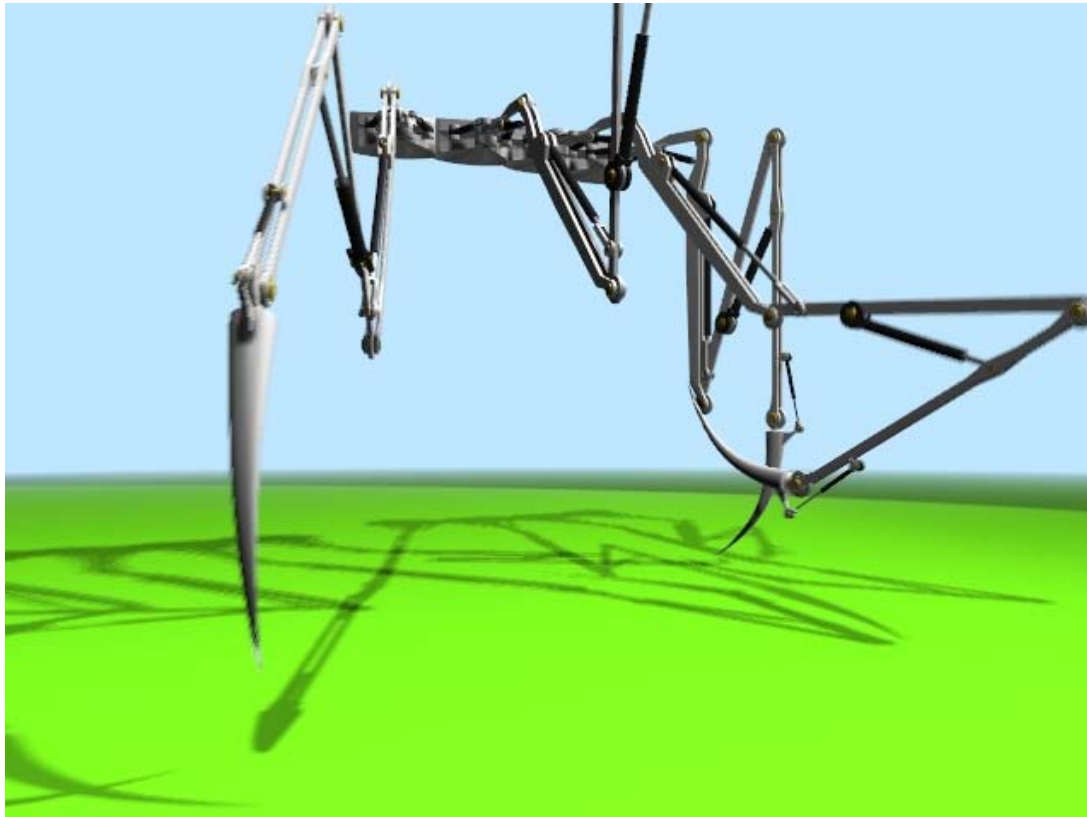
Rigging de elementos mecánicos

16.10. Rigging de elementos mecánicos

Relevante a Blender v2.31

Las armaduras también son geniales para hacer el rigging de elementos mecánicos, tales como robots, guerreros mecánicos, etc. ([Figure 16-55](#)).

Figure 16-55. Araña-robótica de cuatro patas.

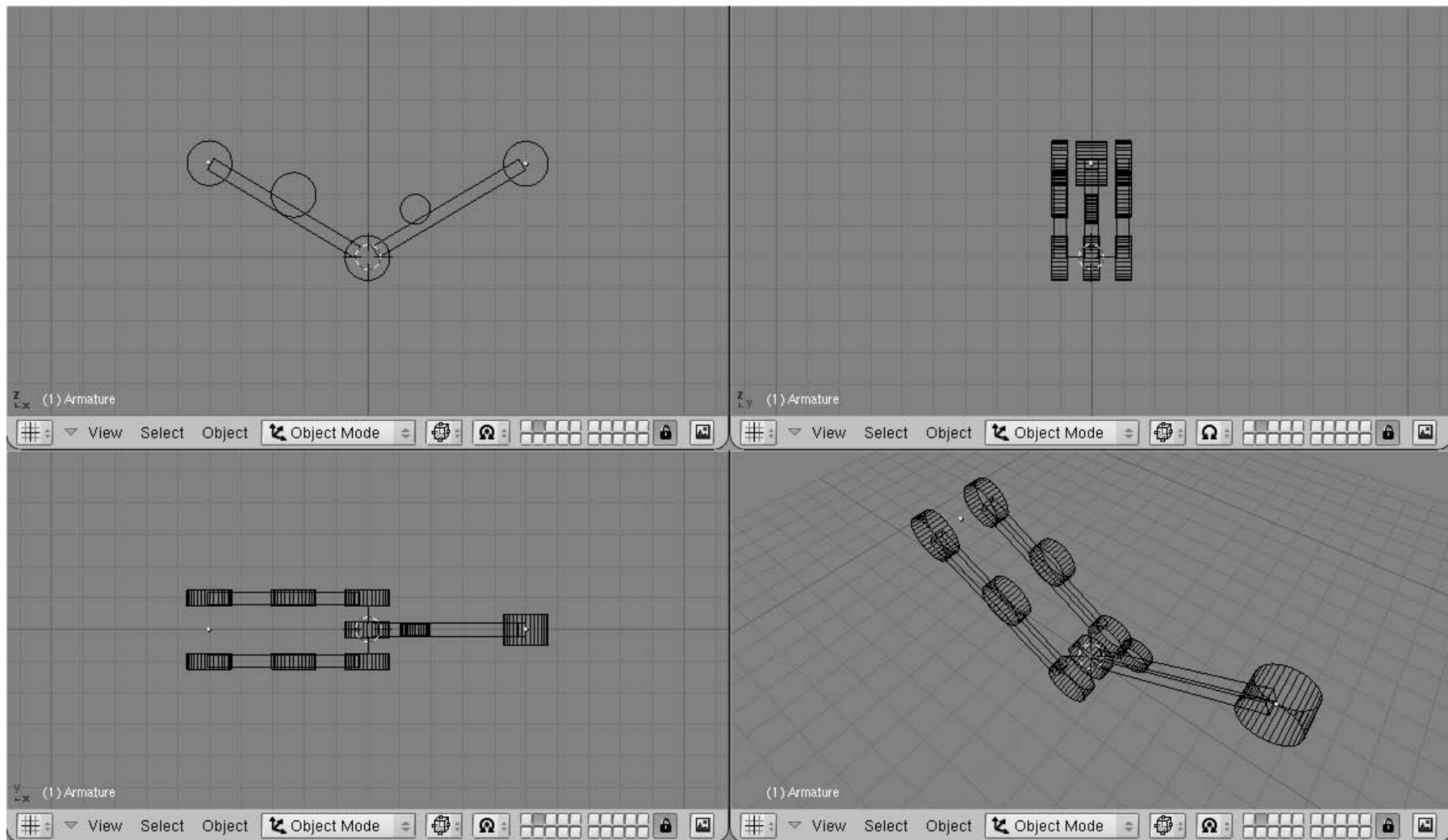


El primer paso es crear la malla para los brazos. Pero no estamos aquí para elementos orgánicos, estamos aquí para elementos mecánicos. Por tanto la malla no es tan simple. El brazo/pierna/loquesea está hecho de partes rígidas, con lo que cada parte es una malla individual, malla que se mueve/rota con respecto a las otras partes.

Aunque [Figure 16-55](#) la araña-robótica tiene cuatro patas, cada una de las cuales tiene 5 secciones, es mucho más sencillo explicar los trucos pertinentes si nos centramos en un única junta de una pata.

Mi sugerencia es la siguiente: haz el brazo con dos secciones iguales, y el antebrazo, en la derecha, hazlo solo de una sección. Dese cuenta de los cilindros que representan el hombro (izquierda), el codo (centro) y la muñeca (derecha) ([Figure 16-56](#)).

Figure 16-56. El modelo del brazo

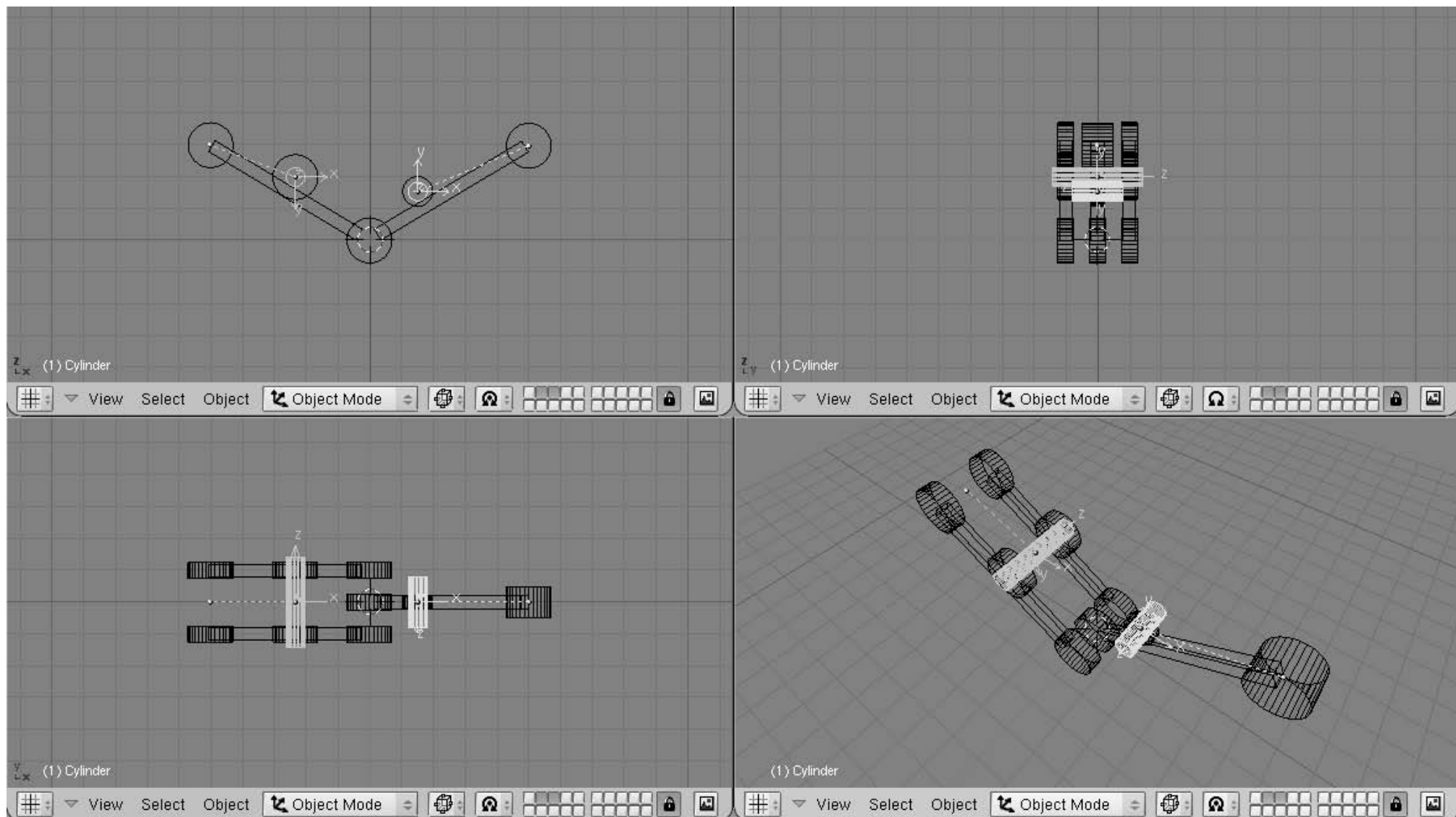


Los otros cilindros en el medio del brazo y del antebrazo están situados donde se enlazarán los pistones.

Dese cuenta que es mucho más sencillo si el eje de rotación propia (hombro, codo, etc.) está exactamente en los puntos de la rejilla. Esto no es absolutamente necesario si dominas bastante bien el menú Snap de Blender.

16.10.1. Ejes pivote

Entonces añade ejes mecánicos en los puntos pivotes. Teóricamente debería añadir uno en cada junta y dos en cada pistón. Para simplificar las cosas aquí solo tendremos los dos ejes en cada pistón, hecho con cilindros ([Figure 16-57](#)).

Figure 16-57. El modelo del brazo con los ejes pivotes.

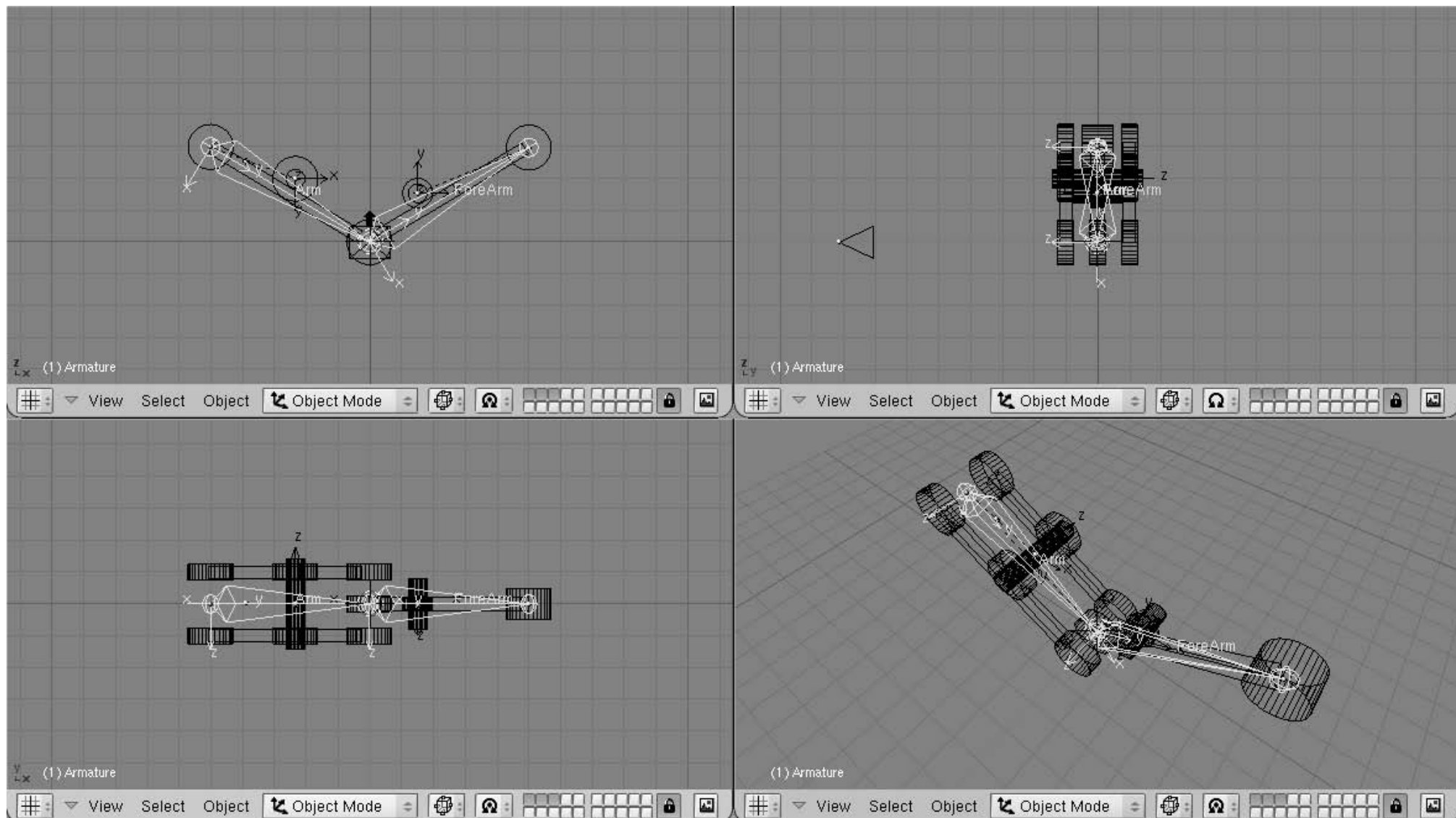
Dese cuenta de dos cosas:

- Es fundamental que el centro de la malla esté exactamente en la mitad y además exactamente en el eje de rotación del pistón.
- Cada eje debe de estar emparentado a su correspondiente malla del brazo.

16.10.2. La armadura

Es hora de configurar la armadura. Con dos huesos habrá suficiente ([Figure 16-58](#)).

Figure 16-58. El brazo y su armadura



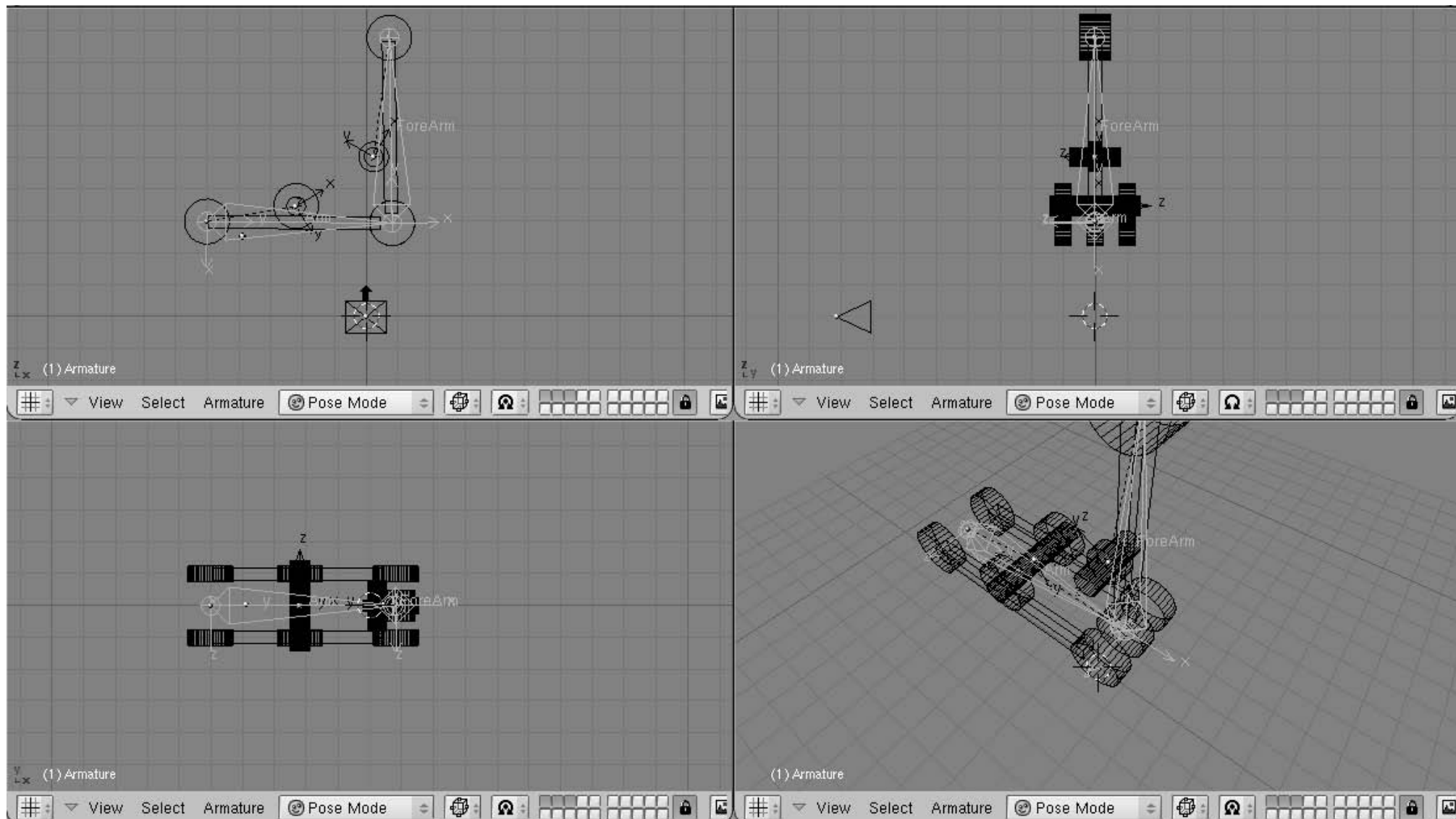
Para un movimiento preciso, las juntas deben situarse justamente en los pivotes de los ejes (Esto es el porqué de situar tales ejes en los puntos de la rejilla, para poder usar la característica "Move Selected To Grid (mover lo seleccionado a la rejilla)")

Nombra los huesos apropiadamente (Arm (brazo) y Forearm (antebrazo), por ejemplo). Emparenta la malla del brazo a la armature, seleccionando la opción *Bone* y el hueso del brazo. Haz lo mismo con la malla y el hueso del antebrazo.

Tip Emparentar a un hueso

Emparentar a un hueso efectivamente hace que el Objeto siga al hueso sin que se produzca ninguna deformación. Esto es lo que debería ocurrir para un robot que está hecho de piezas de acero indeformables!!.

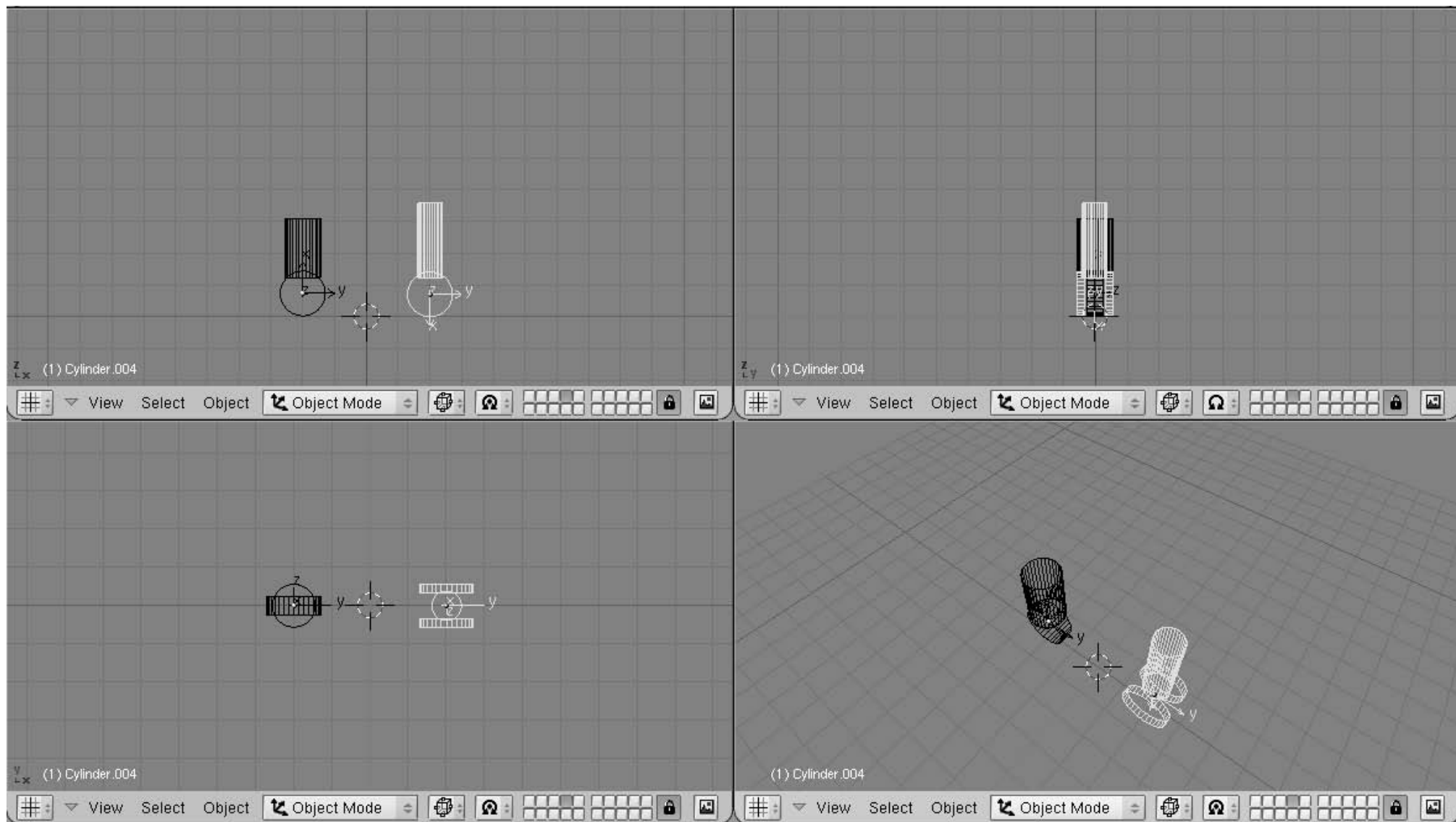
Figure 16-59. El brazo en Modo Posado



Si cambias a modo posado podrás mover el brazo simplemente rotando los huesos ([Figure 16-59](#)). Si lo prefieres puedes añadir un "IK solver" tal y como hicimos en la sección anterior.

16.10.3. Mecanismos hidráulicos

Figure 16-60. Piston hidráulico.

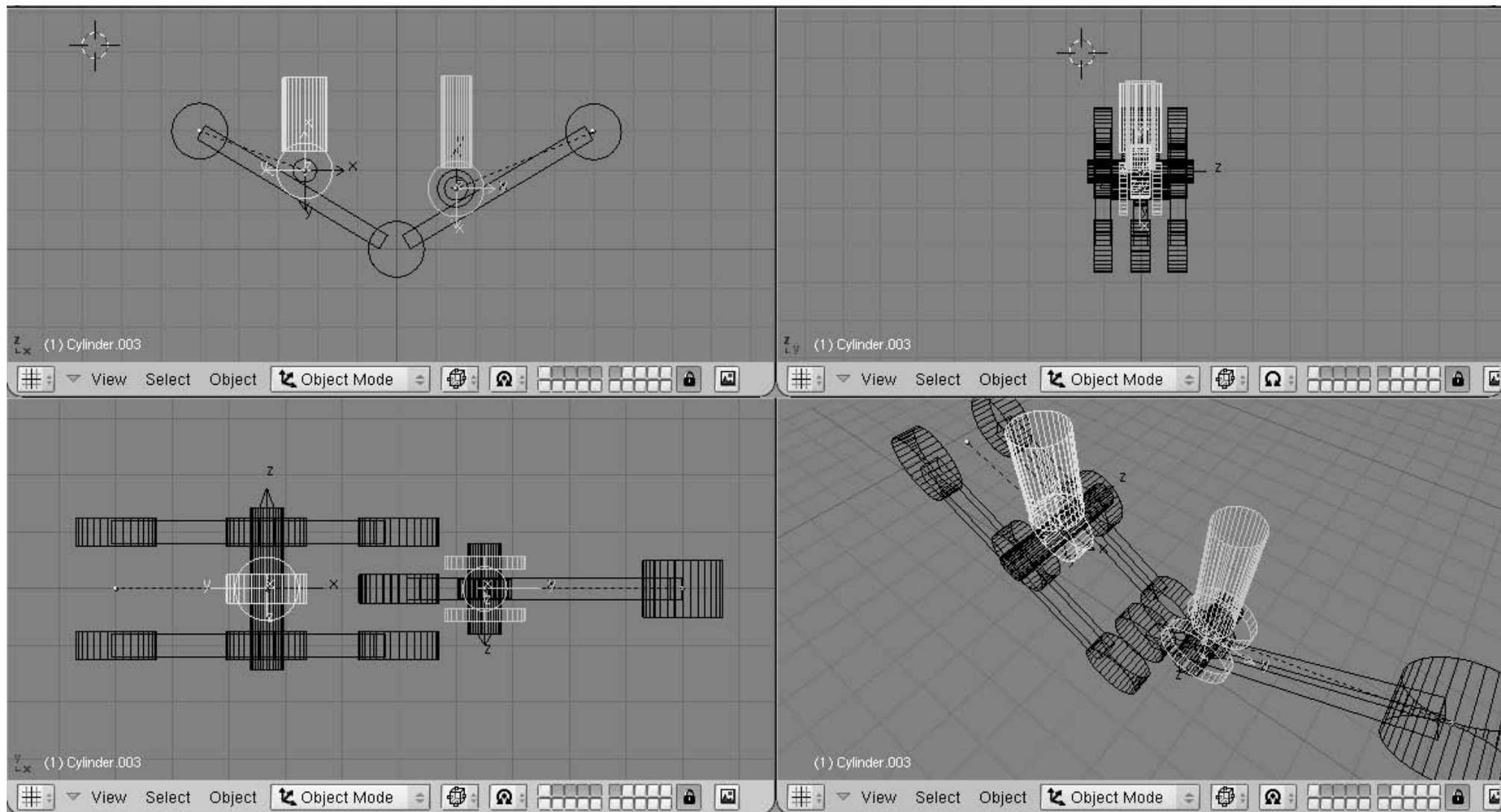


Haz un pistón con dos cilindros, uno más grande y otro más delgado, con alguna clase de elemento en la cabeza para enlazarlos a los puntos pivote ([Figure 16-60](#)).

Es *esencial* para ambas piezas tener el centro de la malla exactamente en sus respectivos ejes pivotes.

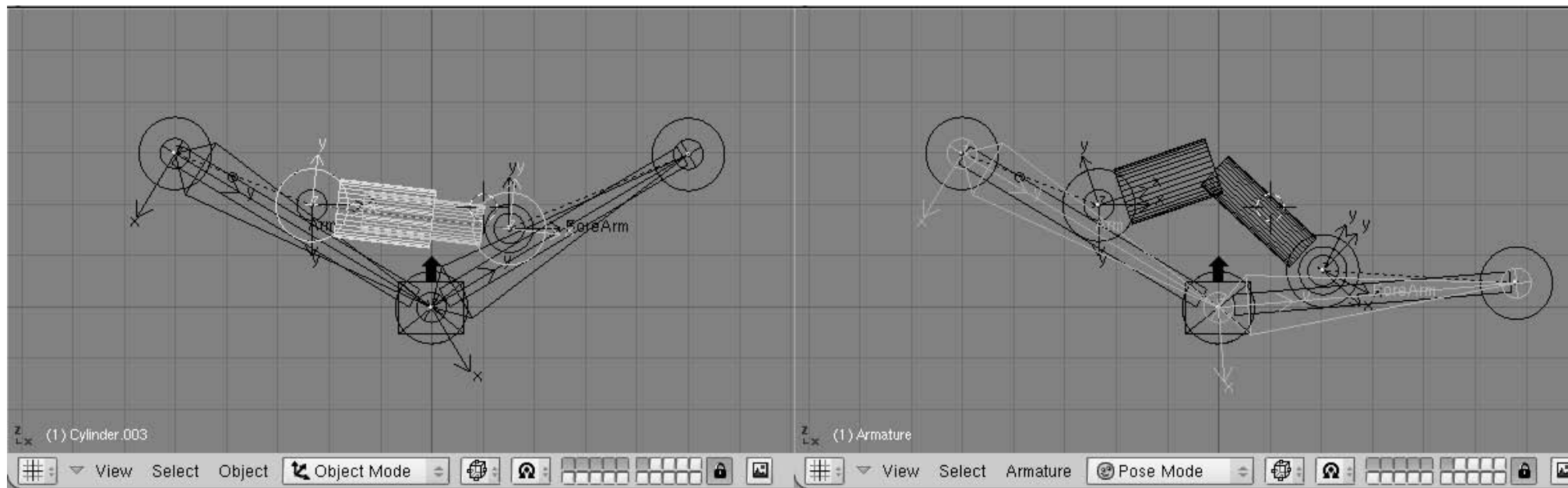
Sítalos en su posición correcta y emparenta cada pieza de pistón al acoplamiento pertinente que representa el eje ([Figure 16-61](#)).

Figure 16-61. Pistón hidráulico en el brazo.



Si ahora rotas las dos piezas y las situas en la posición que deberían tener para formar una correcta imagen *estática* obtendrás un bonito pistón. ([Figure 16-62](#), izquierda).

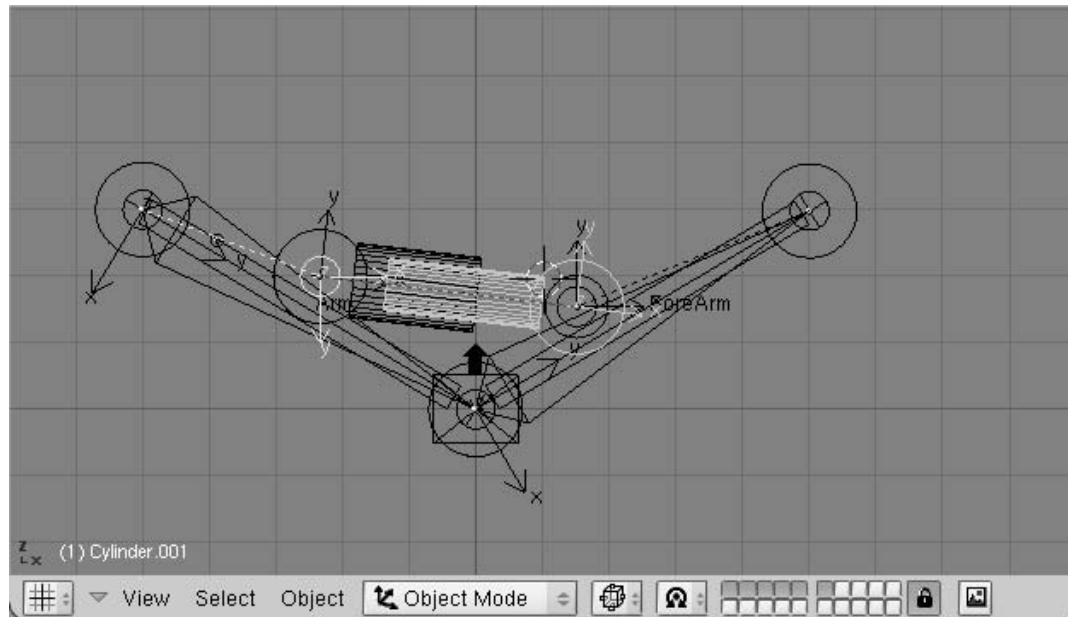
Figure 16-62. Pistón hidráulico en modo posado.



Pero si cambias al modo posado y comienzas a mover el brazo/antebrazo el pistón se estropea... (Figure 16-62, derecha).

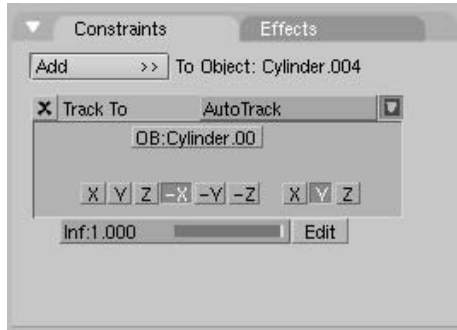
Para hacer un pistón totalmente funcional necesitarás hacer que cada mitad del pistón siga *al eje pivote de la otra mitad del pistón* de la malla cilíndrica (no a la otro mitad del pistón!). Esto crearía un bucle de restricciones) Esto es el porqué de la importancia de todos los centros de las mallas (Figure 16-63).

Figure 16-63. Pistón hidráulico con seguimiento mutuo.



Seleccione la mitad de un pistón, luego el eje de la otra mitad del pistón, y en el Contexto Objeto (**F7**) y en el panel *Constraints (restricciones)* añada "Track To Constraint". Los botones de abajo X,Y... deben configurarse apropiadamente ([Figure 16-64](#)).

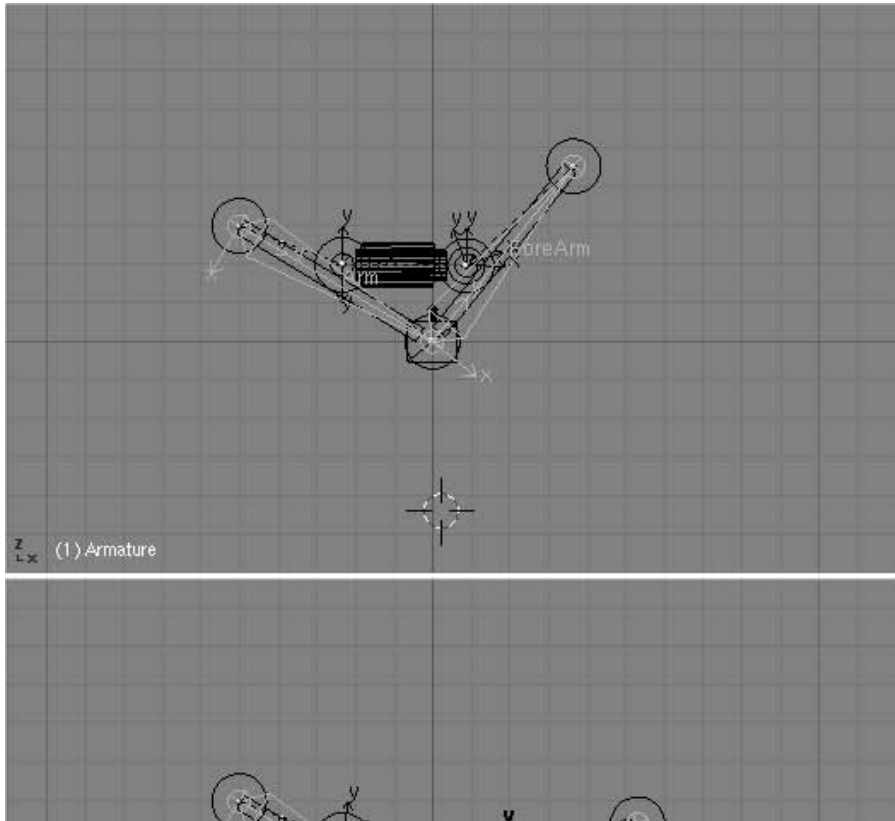
Figure 16-64. Configuración para el Tracking (seguimiento).

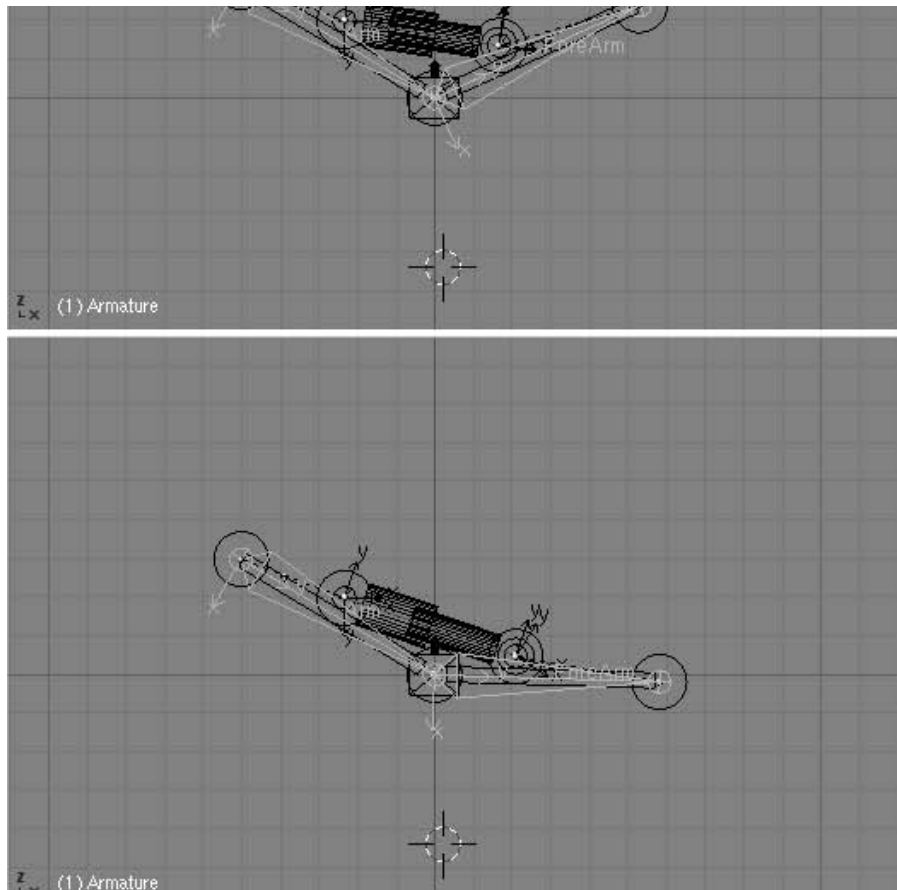


Note Si tu prefieres el viejo "Old Track" método, recuerda que también debes presionar el botón *PowerTrack* en el panel *Anim Setting* para un buen resultado.

Ahora, si cambias al modo posado y rotas los huesos el pistón se extenderá y contraerá adecuadamente, como debería ocurrir en la realidad ([Figure 16-65](#)).

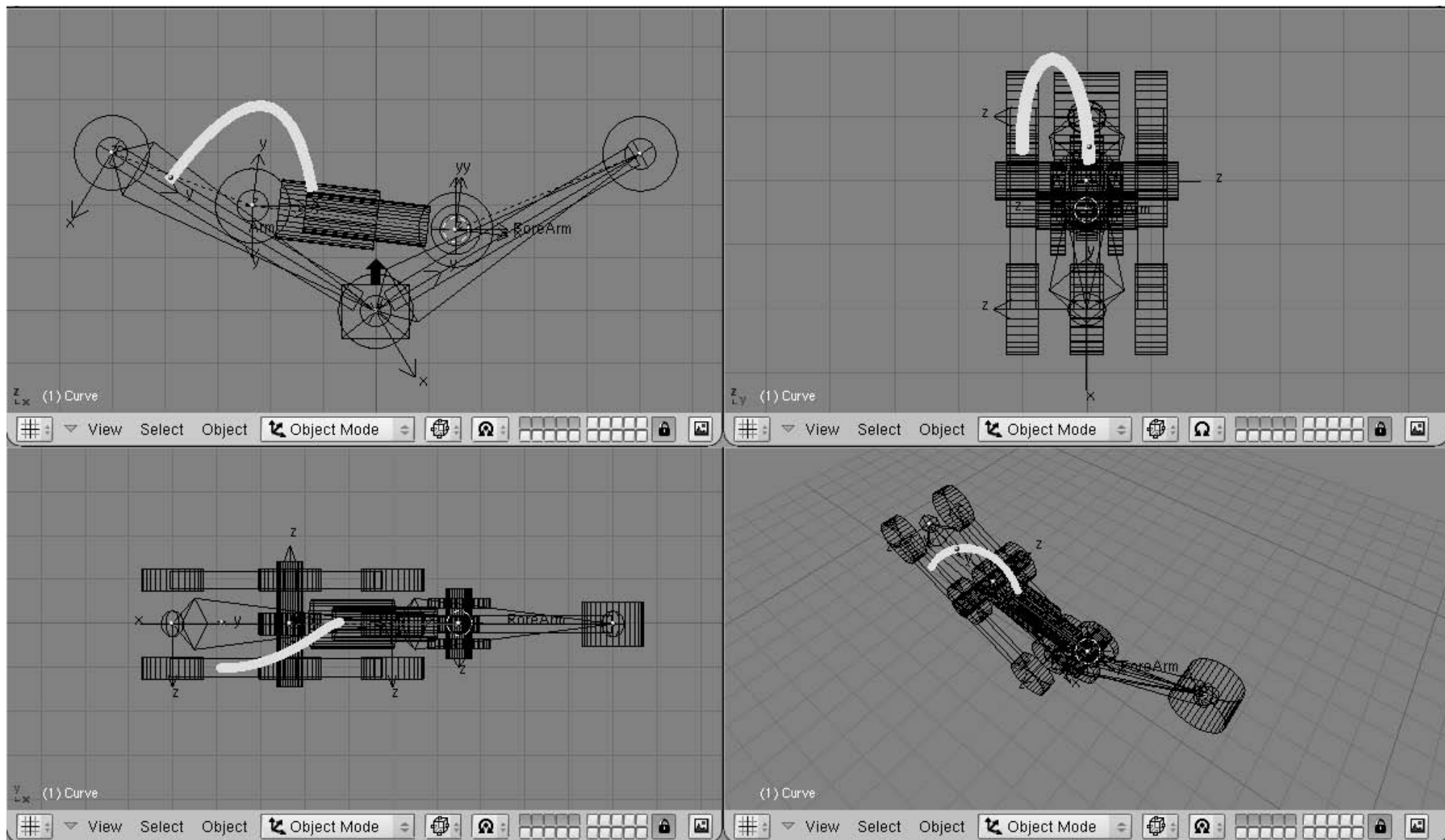
Figure 16-65. Modo Posado para el brazo con elementos hidráulicos.





Ahora el siguiente problema reside en, como los pistones trabajan con aceite presurizado en su interior, para hacer un modelo extremadamente aproximado a la realidad tendremos que añadir algunos tubos. Pero, ¿Cómo situar un tubo deformado que va desde el brazo al pistón? Los dos finales deberían unirse a dos sólidos rígidos rotando reciprocamente. Esto requiere IK!

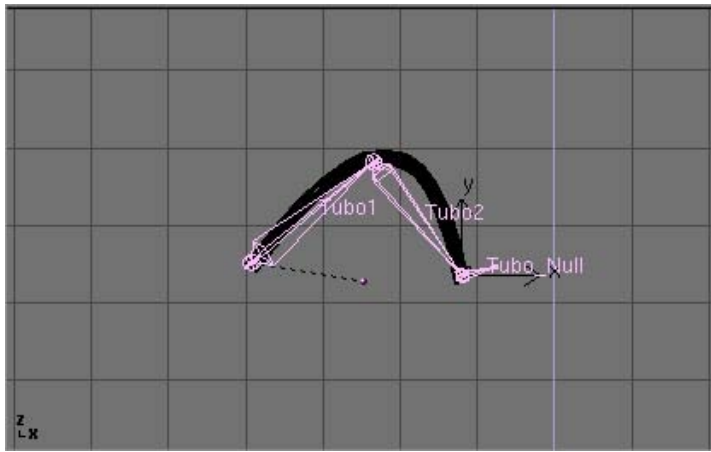
Figure 16-66. Añadiendo un tubo flexible.



Primero añade una malla con la forma del tubo que tu quieras modelar. ([Figure 16-66](#)).

Personalmente prefiero dibujar el tubo en su posición doblada como una curva "bevelled". Esto se hace añadiendo una curva Bézier, añadiendo un círculo Bézier, y usando el círculo Bézier como "BevOb" de la curva Bezier. Entonces conviértela en una malla con **ALT-C** para que seas capaz de deformarlo con una armature.

Figure 16-67. Añadiendo la armature al tubo.



Entonces añada una armature. Un par de huesos bastarán. Esta armature debería ir desde el final del tubo "fijo" al final del tubo "movil". Añada un tercer hueso que se usará para la solución de Cinematica Inversa (IK) ([Figure 16-67](#)).

Aseguraté de que la armature esté emparentada al objeto donde la parte "fijada" del tubo esté, bien, "fijada". En este caso el brazo del robot. Añada también un Empty a la parte "movil" del tubo. ([Figure 16-68](#)).

Figure 16-68. El Empty para la solución IKA.

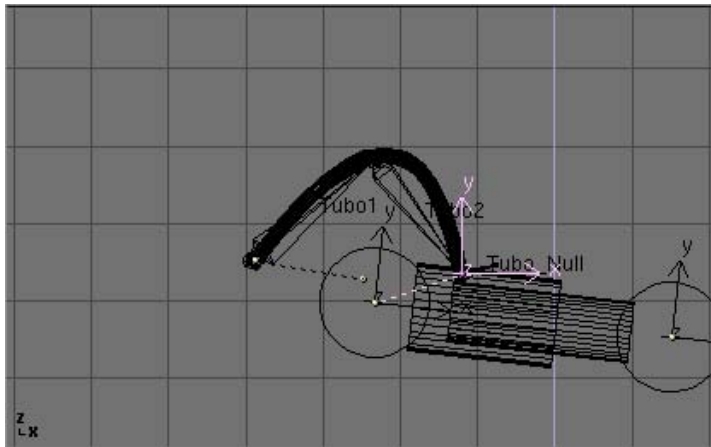
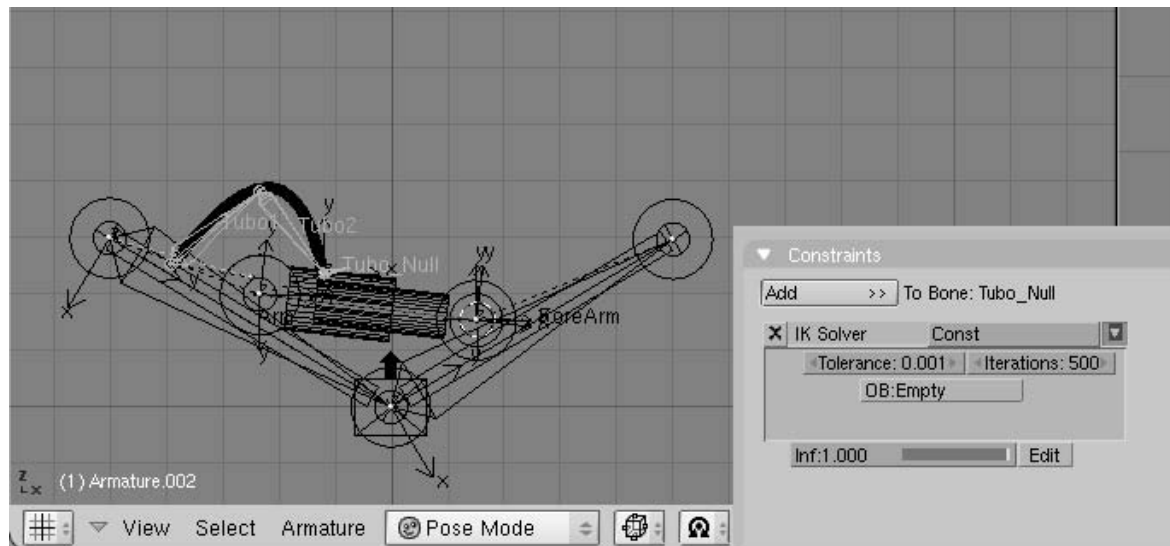


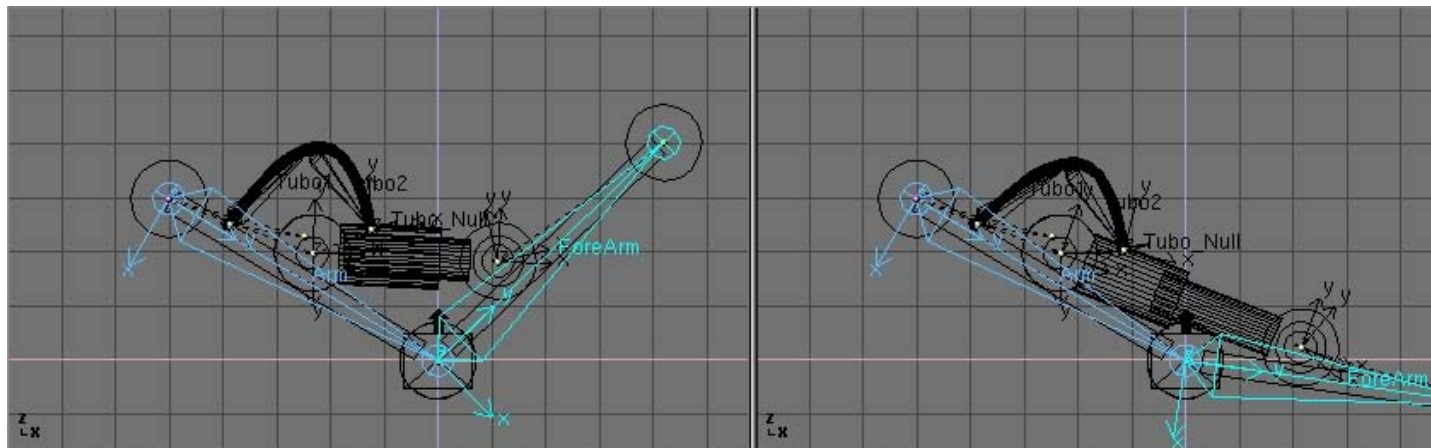
Figure 16-69. Restricción IKA.



Emparenta el Empty a la parte 'móvil' de la estructura. En esta caso, la parte exterior del pistón a la que el tubo está enlazada. En modo posado ve al contexto Objeto y al panel Constrains. Selecciona el último hueso, el que empieza donde el tubo finaliza, y añade una restricción. Selecciona el tipo de restricción "IK solver" y selecciona el Empty como objetivo con OB:. (Figure 16-69). Si te gusta puedes jugar un poco con la "Tolerance (Tolerancia)" y las "Iterations (Iteraciones)".

Por último, emparenta el tubo a la Armature a través de la opción 'Armature'. Create Vertex groups if you like. Now if, in pose mode, you move the arm, the two parts of the piston keeps moving appropriately, and the Empty follows. This obliges the IK Armature of the tube to move, to follow the Empty, and, consequently, the Tube to deform (Figure 16-69).

Figure 16-70. El brazo del robot completo en modo posado.



Note Puedes usar un hueso de la Armature, en vez de un empty, como un "IK solver", pero en este caso no puede emparentar el hueso al objeto en movimiento. Puede por otro lado, usar la restricción "Copy Location", pero esto no es tan fácil desde que "copy location" moverá el final de la armature al centro del objeto en movimiento, el cual no es el sitio correcto.

16.11. Cómo hacer un ciclo de andar usando NLA

por Claudio 'Malefico' Andaur

Relevante to Blender v2.31

En este tutorial vamos a configurar un ciclo de andar y a usarlo con la opción *Path* en el editor NLA de Blender. Antes de empezar aclaremos que necesitarás un conocimiento básico de las herramientas de animación, (configuración de armatures) y tener mucha paciencia para poder seguir el texto. Es muy recomendable que hayas leído todas las partes previas sobre NLA de este libro.

Vamos a utilizar una configuración de personaje como la explicada en el ["tutorial Rigging de Pies y Manos"](#), es decir con los huesos de los pies separados del resto de la pierna y usando un hueso nulo extra para contener una restricción de Solución IK. Para más detalles al respecta porfavor consulta con esa sección!

Teniendo un personaje configurado, la primer cosa que necesitamos es definir las acciones: "WALKCYCLE", "WAVE_HAND" y "STAND_STILL". En WALKCYCLE y STAND_STILL habrá grupos de keyframes para casi todos los huesos de control mientras que en "WAVE_HAND" habrá KeyFrames solamente para los brazos y manos. Esto permitirá a nuestro personaje mover las manos meintras camina.

La idea principal detrás de esto es trabajar soibre cada moviemento separadamente y luego combinarlos todos en la ventana NLA.

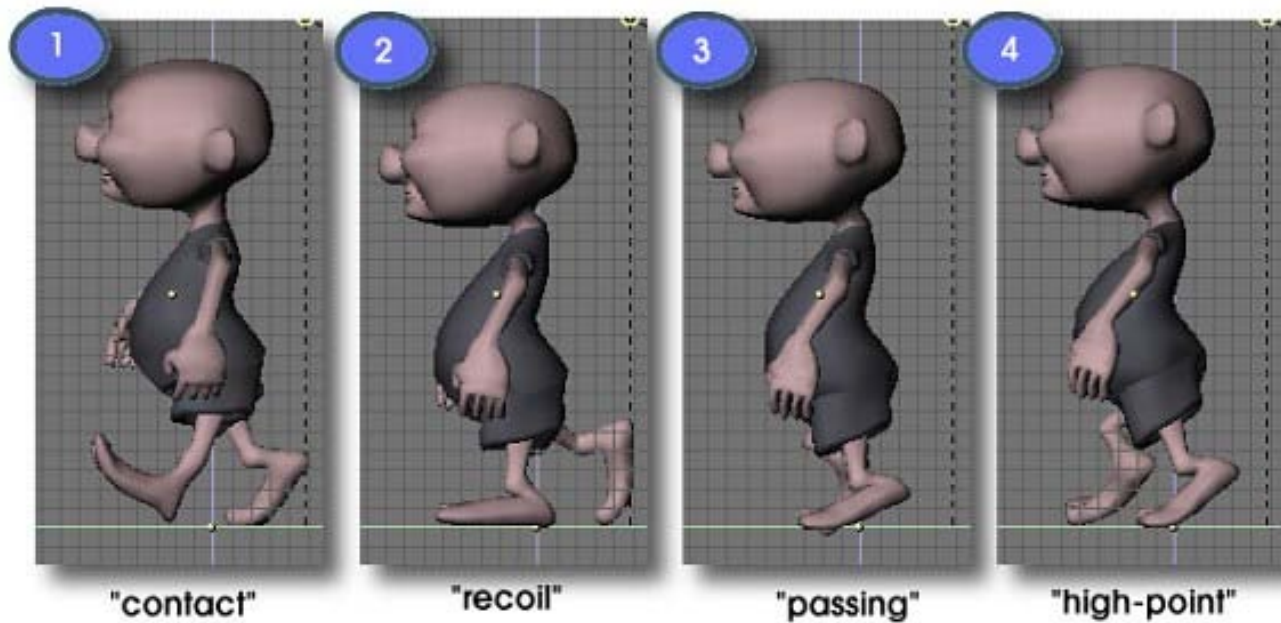
16.11.1. El camino del éxito

Existen dos formas elementales para animar un ciclo de andar, la primera es hacer que el personaje realmente avance a través de las poses del ciclo, la segunda es hacerlo caminar *in situ* es decir, sin desplazamiento real.

La última opción, auqnue un poco más deificl de configurar, es la mejor opción para animación digital y será nuestra elección para este tutorial.

Todo el ciclo de andar será una "acción" de neustro esqueleto, así que vayamos y creemos una nueva acción y cambiemos a Modo Pose para obtener algo parecido a la Pose 1 (la conocida como "*pose de contacto*" en [Figure 16-71](#)).

Figure 16-71. Algunas poses típicas de ciclo de andar.



Note Hay algunos detalles para tener en cuenta a la hora de configurar un esqueleto para ciclo de andar. Si adoptamos la convención de nomenclatura de Blender presentada en [Section 16.3](#) seremos capaces de pegar poses invertidas. Además, antes de emparentar nuestro esqueleto a nuestro personaje, nos aseguraremos de que los ejes locales del mismo estén alineados con los ejes globales seleccionándolos y presionando **CTRL-SHIFT-A**.

Para animar a nuestro modelo andante nos limitaremos a animar sus pies ya que las restricciones IK ajustarán automáticamente la posición de los huesos de las piernas mucho mejor que nosotros mismos. Para asegurarnos que ambos pies se muevan en distancias fijas, activaremos la opción *Ajustar a Rejilla* en los botones *Vista y Controles* en la ventana de Preferencias del Usuario antes de empezar a mover huesos. Reduciremos el tamaño de la rejilla si es necesario.

Un buen método es esconder, con el botón apropiado, todos los huesos a los que no vayamos a ponerle keyframes. De esta manera es más sencillo ver el modelo durante la animación y mantendrá nuestro trabajo lo más simple posible.

Normalmente un ciclo de andar abarca cuatro poses que se conocen comúnmente como poses de *contacto*, *retroceso* o *punto bajo*, *pasante* y *punto alto*. Observa la [Figure 16-71](#).

La pose más importante es la "pose Contacto". La mayoría de los animadores concuerdan en que cada ciclo de andar debe comenzar por definir correctamente esta pose. En ella el personaje alcanza la mayor distancia que es posible recorrer en un solo paso. En la "pose Retroceso", el personaje está en su posición más baja, con todo el peso sobre una pierna. En la "pose Punto Alto", el personaje está en su posición más alta, casi cayendo hacia adelante. La "pose Pasante" es casi una pose automática intermedia entre Retroceso y Punto Alto.

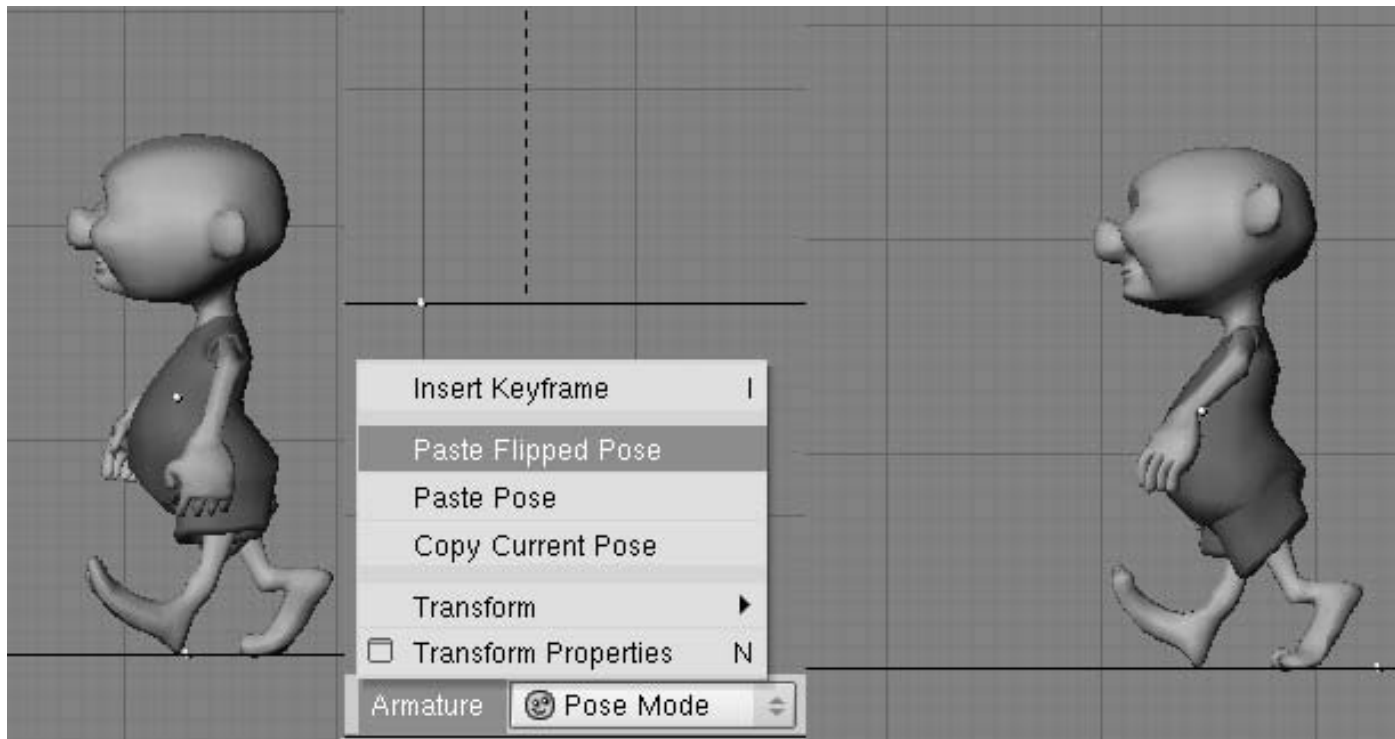
La rutina de trabajo sería como sigue:

1. Posamos el modelo en Pose Contacto en el cuadro 1.
2. Insertamos KeyFrames para los huesos de control de nuestro esqueleto (aquellos que usamos para manipular, principalmente huesos IK).
3. Sin deseleccionar nada, presionamos el botón "Copiar Pose". En ese momento la ubicación y rotación de nuestros huesos ha sido almacenada en la memoria.
4. Avanzamos unos cuadros adelante y pulsamos "Pegar Pose Invertida". La pose invertida se pegará en este cuadro de forma que si en el cuadro anterior la pierna izquierda apuntaba hacia delante, ahora lo hace hacia atrás y viceversa.
5. Ahora seleccionamos otra vez los huesos de control e insertamos KeyFrames.
6. Avanzamos otros tantos cuadros de nuevo (conviene avanzar el mismo número de cuadros que antes, un a opción sencilla es avanzar siempre 10 cuadros) y presionamos "Pegar Pose", esto pegará la pose inicial completando el ciclo. De este modo hemos logrado un estilo de andar "a lo Michael Jackson" ya que nuestro personaje nunca levanta los pies del suelo.
7. Para remediarlo, iremos a alguna posición intermedia entre las dos primeras poses y moveremos los pies a algo similar a la Pose Retroceso en [Figure 16-71](#), donde la cadera alcanza su postura más baja.
8. Insertamos KeyFrames y copiamos la pose.
9. Avanzamos 10 cuadros hasta un cuadro entre las últimas dos poses (Contacto Invertido y Contacto) y pegamos la pose invertida. Insertamos los KeyFrames necesarios y ya estamos.

Tip Si por el contrario ves que la malla se deforma de manera rarísima, no cunda el pánico !, ve a Modo Edición para el esqueleto, selecciona todos los huesos y presiona **CTRL-N**. Esto recalculará la dirección de las articulaciones de los huesos que producen el efecto de torcedura.

Deberías seguir la misma rutina para todas las poses que desees incluir en tu ciclo de andar. Normalmente utilizo las poses Contacto, Retroceso y Punto Alto dejando a Blender que calcule la pose Pasante por sí mismo.

Figure 16-72. Usa los botones Copiar, Pegar y Pegar Pose Invertida y serás feliz!



Ahora si hacemos **ALT-A** verás a nuestro personaje caminar casi naturalmente.

Será muy útil contar cuantas Unidades de Blender (U.B) recorren en cada paso, lo que se puede hacer contando los cuadrados de la rejilla entre los pies cuando el personaje está en la Pose 1 (contacto). Este número es el Avance o **STRIDE** que usaremos más adelante en la ventana NLA.

Ahora nos concentraremos en hacer que nuestro personaje se desplace de hecho en nuestra escena.

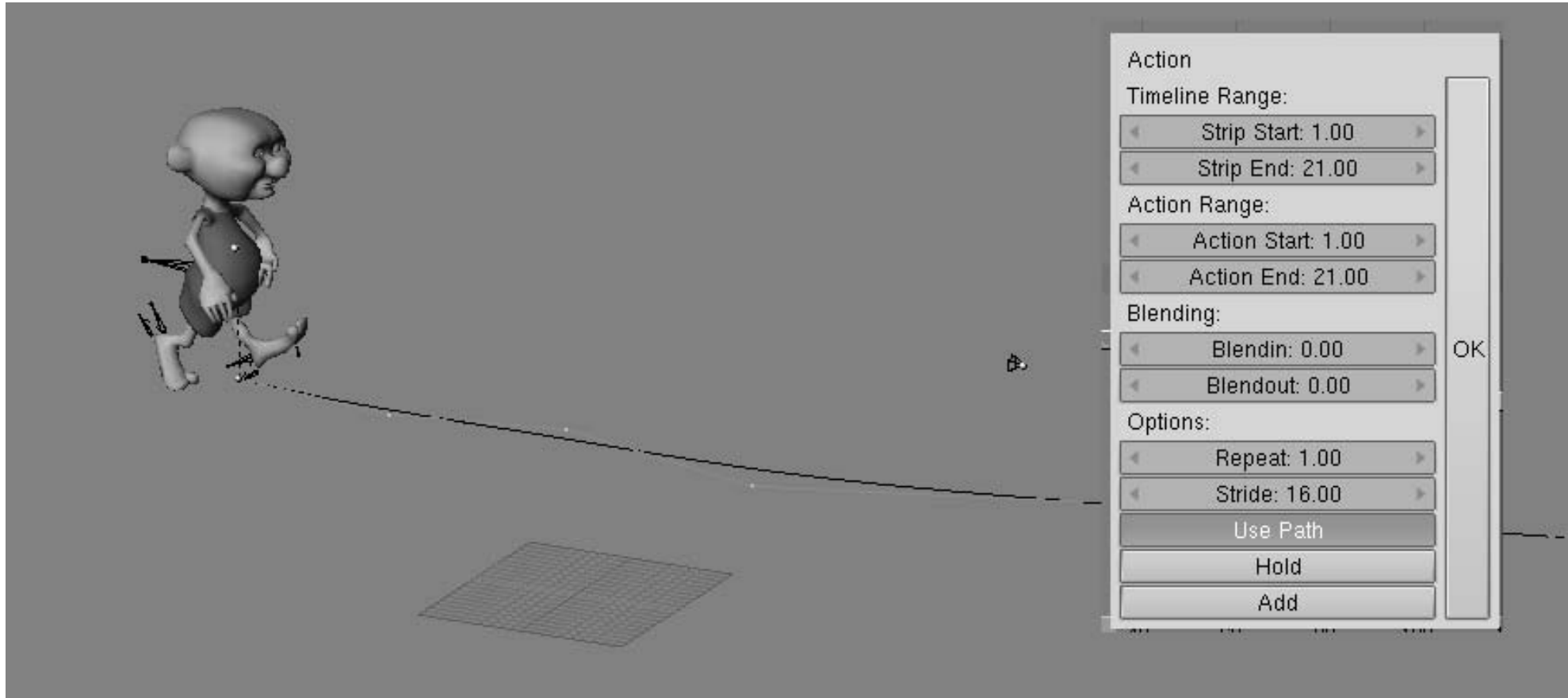
Primero que nada, deseleccionaremos la acción ciclo de andar de nuestro armature, de modo de que deje de moverse cuando hagamos **ALT-A**. Para ello pulsaremos el botón con la pequeña X junto al nombre de la acción en la ventana del Editor de Acciones.

Luego crearemos un objeto CAMINO para nuestro héroe en el plano del suelo intentando no hacerlo demasiado sinuoso por ahora (mientras más rectilíneo mejor), cuando hayamos terminado emparentaremos el armature del personaje al mismo (un emparentado *normal*, no una restricción Follow Path!). Si todo anduvo bien veremos a nuestro personaje moverse rígido a lo largo del camino cuando pulsemos **ALT-A**.

Vayamos a la ventana NLA y añadamos en un canal, como una banda NLA, la acción ciclo de andar. Con la banda seleccionada pulsaremos **N** y luego el botón **Use Path**.

Note Es conveniente que en el momento de añadir acciones como bandas en la ventana NLA no tengamos seleccionada ninguna acción para nuestra armature ¿Por qué? Porque en lugar de una banda NLA veremos aparecer insertados en el canal los KeyFrames individuales de la acción, y estos KeyFrames solaparán completamente toda banda de animación NLA precedente que hayamos agregado hasta el momento. De cualquier manera si lo has hecho, siempre puedes convertir estos KeyFrames en una banda NLA presionando **CKEY**.

Figure 16-73. Un lindo paseo



Si lanzamos la animación de nuevo, pueden pasar algunas cosas raras. Esto se debe a que no hemos establecido el parámetro *Stride*.

Este parámetro es el número de Unidades Blender que debería recorrerse en un solo ciclo de andar y es muy importante que lo estimemos con precisión. Una vez calculado lo entraremos en el botón numérico *Stride* que aparece si seleccionamos la banda NLA y presionamos la tecla **NKEY**.

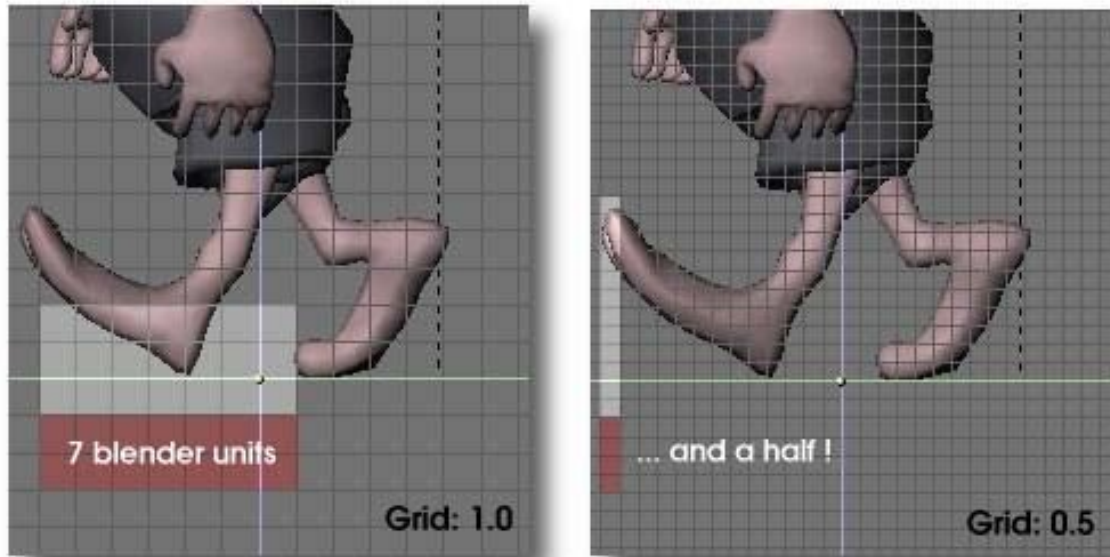
Si lo ajustamos bien y hemos configurado correctamente nuestro ciclo de andar, nuestro personaje no debería "patinar" a lo largo del recorrido.

Una forma de estimar con exactitud es contar cuantos cuadrados de la rejilla hay entre las puntas de los pies en la Pose 1. Este valor multiplicado por 2 y

por la escala de la rejilla (normalmente 1 UB = 1 cuadrado pero podría no ser el caso, por ejemplo en este ejemplo tenemos que 1UB = 2 cuadrados) nos dará el valor de STRIDE requerido.

En el ejemplo hay 7.5 cuadrados, y la escala de la rejilla es GRID=1.0, con lo que tenemos que: $STRIDE = 7.5 \times 1.0 \times 2 = 15$

Figure 16-74. Estimando el STRIDE. Refina la rejilla si es necesario !

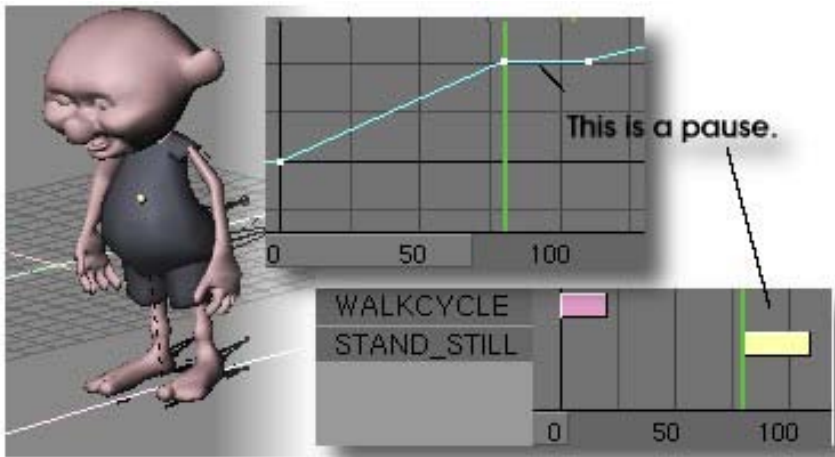


Es posible que queramos que nuestro personaje camine más rápido o más despacio o que incluso se detenga durante un momento. Podemos hacer todo esto editando la curva Speed del camino.

Selecciona el camini y abre una ventana IPO. Allí veremos una curva Speed normalizada entre las coordenadas 0 y 1 (eje Y) que va desde el cuadro 1 hasta el último cuadro de la animación en el eje X. La coordenada Y representa la posición relativa en el camino mientras que la pendiente o inclinación de la curva es la velocidad de los objetos que están emparentados al mismo. En Modo Edición añadiremos 2 nodos con la misma coordenada Y. Esta parte plana representa una pausa en el movimiento y va desde el cuadro 40 al cuadro 60 en la figura.

El problema es que cuando nuestro personaje se detenga a causa de esta pausa en la curva, lo veremos en una pose congelada con un pie en el suelo y el otro en el aire.

Figure 16-75. Having a rest in the walk



Para solucionar este problemita, usaremos la ventana NLA. Lo que debemos hacer es insertar la acción "STAND_STILL" , que es una pose donde nuestro personaje está en reposo. He definido esta acción con un único Keyframe donde he borrado todas las rotaciones y desplazamientos de los huesos (Ver [Borrando Transformaciones](#)), y luego he acomodado un par de huesos para darle una actitud "relajada".

Como la pausa va del cuadro=78 al cuadro=112 deberemos insertar esa acción "quieto" exactamente allí para que coincida perfectamente con la pausa. Para que la animación no empiece ni termine bruscamente podemos utilizar las opciones BlendIn y BlendOut, con las que indicamos la cantidad de cuadros que serán usados para mezclar acciones y de esta manera obtener una transición más natural entre ellas. De este modo el personaje cambiará su pose suavemente y todo se verá bien. Si usamos un valor para *BlendIn* o *BlendOut*, introduciendolo a través del dialogo **NKEY** , deberemos comenzar la acción *BlendIn* cuadros antes y finalizarla *BlendOut* cuadros después, porque el personaje aún debería estar moviendose mientras cambia de poses.

Podemos combinar diferentes ciclos de andar en el mismo camino para por ejemplo cambiar de "caminar" a "correr" en una zona de mayor velocidad.

En todas estas situaciones tendremos que tener en mente que los diferentes efectos se sumarán de una banda NLA a las precedentes. Por eso, la mejor opción es insertar el ciclo de andar y las poses estáticas antes que cualquier otra..

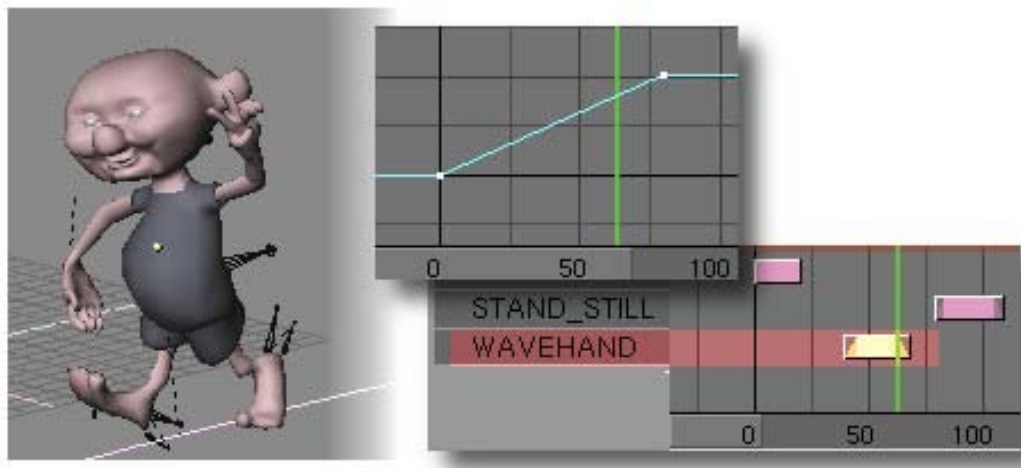
16.11.2. Moviendo las manos mientras caminamos

Para añadir acciones en la ventana NLA tenemos que ubicar el puntero del ratón encima del canal del armature y presionar **SHIFT-A**. Aparecerá un menú con todas las acciones disponibles. Si por lo contrario hemos ubicado el puntero erróneamente aparecerá un mensaje de error **ERROR: Not an armature**.

Entonces, ubiquemos el puntero del ratón sobre el canal del armature, presionemos **SHIFT-A** e insertemos la acción "WAVE_HAND" ("MOVER_MANO").

COMO esta acción en particular es sólo el movimiento del brazo izquierdo para decir "hola" en algunos momentos durante el ciclo de andar, no usaremos la opción "Use Path" sino que la desplazaremos en la línea de tiempo a nuestro gusto y solapará los KeyFrames de los brazos en la acción ciclo de andar. Con el puntero sobre la banda NLA, persionaremos **NKEY** para indicar comienzo y final de la banda o simplemente la arrastraremos y escalaremos hasta lograr el resultado deseado.

Figure 16-76. Hola amigos!



Como esta acción es la última en calcularse (recuerda que Blender evalua las acciones de arriba hacia abajo en el editor NLA) solapará cualquier KeyFrame definido para los huesos comunes a las acciones anteriores.

Ya no hay mucho más que decir sobre NLA y armatures. Es hora de que experimentes por tu cuenta y muestres al mundo tus resultados. Valga sin embargo una última recomendación: Es posible editar KeyFrames en la ventana NLA. Podemos duplicar cuadros (**SHIFT-D**), mover cuadros (**GKEY**) e incluso borrar cuadros (**XKEY**), pero si borras KeyFrames ten cuidado pues aquello que borres será eliminado permanentemente en la acción seleccionada. Por ello asegurate de convertir los cuadros siempre en una banda NLA antes de borrar nada.

Adiós y buena suerte Blenderadictos !!

[Prev](#)

Rigging de elementos mecánicos

[Home](#)

[Up](#)

[Next](#)

Renderizado

IV. Renderizado

Hay más, en el motor de renderizado de Blender de lo que puedes ver con una simple pulsación de **F12**.

Esta parte te guiará a través de todas las características del renderizado que tiene Blender, desde el Motion Blur hasta Cartoon-like edge (dibujos animados). Un capítulo entero está dedicado a su motor de radiosity, el cual permite unas escenas iluminadas muy reales sin necesidad de utilizar ninguna luz.

Table of Contents

17. [Renderizado](#)

- 17.1. [Renderizando por Partes](#)
- 17.2. [Renders panorámicos](#)
- 17.3. [Antialiasing](#)
- 17.4. [Formatos de salida](#)
- 17.5. [Renderizando animaciones](#)
- 17.6. [Desenfoque de Movimiento \(Motion Blur\)](#)
- 17.7. [Profundidad de Campo](#)
- 17.8. [Bordes de Caricatura](#)
- 17.9. [El Renderizador Unificado \(Unified Renderer\)](#)
- 17.10. [Preparando su trabajo para vídeo](#)
 - 17.10.1. [Saturación de Color](#)

18. [Radiosidad](#)

- 18.1. [El método de Radiosidad en Blender](#)
- 18.2. [Renderizando la Radiosidad](#)
- 18.3. [Radiosidad como herramienta de modelado](#)
 - 18.3.1. [Fase 1: Collect Meshes](#)
 - 18.3.2. [Fase 2: límites de Subdivisión](#)
 - 18.3.3. [Fase 3: Subdivisión adaptativa](#)
 - 18.3.4. [Fase 4: Editando la solución](#)
- 18.4. [Un jugoso ejemplo de radiosidad](#)
 - 18.4.1. [Setting up.](#)
 - 18.4.2. [La cúpula del cielo](#)
 - 18.4.3. [La solución de la radiosidad](#)
 - 18.4.4. [Texturizado](#)

19. [Trazado de rayos \(Raytracing\)](#)

[Prev](#)

[Home](#)

[Next](#)

Cómo hacer un ciclo de andar
usando NLA

Renderizado

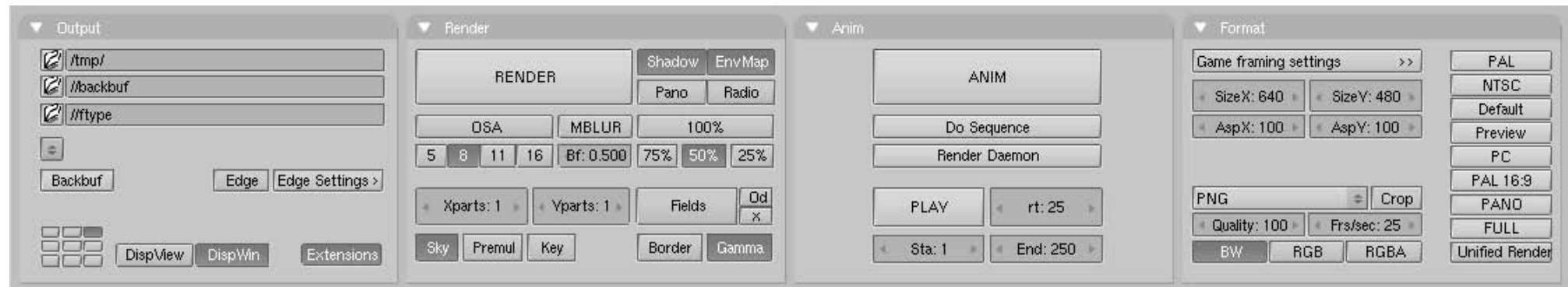
Chapter 17. Renderizado

Relevante a Blender v2.31

El render es el proceso final del CG (antes de la postproducción, por supuesto) y es la fase en la que la imagen que se corresponde con la escena 3D se crea finalmente.

La ventana de botones de render es accesible a través del panel Escena (Scene Context) y el subpanel render. (**F10** o el botón ). Los paneles y botones de render se pueden ver en [Figure 17-1](#).

Figure 17-1. Botones de render.



El render de la escena actual se realiza presionando el botón grande *RENDER* en el panel *Render*, o presionando **F12**. El resultado del render es almacenado en un buffer y se ve en la misma ventana. Puede ser guardado presionando **F3** o a través del menú *File>>Save Image*.

La imagen es renderizada en función de las dimensiones definidas en el panel *Format* ([Figure 17-2](#)).

Figure 17-2. Tipos de imagenes y dimensiones.



Por defecto las dimensiones *SizeX* y *SizeY* son 320x256 y pueden ser modificadas con cualquier botón numérico. Los dos botones más abajo definen el ratio de proporción de los pixels. Éste es un ratio entre las dimensiones X e Y de los pixels de la imagen. Por defecto está en 1:1 dado que los pixels de la pantalla son cuadrados, pero puede ser modificado si es para televisión, ya que sus pixels

no son cuadrados. Para simplificar, el bloque de botones trae algunas configuraciones predeterminadas: [Figure 17-3](#))

Figure 17-3. Dimensiones de imagen predeterminadas.



- *PAL* 720x576 pixels ratio 54:51.
- *NTSC* 720x480 pixels ratio 10:11.
- *Default* Igual que *PAL*, pero con todas las opciones para TV, como se explicará en las siguientes secciones.
- *Preview* 640x512 con ratio de 1:1. Automáticamente reduce la imagen al 50% para producir una imagen real de 320x256.
- *PC* 640x480 con ratio de 1:1.
- *PAL 16:9* 720x576 con ratio 64:45, para renders a TV en widescreen 16:9.
- *PANO* Configuración panorámica estándar 576x176 con ratio de 115:100. Más info sobre el render 'panorámico' en la sección pertinente.
- *FULL* 1280x1024 con ratio de 1:1.

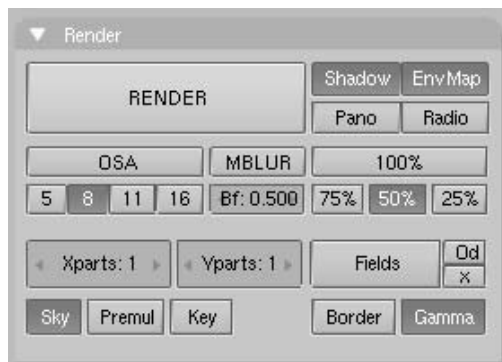
17.1. Renderizando por Partes

Relativo a Blender v2.31

Es posible renderizar una imagen por partes, una después de la otra, en lugar de toda de una vez. Esto puede ser útil para escenas muy complejas, donde renderizar pequeñas secciones, una a continuación de otra, solo necesita el procesamiento de una pequeña parte de la escena, lo cual utiliza menos memoria.

Al asignar valores diferentes de 1 en *Xpart* y en *Ypart* en el Panel *Render* ([Figure 17-4](#)) se obliga a Blender a dividir la imagen en una grilla de *Xpart* por *Ypart* sub-imágenes, que son renderizadas una a continuación de la otra y finalmente ensambladas como una sola imagen.

Figure 17-4. Botones de Renderizado por partes.



Note

Blender no puede manejar más de 64 partes.

[Prev](#)

Renderizado

[Home](#)

[Up](#)

[Next](#)

Renders panorámicos

17.2. Renders panorámicos

Relevante a Blender v2.31

Para conseguir bonitos renders panorámicos, de hasta 360 grados; con vista al horizonte, Blender nos provee de un procedimiento automático.

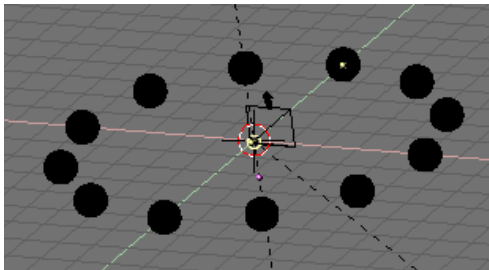
Si *Xparts* es mayor que 1 y el botón *Pano* del panel *Render* está presionado ([Figure 17-5](#)), entonces la imagen renderizada se crea *Xparts* veces *SizeX* de ancho y *SizeY* de altura, renderizando cada parte rotando esa parte la cámara tan lejos como sea necesario para conseguir la imagen con el menor número de uniones posible.

Figure 17-5. Botón panorama.



[Figure 17-6](#) muestra una configuración test con 12 esferas alrededor de una cámara. Dejando la cámara como está obtendrá el render mostrado en [Figure 17-7](#). Si pone *Xparts* en 3 y seleccionando *Panorama* el resultado será una imagen tres veces más ancha, mostrando un plano de cámara a mayores a la derecha y otro a la izquierda ([Figure 17-8](#)).

Figure 17-6. Configuración test panorámico.



Para conseguir algo similar sin la opción Panorámica, el único camino es reducir la longitud focal de la cámara. Por ejemplo [Figure 17-9](#) muestra una vista comparable, a una obtenida con una longitud focal de 7.0, equivalente a un ángulo muy ancho, o lentes de "ojo de pez". La distorsión es evidente.

Figure 17-7. Renders no panorámicos.

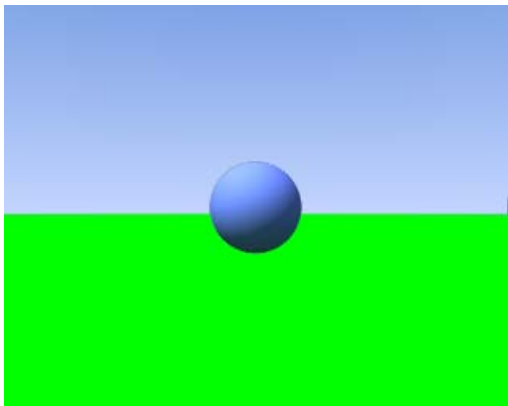


Figure 17-8. Renderizado panorámico.

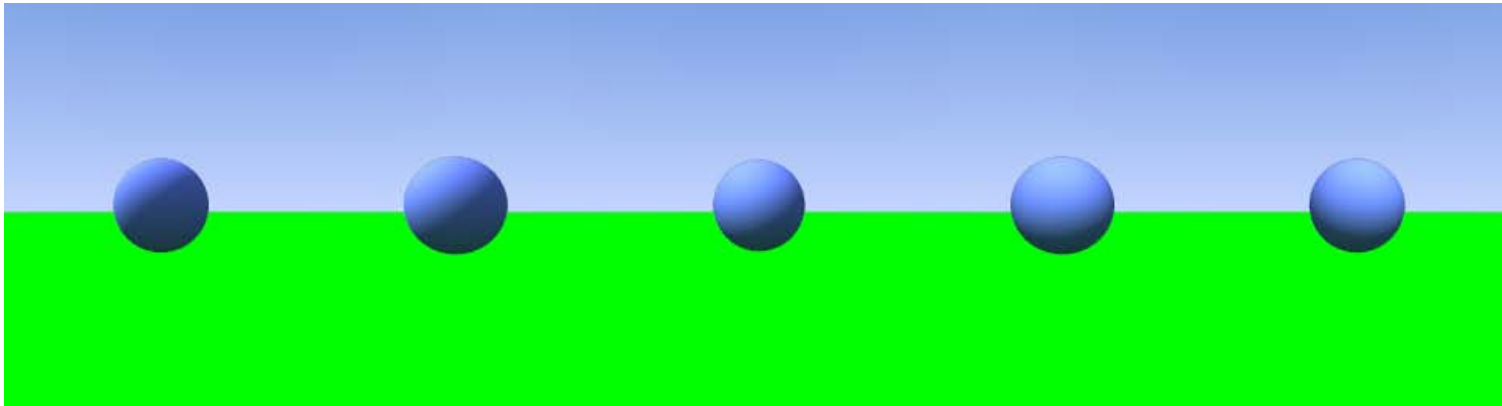
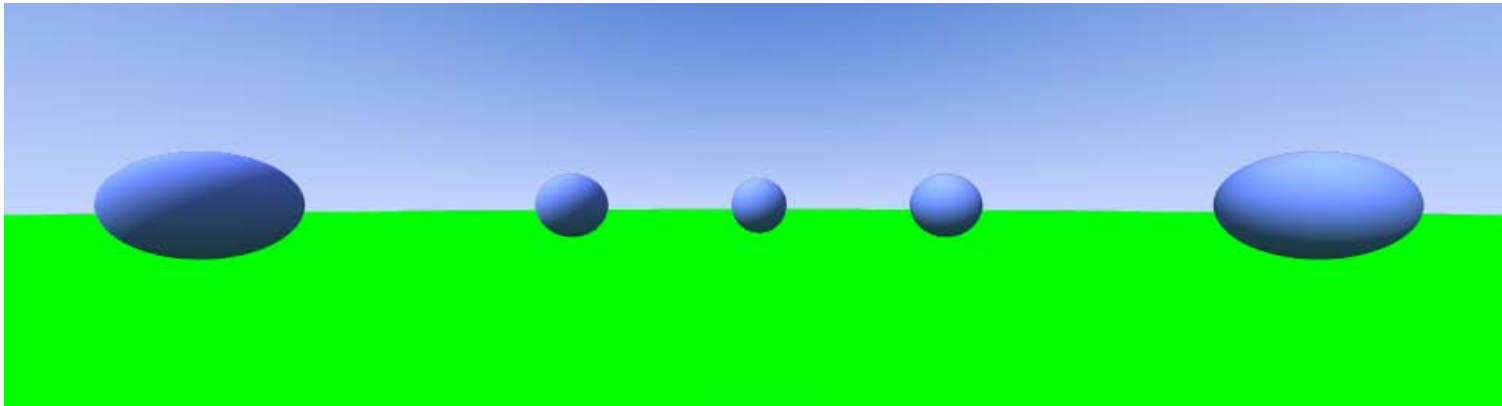
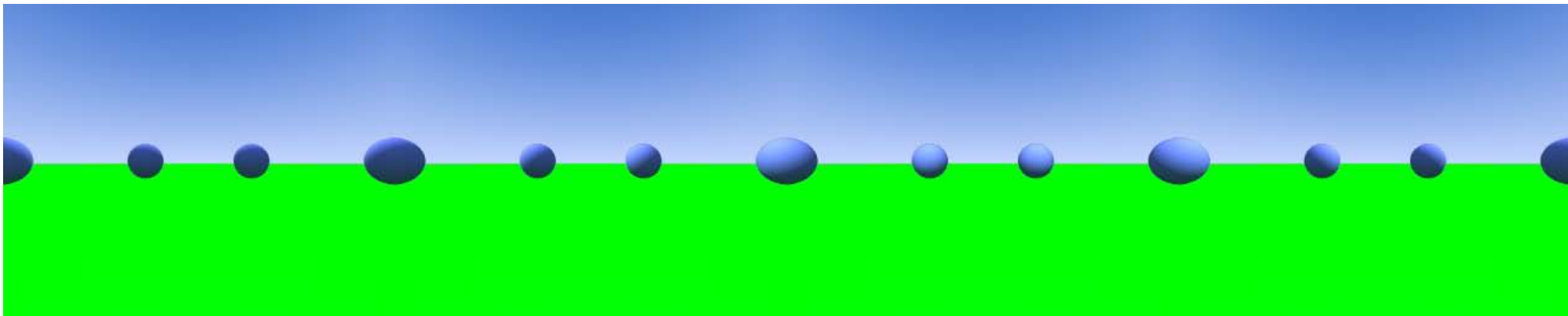


Figure 17-9. Renderizado "ojo de pez".



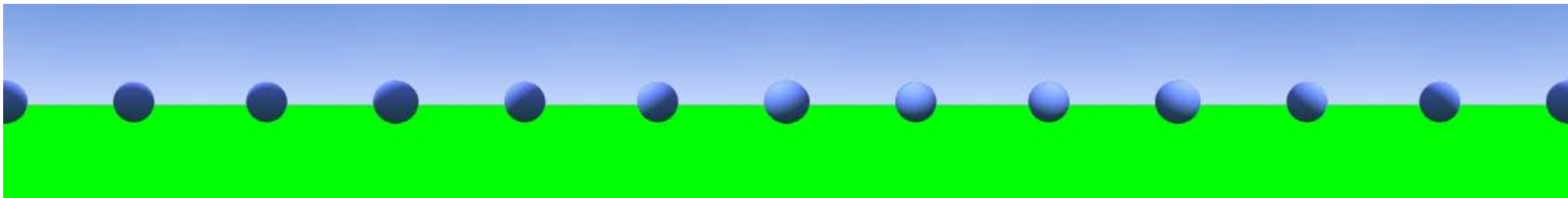
Para conseguir una vista completa de 360 grados; es necesario hacer algunos trucos. Es sabido que una longitud focal de 16.0 corresponde con un ángulo de visión de 90 grados. Entonces un render panorámico con 4 Xparts y una cámara con una lente de 16.0 supone una vista completa de 360 grados, como se ve en [Figure 17-10](#). Queda sumamente distorsionado, ya que una lente de 16.0 es muy ancha, y distorciona en los lados.

Figure 17-10. Panorámica 360 grados con lentes 16.0.



Para conseguir vistas sin distorsionar, la distancia focal debería ser de alrededor de 35.0. [Figure 17-11](#) muestra el resultado de una panorámica con 8 Xparts y una cámara con una lente de 38.5, que corresponde a un ángulo de visión de 45 grados.

Figure 17-11. Panorámica de 360 grados con lente de 38.5.



La imagen está mucho menos distorsinada, pero se debe prestar especial atención a la proporción. La imagen original era de 320x256 pixels. La panorámica en [Figure 17-10](#) es de 4x320 de ancho. Para mantener a esta nueva panorámica con el mismo ancho, el valor *SizeX* de la imagen debe ponerse a 160 de forma que $8 \times 160 = 4 \times 320$. Pero el ancho del ángulo de visión de cámara está para la dimensión mayor, de forma que, si *SizeX* se mantiene en 256 la imagen se prolonga 45 grados verticalmente pero menos que horizontalmente, así el resultado final no es una panorámica de 360 grados. O bien *SizeX* es mayor que *SizeY* o tendrá que estar dispuesto a hacer unos cuantos test.

[Prev](#)[Renderizado](#)[Home](#)[Up](#)[Next](#)[Antialiasing](#)

17.3. Antialiasing

Relativo a Blender v2.31

Una imagen generada por ordenador está compuesta de píxeles. Estos píxeles pueden ser por supuesto sólo de un color. En el proceso de renderizado el motor de render debe asignar un color a cada pixel basándose en qué objeto se muestra en ese pixel.

A menudo esto conduce a unos resultados pobres, especialmente en los límites muy contrastados, o donde hay líneas delgadas, y es particularmente evidente en las líneas oblicuas.

Para solucionar este problema, que se conoce como *Aliasing*, es posible recurrir a una técnica Anti-Aliasing. Básicamente, cada pixel se 'sobremuestra', renderizándolo como si fuera 5 píxeles o más, y asignando un color 'medio' al pixel renderizado.

Los botones para controlar el Anti-Aliasing, o el SobreMuestreo (OSA, OverSampling), se encuentran debajo del botón de render en el panel *Render* ([Figure 17-12](#)). Presionando el botón OSA se activa el antialiasing. Seleccionando uno de los cuatro botones numéricos bajo el, se determina el nivel de sobremuestreo (de 5 a 16).

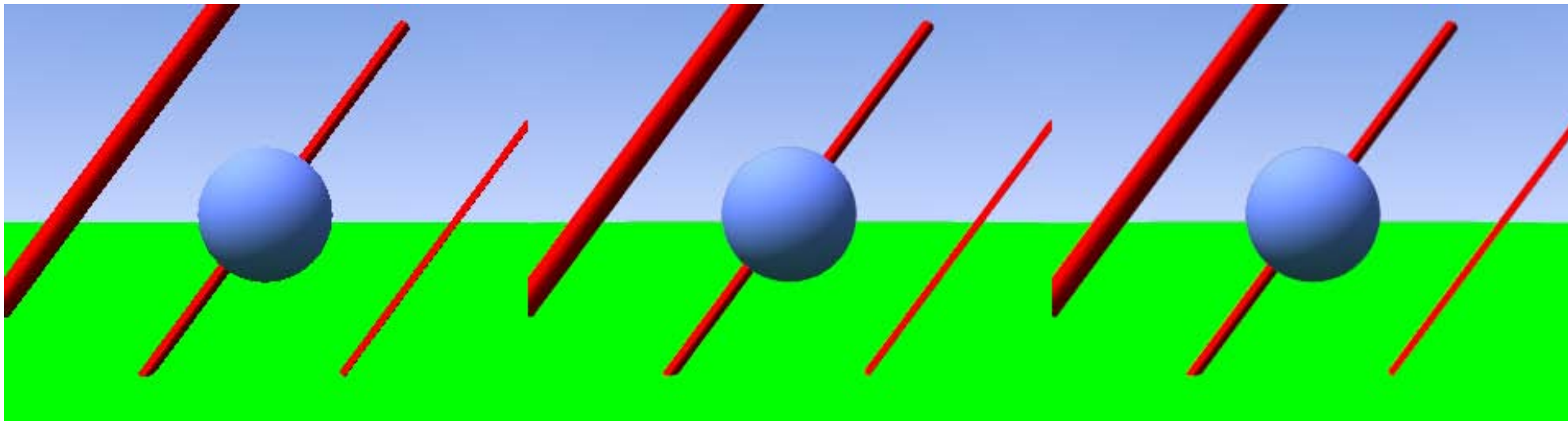
Figure 17-12. Botones OSA.



Blender usa un sistema de renderizado de Acumulación Delta con muestreo jittered. Los valores de OSA (5, 8, 11, 16) son números predefinidos que especifican el número de muestras; un valor más alto produce mejores bordes, pero ralentiza el proceso de renderizado.

[Figure 17-13](#) muestra un render con OSA desactivado y con 5 u 8 muestras OSA.

Figure 17-13. Render sin OSA (izquierda) con OSA=5 (centro) y OSA=8 (derecha).



[Prev](#)

Renders panorámicos

[Home](#)

[Up](#)

[Next](#)

Formatos de salida

17.4. Formatos de salida

Relevante a Blender v2.31

El fichero se salva en el formato que se haya seleccionado en el Panel *Format* ([Figure 17-2](#)). Aquí puede seleccionar varios formatos de imagen o animación ([Figure 17-14](#)).

Figure 17-14. Formatos de imagen y animación.



El tipo de imagen por defecto es *Targa*, pero desde que la imagen se guarda en un buffer y luego se guarda, es posible cambiar el tipo de formato de imagen después de que se haya renderizado usando este menú.

Por defecto Blender renderiza imagenes en color (*RGB*) (línea al final en [Figure 17-2](#)) pero también es posible renderizar en blanco y negro (*BW*) y en color con canal alpha (*RGBA*).

Blender *no* añade automáticamente la extensión a los ficheros, con lo que cualquier extensión *.tga* o *.png* se tiene que escribir en la ventana donde se guardan los ficheros.

Excepto el formato Jpeg, que es de compresión con pérdidas, todos los demás formatos son más o menos

equivalentes. Generalmente es una mala idea utilizar el formato Jpeg ya que es un formato con pérdidas. Es mejor usar el formato Targa y entonces convertirlo a Jpeg, ya que de esta manera mantenemos el original.

Para lo que concierne a los otros formatos: *TARGA raw* es el formato Targa sin compresión y usa un montón de espacio de disco. *PNG* es Portable Network Graphics, y es un estándar que se creó para intentar reemplazar al formato GIF y soporta imágenes con color verdadero. *HamX* es un formato de 8 bits con compresión RLE (Run Length Encoded bitmap); crea ficheros extremadamente compactos que se pueden mostrar rápidamente. Para usarse únicamente con la opción "Play". *Iris* es el formato estándar de SGI e *Iris + Zbuffer* es el mismo formato añadiéndole la información del Zbuffer (buffer de profundidad).

Finalmente *Ftype* usa un fichero "Ftype", para indicarle que este fichero sirve como ejemplo del tipo de formato gráfico en el cual Blender debe guardar las imágenes. Este método permite procesar el 'mapa de color' (colormap). El mapa de color se lee del fichero y se usa para convertir a 24 o 32 bit. Si la opción "RGBA" se especifica, el número de color estándar '0' se usará como color transparente. Blender lee y escribe mapas de colores en los siguientes formatos: (Amiga) IFF, Targa, (SGI) Iris y CDinteractive (CDi) RLE.

Por lo que concierne a las animaciones:

- *AVI Raw* - guarda un AVI sin compresión, sin pérdidas pero con un enorme tamaño.
- *AVI Jpeg* - guarda una AVI como una serie de imágenes Jpeg. Tiene pérdidas, tamaño pequeño pero no tan pequeño como puede obtenerse con un algoritmo de compresión mejor. Además el formato AVI Jpeg no lo lee la mayoría de los reproductores.
- *AVI Codec* - guarda un AVI comprimido con un codec. Blender automáticamente le proporciona la lista de los codecs disponibles en su sistema y le permite seleccionar los distintos parámetros. También es posible cambiarlo o cambiar sus parámetros, una vez seleccionado, a través del botón *Set Codec* que aparece en ([Figure 17-15](#)).

Figure 17-15. Configuración de los codec del AVI.



Para una animación en formato AVI también es posible seleccionar el ratio de frames (framerate) ([Figure 17-15](#)) que, por defecto, es de 25 frames por segundo.

[Prev](#)

Antialiasing

[Home](#)

[Up](#)

[Next](#)

Renderizando animaciones

17.5. Renderizando animaciones

Relevante a Blender v2.31

El renderizado de una animación se controla a través del panel *Anim* ([Figure 17-16](#)).

Figure 17-16. Botones para renderizar una animación.

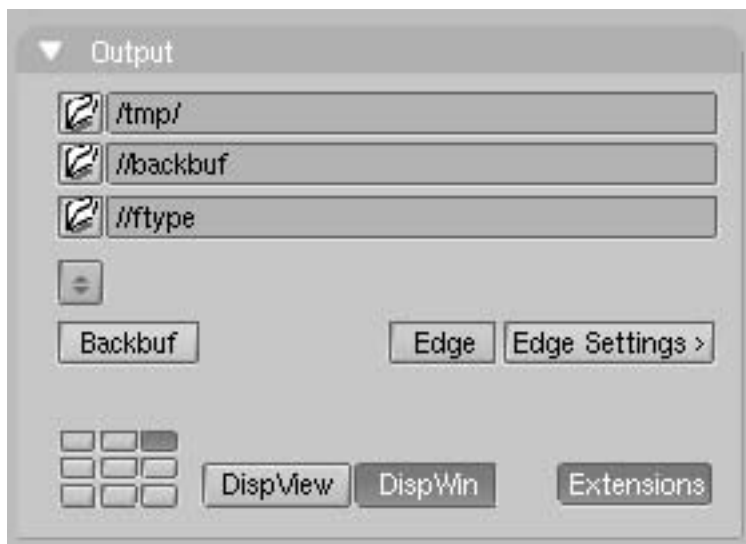


El botón *ANIM* comienza el renderizado. El primero y el último frame de la animación se configura con los dos botones numéricos del fondo (*Sta:* y *End:*), que por defecto están a 1 y a 250.

Por defecto la animación de la escena 3D se renderiza, entonces para hacer uso del editor de secuencia el botón *Do Sequence* debe estar seleccionado.

Por defecto la animación se renderiza en el directorio especificado en el Panel *Output* ([Figure 17-17](#)). Si se ha seleccionado el formato AVI, entonces el nombre será #####_#####.avi donde '#####' indica el frame inicial y el final de la animación con 4 numeros enteros añadiendo deros a la izquierda si fuese necesario.

Figure 17-17. Localización de la animación y las extensiones.



Si se elige, por otro lado, un formato de imagen, una serie de imagenes llamadas #####, (siendo '#####' el numero de frame correspondiente) se crea en el directorio. Si fuese necesaria la extension del fichero, esta puede obtenerse presionando el botón *Extensions* ([Figure 17-17](#)).

Tip Animaciones complejas

A menos que su animación sea muy sencilla, y espere renderizarla en media hora o menos, siempre es una buena idea, desde el principio, renderizar la animación como frames separados en formato Targa más que en un formato AVI.

Esto permite una recuperación sencilla si algo falla en la computadora y tiene que comenzar de nuevo a renderizar, ya que los frames que se han renderizado siempre estarán ahí.

Este método también le vendrá bien, en el caso de que un error ocurra en unos pocos frames, ya que puede hacer las correcciones oportunas y renderizar de nuevo solo los frames afectados.

Puede entonces crear el AVI con los frames afectados con el editor de secuencia de Blender o con otro programa que se lo posibilite.

[Prev](#)

Formatos de salida

[Home](#)

[Up](#)

[Next](#)

Desenfoco de Movimiento
(Motion Blur)

17.6. Desenfoco de Movimiento (Motion Blur)

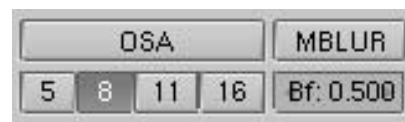
Relativo a Blender v2.31

Las animaciones de Blender son, por defecto, renderizadas como una secuencia de imágenes *perfectamente quietas*.

Esto no es realista, ya que los objetos que se mueven a gran velocidad, realmente parecen estar "en movimiento", es decir, desenfocados por su propio movimiento, tanto en un fotograma de una película como en una fotografía obtenida con una "cámara del mundo real".

Para obtener ese tipo de efecto de Desenfoco de Movimiento (Motion Blur), se le puede decir a Blender que haga el render del cuadro actual y de algunos cuadros más entre los cuadros reales, y combinarlos para obtener una imagen donde los detalles de los objetos que se mueven rápidamente aparecen "desenfocados".

Figure 17-18. Botones de Motion Blur.



Para acceder a esta opción hay que seleccionar el botón *MBLUR* que se encuentra al lado del botón *OSA* en el Panel *Render* ([Figure 17-18](#)). Esto hace que Blender renderice tantos cuadros "intermedios" como indique el valor en oversampling (5, 8, 11 o 16) y los acumule, uno sobre otro, en un sólo cuadro. El botón-número *Bf*: o Factor de Desenfoco (Blur Factor) define la duración del tiempo de obturación, como se mostrará en el ejemplo de abajo. Activar los botones de *OSA* no es necesario, ya que el proceso de Desenfoco de Movimiento (Motion Blur) agrega algo de antialiasing de todas formas, pero para tener una imagen realmente suavizada, *OSA* puede ser también activado. Esto hace que cada imagen acumulada tenga antia-aliasing.

Para entender mejor este concepto, asumamos que tenemos un cubo moviéndose en uniformemente una (1) unidad de Blender hacia la derecha en cada cuadro. Esto es realmente rápido, especialmente porque el cubo tiene un lado de sólo 2 unidades de Blender.

[Figure 17-19](#) muestra el render del primer cuadro, sin Desenfoco de Movimiento (Motion Blur), [Figure 17-20](#) muestra el render del cuadro 2. La escala debajo del cubo ayuda a apreciar el movimiento de una (1) unidad de Blender.

Figure 17-19. Cuadro 1 del cubo en movimiento, sin Desenfoco de Movimiento (motion blur).

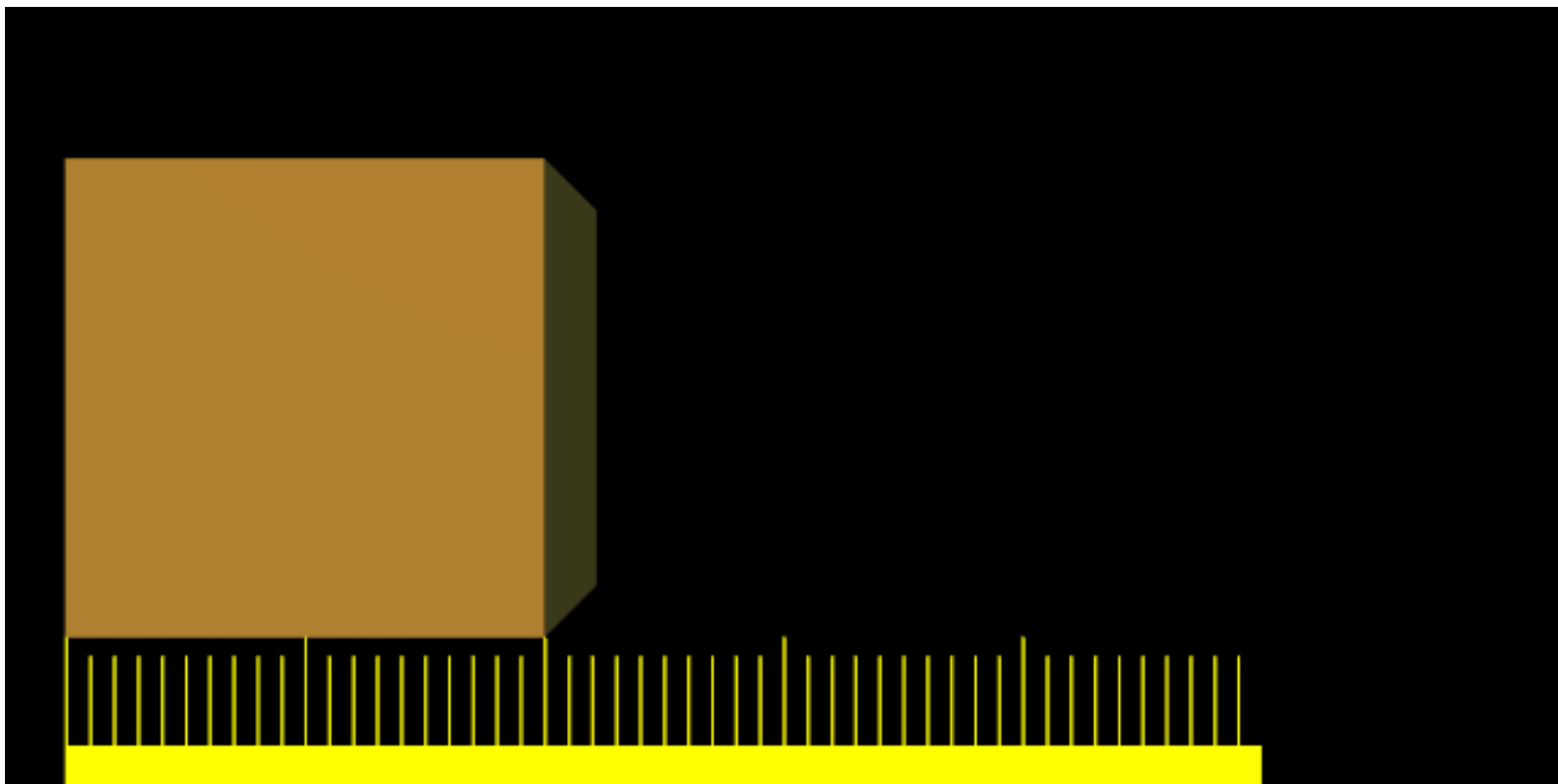
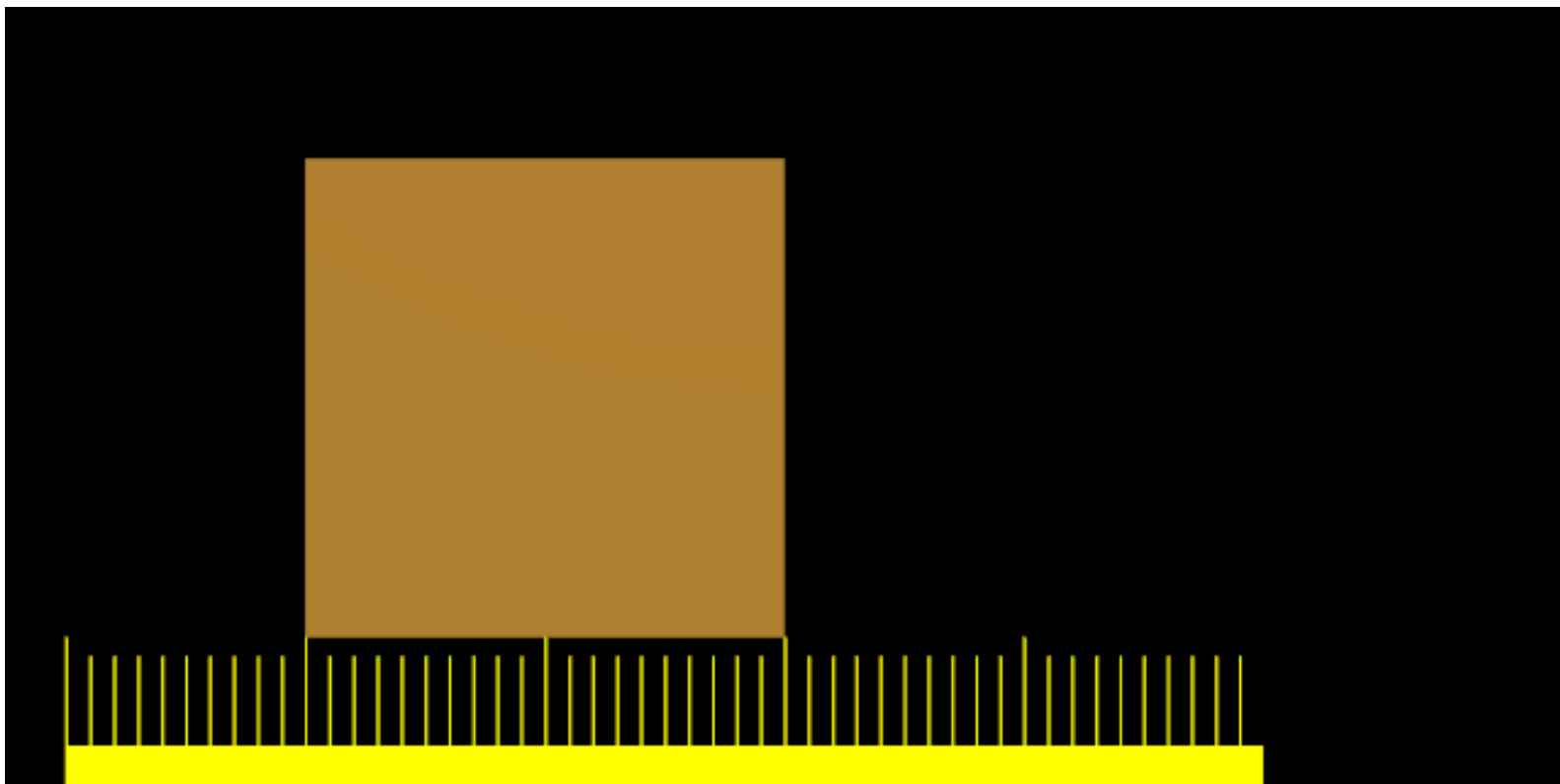
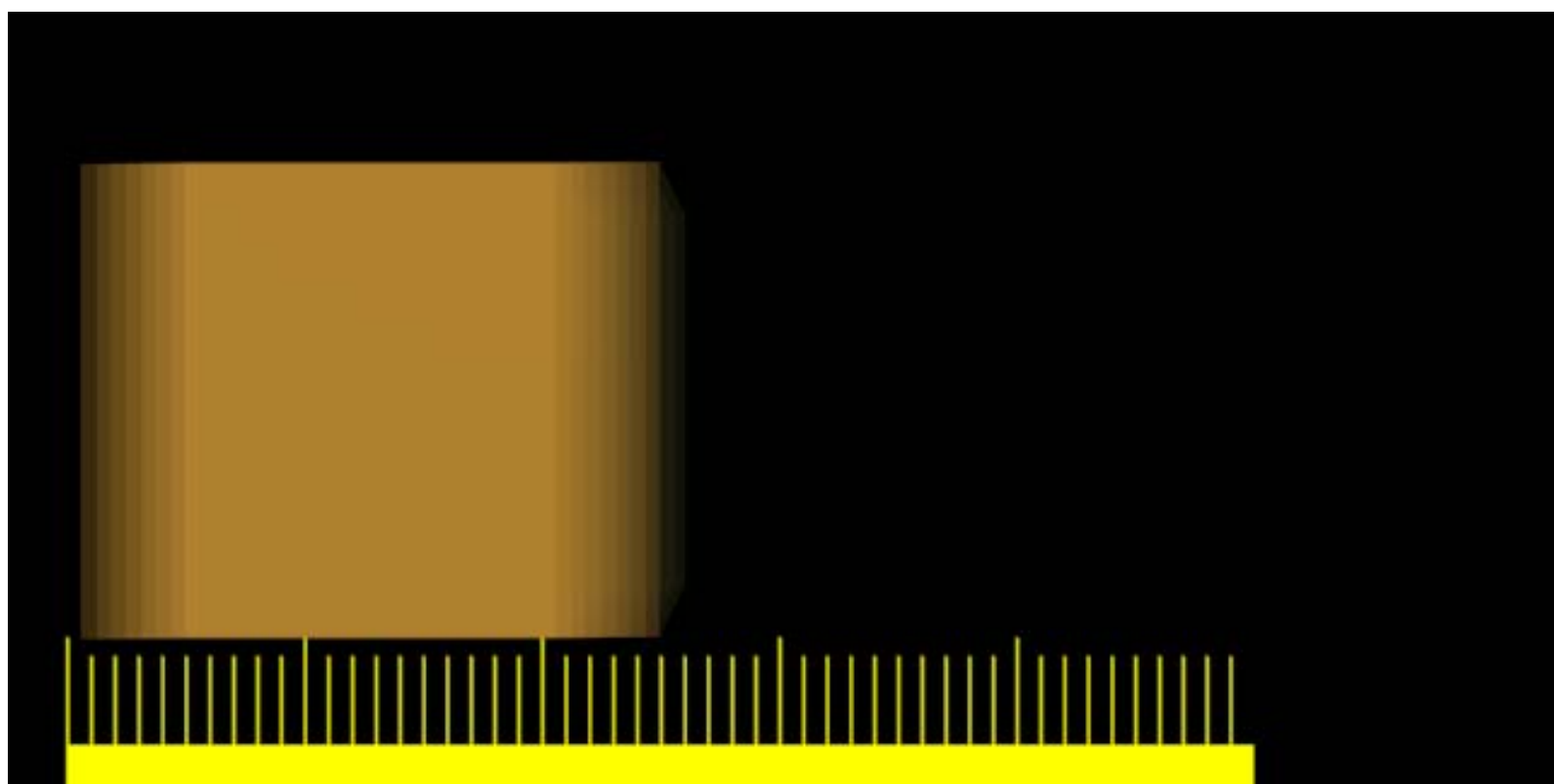


Figure 17-20. Cuadro 2 del cubo en movimiento, sin Desenfoque de Movimiento (motion blur).



[Figure 17-21](#) por otra parte, muestra el render del cuadro 1 cuando Desenfoque de Movimiento (Motion Blur) está activado y 8 cuadros "intermedios" son computados con un periodo entre cuadros de 0,5, empezando desde el cuadro 1. Esto es bastante evidente ya que todo el "desenfoque" del cubo ocurre media unidad antes y media unidad después del cuerpo principal del cubo.

Figure 17-21. Cuadro 1 del cubo en movimiento, con Desenfoque de Movimiento (motion blur), 8 muestras, $B_f=0,5$.



[Figure 17-22](#) y [Figure 17-23](#) muestran el efecto que produce aumentar los valores de B_f . Un valor más grande que 1 implica una obturación "lenta".

Figure 17-22. Cuadro 1 del cubo en movimiento con Desenfoque de Movimiento (motion blur), 8 muestras, $B_f=1,0$.

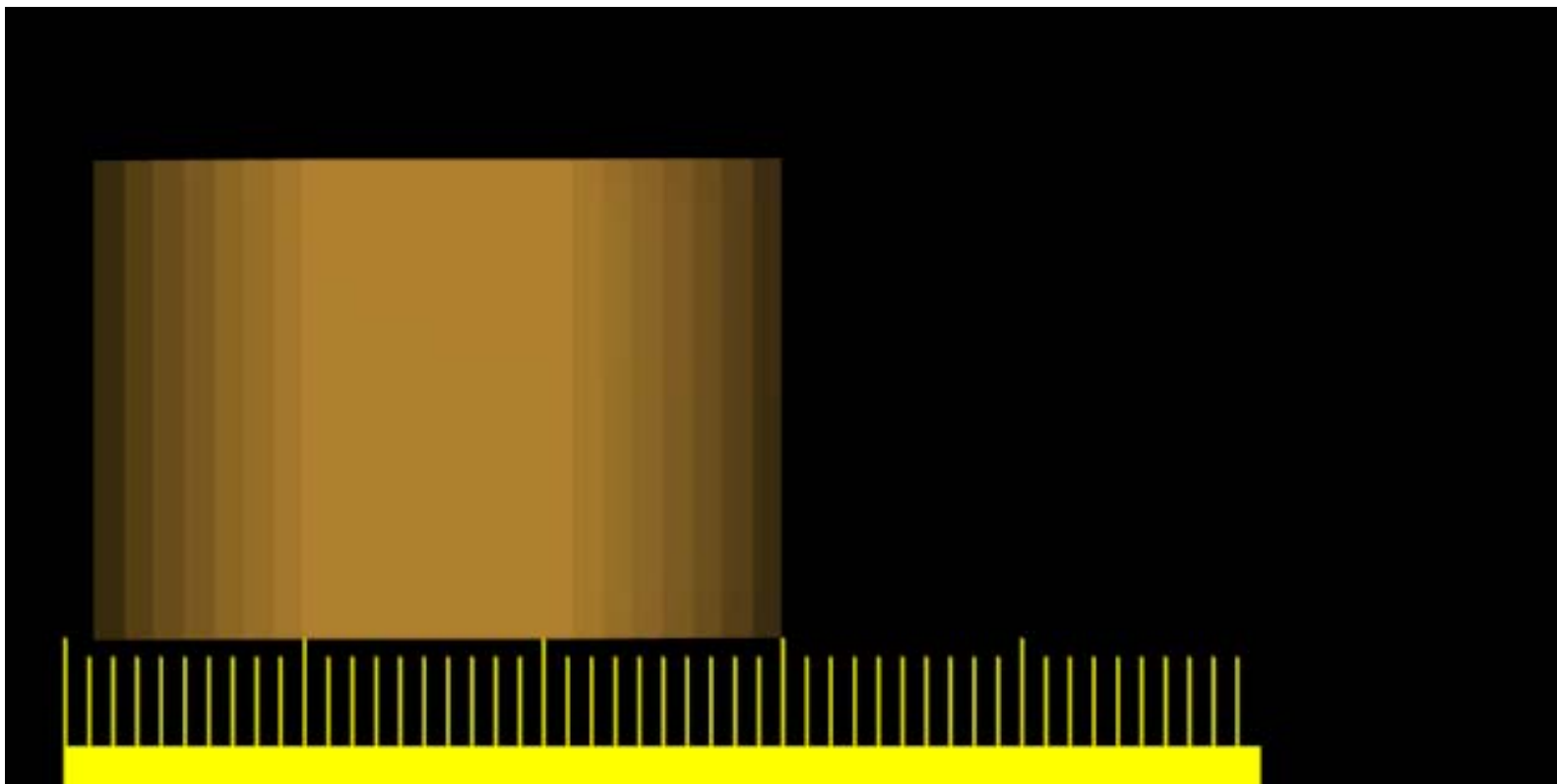
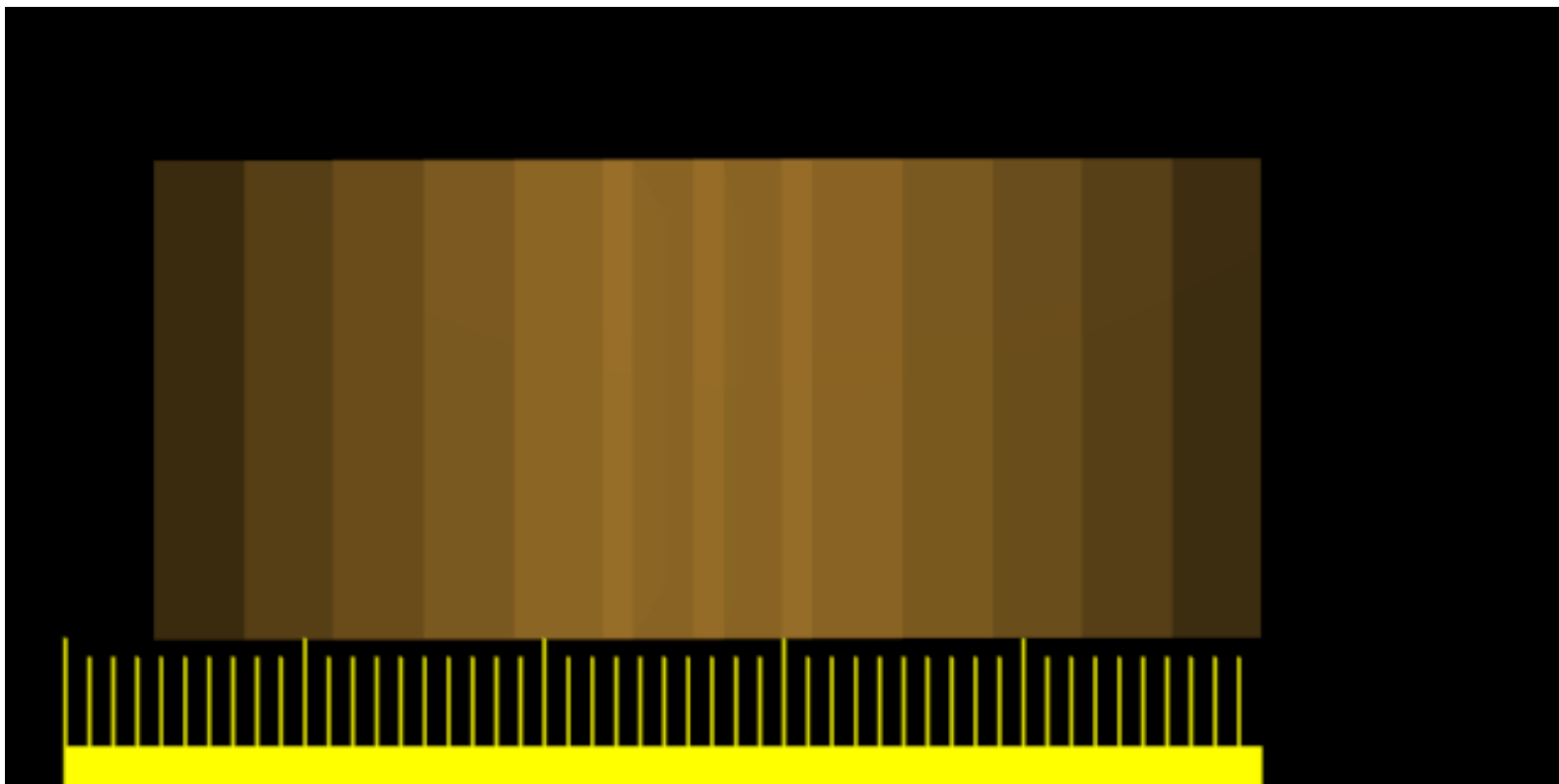


Figure 17-23. Cuadro 1 del cubo en movimiento, con Desenfoque de Movimiento (motion blur), 8 muestras, $B_f=3,0$.



Se pueden obtener mejores resultados que los mostrados anteriormente utilizando 11 o 16 muestras en lugar de 8, pero, por su puesto, ya que se necesitan tantos renders *separados* como muestras, un render con Desenfoque de Movimiento (Motion Blur) tarda bastante más tiempo que uno sin él.



Mejor Anti-Aliasing

Si Desenfocado de Movimiento (Motion Blur) está activo, aún si nada se mueve en la escena, Blender igualmente "agita" la cámara un poco entre un cuadro y el siguiente. Esto implica que, aun si OSA está desactivado, la imagen resultante tiene un buen Anti-Aliasing. Un Anti-Aliasing obtenido a través de Desenfocado de Movimiento (Motion Blur) es comparable a un Anti-Aliasing OSA del mismo nivel, pero generalmente menor.

Esto es interesante, ya que para escenas muy complejas donde un nivel de OSA de 16 no da resultados satisfactorios, se pueden obtener mejores resultados utilizando OSA y MBLUR. De esta forma se tienen tantas muestras por cuadro como cuadros "intermedios", dando de forma efectiva oversampling a niveles 25, 64, 121 y 256, si se eligen muestras de 5, 8, 11 y 16 respectivamente.

[Prev](#)

Renderizando animaciones

[Home](#)

[Up](#)

[Next](#)

Profundidad de Campo

17.7. Profundidad de Campo

Relativo a Blender v2.31

La Profundidad de Campo (Depth of Field) o DoF es un efecto interesante en Fotografía que realza mucho las imagenes generadas por computadora. También se conoce este efecto con el nombre de Borroneado Focal.

El fenómeno está vinculado al hecho de que una cámara fotográfica real puede enfocar un objeto a una distancia determinada, de manera que los objetos más cercanos y los muy lejanos a la cámara quedarán fuera del plano focal, por lo que se los verá ligeramente borrosos en la fotografía resultante.

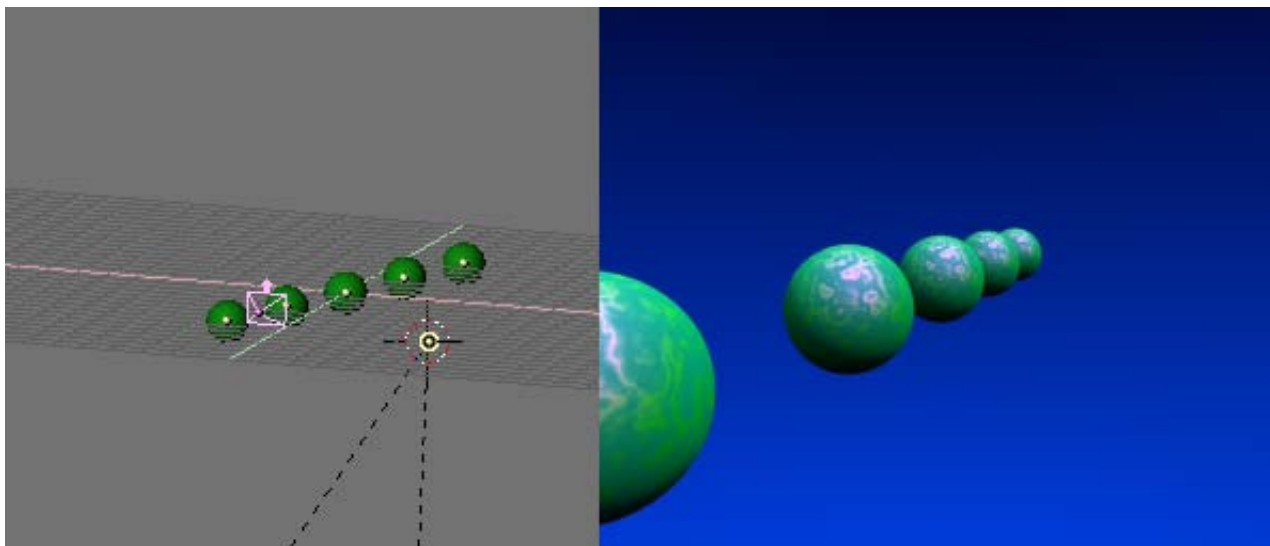
El nivel de borroneado de los objetos más cercanos y los más lejanos varía muchísimo con las dimensiones de la longitud y la apertura focal de la lente, y si se la usa con destreza, puede dar efectos muy agradables.

El motor de render de Blender no ofrece un mecanismo automático para obtener el efecto DoF, pero existen dos caminos alternativos para lograrlo. Uno se basa solamente en los recursos internos de Blender y será descrito en estas lineas. El otro requiere un plugin de secuencia externo y será someramente descrito en el Capítulo Editor de Secuencias.

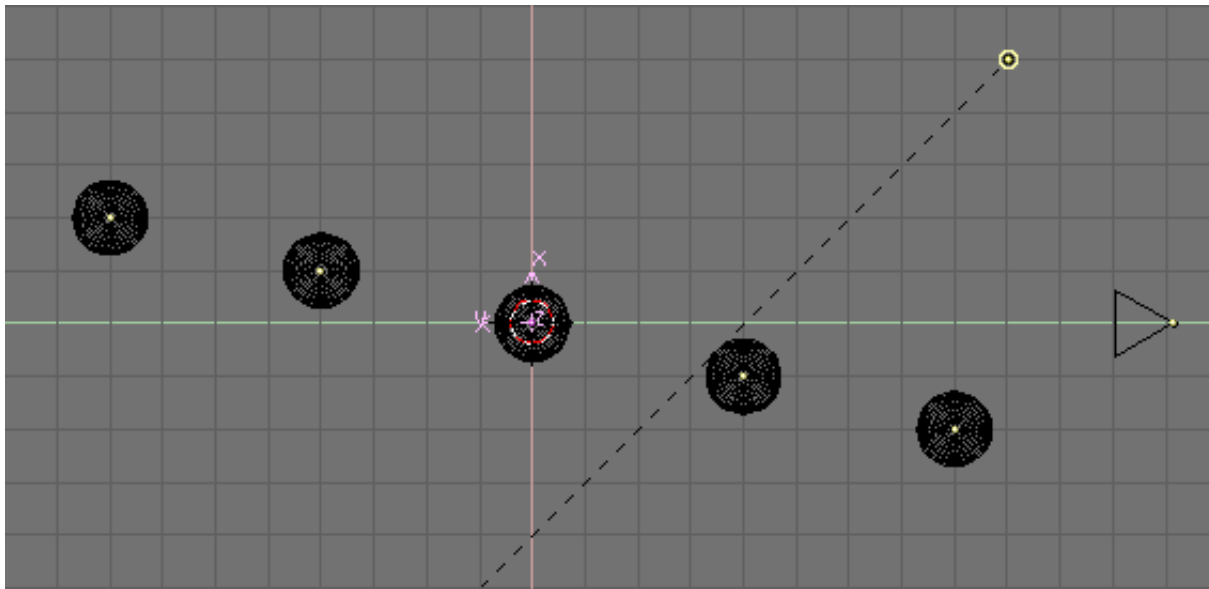
El truco para obtener el efecto DoF en Blender consiste en el uso hábil del efecto de Motion Blur o Borroneado de Movimiento descrito anteriormente, haciendo que la cámara se mueva circularmente alrededor de los que sería la apertura de la lente en una "cámara en el mundo real", apuntando constantemente hacia un punto donde se desea tener el foco "perfecto".

Supongamos que tiene una escena con esferas alineadas como se ve en la parte izquierda de la [Figure 17-24](#). Un render estándar de Blender dará como resultado la imagen en la parte derecha de la [Figure 17-24](#), con todas las esferas perfectamente nítidas y en foco.

Figure 17-24. Escena de prueba de Profundidad de Campo.



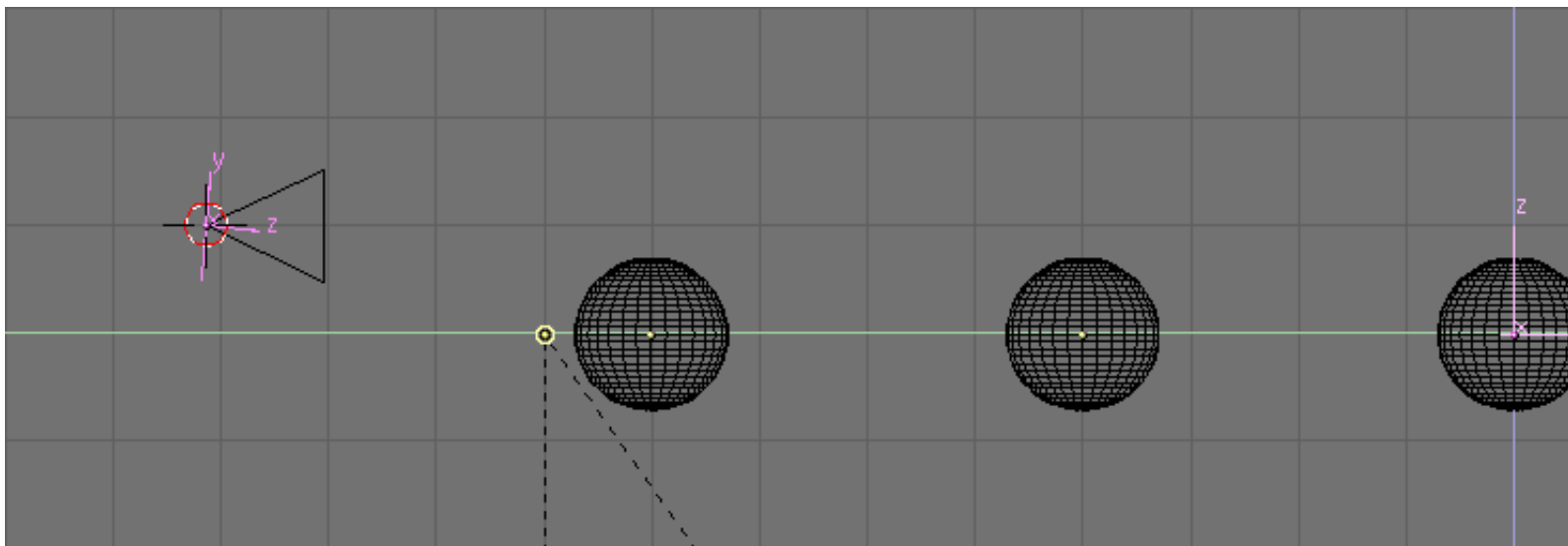
El primer paso es ubicar un objeto Empty (**SPACE**>>Add>>Empty) donde se encontrará el foco. En nuestro caso en el centro de la esfera del medio ([Figure 17-25](#)).

Figure 17-25. Poniendo el Empty Foco.

Luego, asumiendo que su cámara esté ya en la posición correcta, ponga el cursor sobre la cámara (seleccione la cámara, **SHIFT-S**>>Curs->Sel) y cree un círculo NURBS (**SPACE**>>ADD>>Curve>>NURBS Circle).

Salga de ModoEdición (**TAB**) y escale el círculo. Esto es muy arbitrario, y quizás desee re-escalarlo luego para lograr un resultado mejor. Básicamente, el tamaño del círculo está vinculado a la diámetro físico de la apertura o diafragma de su "cámara real". Mientras mayor sea el círculo, más estrecha será la región de foco perfecto, y más borronados se verán tanto los objetos cercanos como los lejanos. Mientras más chico sea el círculo, menos notorio será el efecto DoF.

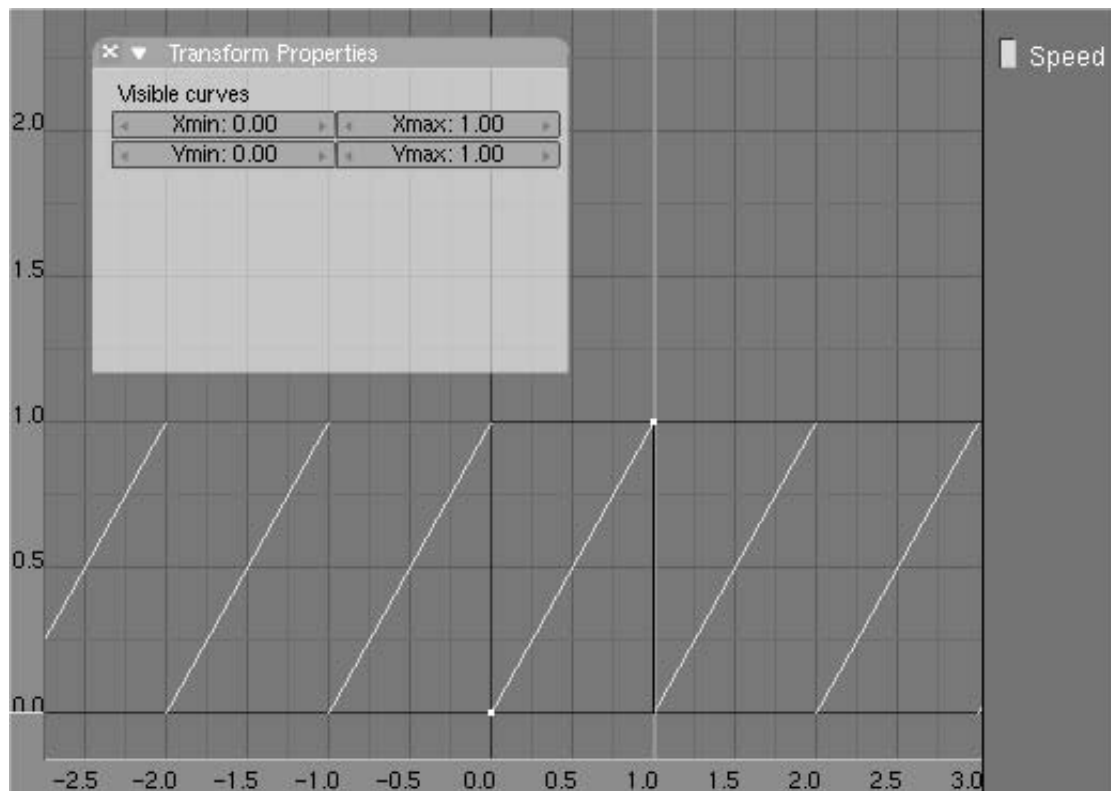
Ahora haga que el círculo siga al Empty con una restricción de seguimiento o bien con el viejo Seguimiento (Tracking) visto en [Figure 17-26](#). Como la normal al plano que contiene al círculo está sobre el eje local Z, deberá configurar el seguimietno correctamente de modo que el eje local Z del círculo apunte al Empty y el círculo sea ortogonal con la línea que conecta su centro con el Empty.

Figure 17-26. NURBS circle tracking the focus Empty.

Seleccione la Cámara y luego el círculo y emparente la Cámara al círculo (**CTRL+P**). El círculo será el Recorrido de la cámara, por lo que puede utilizar una relación de parentesco normal y luego activar el botón *CurvePath* o emplear una relación de parentesco *Follow Path (Seguir Recorrido)*.

Con el círculo aún seleccionado, abra una ventana IPO, seleccione el tipo de curva IPO Curve. La única IPO disponible es "Speed" (Velocidad). Con **CTRL+LMB** dos veces sobre un lugar al azar en la Ventana IPO agregue dos puntos cualquiera. Luego, usando **NKEY**, ponga numéricamente en estos dos puntos X_{min} e Y_{min} a 0, X_{max} e Y_{max} a 1. Para completar la edición de la IPO, hágala acíclica usando la opción del menú *Curve>>Extend Mode>>Cyclic*. El resultado final debería verse como se muestra en [Figure 17-27](#).

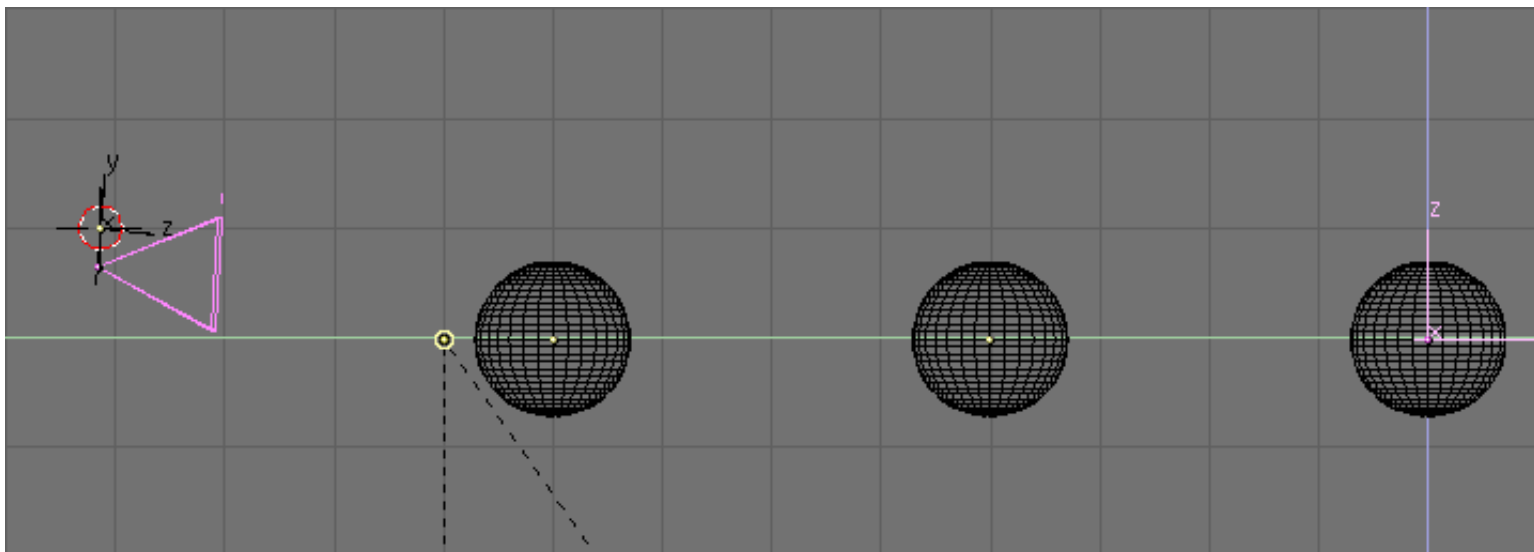
Figure 17-27. IPO Speed para el recorrido circular NURBS.



Con estos valores hemos hecho que efectivamente la cámara circule a lo largo del recorrido del círculo NURBS y alrededor de su posición original, en exactamente 1 cuadro. Esto hace que la opción Motion Blur, vaya tomando distintas vistas de la escena ligeramente diferentes y cree al final un efecto similar al de Borroneado Focal.

Falta realizar un ajuste más. Seleccione primero la cámara y luego el Empty Focal, y haga que la cámara siga al Empty de la manera que prefiera. La Cámara debería realizar un seguimiento del Empty como se ve en [Figure 17-28](#).

Figure 17-28. Cámara con seguimiento sobre el Empty Focal.



Si pulsa **ALT-A** no habrá ningún movimiento aparente ya que la cámara hace un giro en exactamente una vuelta completa al recorrido circular en un cuadro, por lo que parece estar quieta, sin embargo el motor de Motion BLur sí detectará estos movimientos.

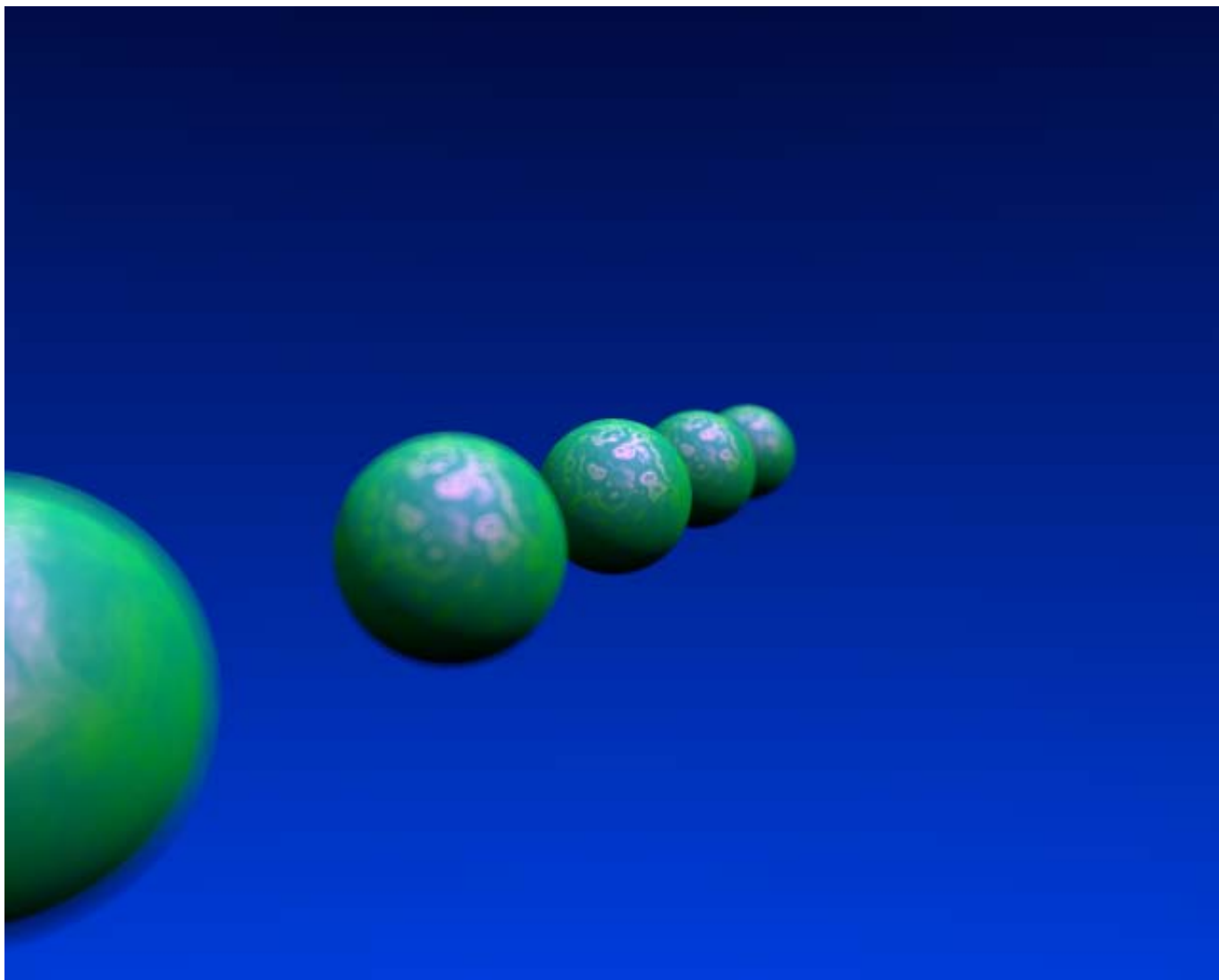
El último retoque es luego ir a la ventana de botones de Renderizado (**F10**) y pulsar sobre el botón *MBLUR*. Muy probablemente no necesite activar la opción *OSA* ya que el Motion Blur realizará algún antialiasing implícitamente. Se recomienda enérgicamente que ponga el factor de Motion BLur a 1, ya que de esta manera recorrerá un frame completo para borronear, tomando el perímetro completo del círculo. Para resultados óptimos es necesario también poner el número de sobremuestreos (oversamples) al nivel máximo (16) ([Figure 17-29](#)).

Figure 17-29. Valores de Motion blur.



Renderizando la escena (**F12**) obtendremos el resultado deseado. Esto puede tardar mucho más que un render sin DoF teniendo en cuenta que Blender renderizará 16 imágenes y luego las combinará en una sola. La [Figure 17-30](#) muestra el resultado para comparación con la [Figure 17-24](#). Se debe observar que el círculo ha sido escalado mucho menos para obtener esta imagen que lo que se ha mostrado en las capturas de pantalla de ejemplo. Estas últimas fueron hechas con un radio grande (igual a 0.5 unidades de Blender) para demostrar mejor la técnica. Por otro lado, la [Figure 17-30](#) tiene un círculo cuyo radio es de 0.06 unidades de Blender.

Figure 17-30. Render final con Motion blur.



Esta técnica es interesante y con ella es bastante sencillo obtener pequeños grados de Profundidad de Campo. Para grandes Nieblas Focales está limitada por el hecho de que no es posible tener más de 16 sobremuestreos.

[Prev](#)

Desenfoco de Movimiento (Motion Blur)

[Home](#)

[Up](#)

[Next](#)

Bordes de Caricatura

17.8. Bordes de Caricatura

Relativo a Blender v2.31

Los nuevos sombreadores de materiales de Blender de la versión 2.28, incluyen sombreadores especulares y difusos de tipo caricatura (toon).

Utilizando estos sombreadores, le puede dar a sus renderizados un aspecto similar al de album de historieta o Manga, afectando los tonos de los colores como quizás pueda ver en la [Figure 17-31](#).

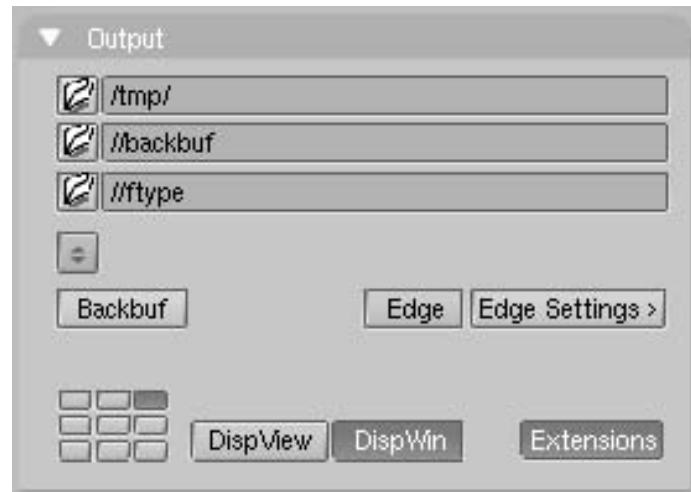
Figure 17-31. Una escena con materiales de Caricatura.



El efecto no es perfecto ya que las historietas y manga reales, también tienen normalmente contornos en tinta china. Blender puede añadir esta característica como una operación de post-procesado.

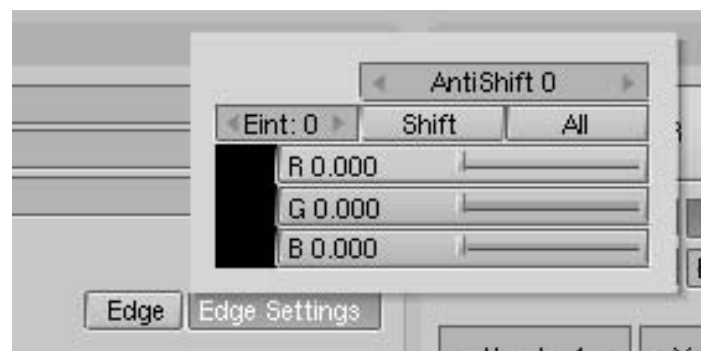
Para acceder a esta opción seleccione el botón *Edge* en el Panel *Output* de los Botones de Renderizado (**F10**) ([Figure 17-32](#)). Esto hace a Blender buscar los bordes en su renderizado y le añade una línea de contorno a los mismos.

Figure 17-32. Botones de borde de caricatura.



Antes de repetir el renderizado es necesario establecer algunos parametros. El botón *Edge Settings* abre una ventana para establecer estos valores ([Figure 17-33](#)).

Figure 17-33. Valores del borde de caricatura.



En esta ventana es posible indicar el color del contorno, que es negro por omisión, y su intensidad, *Eint* que es un número entero que va de 0 (más débil) a 255 (más intenso). Los otros botones son útiles si se usa el Renderizador Unificado (ver sección siguiente).

La [Figure 17-34](#) muestra la misma imagen que la [Figure 17-31](#) pero con los contornos de caricatura habilitados en color negro y con máxima intensidad (*Eint*=255).

Figure 17-34. Escena re-renderizada con bordes de caricatura activados.



[Prev](#)

Profundidad de Campo

[Home](#)

[Up](#)

[Next](#)

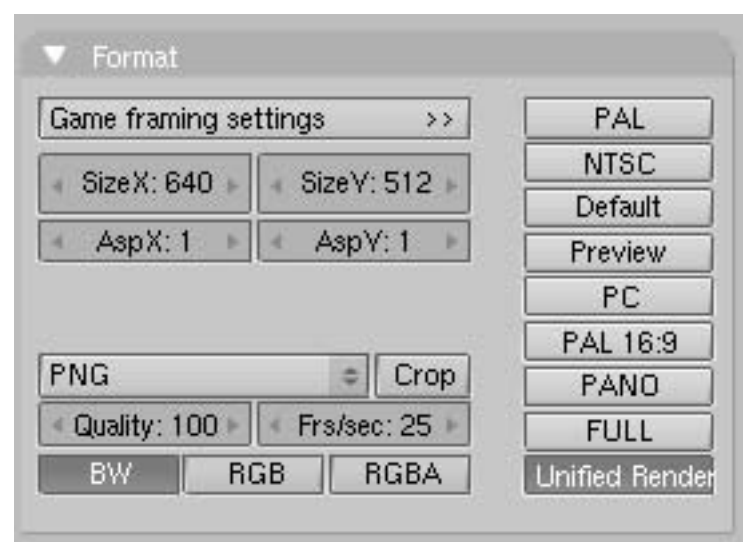
El Renderizador Unificado (Unified
Renderer)

17.9. El Renderizador Unificado (Unified Renderer)

Relativo a Blender v2.31

Una característica no muy conocida de Blender es el botón Renderizador Unificado (Unified Renderer) que se encuentra en la esquina derecha inferior del Panel *Format* de los Botones de Renderizado ([Figure 17-35](#)).

Figure 17-35. El botón Renderizador Unificado.



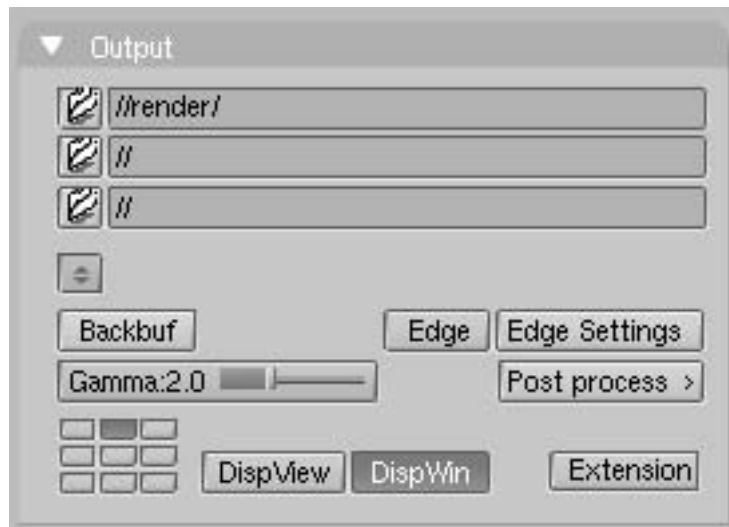
El motor de render de Blender *por omisión* está muy optimizado hacia la mejora en velocidad. Esto se ha conseguido subdividiendo el proceso de renderizado en varias "pasadas". Primero se procesan los materiales "normales", luego los materiales con transparencia (alfa); y, finalmente, se añaden los Halos y Flares.

Esto es rápido, pero puede llevar a resultados no óptimos, especialmente con Halos. El Renderizador Unificado, por el contrario, renderiza la imagen en una pasada única. Esto es más lento, pero da mejores resultados, particularmente en el caso de Halos.

Aún más, como los materiales transparentes se renderizan con los materiales convencionales, se le pueden aplicar también Bordes de Caricatura, pulsando el botón *All* en el diálogo de Edge Setting.

Si se selecciona el Renderizador Unificado, aparecerá un nuevo grupo de botones en el Panel *Output* ([Figure 17-36](#)).

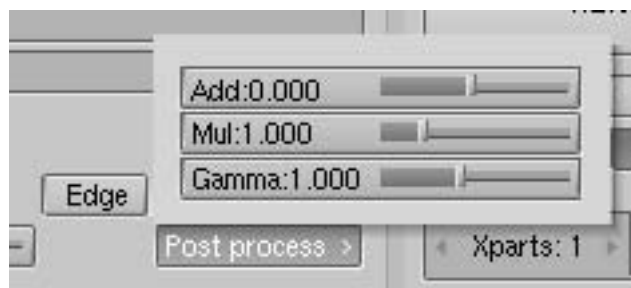
Figure 17-36. Botones adicionales del Renderizador Unificado.



El deslizador *Gamma* está relacionado con el procedimiento de OSA. Los sobremuestreos (oversampling) de píxeles se mezclan para generar el pixel renderizado final. El renderizador convencional tiene un $\text{Gamma}=1$, pero en el Renderizador Unificado se puede variar este valor.

Los botones de *Post process* (*Post-procesamiento*) hacen aparecer una caja de diálogo ([Figure 17-37](#)). Desde allí puede controlar tres tipos de postprocesamiento: el deslizador *Add* define una cantidad constante que será sumada al valor de color RGB de cada pixel. Los valores positivos hacen la imagen más brillante uniformemente, los valores negativos la hacen más oscura.

Figure 17-37. El submenú de postprocesamiento del Renderizador Unificado.



El deslizador *Mul* define un valor por el que se multiplican todos los valores RGB de todos los píxeles. Valores mayores que 1 hacen la imagen más brillante mientras que valores menores que 1 la hacen más oscura.

El deslizador *Gamma* realiza la corrección de contraste gamma estándar de cualquier programa de

dibujo.

[Prev](#)

Bordes de Caricatura

[Home](#)

[Up](#)

[Next](#)

Preparando su trabajo para vídeo

17.10. Preparando su trabajo para vídeo

Relativo a Blender v2.31

Una vez que ha dominado las técnicas de animación, seguramente comenzará a producir maravillosas animaciones codificadas con sus codecs favoritos, y probablemente los compartirá en Internet con el resto de la comunidad.

Pero tarde o temprano será tentado por el deseo de realizar una animación para Televisión o quizás para grabar sus propios DVDs.

Para ahorrarle alguna desilusión, aquí se presentan algunas pistas y secretos específicamente orientados a la preparación de vídeo. La primera y principal es recordar las dobles líneas blancas punteadas en la vista de cámara.

Si renderiza para PC, verá en su totalidad la imagen renderizada que es lo abarcado dentro del rectángulo punteado *exterior*. Para Televisión, algunas líneas y algunas partes de esas líneas se perderán debido a la naturaleza del barrido del haz de electrones en el tubo de rayos catódicos de su TV. Se garantiza que aquello que esté dentro del rectángulo punteado *interior* en la vista de cámara será visible en la pantalla. Todo aquello dentro del espacio entre los dos rectángulos, puede o no ser visible, dependiendo del equipo receptor de TV en el que esté viendo su vídeo.

Más aún, el tamaño del renderizado es dictado estrictamente por el estándar de TV. Blender ofrece tres configuraciones preestablecidas para su comodidad:

- *PAL* 720x576 píxeles con relación de aspecto de 54:51.
- *NTSC* 720x480 píxeles con relación de aspecto de 10:11.
- *PAL* 16:9 720x576 con relación de aspecto de 64:45 , para renderizado de TV en pantalla ancha de 16:9.

Recuerde lo relativo a "Relación de Aspecto". Las pantallas de TV *no* tienen píxeles cuadrados como tienen los monitores de computadora, sus píxeles son más bien rectangulares, por lo que es necesario generar imágenes *pre-distorsionadas* que se verán mal en una computadora, pero se verán perfectas en un equipo de TV.

17.10.1. Saturación de Color

La mayoría de las cintas de vídeo y señales de vídeo no se basan en modelos RGB sino en modelos YUV o YCrCb en el caso de Europa o YIQ en los Estados Unidos, siendo este último muy similar al primero. De ahí que también se necesite algún conocimiento al respecto de este tema.

El modelo YUV envía información como 'Luminancia', o intensidad (Y) y dos señales de 'Crominancia', rojo y azul. En realidad, un equipo de TV blanco y negro sólo muestra luminancia; en cambio, los televisores a color, reconstruyen el mismo a partir de crominancias, siendo:

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = Cr = R - Y$$

$$V = Cb = B - Y$$

Aunque una imagen estandar de 24 bits tiene 8 bits por canal, para mantener bajo el ancho de banda y, considerando que el ojo humano es más sensible a la luminancia que a la crominancia, la primera es enviada con más bits que las otras dos.

Esto lleva a un rango dinámico de colores menor, en Vídeo, que aquel al que está acostumbrado en los monitores. Debe, por lo tanto, tener en cuenta que no todos los colores se mostrarán correctamente. Una regla práctica es mantener los colores lo más "grisáceos" o "insaturados" que se pueda, esto se puede traducir básicamente en mantener el rango dinámico de sus colores por debajo de 0.8.

En otras palabras, la diferencia entre el valor de RGB más alto y el valor más bajo no debería sobrepasar 0.8(rango [0-1]) ó 200 (rango [0-255]).

No es una regla estricta, valores algo mayores que 0.8 son aceptables, pero un material con RGB=(1.0,0,0) se verá muy mal.

17.10.1.1. Renderizando por Campos

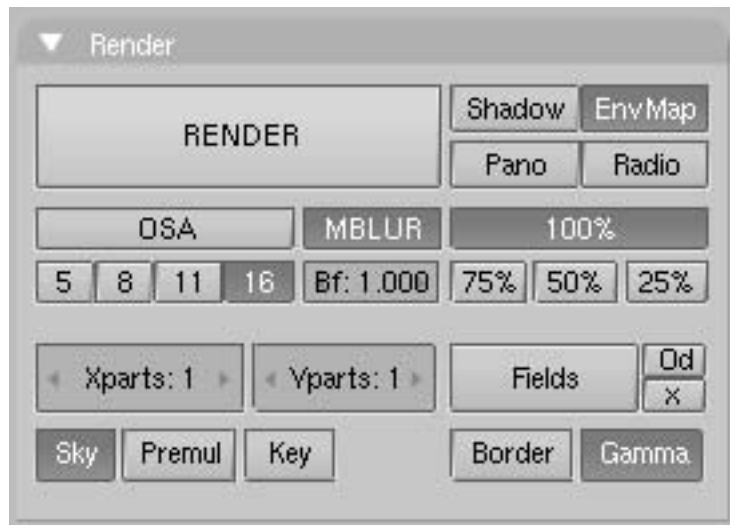
El estándar de TV indica que debería haber 25 cuadros por segundo (PAL) ó 30 cuadros por segundo (NTSC). Como el fósforo de la pantalla no mantiene la luminosidad por mucho tiempo, esto puede provocar un parpadeo apreciable. Para minimizar este efecto, los televisores no representan cuadros como lo hace una computadora, sino que representan semi-cuadros, o *campos (fields)* a una velocidad de refresco doble, osea 50 semi-cuadros por segundo en PAL y 60 semi-cuadros por segundo en NTSC. Originalmente esto estaba relacionado con la frecuencia de las líneas de tensión en Europa (50Hz) y en los Estados Unidos (60Hz).

Los campos están "entrelazados" en el sentido que un campo presenta todas las líneas pares de un cuadro

completo y el campo siguiente muestra todas las líneas impares.

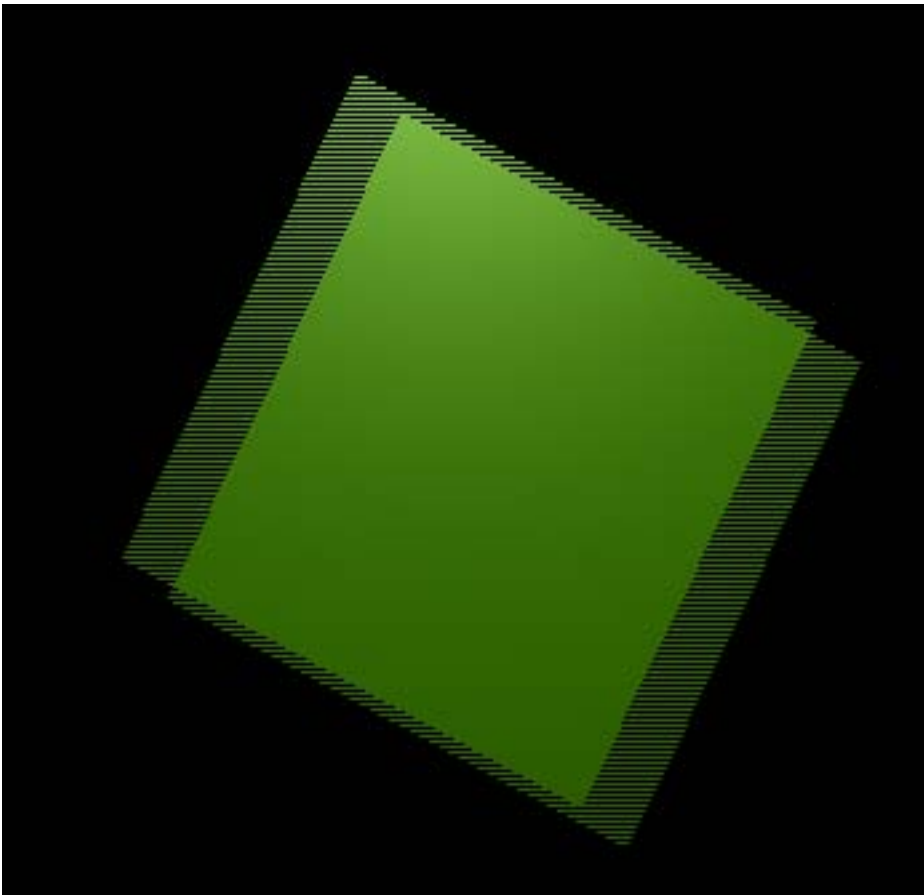
Como no hay una diferencia de tiempo despreciable entre cada campo (1/50 ó 1/60 avas de segundo), normalmente no sirve renderizar un cuadro y separarlo en dos semi-cuadros. Se hará presente un temblor de los bordes en los objetos móviles claramente apreciable.

Figure 17-38. Configuración de Render por campos.



Para manejar de forma óptima este tema, Blender ofrece renderizado por campos. Cuando se presiona el botón *Fields* en el Panel *Render*([Figure 17-38](#)), Blender genera cada cuadro en dos pasadas. En la primera pasada, renderiza sólomente las líneas pares, luego *avanza en el tiempo un medio paso* y renderiza todas las líneas impares.

Figure 17-39. Resultado de renderizar por campos.



Esto produce resultados extraños en una pantalla de PC ([Figure 17-39](#)), pero se verá perfectamente en un televisor.

Uno de los dos botones al lado del botón *Fields* fuerza el renderizado del campo Impar primero (*Odd*) y el otro deshabilita el paso de medio cuadro entre campos (*x*).

Tip Configurando el orden de campo correcto

Por omisión, Blender producirá los campos Pares antes que los Impares, de acuerdo a los estándares PAL europeos. En cambio, en NTSC los campos impares se barren primero.

Por supuesto, si se equivoca al elegir las cosas se verán peor que si no hubiera renderizado por campos.

[Prev](#)

El Renderizador Unificado
(Unified Renderer)

[Home](#)

[Up](#)

[Next](#)

Radiosidad

Chapter 18. Radiosidad

Muchos modelos de render, incluyendo el trazado de rayos, asumen un modelo espacial simplificado, altamente optimizado para la luz que entra en el "ojo" para dibujar la imagen. Puede añadir reflexiones y sombras a este modelo para conseguir un resultado más realista. Mientras, hay un aspecto importante que falta. Cuando una superficie tiene un componente de luz reflexiva, no solamente muestra nuestra imagen, también brilla con la luz de las superficies en su alrededor. Y viceversa. De hecho, la luz rebota por todo el entorno hasta que toda la energía de la luz sea absorbida (o haya escapado!).

La luz re-irradiada porta información sobre el objeto que la ha rebotado, normalmente color. Por lo tanto no solo las sombras son 'más oscuras' por la luz re-irradiada, sino que también tienden a mostrar el color de objeto más cercano y más iluminado. Un fenómeno comunmente llamado como 'escape de color'. ([Figure 18-1](#)).

Figure 18-1. Ejemplo de radiosidad



En entornos cerrados, la energía de la luz es generada por 'emisores' y es considerada como la reflexión o absorción de las superficies en el entorno. El parámetro el cual la energía queda en una superficie se llama la 'radiosidad' de la superficie. Al contrario de los métodos tradicionales de render, los métodos de radiosidad primero calculan todas las interacciones de la luz de un entorno de forma independiente de la

vista. Así, pueden ser renderizadas varias vistas en tiempo real.

En Blender, desde la versión 2.28, la Radiosidad es una herramienta tanto de modelado como de render. Esto quiere decir que puede activar Radiosidad sin un renderizado o usar la Radiosidad para pintar los colores de vértice y luces de vértice de sus mallas, para un uso posterior.

18.1. El método de Radiosidad en Blender

Pertinente a Blender v2.31

Primero, ¡un poco de teoría! Puede saltar a la siguiente sección si lo desea, y volver aquí si surgen dudas.

A finales de los ochenta y principios de los noventa, la Radiosidad era un tema candente en el mundo de los gráficos 3D por ordenador. Se desarrollaron métodos diversos; la más exitosa de esas soluciones estaba basada en el método del "refinamiento progresivo" con una esquema de "subdivisión adaptativa". Y esto es lo que Blender utiliza.

Para poder sacar el máximo partido del método de Radiosidad de Blender, es importante comprender los siguientes principios:

- *Método de Elemento Finito*

Muchos métodos de simulación o de gráficos por ordenador asumen una simplificación de la realidad con "elementos finitos". Para una solución visualmente atractiva (e incluso científicamente válida), no es siempre necesario zambullirse hasta un nivel de detalle molecular. En su lugar, puede reducirse el problema a un número finito de elementos representativos y bien descritos. Es un hecho común el que tales sistemas convergen rápidamente en una solución estable y fiable.

El método de Radiosidad es un ejemplo de método de elemento finito, puesto que cada cara está considerada un "elemento finito", y su emisión de luz está considerada como un todo.

- *Parches y Elementos*

En el universo de la Radiosidad, distinguimos entre dos tipos de caras 3D:

Parches. Son triángulos o cuadrángulos capaces de *emitir energía*. Para una solución rápida es importante tener cuantos menos de estos Parches como sea posible. Pero para acelerar las cosas, la energía es modelada como si la energía se emitiera desde el centro del Parche; el tamaño de los Parches debería, por lo tanto, ser lo suficientemente pequeño para que hubiera una distribución realista de energía. Por ejemplo, cuando un pequeño objeto está situado encima del centro del

Parche, toda la energía que el Parche envía es oscurecida por este objeto, ¡incluso si el Parche es grande! Este Parche debería ser subdividido en Parches más pequeños.

Elementos. Estos son los triángulos o cuadrángulos que *reciben energía*. Cada Elemento está asociado a un Parche. En realidad, los Parches están subdivididos en pequeños Elementos. Cuando un Elemento recibe energía, absorbe parte de la misma (dependiendo de su color) y pasa el remanente al Parche, para su posterior radiación. Puesto que los Elementos son también las caras que se muestran, es importante mantenerlas cuanto más pequeñas posible, para producir bordes de sombras y gradientes de luz sutiles.

- *Refinamiento Progresivo*

Este método empieza examinando todos los Parches disponibles. El Parche con más cantidad de energía "no disparada" es seleccionado para disparar toda su energía al entorno. Los Elementos del entorno reciben esta energía, y la añaden a la energía "no disparada" de los Parches que tienen asociados. Luego, el proceso se inicia una vez más para el Parche que tiene *ahora* más energía por disparar. Esto se sucede para todos los Parches hasta que no se recibe más energía, o hasta que la cantidad de energía "no disparada" converge por debajo de cierto valor.

- *El método del hemicubo*

El cálculo de cuánta energía da cada Parche a un Elemento se realiza mediante el uso de "hemicubos". Exactamente situado en el centro del Parche, un hemicubo (literalmente "medio cubo") consiste en 5 pequeñas imágenes del entorno. Para cada píxel en estas imágenes, cierto Elemento visible es codificado mediante color, y la cantidad de energía transmitida puede calcularse. Especialmente con el uso de hardware especializado, el método del hemicubo puede acelerarse significativamente. En Blender, sin embargo, los cálculos del hemicubo se realizan por software.

Este método es, de hecho, una simplificación y optimización de la fórmula "real" de la Radiosidad (diferenciación del factor forma). Por esta razón, la resolución del hemicubo (el número de píxeles de sus imágenes) es aproximado, y su cuidadosa configuración es importante para prevenir defectos de aliasing.

- *Subdivisión adaptativa*

Puesto que el tamaño de los parches y los elementos en una malla define la calidad de la solución de la Radiosidad, se han desarrollado esquemas de subdivisión automática para definir el tamaño óptimo de Parches y Elementos. Blender posee dos métodos de subdivisión automática:

1. Parches Subdivisión-disparo. Disparando energía al entorno, y comparando los valores del hemicubo con el "factor forma" matemático actual, pueden detectarse errores que indican la necesidad de una nueva subdivisión del Parche. El resultado son Parches menores y un mayor tiempo de resolución, pero un realismo de la solución más alto.

2. *Elementos Subdivisión-disparo*. Disparando energía al entorno, y detectando cambios altos de energía (gradientes) dentro de un Parche, los Elementos de este Parche son subdivididos un nivel extra. El resultado son Elementos menores y un mayor tiempo de resolución, y quizá más aliasing, pero un nivel de detalle más alto.

- *Visualización y Post-procesado*

La subdivisión de Elementos en Blender está "equilibrada"; ello significa que cada elemento difiere un máximo de 1 nivel de subdivisión respecto a sus vecinos. Esto es importante para una visualización agradable y correcta de la solución de Radiosidad con caras sombreadas mediante Gouraud. Normalmente después de la resolución, la solución consiste en miles de pequeños Elementos. Filtrándolos y eliminando "dobles", el número de Elementos puede reducirse significativamente sin destruir la calidad de la solución de Radiosidad. Blender guarda los valores de energía en números de punto flotante. Ello hace posible la configuración de situaciones de iluminación dramática, cambiando los valores estándar de multiplicación y gamma.

- *Radiosidad para modelado*

El último paso puede ser sustituir las mallas de entrada con la solución de Radiosidad (botón *Replace Meshes* - Sustituir Mallas). En ese momento los colores de los vértices se convierten de un valor de punto flotante a un valor RGB de 24 bits. Los Objetos de Malla antiguos son eliminados y reemplazados por uno o más Objetos de Malla nuevos. Se pueden eliminar los datos de Radiosidad con *Free Data* (Liberar Datos). Los nuevos objetos reciben un material por defecto que permite un renderizado inmediato. Hay dos parámetros importantes en un Material para trabajar con colores de vértice:

VColPaint. Esta opción trata los colores de los vértices como un reemplazo para el valor RGB normal en el Material. Deben añadirse Lamps para poder ver los colores de Radiosidad. De hecho, se puede usar la iluminación y sombreado de Blender como de costumbre, y conservar todavía una apariencia neta de Radiosidad en el render.

VColLight. Los colores de los vértices se añaden a la luz al renderizar. Puede verse el resultado incluso sin Lamps. Con esta opción, los colores de los vértices son pre-multiplicados por el color RGB del Material. Esto permite un ajuste fino de la cantidad de "luz de Radiosidad" en el render final.

Como todo en Blender, la configuración de Radiosidad se almacena en un bloque de datos. Queda adjuntado a la escena, y cada escena de Blender puede tener un "bloque" de Radiosidad diferente. Puede usarse esta característica para dividir entornos complejos en escenas con solucionadores de Radiosidad independientes.

18.2. Renderizando la Radiosidad

Pertinente a Blender v2.31

Asumamos que tenemos una escena lista, y que queremos renderizarla con Renderizado de Radiosidad. Lo primero que debe comprenderse al usar Radiosidad es que *no es necesaria ninguna Lamp*, aunque se requieren algunas mallas con la propiedad de material Emit mayor que cero, pues serán las fuentes luminosas.

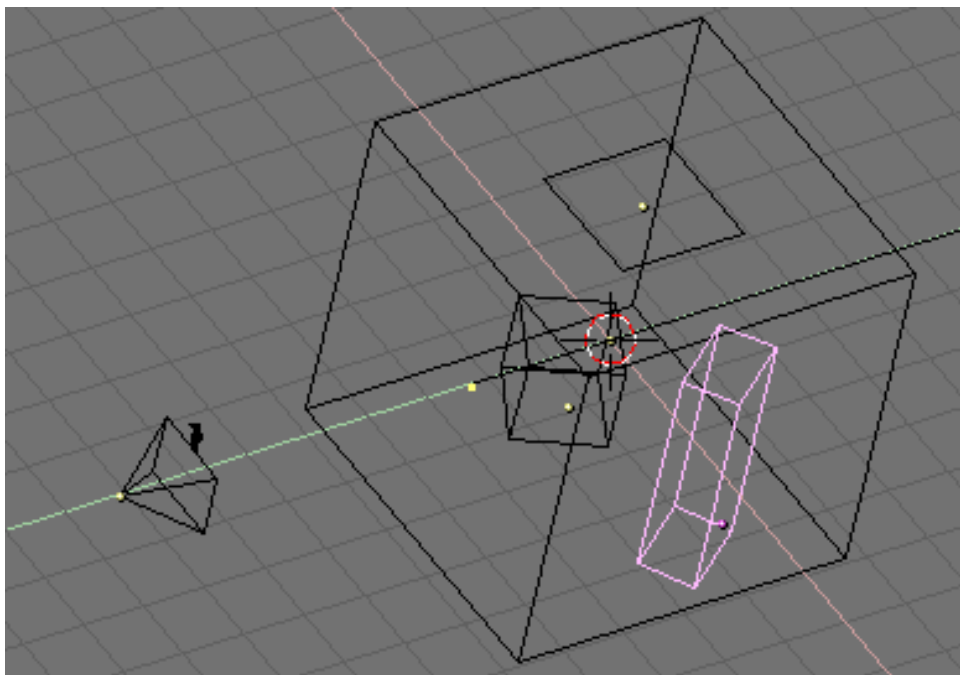
Puede construirse la escena de prueba mostrada en [Figure 18-1](#); es bastante sencillo. Simplemente debe crearse un cubo grande para la habitación, darle materiales diferentes a las paredes laterales, añadir un cubo y un cubo estirado en su interior, y añadir un plano con un valor Emit diferente de cero junto al techo para simular la luz de área ([Figure 18-2](#)).

Se asignan materiales como de costumbre a los modelos de entrada. El valor RGB del Material define el color del Parche. El valor 'Emit' de un Material define si un Parche es cargado con energía al principio de la simulación de Radiosidad. El valor 'Emit' es multiplicado por el área de un Parche para calcular la cantidad inicial de energía no disparada.

Tip**Caras emisoras:**

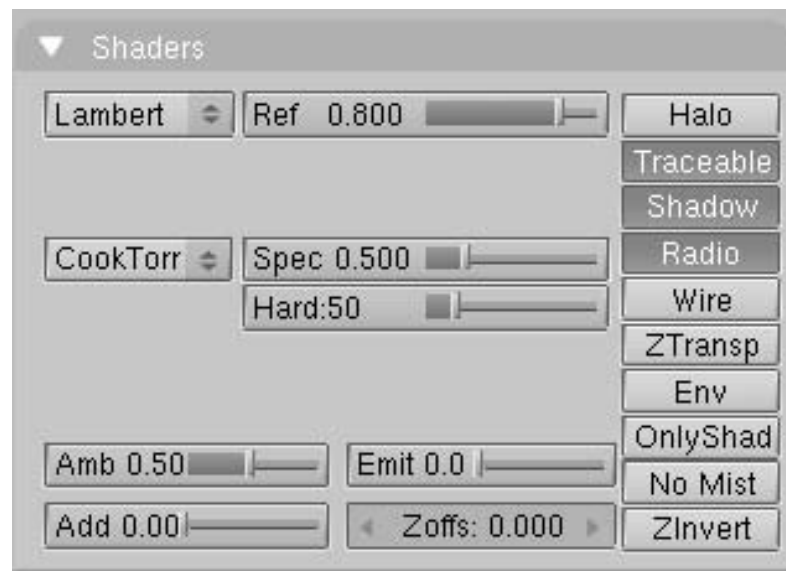
¡Compruebe el número de 'emisores' en la consola de Blender! Si es cero, no puede suceder nada interesante. Es necesario como mínimo un Parche emisor para obtener luz, y en consecuencia, una solución.

Figure 18-2. Decorado para la prueba de Radiosidad.



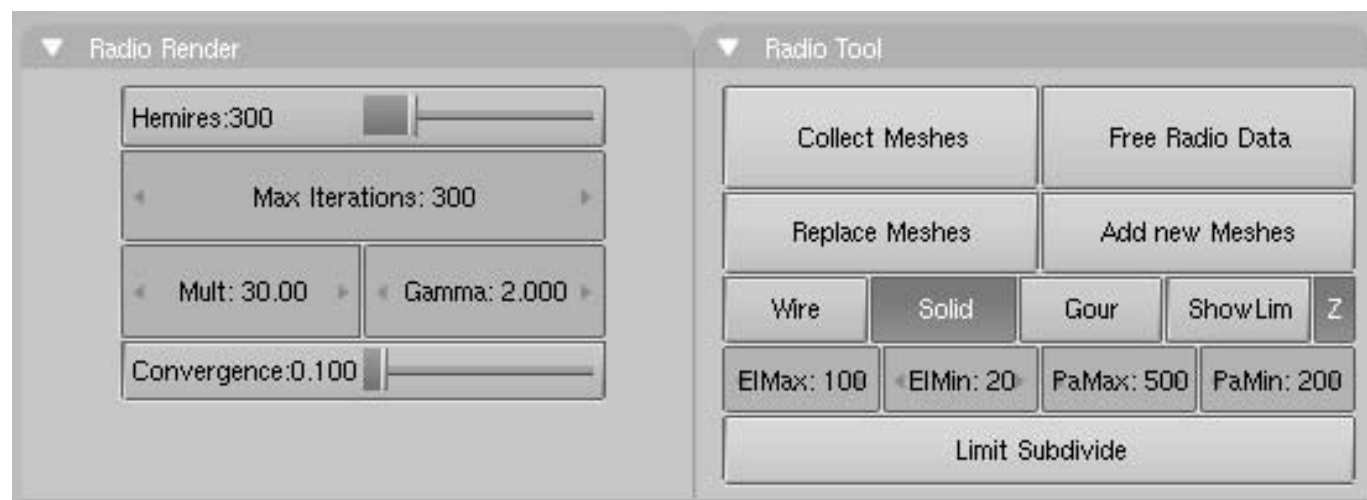
Al asignar materiales debemos asegurarnos de que todos tienen el conmutador *Radio* activado, para habilitar el Panel *Shaders* de los botones del subcontexto Material ([Figure 18-3](#)).

Figure 18-3. Material con la Radiosidad habilitada.



Nótese que la emisión de luz está gobernada por la dirección de las normales de una malla, por lo que el plano emisor de luz debería tener una normal apuntando *hacia abajo*, y el cubo *exterior* (la habitación) debería tener las normales apuntando hacia adentro (¡inviértalas!).

Seleccione el subcontexto Radiosidad  del Contexto Shading. Los Paneles, mostrados en [Figure 18-4](#), son dos: *Radio Rendering* (renderizado de radiosidad) el cual gobierna la Radiosidad cuando es usado como herramienta de renderizado (caso presente), y *Radio Tool* (herramienta de radiosidad), el cual gobierna la Radiosidad como una herramienta de modelado (próxima sección).

Figure 18-4. Botones Radiosidad para renderizado de radiosidad.

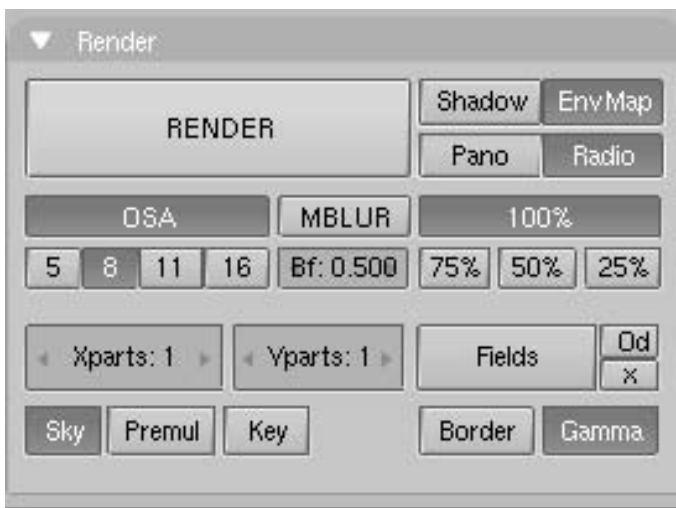
Los botones definen:

- *Hemires*: - La resolución del hemicubo, es decir, de las imágenes codificadas con color, utilizadas para encontrar los Elementos que son visibles desde un "Parche de disparo", y así recibir energía. Los Hemicubos no son almacenados, sino recalculados cada vez para cada Parche que dispara energía. El valor 'Hemires' determina la calidad de la Radiosidad e influye significativamente en el tiempo de resolución.
- *Max Iterations*: - El número máximo de iteraciones de Radiosidad. Si se pone a cero, la Radiosidad seguirá hasta que se cumpla el criterio de convergencia. Se recomienda fuertemente asignarle un número diferente de cero, normalmente mayor que 100.
- *Mult*., *Gamma*: - El espacio de color de la solución de Radiosidad, es muchísimo más detallado de lo que puede expresarse con simples valores RGB de 24 bits. Cuando los Elementos se convierten en caras, sus valores de energía son convertidos a un color RGB usando los valores *Mult* y *Gamma*. Con el valor *Mult* puede multiplicarse el valor de la energía; con *Gamma* puede ajustarse el contraste de los valores de la energía.
- *Convergence*: - Cuando la cantidad de energía no disparada en un entorno es inferior a este valor, la resolución de Radiosidad se detiene. La energía inicial no disparada en un entorno se multiplica con el área de los Parches. Durante cada iteración, parte de la energía es absorbida, o desaparece cuando el entorno no es un volumen cerrado. En el sistema de coordenadas estándar de Blender, un emisor típico (como en los archivos de ejemplo) tiene un área relativamente pequeña. Por esta razón, el valor de convergencia es dividido por un factor de 1000 antes de la prueba.

Ponga *Max Iterations* a 100 y seleccione el Contexto Scene y el Subcontexto Render (**F10**).

Localice el botón conmutador *Radio* ([Figure 18-5](#)) en el Panel *Render* y póngalo a 'on' para habilitar la Radiosidad, y luego ¡Render! (**F12**).

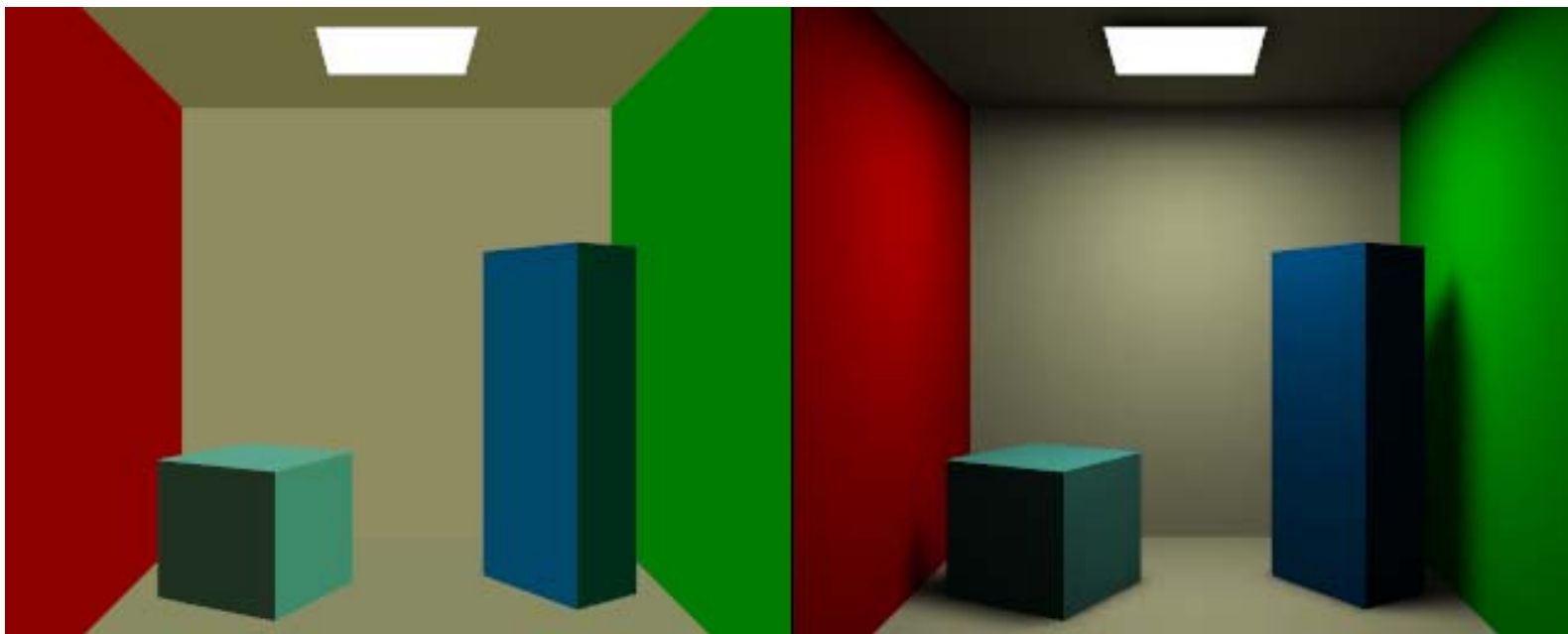
Figure 18-5. Habilitando la Radiosidad en los botones de Render.



El render tardará más que de costumbre; en la consola apreciará un contador subiendo. El resultado será bastante pobre ([Figure 18-6](#), izquierda) porque el render automático de radiosidad ;no realiza un refinado adaptativo!

Seleccione todas las mallas, una tras otra, y en EditMode subdivídalas al menos tres veces. La habitación, que es mucho mayor que las otras mallas, puede ser subdividida incluso cuatro veces. Aumente *Max Iterations* un poco, 300 o más. Intente renderizar una vez más (**F12**). Esta vez el renderizado tardará incluso más, pero los resultados serán mucho mejores, con sombras suaves y derrame de colores. ([Figure 18-6](#), derecha).

Figure 18-6. Render de Radiosidad para mallas rudas (izquierda) y mallas refinadas (derecha).



Note En el renderizado de Radiosidad, Blender actúa como en un renderizado normal; esto significa que las texturas, curvas, superficies e incluso Objetos Dupliframe son correctamente tratados.

18.3. Radiosidad como herramienta de modelado

Relativo a Blender v2.31

La radiosidad puede utilizarse como herramienta de modelado para definir Vértices de colores y luces. Esto puede resultar muy útil si lo que quieres es refinar tus modelos, o si pretendes utilizarlos en el Game Engine. Es más, el Modelado Radiosivo permite Refinamiento Adaptativo, lo que no permite el Renderizado Radiosivo.

Hay unos pocos puntos importantes que hay que comprender para el trabajo práctico con Modelado Radiosivo:

En Blender sólo las Mallas pueden actuar como entrada del Modelado Radiosivo. Esto es debido a que el proceso genera Vértices de colores... por tanto las entradas deben de ser Vértices. También es importante darse cuenta de que *cada* Cara de una Malla se convierte en un Parche, y por tanto en un emisor potencial de energía y reflector. Típicamente, los Parches largos envían y reciben más energía que los pequeños. Es importante, por tanto, tener un modelo de entrada equilibrado con Parches lo suficientemente grandes para crear una diferencia. Cuando se añaden Caras excesivamente pequeñas, casi nunca recibirán la suficiente energía para ser apreciadas por el método de "Refinamiento Progresivo", el cuál solo selecciona Parches con la suficiente cantidad de energía no disparada.

Tip Objetos No-Malla

El método anterior trabaja *sólo con Mallas*, por lo tanto, hay que convertir las Curvas y Superficies en Mallas (**CTRL+C**) antes de aplicarlo.

18.3.1. Fase 1: Collect Meshes

Todas las Mallas seleccionadas y visibles en la Escena actual se convierten en Parches apretando el botón *Collect Meshes* del panel *Radio Tool* ([Figure 18-4](#)). Como resultado un nuevo panel, *Calculation*, aparece. En este momento Blender ha entrado en el modo de modelado con radiosidad, y las otras funciones de edición han sido bloqueadas hasta que el botón recién creado *Free Data* sea presionado. El texto *Phase* sobre los botones ahora pone *Init* y muestra el número de Parches y Elementos.

Tras recopilar todas las Mallas, éstas son dibujadas en un pseudo modo de iluminación que difiere de manera clara del dibujo normal.

El panel *Radio Tool* ([Figure 18-7](#)) muestra tres botones: *Wire*, *Solid*, *Gour* que son tres opciones de modo de dibujo independientes del modo de dibujo indicado en una ventana 3D. El display Gouraud se muestra solamente después de que el Proceso Radiosivo ha comenzado. Presiona el botón *Gour* para obtener resultados más suaves en superficies curvas.

Figure 18-7. Botón Gourad



18.3.2. Fase 2: límites de Subdivisión

Blender permite configurar los tamaños máximo y mínimos de los Parches y Elementos en las *Radio Tools* y paneles *Calculation* ([Figure 18-8](#)).

Figure 18-8. Botones Radiosivos para Subdivisión

Limit Subdivide Los Parches son subdivididos respecto a los valores "PaMax" y "PaMin". Esta subdivisión es automáticamente realizada cuando ha comenzado una acción "GO".

PaMax, PaMin, ElMax, ElMin El tamaño máximo y mínimo de un Parche o Elemento. Estos límites son usados durante todas las fases de Radiosidad. La unidad se expresa en un 0.0001 del tamaño de marco (bounding) del entorno completo. Por lo tanto, con la configuración por defecto de 500 y 200 de tamaño máximo y mínimo del tamaño del Parche, 0.05 del modelo completo (1/20) y 0.02 del modelo completo (1/50)

ShowLim, Z Esta opción visualiza los límites de Parche y Elemento. Presionando la opción Z, los límites son dibujado y rotados de manera diferente. Las líneas blancas muestran los límites de los Parches, las líneas magenta muestran los límites de los Elementos.

18.3.3. Fase 3: Subdivisión adaptativa

Últimos ajustes antes de comenzar el análisis ([Figure 18-9](#)).

Figure 18-9. Botones de Radiosidad

MaxEl Máximo número de elementos permitidos. Puesto que los elementos son divididos automáticamente por Blender, la cantidad de memoria utilizada durante el tiempo de resolución puede ser controlada con este botón. Como regla general 20.000 elementos requieren 10Mb de memoria.

Max Subdiv Shoot El número máximo de parches de disparo que son evaluados para el proceso de "subdivisión adaptativa" (descrito arriba). Si es cero todos los parches con 'Emit' son evaluados.

Subdiv Shoot Patch El disparo de energía hacia el entorno permite la detección de errores que indican la necesidad de una mayor subdivisión de los parches. El resultado son parches menores y un tiempo de resolución mayor, pero a cambio se obtiene un mayor realismo de la solución. Esta opción puede ser también llevada a cabo de manera automática cuando la acción *GO* ha comenzado.

Subdiv Shoot Element Disparando energía al entorno, y detectando fuertes variaciones de energía (frecuencias) dentro de un parche, los elementos de este parche son seleccionados para ser divididos otro nivel extra. La subdivisión es llevada a cabo solamente cada vez que llares a esta función. El resultados son elementos más pequeños, un mayor tiempo de resolución, y probablemente más aliasing, pero a cambio se obtiene un mayor nivel de detalle. Esta opción puede ser también llevada a cabo de manera automática cuando la acción *GO* ha comenzado.

SubSh P El número de veces que el entorno es comprobado para detectar Parches que necesitan subdivisión.

SubSh E El número de veces que el entorno es comprobado para detectar elementos que necesitan subdivisión.

Note *Hemires*, *Convergence* y *Max iterations* en el panel *Radio Render* están todavía activos y tienen el mismo significado que en el Renderizado Radiosivo.

GO Con este botón comienza la Simulación Radiosiva. Las fases son :

1. *Limit Subdivide* Cuando los Parches son demasiado largos , éstos son divididos.
2. *Subdiv Shoot Patch*. El valor de *SubSh P* define el número de veces que la función *Subdiv Shoot Patch* es llamada. Como consecuencia los Parches son subdivididos.
3. *Subdiv Shoot Elem*. El valor de *SubSh E* define el número de veces que la función *Subdiv Shoot Element* es llamada. Como consecuencia los Elementos son subdivididos.
4. *Subdivide Elements*. Cuando los Elementos son todavía mayores que el tamaño mínimo son subdivididos. Ahora, la cantidad máxima de memoria está normalmente asignada.
5. *Solve*. Este es el método actual de 'refianamiento progresivo'. El puntero del ratón visualiza el paso de la iteración, el número actual de Parches que dispararon su energía en el entorno. Este proceso continúa hasta que la energía no disparada del entorno es menor que el valor *Convergence* o cuando el máximo número de iteraciones han sido alcanzada.
6. *Convert to faces* Los puntos son convertidos en triángulos o cuadrados con Aristas 'ancladas', para asegurar que sea posible mostrar un agradable Gouraud no-discontinuo.

Este proceso puede ser terminado con **ESC** durante cualquier fase.

18.3.4. Fase 4: Editando la solución

Una vez que la solución de radiosidad ha sido computada aún se pueden ejecutar una serie de acciones. ([Figure 18-10](#)).

Figure 18-10. Post proceso de radiosidad



Element Filter Esta opción filtra los elementos para eliminar el aliasing, para suavizar los contornos de las sombras, o para forzar colores ecualizados para la opción *RemoveDoubles*.

RemoveDoubles Cuando dos elementos próximos muestran un color que difiere menos que *Lim*, los elementos son juntados. El valor *Lim* usado anteriormente es expresado en un estándar de 8 bits de resolución; un rango de color de 0-255.

FaceFilter Los Elementos son convertidos en Caras para su visualización. *FaceFilter* fuerza un nivel extra de suavizado (smoothing) en el resultado mostrado, sin ningún cambio en los valores de los Elementos.

Mult, *Gamma (NumBut)* Tienen el mismo significado que en el Renderizado Radiosivo.

Add New Meshes Las Caras de la solución Radiosiva son convertidas a Objetos de Malla con Vértices de colores. Un nuevo material que permite el renderizado inmediato es añadido. *Las Mallas de entrada permanecen sin cambios.*

Replace Meshes (But) Como antes, pero las Malla de entrada son borradas.

Free Radio Data (But) Todos los Parches, Elementos y Caras son liberados de Memoria. Siempre se debe realizar esta acción después de usar Radiosidad para ser capaz de volver al modo normal de edición.

[Prev](#)

Renderizando la Radiosidad

[Home](#)

[Up](#)

[Next](#)

Un jugoso ejemplo de radiosidad

18.4. Un jugoso ejemplo de radiosidad

Relevante a Blender v2.31

Para alejarnos de una vez de la árida teoría y ver que es lo que el modelado de Radiosidad puede realmente conseguir, veamos un ejemplo.

Esto realmente le mostrará una verdadera escena de iluminación global, con resultados más suavizados que las 'luces focales de dupliverts'. Técnica mostrada en el capítulo de iluminación para conseguir algo como [Figure 18-11](#).

Figure 18-11. Cylon Raider renderizado con radiosidad.



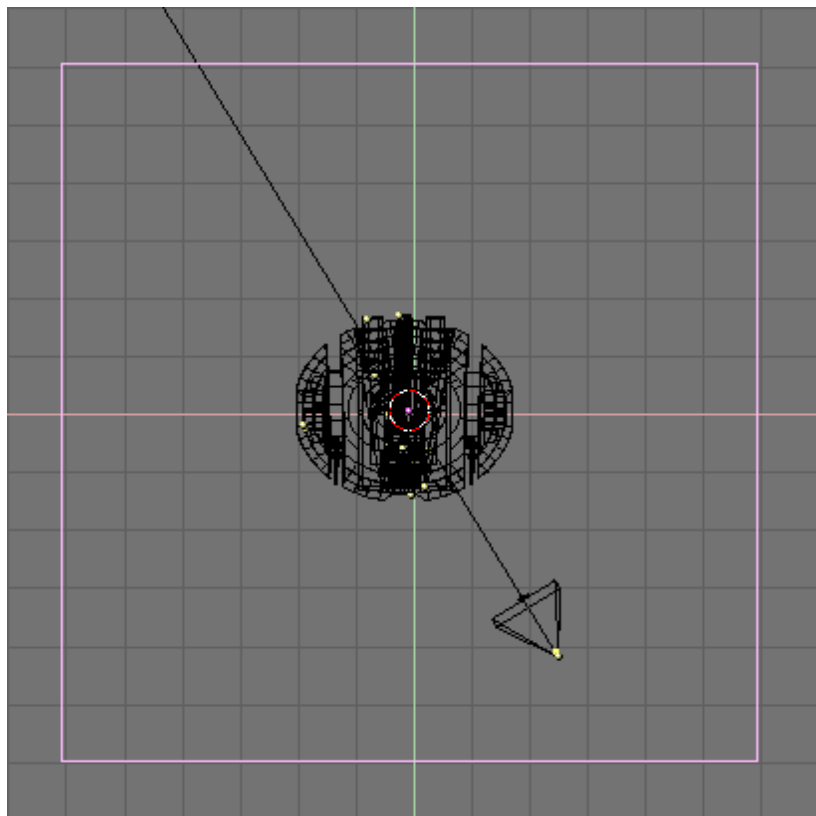
18.4.1. Setting up.

Tenemos únicamente dos elementos en la escena al comienzo: una nave de asalto (si recuerda alguna película de ciencia ficción...) y una cámara. La nave de asalto tiene el material gris por defecto, excepto las ventanas principales de la cabina que son negras. Para esta técnica no

necesitaremos ninguna lámpara.

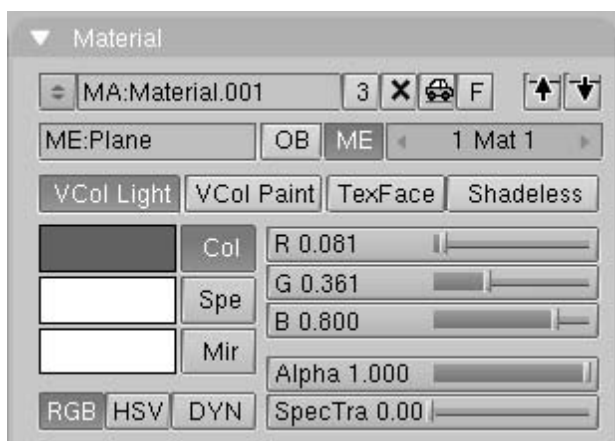
Lo primero que debemos hacer de añadir a la escena es un plano. Este plano será usado como el suelo en nuestra escena. Cambie el tamaño del plano como se ve en [Figure 18-12](#) y sitúelo justo debajo de la nave. Deje un poco de espacio entre el plano y el plano de la nave. Esto le dará un buen toque de "levitación".

Figure 18-12. Añadir un plano



Después, tendrá que dar al plano un material y seleccionar un color para él. Lo intentaremos con un bonito azul. Puede usar la propiedad que ve en [Figure 18-13](#) para ello.

Figure 18-13. Color del plano



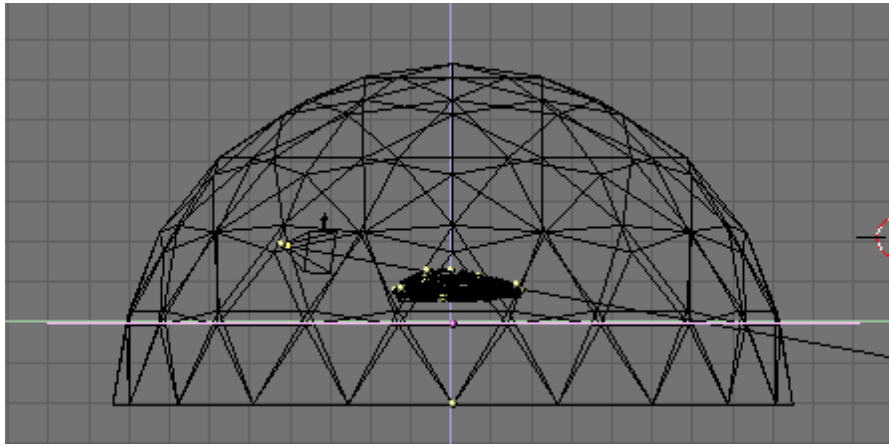
18.4.2. La cúpula del cielo

Queremos hacer un render GI, de modo que lo siguiente que tenemos que hacer es añadir una icoesfera. La esfera será nuestra fuente de luz en lugar de las lámparas típicas. Lo que vamos a hacer es usar sus caras como *emisores* que proyectarán luz para nosotros en multitud de

direcciones, en lugar de en una sola como una única lámpara típica. Esto nos dará el efecto deseado.

Para conseguirlo, añada una icoesfera con 3 subdivisiones. Mientras siga en el modo edición, use el modo selección **BKEY** para seleccionar la porción inferior de la esfera y eliminarla. Así nos quedaremos con nuestra cúpula. Cambie el tamaño de la cúpula para que encaje bien con la escena y hágala coincidir con el plano. Se debería parecer a [Figure 18-14](#).

Figure 18-14. Cúpula del cielo.



Lo siguiente, es asegurarnos de que tenemos seleccionados todos los vértices de la cúpula, para hacer click en Edición de Botones (**F9**) y seleccionar *Dibujar Normales* (*Draw Normals*). Esto nos permitirá ver en qué dirección están "emitiendo" los vértices. Por defecto será hacia afuera, así que presione el botón *Flip Normals*, que cambiará el emisor del vértice para proyectarlo desde fuera hacia dentro de la cúpula ([Figure 18-15](#)).

Figure 18-15. Cúpula del cielo.



Ahora que hemos creado la cúpula, necesitamos un nuevo material. Cuando cree el material para la cúpula, cambie las siguientes propiedades en los Botones de Material (**F5**):

$Add = 0.000$

$Ref = 1.000$

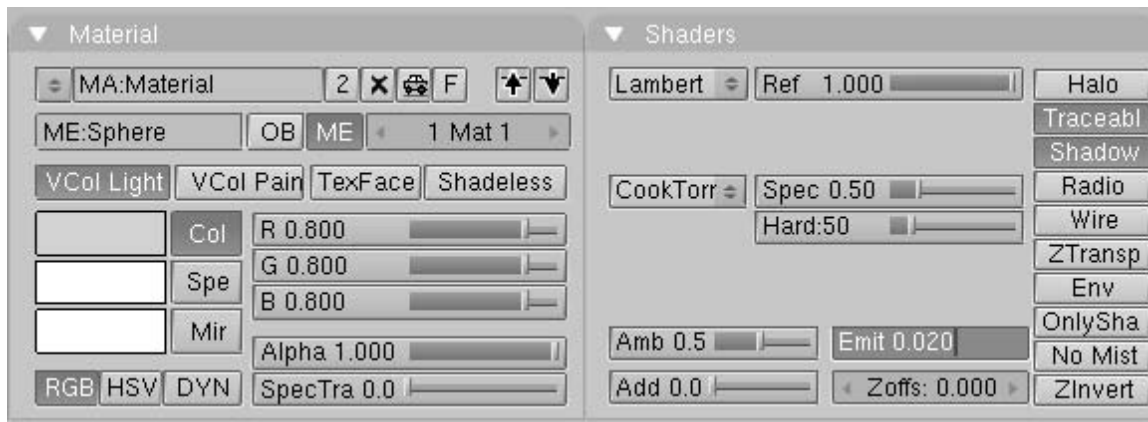
$Alpha = 1.000$

$Emit = 0.020$

El deslizador *Emit* es la clave aquí. Esta propiedad controla la cantidad de luz "emitida" desde nuestra cúpula. 0.020 está bien por defecto. ¡Recuerde que la cúpula es la parte más grande de la escena! ¡Tampoco queremos demasiada luz! Usted puede experimentar con esta

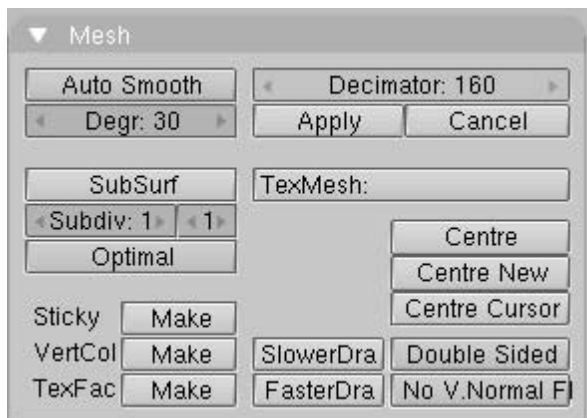
propiedad para conseguir diferentes resultados. The *Emit* slider here is the key. This setting controls the amount of light "emitted" from our dome. 0.020 is a good default. Remember that the dome is the bigger part of the scene! you don't want too much light! But you can experiment with this setting to get different results. Cuanto menor sea ésta propiedad, piense que durará más el tiempo de resolución después. ([Figure 18-16](#)).

Figure 18-16. Material de la cúpula.



En este momento hemos creado todo lo que necesitamos para nuestra escena. El siguiente paso que haremos tiene que ser alterar la cúpula y el plano de "doble cara" a "una cara". Para conseguir esto, seleccionaremos la malla de la cúpula y volveremos a los Botones de Edición (**F9**). Haga click en el boton *Double Sided* (*doble cara*) y desactívelo. ([Figure 18-17](#)). Repita este proceso para el plano.

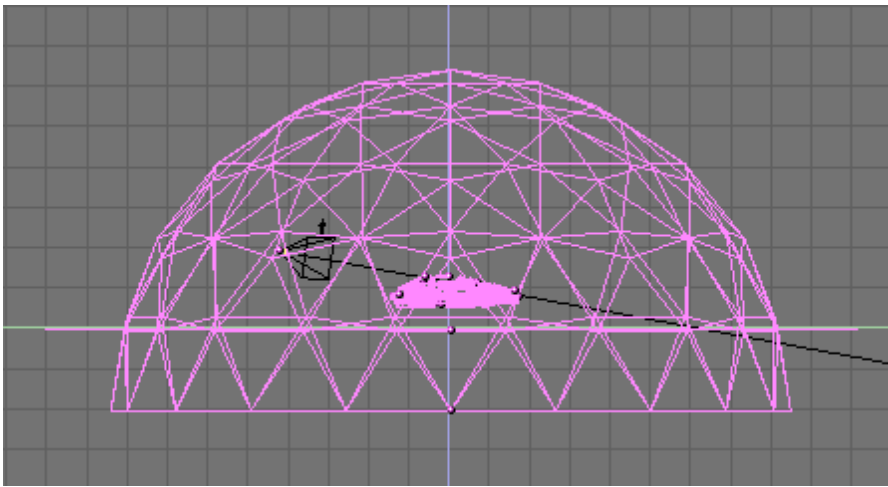
Figure 18-17. Configurando la cúpula y el plano como de 'una cara'.



18.4.3. La solución de la radiosidad

Los siguientes pasos son el corazon y alma de la iluminación global. Vaya a la vista lateral con **NUM 3** y use **AKEY** para seleccionar todas las mallas de nuestra escena. Ahora haga doble click en la cámara mientras mantiene **SHIFT** pulsado. No queremos que esté seleccionada. Debería parecerse a algo similar a [Figure 18-18](#).

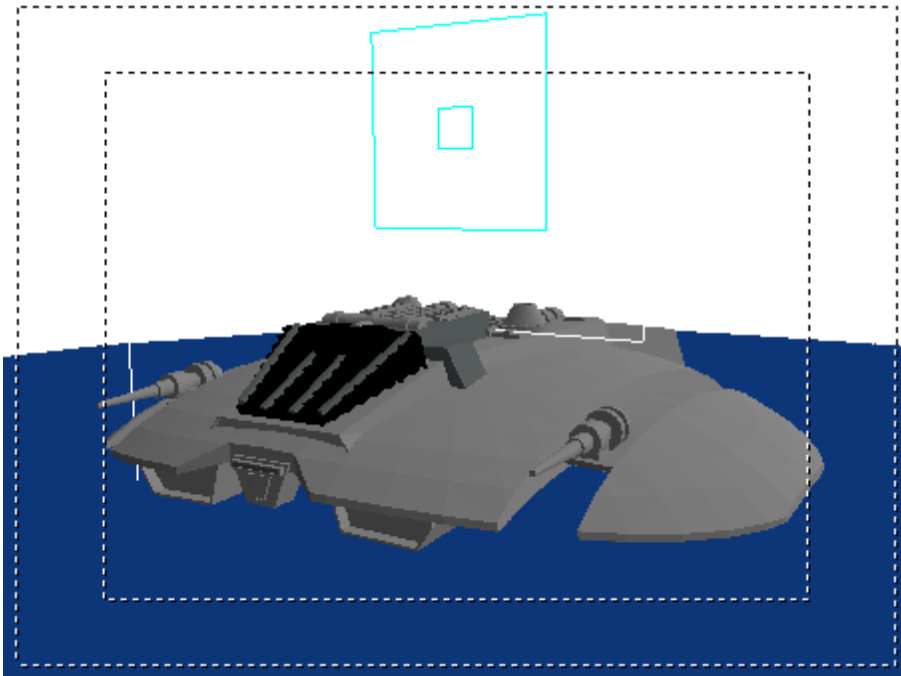
Figure 18-18. Seleccionando todas las mallas.



Después de seleccionar las mallas, vaya a la vista de cámara con **NUM 0** y active el modo sombreado con **ZKEY** ahora podemos ver dentro de la cúpula.

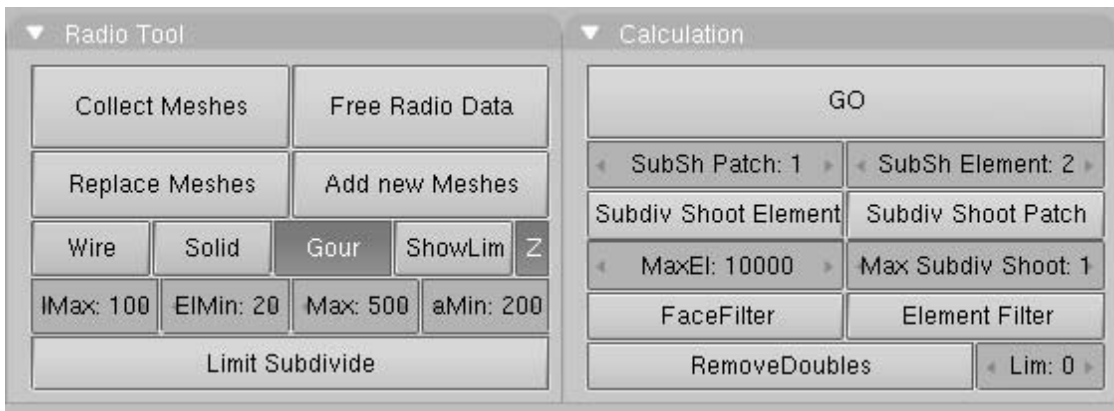
Seleccione el menu sombreado (**F5**) y el submenu de botones radiosity ). En el panel *Radio Tool*, haga click sobre el botón *Collect Meshes*. Debería darse cuenta de un cambio de colores en la vista. Debería ser algo similar a [Figure 18-19](#).

Figure 18-19. Preparando la solución de radiosity.



Ahora, para mantener el suavizado de la nave de asalto como la malla original, cambiaremos de *Solid* a *Gour*. Esto devolverá a nuestra nave sus bellas curvas de nuevo, sobre lo mismo *Set Smooth* debería estar en los botones de Edición. También necesitará cambiar el valor de *Max Subdiv Shoot* a 1 ([Figure 18-20](#)). ¡No olvide este paso!

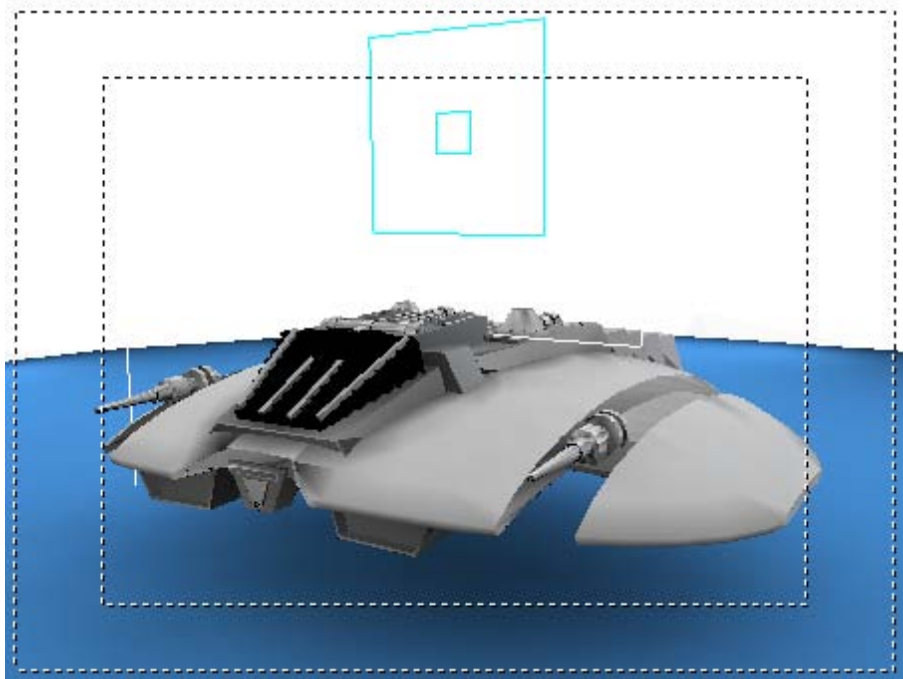
Figure 18-20. Propiedades de radiosity.



Después de que haya seleccionado *Gour* y *Max Subdiv Shoot*, haga click en *Go* y espere. Blender empezará a calcular la parte emisora de la cúpula, cara por cara, de este modo "soluciona" el render. Mientras lo hace, verá el cambio en la escena como mas y mas luz se va añadiendo a la escena y las mallas están cambiadas. También se dará cuenta de que el cursor en Blender cambia a un contador como si fuera una animación. Deje que Blender trabaje, resolviendo el problema de Radiosidad.

Dejando a Blender la cuenta en algún punto entre 50 y 500, dependiendo de la escenal, puede estar hecho en la mayoría de los casos. El tiempo de resolución depende de cuanto tiempo quiere permitirle trabajar... recuerde que puede presionar **ESC** en cualquier momento para parar el proceso. Éste es un área con el que se puede expmerintiar para obtener diferentes resultados. Puede tomar desde 5 hasta 10 minutos y la velocidad de su sistema también determinará en gran meidda cuanto tiempo tomará el proceso. [Figure 18-21](#) Es nuestra nave después de 100 iteraciones. is our Raider after 100 iterations.

Figure 18-21. Solución de radiosidad.

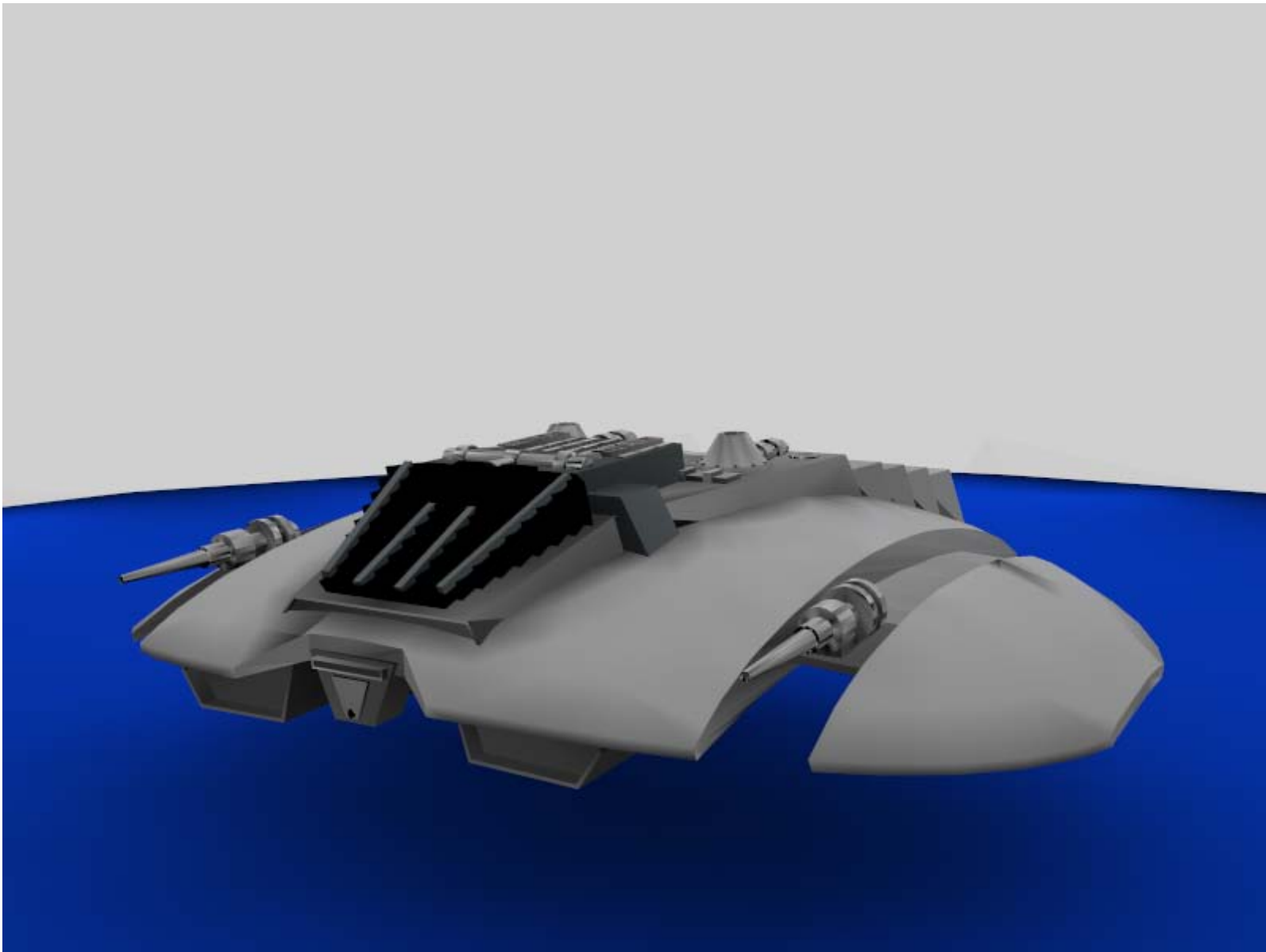


Después de presionar la tecla **ESC** y parar la resolución, haga click en *Replace Meshes* (reemplazar mallas) (o *Add New Meshes* añadir nueva malla) y después *Free Radio Data*. Esto finaliza la resolución y reemplaza la escena anterior opr la nueva escena de resolución Radiosidad.

Note Añadir en lugar de reemplazar las mallas es una fomra de deshacer cambios. Usted mantiene las mallas antiguas y puede volver a ejecutar radiosidad de nuevo. Pero tiene que mover las mallas nuevas a una nueva capa y ocultar las viejas antes de renderizar.

Ahora estamos preparados para pulsar **F12** y renderizar ([Figure 18-22](#)).

Figure 18-22. Render de la solución de radiosidad.



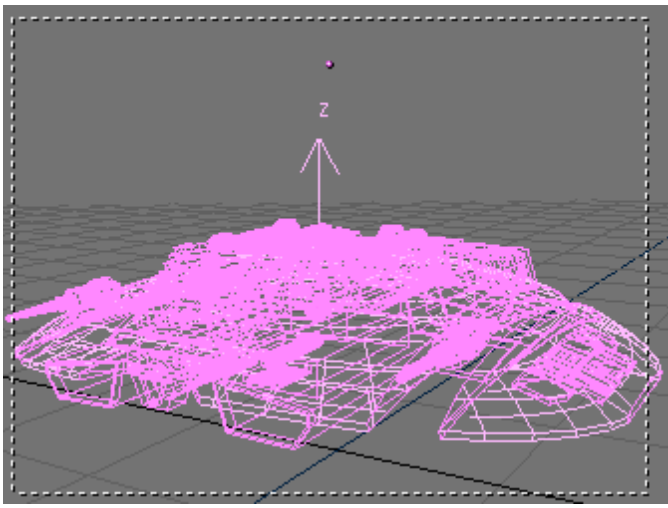
18.4.4. Texturizado

¡Allá vamos amigos! Ahora tiene un render muy limpio con una luz ténue de 360 grados usando radiosidad. Muy bonito... pero lo siguiente que tenemos que hacer es añadir texturas a la malla. Así que volvamos al área de la pantalla principal.

Ahora intente seleccionar su malla. Se dará cuenta de que no solo se selecciona la nave, sino también el plano y la cúpula. Esto es debido a que la radiosidad ha creado una nueva malla *única* durante el proceso de resolución. Para añadir la textura, solo necesitamos la nave.

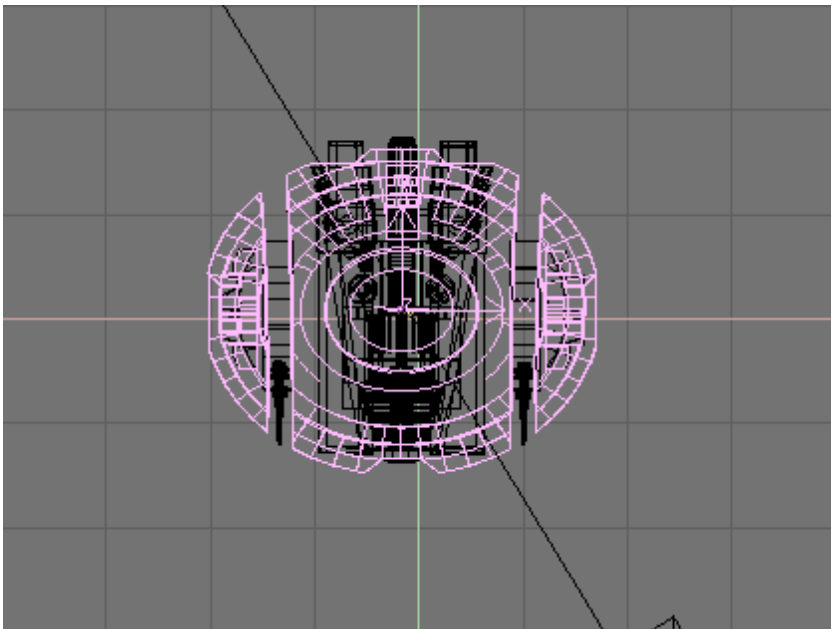
Así, seleccione la malla y vaya al Modo Edición. En este modo, podemos eliminar la cúpula y el plano ya que no les vamos a volver a necesitar. Puede usar **LKEY** para seleccionar los vértices adecuados y presionar **XKEY** para eliminarlos. Siga seleccionando y eliminando hasta que quede solo la nave. Debería aparecer algo como [Figure 18-23](#). Si quisieramos renderizar ahora, con **F12**, solo obtendríamos un fondo negro y nuestra nave. Es bonito... pero de nuevo, ¡queremos texturas!

Figure 18-23. La malla de la nave.



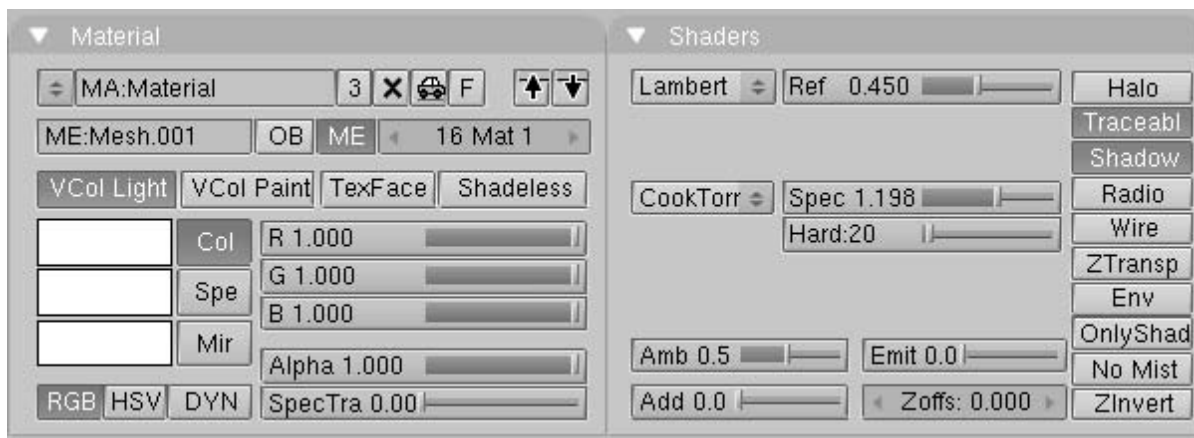
Para añadir texturas a la malla, debemos separar las áreas sobre las que vamos a aplicar materiales y texturas. Para la nave, queremos añadir texturas a las alas y una sección media. Para hacerlo seleccione la malla de la nave, y vuelva al Modo Edición. Seleccione un vértice cerca del final del ala y presione **LKEY** para seleccionar los vértices enlazados. Haga lo mismo con el otro lado. Ahora seleccione las áreas vistas en [Figure 18-24](#). Cuando lo tenga, presione **PKEY** para separar los vértices seleccionados.

Figure 18-24. Separando las partes de la nave que serán texturizadas.



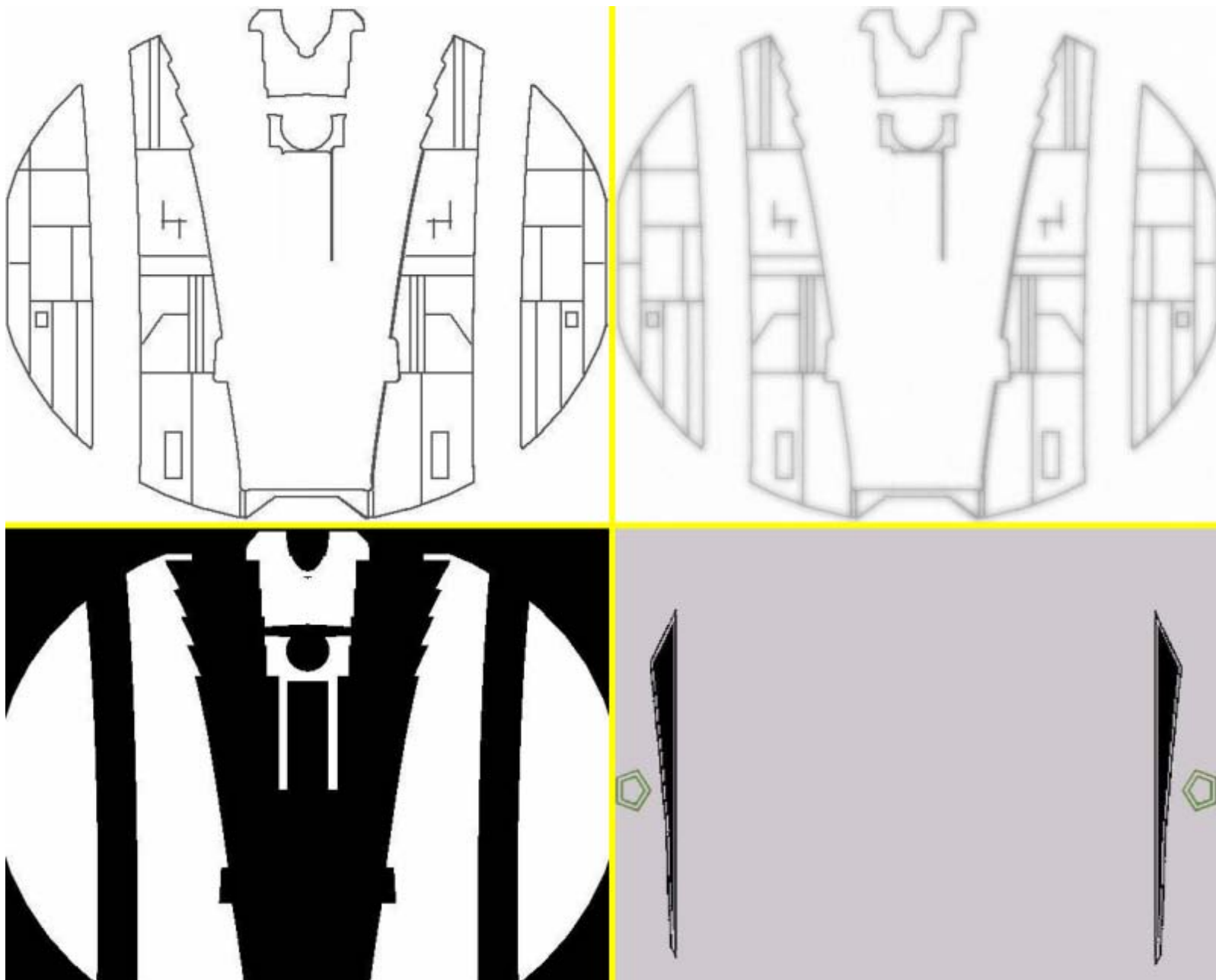
Ahora tenemos nuestra sección del ala separada y estamos preparados para añadir materiales y texturas. Queremos crear un nuevo material para esta malla. Para conseguir un resultado de aspecto metálico, podemos usar las propiedades que puede ver en [Figure 18-25](#).

Figure 18-25. Material "Metálico".



Es el momento de añadir las texturas. Queremos conseguir algunos resultados bien elaborados. Necesitaremos dos mapas de relieve para crear ranuras y dos máscaras para pintar y 'calcar'. Hay, de este modo, cuatro texturas para las alas de la nave que tienen que ser creadas, como se puede ver en [Figure 18-26](#).

Figure 18-26. Cuatro texturas, desde la esquina superior izquierda, en el sentido de las agujas del reloj: RaiderBM, RaiderDI, Markings, Raider.





Las texturas deberían ir situadas en cuatro canales de material en la malla global de la nave. 'RaiderBM' y 'RaiderDI' deberían ponerse en negativo *NOR* ([Figure 18-27](#) un click *NOR* se volverá amarillo). 'Raider' debería ponerse en negativo el *REF* ([Figure 18-27b](#)).

Tip ¿Qué material?

Una malla importada de una resolución de radiosidad tiene normalmente más de un material sobre ella. Es importante operar en el material "original" correcto.

Figure 18-27. Configuración de texturas.



El resultado es el plateado metálico deseado para la cabina de la nave. Finalmente la cuarta textura, 'Markings', se sitúa en *COL* en los botones de material ([Figure 18-27c](#)). Esto dará a la nave su propia marcas e insignias. Nuestra nave es totalmente plana, así que la proyección *Flat* (Plana) es adecuada. Cuando sea una forma más compleja será necesario algo de mapeado UV para conseguir buenos resultados. La previsualización del material para la malla debería ser algo como [Figure 18-28](#).

Figure 18-28. Previsualización completa de material.



Nuestras texturas *no* se verán en la renderización por ahora (excepto markings) porque los tipos de textura "Nor" y "Ref" reaccionan a la luz, ¡y no hay fuente de iluminación en la escena! De este modo, ahora necesitamos añadir una lámpara o dos, teniendo en mente que nuestra nave ya está bien iluminada por la solución de radiosidad, de modo que la energía de las lámparas debe ser muy débil. Una vez tenga sus lámparas, intente un test de render. Experimente con las lámparas hasta obtener el resultado deseado.

El render final muestra una bonita y bien iluminada nave de asalto con texturizado suave. ([Figure 18-11](#))

[Prev](#)

[Next](#)

Chapter 19. Trazado de rayos (Raytracing)

Aún por escribir

[Prev](#)

Un jugoso ejemplo de radiosidad

[Home](#)

[Up](#)

[Next](#)

Herramientas Avanzadas

V. Herramientas Avanzadas

Hay herramientas y técnicas que no pueden ser encapsuladas solamente en las partes de Modelado o Animación ya que son una mezcla de las dos.

Por tanto, esta parte está dedicada a la utilización de la Animación como una herramienta de modelado y a los efectos especiales proporcionados por Blender.

Dentro de la categoría de herramientas avanzadas hay una muy importante, el Editor de Secuencias, un completo editor de video capaz de proporcionar a tus animaciones un acabado totalmente profesional.

Table of Contents

20. [Efectos](#)

20.1. [Introducción](#)

20.2. [Partículas sencillas](#)

20.2.1. [El primer sistema de partículas](#)

20.2.2. [Renderizando un sistema de partículas](#)

20.2.3. [Objetos como partículas](#)

20.2.4. [Haciendo fuego con las partículas](#)

20.2.5. [Una simple explosión](#)

20.2.6. [Fuegos Artificiales](#)

20.2.7. [Controlando las partículas mediante celosía](#)

20.2.8. [Partículas estáticas](#)

20.3. [Interacción entre Partículas](#)

20.3.1. [Introducción](#)

20.3.2. [La interfaz](#)

20.3.3. [Ejemplo](#)

21. [Otros Efectos](#)

21.1. [Introducción](#)

21.2. [Efecto Build](#)

21.3. [Efecto Onda](#)

22. [Tecnicas especiales de modelado](#)

22.1. [Introducción](#)

22.2. [DupliVerts](#)

22.2.1. [DupliVerts as an Arranging Tool](#)

22.2.2. [DupliVerts para modelar un objeto](#)

22.3. [DupliFrames](#)

22.3.1. [Modelando con Dupliframes](#)

22.3.2. [Colocando objetos con DupliFrames](#)

22.3.3. [Más modelado y animación](#)

22.4. [Modelando con lattices](#)

22.4.1. [¿ Cómo se hace ?](#)

23. [Efectos Volumétricos](#)

24. [Cuerpos blandos](#)

24.1. [Introducción](#)

24.2. [El sistema SoftBody de Blender](#)

24.2.1. [Las bases](#)

24.2.2. [Interfaz](#)

24.2.3. [Ejemplo](#)

25. [Editor de secuencia](#)

25.1. [Conociendo el editor de secuencia](#)

25.1.1. [Primera animación: dos cubos](#)

25.1.2. [Primera secuencia: alambres retrasados](#)

25.1.3. [Segunda animación: un cubo solido desfasado](#)

25.1.4. [Tercera animación: Un túnel](#)

25.1.5. [Segunda secuencia: Usando el túnel como telón de fondo](#)

25.1.6. [Cuarta animación: un logo saltarín](#)

25.1.7. [Quinta animación: Barras de partículas](#)

25.1.8. [Tercera secuencia: Combinando el logo y las barras de partículas](#)

25.1.9. [Sexta animación: Haciendo zoom al logo](#)

25.1.10. [Ensamblandolo todo](#)

25.1.11. [Conclusión](#)

25.2. [Editor de Secuencia de sonidos](#)

25.3. [Plugins del editor de secuencia](#)

[Prev](#)

[Home](#)

[Next](#)

Trazado de rayos (Raytracing)

Efectos

Chapter 20. Efectos

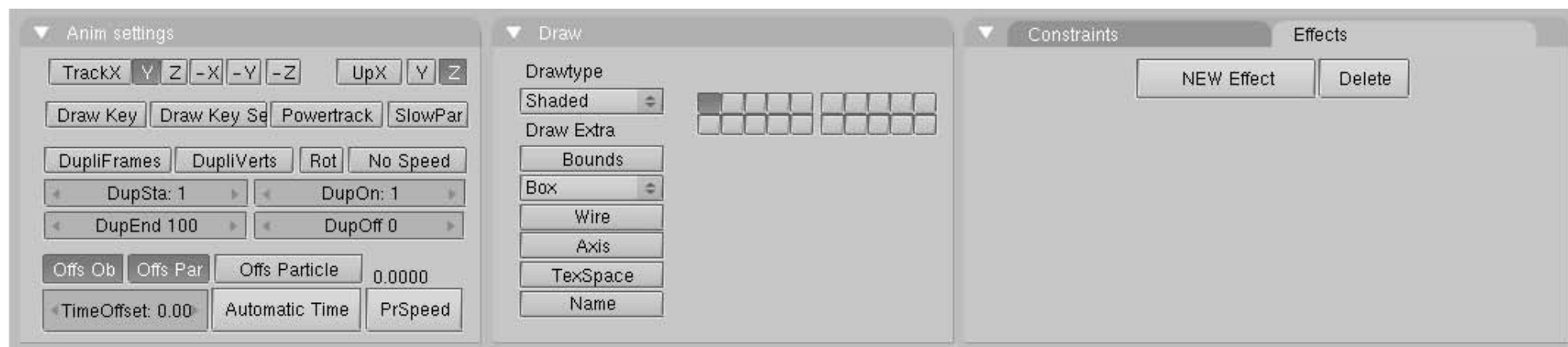
20.1. Introducción

Relativo a Blender v2.31

Hay tres tipos de efectos que pueden ser asociados a un objeto, idealmente trabajando a lo largo de animaciones pero en la practica se aprovechan en las imágenes estáticas.

Los efectos son añadidos a un objeto seleccionándolos, cambiando al Object Context y localizando el panel *Effects* en el panel *Constraints* (F7 o ). Pulsando el botón *New Effect* [Figure 20-1](#) El efecto se añade.

Figure 20-1. Ventana de botones de animación



El botón *Delete* elimina un efecto, si hay alguno, mientras que la lista desplegable que aparece a la derecha una vez se ha añadido un efecto ([Figure 21-1](#)) selecciona el tipo de efecto.

Se pueden asociar varios efectos a un único Objeto. Se crea una fila de pequeños botones, una por cada efecto, junto al botón *New Effect* permitiendo cambiar entre ellos para modificar sus propiedades.

Los tres efectos son *Build*, *Particles* y *Wave*, el segundo, es el más versátil y se le reservará un capítulo entero para él.

20.2. Partículas sencillas

Relativo a Blender v2.31

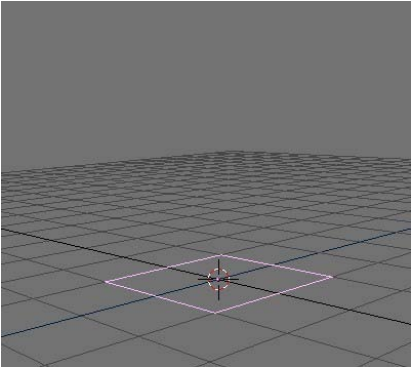
El sistema de partículas de Blender es rápido, flexible y potente. Cada Objeto de Malla puede servir como emisor de partículas. Los Halos pueden ser usados como partículas y con la opción DupliVert, también los objetos. Estos objetos con DupliVert pueden ser de cualquier tipo de Blender, por ejemplo, Mallas, Curvas Metaballs, o incluso focos. Las partículas pueden ser influenciadas por fuerzas globales para simular efectos físicos, como gravedad o viento.

Con estas posibilidades se pueden generar humo, fuego, explosiones, fuegos artificiales o incluso bandadas de pájaros. Con las partículas estáticas se puede generar pelo, pieles, hierba o plantas.

20.2.1. El primer sistema de partículas

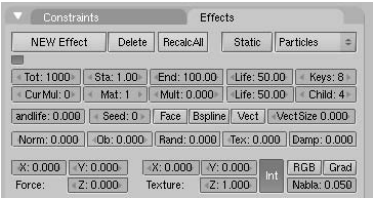
Reinicia Blender a la escena por defecto, o crea una escena con un solo plano desde la vista superior. Este plano será nuestro emisor de partículas. Rota la vista hasta que tengas una buena visión del plano y el espacio sobre él. ([Figure 20-2](#)).

Figure 20-2. El emisor.



Cambia al panel *Effects* en el Object Context (F7 o ) y haz click en el botón *NEW Effect* en la mitad del Panel. Cambia la lista desplegable de MenuButton desde *Build* a *Particles*. Ahora se mostrarán los botones de partículas. ([Figure 20-3](#)).

Figure 20-3. Botones de partículas.



En el NumButton *Norm*: ponga 0.100 con un click en la parte derecha del botón o use **SHIFT-LMB** para introducir el valor por teclado.

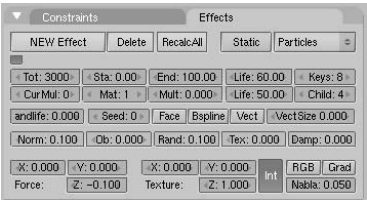
Reproduzca la animación presionando **ALT-A** con el ratón sobre la ventana 3D. Verá una cadena de partículas ascendiendo verticalmente desde los cuatro vértices.

Felicidades - ¡Acaba de crear su primer sistema de partículas en unos sencillos pasos!.

Para crear un sistema un poco mas interesante, es necesario adentrarse mas en el sistema y sus botones ([Figure 20-4](#)):

- El parámetro *Tot*: cpmtrola la cantidad total de partículas. En CPUs nuevas y rápidas podrá ampliar este número sin notar una disminución del rendimiento.
- El número total de partículas indicadas en el botón *Tot*: se crean uniformemente a lo largo de un intervalo de tiempo. Dicho intervalo está definido por los NumButtons *Sta*: y *End*:, que controlan el frame de inicio y el final en el que las partículas son generadas.
- Las partículas tienen un tiempo de vida, duran un número de frames, desde que se crean hasta que desaparecen. Puede cambiar el tiempo de vida de las partículas con el NumButton *Life*:-.
- El NumButton *Norm*: es usado para dar a las partículas una velocidad inicial con un valor constante (0.1), direccionado a lo largo de las normales del vértice. Para hacer ésto más "aleatorio" puede modificar el NumButton *Rand*:- hasta 0.1 también. Ésto también hace que las partículas empiecen con una variación de velocidad aleatoria.
- Use el grupo de NumButtons *Force*: para simular una fuerza constante, como viento o gravedad. Un valor de *Force*: Z: de -0.1 hará que las partículas caigan al suelo, por ejemplo.

Figure 20-4. Propiedades de partículas.



Esto debería ser suficiente para que empezara, pero no tenga miedo de manipular algunos de los otros parámetros mientras está experimentando. Los cubriremos en detalle en secciones posteriores.

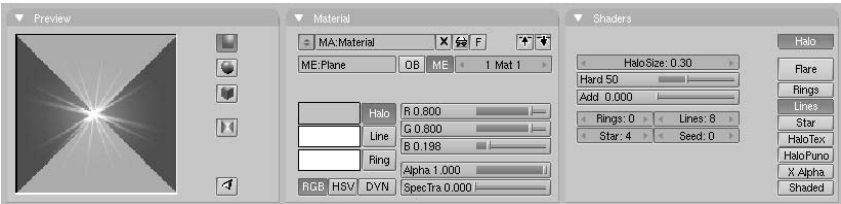
20.2.2. Renderizando un sistema de partículas

Es posible que haya intentado a renderizar una imagen con el ejemplo de antes. Si la cámara estaba correctamente alineada, habrá visto una imagen negra con unos puntos grisáceos en ella. Éste es el Halo-material estandard que Blender asigna a un sistema de partículas creado nuevo.

Posicione la cámara de forma que obtenga una buena visión del sistema de partículas. Si desea añadir un ambiente sencillo, recuerde añadir algunas luces. Los Halos son renderizados sin luces, mientras no se hayan indicado, pero otros objetos necesitan luces para ser visibles.

Vaya a los Botones de Materiales (F5) y añada un nuevo material para el emisor si no lo ha hecho antes. Haga click en el Boton "Halo" en el medio de la tabla (Figure 20-5).

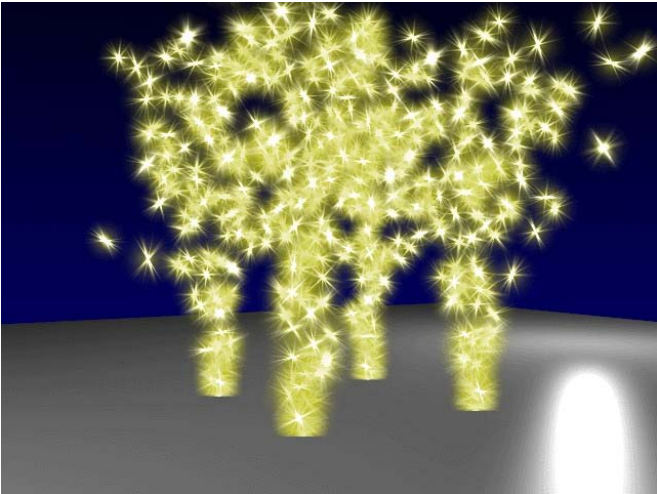
Figure 20-5. Propiedades del Halo



Los Botones de Material cambian a los Botones de Halo. Elija *Line*, y ajuste *Lines*: a un valor de su elección (puede ver el efecto directamente en en Material-Preview). Disminuya *HaloSize*: a 0.30, y elija un color para el Halo y para las líneas (Figure 20-5).

Ahora puede renderizar una imagen con **F12**, o una animación completa y ver miles de estrellas volando alrededor (Figure 20-6).

Figure 20-6. Estrellas disparándose



20.2.3. Objetos como partículas

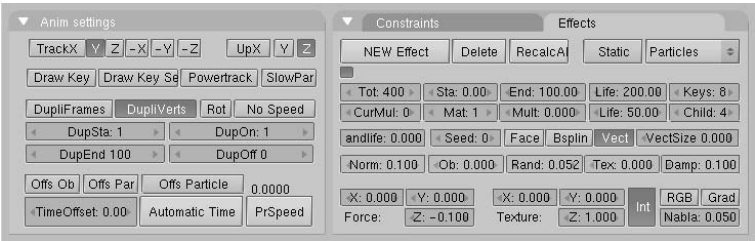
Es muy fácil crear objetos como partículas, es exactamente la misma técnica descrita en [Section 22.2](#).

Empiece creando un cubo, o cualquier otro objeto que desee, en su escena. Merece la pena pensar sobre la potencia de su ordenador, dado que vamos a tener tantos objetos como *Tot*: indique, en la escena. ¡Ésto significa tener tantos vértices como el objeto elegido por el valor de *Tot*!

Escale el nuevo objeto disminuyendolo hasta que encaje con la escala general de la escena

Ahora seleccione el objeto, y haga **SHIFT-RMB** con el emisor y hagalo padre del cubo usando **CTRL-P**. Elija el emisor solo y seleccione la opción "DupliVerts" en el Panel *Anim Settings* del Object Context (F7). Los cubos con la opción "DupliVert" aparecerán inmediatamente en la ventana 3D.

Figure 20-7. Colocando las partículas con DupliVert.



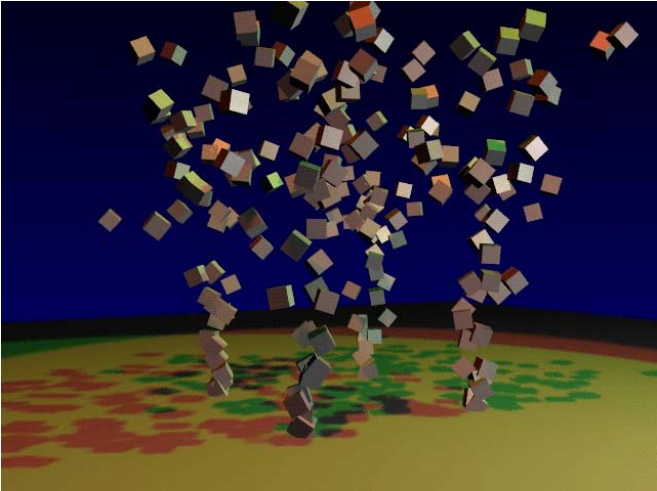
Usted podría querer reducir el número de partículas presionando **ALT-A** ([Figure 20-7](#)).

En la animación se dará cuenta que todos los cubos comparten la misma orientación. Esto puede ser interesante, pero también puede serlo tener los cubos orientados aleatoriamente.

Ésto lo podemos hacer seleccionando la opción *Vect* en los parámetros de partícula, que causa que los dupli-objetos sigan la rotación de las partículas, resultando un movimiento más natural ([Figure 20-7](#)). Podemos ver un frame de la animación ([Figure 20-8](#)).

Tip **Objeto original**
Recuerde mover el objeto original fuera de la vista de cámara, ya que al contrario que una Malla de DupliVerts normal, las partículas con DupliVert también serán renderizadas.

Figure 20-8. Renderizado de partículas con DupliVert.

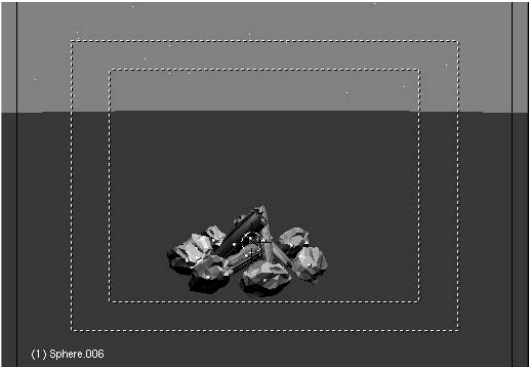


20.2.4. Haciendo fuego con las partículas

El sistema de partículas de Blender es muy útil para hacer fuego y humo realista. Esto podría ser una vela, una hoguera, o una casa ardiendo. Es interesante conocer cómo el fuego esta condicionado por la física. Las llamas de un fuego son gases calientes. Ascenden porque su densidad es menos que la del aire frío circundante. Las llamas son calientes y brillan en el medio, y se vna haciendo mas oscuras hacia su perímetro.

Prepare un escenario sencillo para nuestro fuego, con algunos trozos de madera, y algunas piedras. ([Figure 20-9](#)).

Figure 20-9. Configuración de la hoguera.

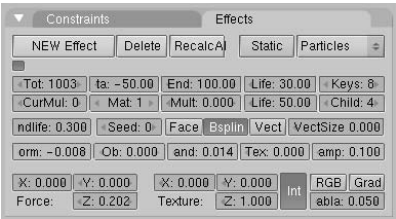


20.2.4.1. El sistema de partículas

Añada un plano en el medio del círculo de piedras. Este plano será nuestro emisor de partículas. Subdivida el plano una vez. Ahora podrá mover los vértices a una posición en la madera donde las llamas (partículas) se deberían originar.

Ahora vaya al Object Conext **F7** y añada un nuevo efecto de partículas al plano. Los números aquí indicados ([Figure 20-10](#)) deberían valer para un fuego realista, pero alguna puede ser necesaria alguna modificación, dependiendo del tamaño del emisor acutal.

Figure 20-10. Configuración de las partículas de fuego.



Algunas notas:

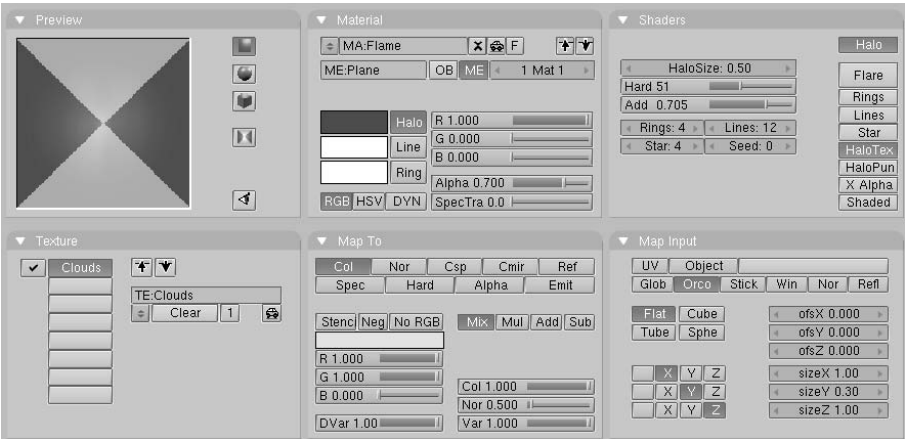
- Para tener el fuego ardiendo desde el principio de la animación haga *Sta*: negativo. Por ejemplo, intente -50. El valor de *End*: debería reflejar la longitud deseada para la animación.
- El valor de las partículas *Life*: es 30. Normalmente podría estar sobre 50 por ahora. Usaremos este parámetro después cuando ajustemos la altura de las llamas.
- Haga el parámetro *Norm*: a poco negativo como por ejmplo (-0.008) que resultará un fuego mas potente en su base.
- Use *Force*: Z: alrededor de 0.200. Si su fuego parece demasiado lento, ajuste este parámetro.
- Cambie *Damp*: a 0.100 para frenar las llamas al cabo de un rato.
- Active el botón *Bspline*. Esto usará un método de interpolación que dará mucho más movimiento de fluido.
- Para añadir algo de aleatoriedad a nuestras partículas, ajuste el parámetro *Rand*: alrededor de 0.014. Use el parámetro *Randlife*: para añadir aleatoriedad e la vida de las partículas; un valor muy alto nos dará una llama muy viva.
- Use alrededor de 600-1000 particulas en total para la animación (*Tot*:).

El la ventana 3D, obtendrá una primera impresión de como de realistas quedará el movimiento de las llamas. Pero lo más importante para nuestro fuego será el material.

20.2.4.2. El material de fuego

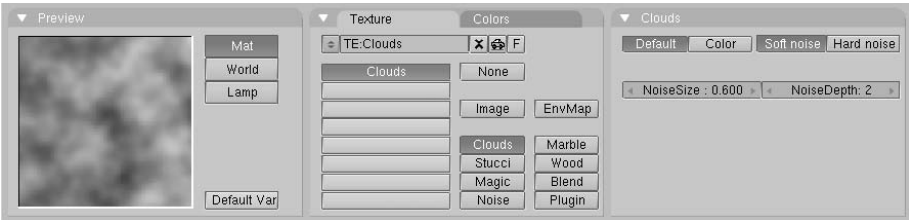
Con el emisor de partículas seleccionado, vaya al panel de sombreado **F5** y añada un nuevo Material. Haga este nuevo material de tipo Halo activando el botón **Halo**. Active también *HaloTex*, situado debajo de dicho botón. Ésto nos permitirá usar una textura más adelante.

Figure 20-11. Material de Llamas.



Asigne al material un color rojo completamente saturado con los deslizadores RGB (RGB-sliders). Disminuya el valor Aplha a 0.700; ésto hará las llamas un poco transparentes. Aumente el deslizador *Añadir* (*Add*) hasta 0.700, así los Halos potenciarán a los demás, dando un brillo interior a las llamas, y mas oscuro en exterior. ([Figure 20-11](#)).

Figure 20-12. Textura de las llamas.



Si ahora hace un render de prueba, verá que solo puede ver una brillante llama roja. Para darle un toque más realista, necesitaremos una textura. Mientras el emisor siga seleccionado, vaya al Panel de Texturas (Texture Panel) y añada una nueva textura del tipo *Nube* (*Cloud*), en los botones de textura (**F6**). Ajuste el *Tamaño de Ruido* (*NoiseSize*:.) a 0.600. ([Figure 20-12](#)).

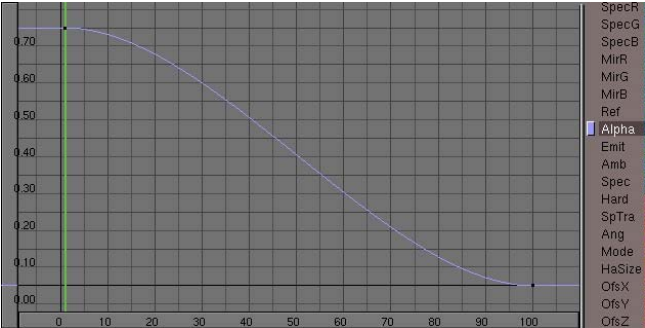
Vuelva a los botones de Material **F5** y haga el color de textura a un amarillo con los deslizadores RGB en el lado derecho de los botones de material. Para ajustar los puntos amarillos en la textura de nube, disminuya el valor de *Tamaño Y (SizeY)* a 0.30.

Un render de prueba ahora nos mostrará un bonito fuego. Pero seguimos necesitando crear partículas apagándose en lo alto del fuego. Podemos conseguir ésto con una animación del Alpha y del Tamaño del Halo del material.

Asegurese que su animación está en el frame 1 (**SHIFT-LEFTARROW**) y mueva el ratón hasta la ventana de materiales. Ahora presione la tecla **IKEY** y elija Alpha del menú emergente. Avance el barra de tiempos hasta el frame 100, e introduzca un valor Alpha de 0.0 e inserte otra clave para el Alpha con la tecla **IKEY**. Cambie una de las ventanas a una ventana IPO. Active el tipo de material IPO haciendo click en la sección del menú correspondiente. Ahora verá una curva del canal Alpha del material ([Figure 20-13](#)).

Note Una animación para un material de partículas siempre está mapeada desde los primeros 100 frames de animación hasta el período de vida de la partícula. Ésto significa que cuando desvanecemos un material en un frame del 1 al 100, una partícula con un período de vida de 50, se desvanecerá en ése tiempo.

Figure 20-13. Material IPO del fuego



Ahora puede renderizar la animación. Quizá deba refinar algunos parámetros como el período de vida de las partículas. Puede añadir un mayor realismo a la escena animando las luces (o usando luces direccionales con sombra) y añadiendo un sistema de partículas de chispas al fuego. También es recomendado animar el emisor para obtener llamas mucho más vivas, o usar más de un emisor ([Figure 20-14](#)).

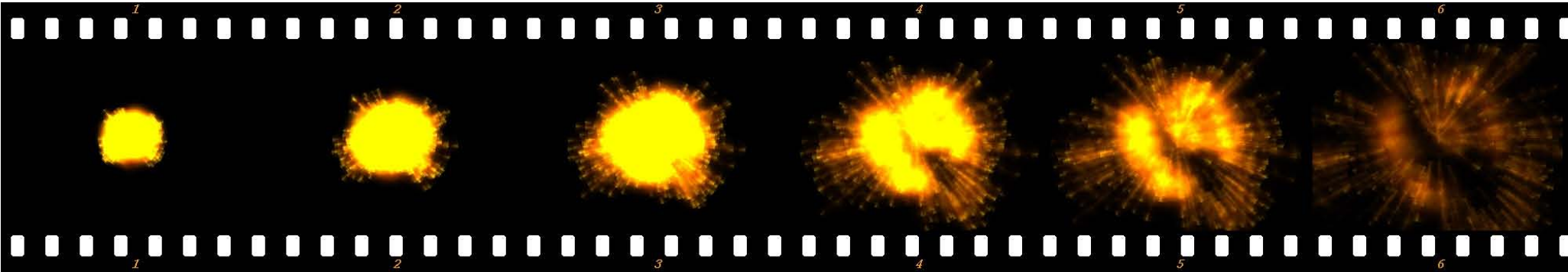
Figure 20-14. Render Final.



20.2.5. Una simple explosión

Esta explosión está diseñada para ser usada como textura animada, para insertarla en la escena actual o para ser usada como textura animada. Para un render estático, o una cámara lenta de una explosión, es posible que necesitemos hacer un poco más de trabajo para hacer que se vea realmente bien. Pero tenga en cuenta, que nuestra explosión sólo será vista durante medio segundo ([Figure 20-15](#)).

Figure 20-15. La explosión



Como emisor para la explosión he elegido una IcoEsfera (IcoSphere). Para hacer la explosión un tanto irregular, he eliminado patrones de vertices con la función de selección circular en el modo Edición (Edit Mode). Para una escena concreta podría ser mejor usar un objeto como emisor, que esté formado de forma diferente, por ejemplo, el objeto actual que se quiera explotar.

Mi explosión está compuesta por dos sistemas de partículas, uno para la nube de gases calientes, y otro para las chispas. He cogido una versión girada del emisor para generar las chispas. Además, he animado la rotación de los emisores mientras las partículas son generadas.

20.2.5.1. Los materiales

Las partículas para la explosión son de un material Halo muy claro, con una textura de nube para añadir aleatoriedad, las chispas también tienen un material similar, mire [Figure 20-16](#) a [Figure 20-18](#).

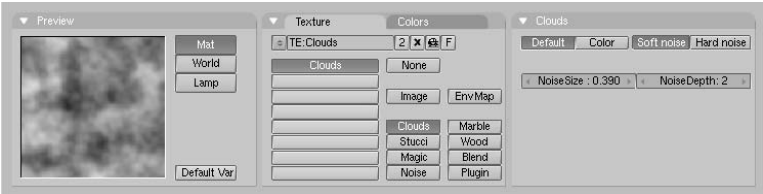
Figure 20-16. Material para la nube de la explosión.



Figure 20-17. Material para las chispas.



Figure 20-18. Textura de ambos.



Anime el valor Alpha del de las partículas del Halo de 1.0 a 0.0 en los primeros 100 frames. Esto hará que sean mapeadas al período de vida de las partículas, como es normal. Dese cuenta de la propiedad Star en el material de las chispas ([Figure 20-17](#)). Ésto modela un poco las chispas. También podríamos haber usado una textura especial para conseguir esto, de todas maneras, en este caso, la manera más fácil es usando la propiedad Star.

20.2.5.2. Los sistemas de partículas

Figure 20-19. Particle system for the cloud

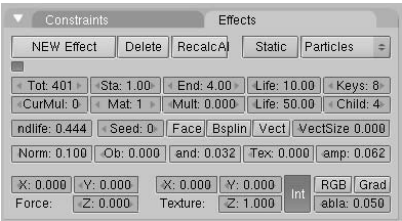
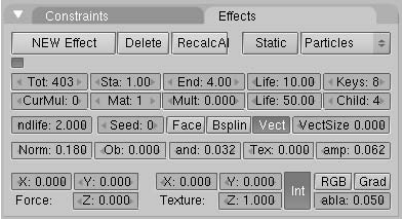


Figure 20-20. Sistema de partículas para las chispas



Como puede ver en (Figure 20-19) y (Figure 20-20), los parámetros son básicamente los mismos. La diferencia está en la propiedad *Vect* para las chispas, y una mayor cantidad en *Norm*: para provocar mayor velocidad a dichas chispas. También he situado el valor de *Randlife*: para las chispas a 2.000 dando una forma irregular.

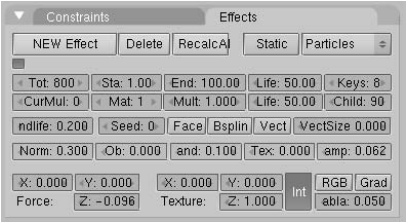
Le sugiero que empiece a experimentar, usando estos parámetros como inicio. Los valores a utilizar dependen de lo que quiera conseguir. Intente añadir más emisores para obtener escombros, humo, etc.

20.2.6. Fuegos Artificiales

Un botón que todavía no hemos usado es el *Mult*:. Toda la tercera línea de botones del panel está relacionada con ésto. Cree un plano y añada un sistema de partículas al mismo.

Ajuste los parámetros de forma que obtenga algunas partículas volando en el cielo, y aumente el valor de *Mult*: a 1.0. Ésto provocará que el 100% de las partículas generen partículas hijas cuando su período de vida acabe. Así, cada partícula generará cuatro hijos. Entonces necesitamos aumentar el valor *Hijos* (*Child*:) a alrededor de 90 (Figure 20-21). Ahora debería ver un fuego artificial convincente hecho con partículas cuando previsualice la escena con **ALT-A**.

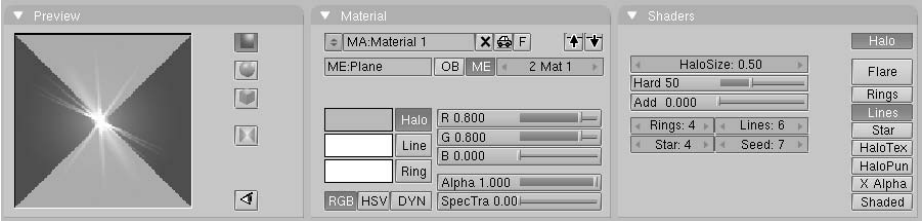
Figure 20-21. Botones de multiplicación de partículas



Cuando renderice los fuegos artificiales, no parecerán muy impresionantes. Ésto es debido al material estandard de Halo asignado por Blender. Por lo tanto el siguiente paso es asignar un mejor material.

Asegurese que tiene el emisor seleccionado y vaya al panel de sombreado (Shading Context) y a los botones de material (**F5**). Añada un nuevo material con Botón de Menú, y seleccione el tipo *Halo*.

Figure 20-22. Material de los fuegos artificiales 1

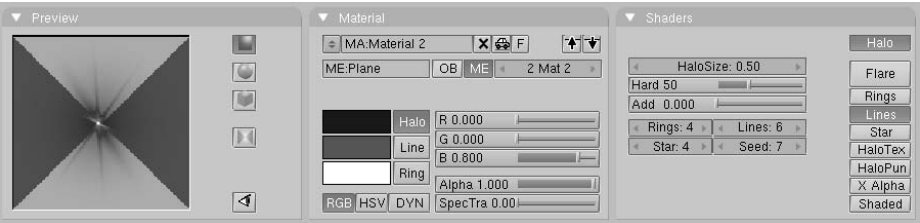


He usado un material halo muy claro; puede ver los parámetros en Figure 20-22. La animación renderizada ahora se verá mucho mejor, pero aún queda algo que podemos hacer.

Mientras el emisor esté seleccionado, vaya al panel Edición (Editing Context) **F9** y añada un nuevo índice de material haciendo click en el botón *New* en el panel *Enlace y materiales* (*Link and Materials*) (Figure 20-23).

Figure 20-23. Añadiendo un segundo material al emisor.

Figure 20-24. Material 2



Tip Más mejoras

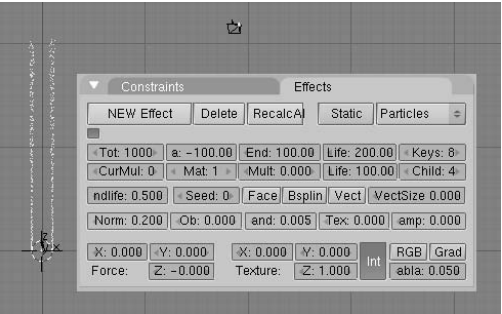
Aparte de cambiar los materiales, también puede usar el los IPOs del material para animar las propiedades de cada uno de los materiales.

20.2.7. Controlando las partículas mediante celosía

El sistema de partículas de Blender es muy potente, y el camino de las partículas no solo estará determinado por las fuerzas sino canalizado por una celosía.

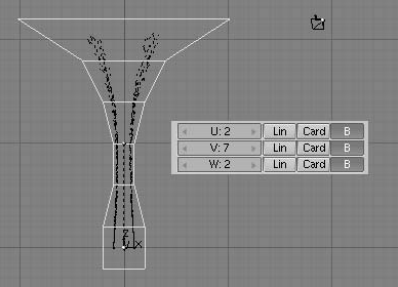
Cree una simple maya cuadrada y añada un sistema de partículas en ella con un una fuerza-z negativa y los parámetros generales así [Figure 20-25](#).

Figure 20-25. Propiedades de partícula



Esto podría ser un buen humo para cuatro pequeños fuegos en un día con poco viento, pero queremos arremolinarlo. Añada una celosía y deformela como ve en [Figure 20-26](#).

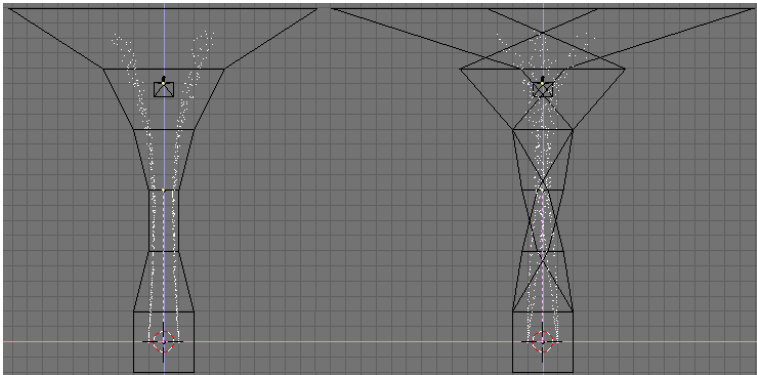
Figure 20-26. Propiedades de celosía



Asocie el emisor de partículas con la celosía (**CTRL-P**). Si ahora selecciona el emisor de partículas cambie a los botones de animación (**F7**) y presione *RecalcAll* ahora se dará cuenta que las partículas siguen, en mayor o menor medida, la celosía ([Figure 20-27](#) en la izquierda).

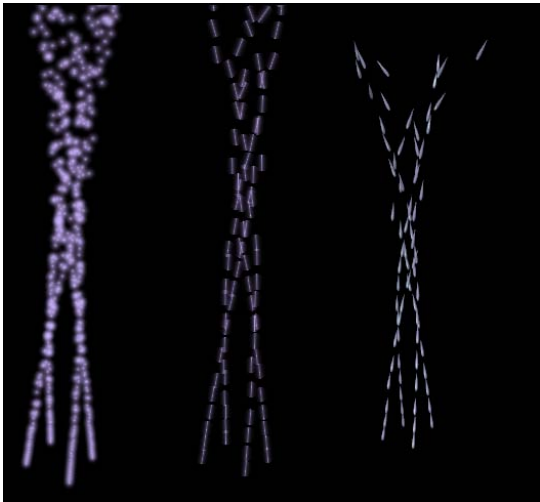
Una mejora añadida, rote cada sección horizontal de la celosía 60 grados en el sentido horario desde la vista superior incrementalmente, como si estuviera haciendo un tornillo. Después, recalculé de nuevo las partículas. El resultado está en [Figure 20-27](#) en la derecha.

Figure 20-27. Efectos de deformación de celosía



El remolino es evidente, y por supuesto, puede conseguir mayores efectos rotando más la celosía, o usando una con más subdivisiones. Si le da al emisor un material Halo y renderiza verá algo como [Figure 20-28](#) en la izquierda.

Figure 20-28. Partículas normales, izquierda; Partículas vectoriales, centro; Y objetos con vertices duplicados siguiendo las partículas, derecha.



Si selecciona el emisor, vaya a los botones de animación y presione el botón de partícula Vectr. Las partículas cambiarán de ser puntos a segmentos, con una longitud y dirección proporcional a la velocidad de la partícula. Un renderizado debería dar un resultado como [Figure 20-28](#) en el medio.

Si ahora duplica los vertices (DupliVert) de un objeto al emisor, asociándolo (Parent) y haciendo click en el botón DupliVert, el objeto con vertices duplicados tendrá la misma orientación que el objeto original si las partículas son normales, pero será alineado con la dirección de las mismas si están seleccionadas como Vert. Seleccionando el Objeto Original y jugando con los botones Track puede cambiar la orientación ([Figure 20-28](#) a la derecha).

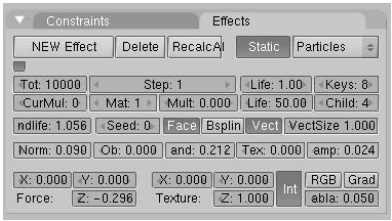
20.2.8. Partículas estáticas

Las partículas estáticas son muy útiles cuando creamos objetos como fibras, hierba, pelo y plantas.

Intente crear un pequeño personaje, o simplemente una pelota, para probar las partículas estáticas. Intente por ejemplo, una pequeña 'bola de pelo'. El emisor no se renderiza, así que duplique la maya (o el objeto que quiera usar y conviértalo en una malla (ALT-C)). Podría ser buena idea subdividir la malla con un fractal para obtener algo de aleatoriedad en ella. Si termina con una malla demasiado densa, use "Eliminar duplicados (Remove Doubles)" con un límite elevado. Elimine las partes del círculo donde no quiera tener pelo.

Ahora, asigne el sistema de partículas y cambie a la opción Static.

Figure 20-29. Propiedades de partículas estáticas



Use los parámetros que puede ver en [Figure 20-29](#). Con la combinación de *Life* y *Norm* puede controlar la longitud del pelo. Use una fuerza en una dirección Z negativa para permitir que el pelo se combe. Active la propiedad *Face* para generar las partículas, no solo en los vértices, sino también distribuidas por las caras. Active también *Vect*; estáó generará la fibra como partículas. La importancia del paso indica cuantas partículas por periodo de vida son generadas. Sitúelo en un valor pequeño para obtener curvas más suaves para las partículas, y asegúrese de no pasar por alto el valor de la propiedad *Rand*.

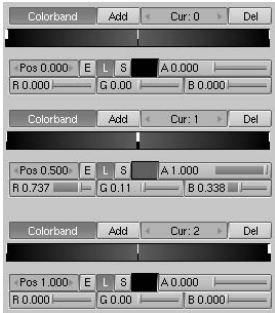
Cuando renderice, verá que obtiene partículas muy difuminadas. El material usado para las partículas estáticas es muy importante, de modo que añada un material para el emisor en el panel de sombreado (Shading Context) (**F5**).

Figure 20-30. Propiedades de material



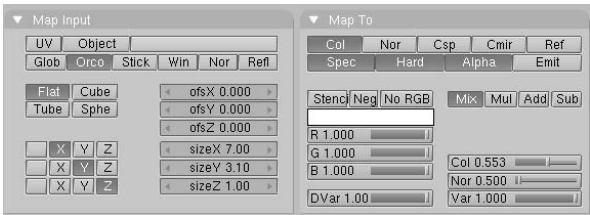
He usado un tamaño de Halo (HaloSize) muy pequeño (0.001). En el botón numérico no se podrá ver así que para ajustarlo, haga click en el botón con **LMB** pulsado mientras mantiene **SHIFT** apretado. Active la opción *sombreado* (*Shaded*) para que las partículas se vean influenciadas por las luces de la escena, después active *HaloTxt*. Ahora vamos a usar una textura para sombrear los pelos ([Figure 20-30](#)).

Figure 20-31. Propiedades de banda de colores de textura



Vaya al subpanel de textura (Texture sub-context) (**F6**) y añada un nuevo tipo de textura *Blend*. Elija como subtipo *Lin*. Active la opción banda de color (colorband) y ajuste los colores como se ve en [Figure 20-31](#). Obtendrá un bonito degradado, desde transparente hasta púrpura, volviendo de nuevo a transparente.

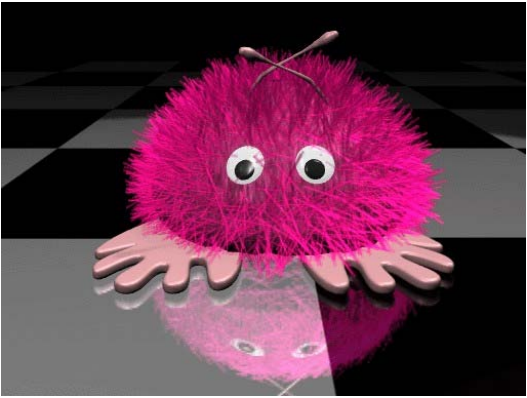
Figure 20-32. Propiedades de textura en los botones de material



Vuelva al subpanel de los botones de material y asegúrese que *Alpha* está activado en la salida del mapeado de textura (texture mapping output) a la derecha de los botones de material. Use *sizeX* y *sizeY* para definir la forma del halo en la previsualización del material como una peñaña fibra ([Figure 20-32](#)).

Si su pelo no es suficientemente denso, incremente el número de partículas con *Tot* o añada más emisores. También cambie los parámetros para los emisores adicionales para obtener algo de variación en los pelos some variation in the hairs ([Figure 20-33](#)).

Figure 20-33. Resultado final



[Prev](#)
Efectos

[Home](#)
[Up](#)

[Next](#)
Interacción entre Partículas

20.3. Interacción entre Partículas

por Kenneth Styrberg

Relativo a Blender v2.34

20.3.1. Introducción

El sistema de partículas de Blender permite que las partículas interactúen de dos maneras - campos de fuerza (force fields) y desvío (deflection).

Campos de Fuerza

Existen dos tipos de campos de fuerza: un *Campo de Fuerza* estándar, que se comporta como un campo de fuerza gravitacional, y el *Campo de Vértice (Vortex field)*. Un campo de vértice produce un efecto parecido al de un tornado, haciendo que las partículas se muevan en forma de espiral alrededor del centro del vértice (vortex). Los campos de fuerza están disponibles para todos los tipos de objeto. Actualmente está disponible un campo basado en puntos (point-based field) con una pérdida (fall-off) esférica.

Los campos de fuerza pueden ser asignados a cualquier objeto. Las partículas serán entonces: atraídas si se usa un valor negativo de *Fuerza (Strength)*, o repelidos si se utiliza un valor positivo de *Fuerza (Strength)*. También hay un parámetro de *Pérdida (Fall-off)* que define cuanta fuerza se pierde de acuerdo a la distancia con respecto al origen del objeto.

Desvío (Deflection)

Permite configurar cualquier objeto de **mall**a como un desviador (deflector) de partículas. Así, las partículas rebotarán sobre la superficie de la malla. Se puede controlar cuanto rebotan las partículas mediante el valor *Damping*, hacer aleatoria la manera en que las partículas rebotan con *Rnd Damping*, y se puede definir el porcentaje de partículas que pasarán a través de la malla, usando el parámetro *Permeabilidad (Permeability)*.

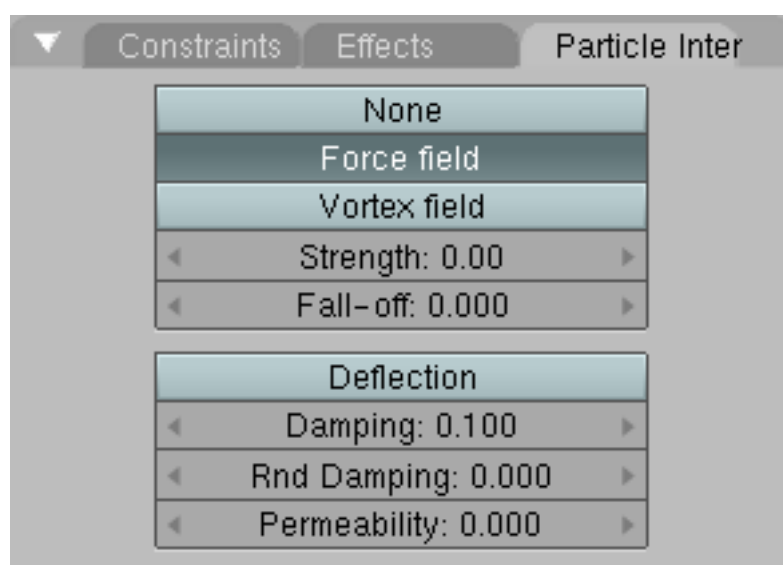
Note

Hay que asegurarse que las normales de la superficie de la malla miran en dirección a las partículas si se quiere obtener un desvío correcto.

20.3.2. La interfaz

La configuración de la interacción de partículas se controla mediante el panel *Interacción de Partículas* (*Particle Interaction*), dentro del contexto Objeto (F7).

Figure 20-34. Panel Interacción de Partículas (Particle Interaction panel).



Campo de Fuerza/Vértice (Force/Vortex field)

- *Fuerza (Strength)* - Fuerza del efecto del campo.
- *Pérdida (Fall-off)* - Cuanto disminuye la fuerza con respecto a la distancia.

Cuando se agrega un campo de fuerza/vértice (force/vortex) a un objeto, el objeto mostrará un pequeño gráfico indicando que existe una interacción de partículas conectada al objeto. Los campos de fuerza mostrarán unos pequeños círculos grises, y los campos de vértice mostrarán el dibujo de una espiral ([Figure 20-36](#)).

Figure 20-35. Indicador del Campo de Fuerza.

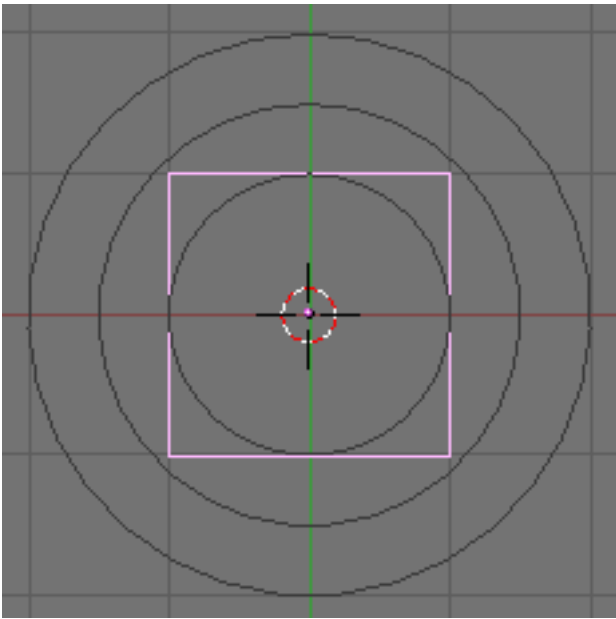
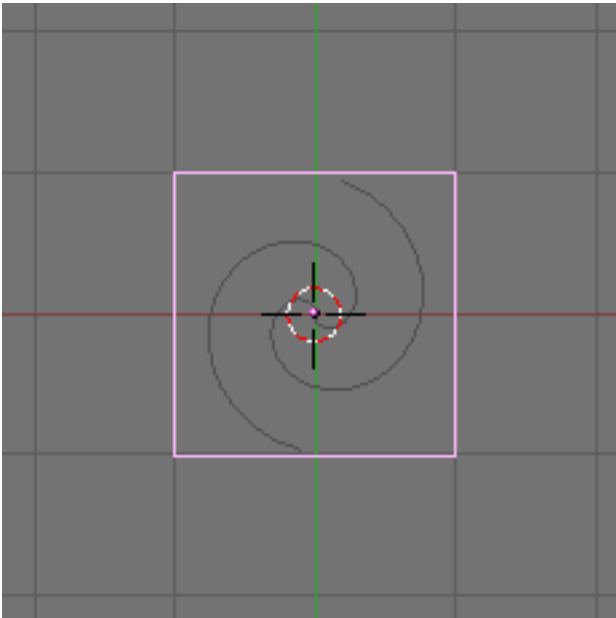


Figure 20-36. Indicador del Campo de Vértice.



La sección de abajo del panel en [Figure 20-34](#) maneja la configuración para desvíos (deflection).

Note Los desviadores (deflectors) no muestran indicadores gráficos como lo hacen los campos de fuerza.

Parámetros de Desvío

- *Damping* - Controla la cantidad de rebote que produce la superficie.

0.0 - Sin damping, las partículas rebotarán al máximo.

1.0 - Máximo damping, las partículas no rebotarán.

- *Rnd Damping* - Agrega el elemento aleatorio al rebote. Por ejemplo, con un valor de *Damping* de 1.0 y un valor de *Rnd Damping* de 0.5, el damping variará entre los valores 1.0 y 1.5.
- *Permeabilidad (Permeability)* - Porcentaje de partículas que atravesarán la malla.

0.0 - No pasa ninguna partícula.

1.0 - Todas las partículas pasan a través del desviador (deflector).

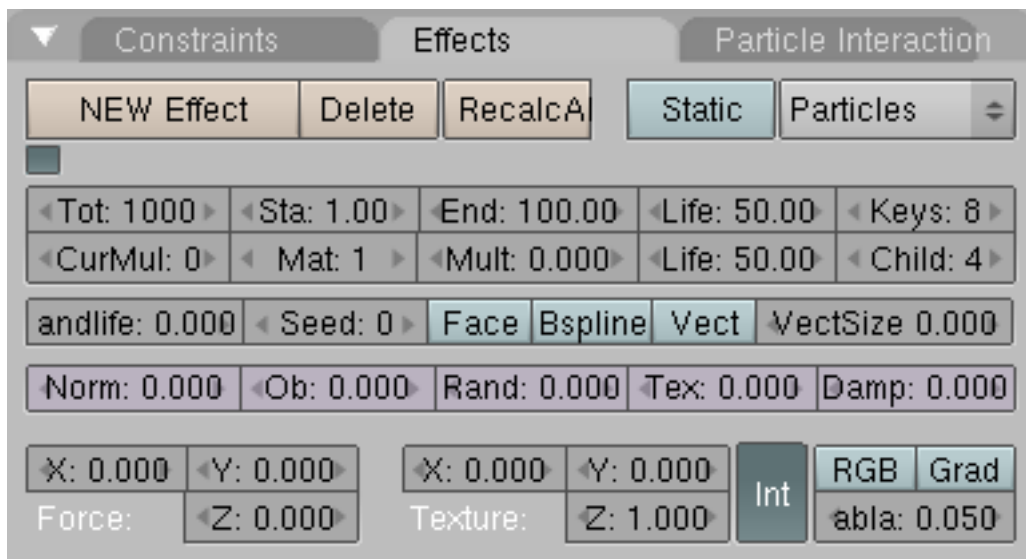
Si se configura un desviador de partículas, hay que asegurarse de tener suficientes cuadros clave para que Blender calcule las colisiones con suficiente detalle. Si se ven partículas moviéndose a través del desviador o rebotando de manera extraña, es probable que haya pocos cuadros clave, o las partículas o el desviador se está moviendo demasiado rápido.

Note Se pueden animar desviadores en movimiento, pero las partículas pueden atravesar la malla si el desviador se mueve demasiado rápido o si su malla es demasiado compleja. Esto puede resolverse parcialmente aumentando los *Cuadros Clave (Keys)* del emisor de partículas.

Después de hacer cualquier cambio en los parámetros, es necesario seleccionar el emisor de partículas, volver a la pestaña *Efectos (Effects)* y presionar el botón **RecalcA**. ([Figure 20-37](#)).

Más cuadros claves (keys) significa un tiempo mayor de cálculo y un mayor uso de memoria. Véase [Section 20.1](#) para conocer cómo configurar los emisores de partículas.

Figure 20-37. Panel de Efectos (Effects panel).



Se pueden insertar cuadros clave *Ipo* para todos los parámetros de los campos de fuerza y desviadores. Las correspondientes curvas *Ipo* se editan como *Object Ipo* en la ventana *Ipo*. Véase [Chapter 14](#) para más información sobre *Animación e Ipo*.

20.3.3. Ejemplo

Aquí hay un pequeño ejemplo que muestra el funcionamiento de los desviadores de partículas.

1. Comenzar con la escena por defecto, con el cubo.
2. En la vista de arriba (top), agregar una malla *Círculo (Circle)* y aceptar los 32 vértices por defecto. Presionar **TAB** para salir del Modo Edición. Cambiar a la vista lateral y mover el círculo 5 unidades hacia arriba.
3. Con el círculo seleccionado, presionar **S**, y escalarlo a 0.1 en todas las direcciones.
4. Ahora, presionar **F7** e ir a la pestaña *Efectos (Effects)*. Presionar **NUEVO Efecto (NEW Effect)** y elegir *Partículas (Particles)* de la *Lista (Popup-list)*. Debe verse lo mismo que en [Figure 20-37](#).
5. Incrementar la vida de la partícula (particle life) de 50 a 100 cambiando el campo *Vida (Life)* a **100**. En el campo de fuerza, en la sección más baja de la pestaña, asignar un valor de **-0.100** a *Force Z*. Si se presiona **ALT-A**, debería verse una animación de partículas moviéndose hacia abajo, en dirección al cubo. Ver [Figure 20-38](#). Hasta el momento, las partículas viajan en línea recta, atravesando el cubo.

Figure 20-38. Cuadro 40 del Ejemplo.



6. Seleccionar el cubo. Con el contexto *Efecto (Effect)* todavía activo, ir a la pestaña *Interacción de Partículas (Particle Interaction)* y configurar los parámetros del *Desviador* como se muestra en [Figure 20-39](#).

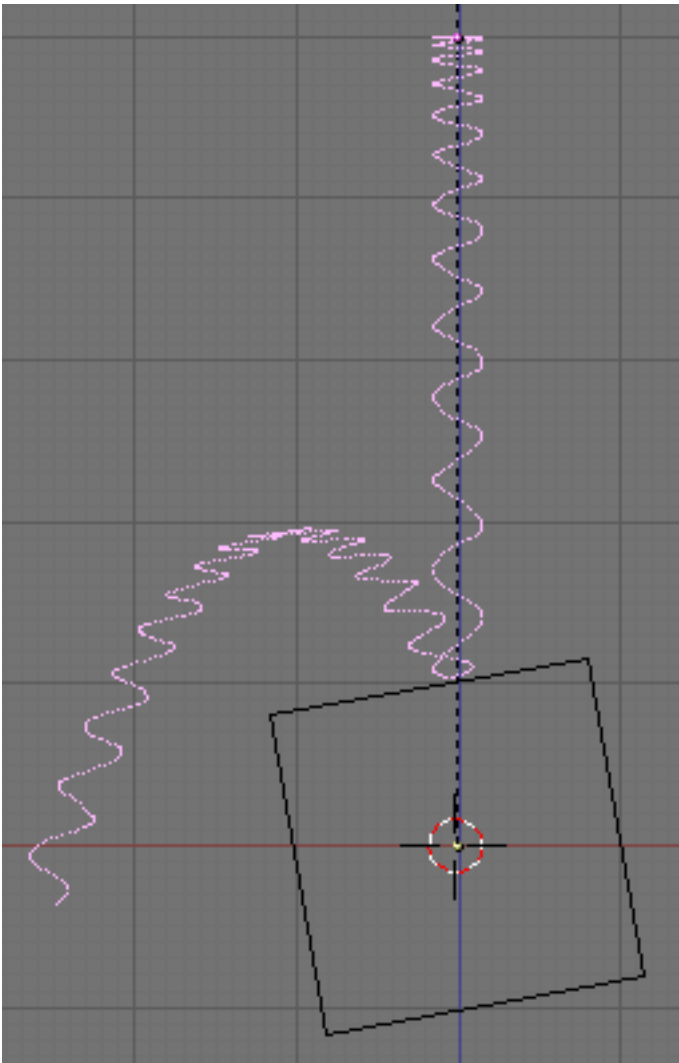
Figure 20-39. Ejemplo de Configuración de Desviación (Deflection).

Deflection	
<	Damping: 0.700 >
<	Rnd Damping: 0.000 >
<	Permeability: 0.000 >

7. Seleccionar el círculo y presionar **RecalcA** en la pestaña *Efectos (Effects)*. Al correr la animación, **ALT-A**, se verá que las partículas rebotan en el cubo hacia arriba y hacia abajo. El problema es que el rebote está un poco alejado del cubo. Esto se debe a que hay muy pocos *Cuadros Clave (Keys)*. En el campo de *Cuadros Clave (Keys)*, ingresar un valor de **50**, y pulsamos **RecalcA**. Al correr nuevamente la animación, las partículas rebotaran en el cubo, o muy cerca de él.

8. Seleccionar el cubo e inclinarlo 10 grados en cualquier dirección. Seleccionar el círculo y recalcular las partículas. Al correr nuevamente la animación, las partículas rebotan en la dirección en la que se giró el cubo ([Figure 20-40](#)).

Figure 20-40. Resultado final del ejemplo.



[Prev](#)

Partículas sencillas

[Home](#)

[Up](#)

[Next](#)

Otros Efectos

Chapter 21. Otros Efectos

21.1. Introducción

Relevante a Blender v2.31

Hay otras dos clases de efectos, además de las partículas, que pueden ser enlazados a un objeto, teoricamente para cuando se trabaja con animaciones pero en la practica sirve incluso para imagenes estáticas.

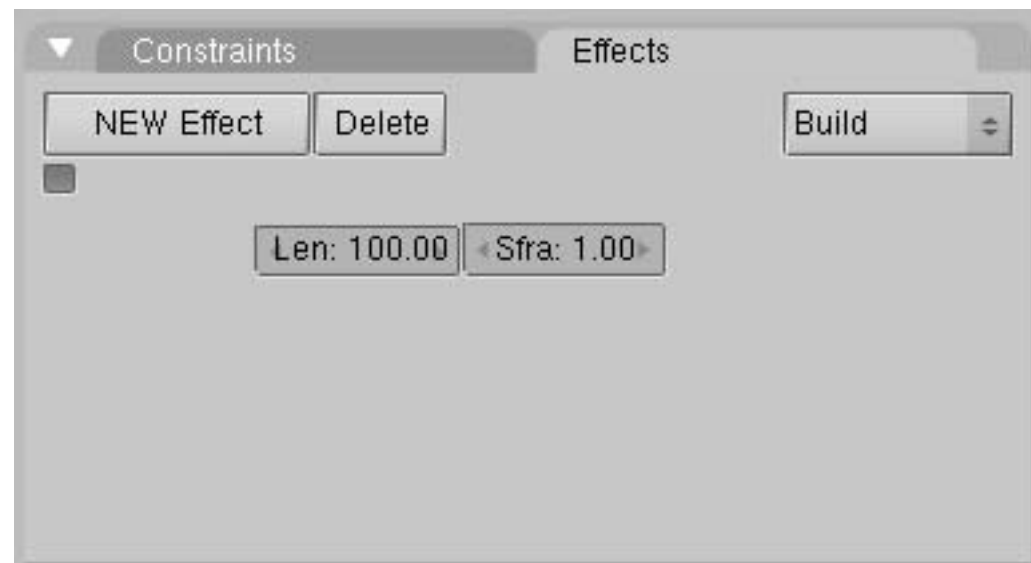
Esos dos efectos son *Build*, y *Wave*, las siguientes secciones describirán en detalle cada uno de ellos.

21.2. Efecto Build

Relativo a Blender v2.31

El efecto Build trabaja sobre mallas que hacen aparecer las caras del objeto, unas detras de otras, todo el tiempo. Si el material de la malla es tipo Halo, en vez del estandard, son los vertices de la malla, y no las caras, las que se solapan.

Figure 21-1. Efecto Build



Las caras, o vertices, aparecen en el orden en el que han sido almacenados en memoria. Este orden puede ser alterado seleccionando el objeto con la tecla **CTRL-F** fuera del modo edicion. Esto hace que las caras sean reordenadas en funcion de su propio valor (coordenada z) en la referencia local de la malla.

Tip Reordenando

Si crea un plano y añade el efecto Build para ver como funciona no quedara satisfecho. Primero, debe subdividirlo de forma que se creen algunas caras, no solo una. Entonces presionar la tecla **CTRL-F** no hara mucho, puesto que el eje Z esta ortogonal al plano. Debe rotar en el modo edicion para obtener alguna diferencia numerica entre las coordenadas de las caras, para poder ser capaz de reordenarlas.

El efecto Build solo tiene dos controles numericos ([Figure 21-1](#)):

Len - Define cuantos frames tomara el efecto Build.

Sfra - Define el frame inicial del proceso Build.

[Prev](#)

Otros Efectos

[Home](#)

[Up](#)

[Next](#)

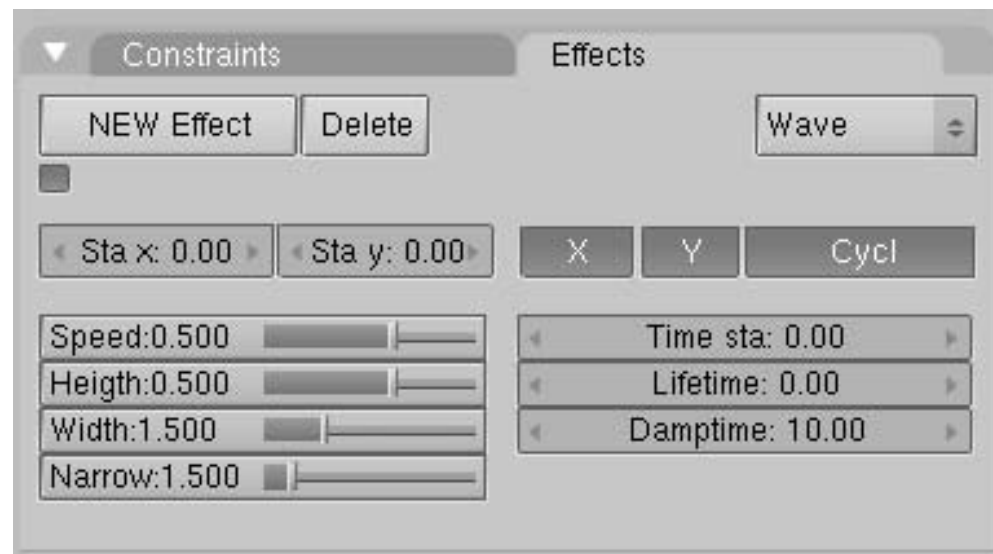
Efecto Onda

21.3. Efecto Onda

Relativo a Blender v2.31

El efecto onda añade movimiento a la coordenada Z del objeto malla.

Figure 21-2. Panel de control de Onda



La influencia del efecto onda viene generada por un punto de inicio dado, definido en los botones numericos *Sta X* y *Sta Y*. Estas coordenadas son de la referencia local de la malla ([Figure 21-3](#)).

Figure 21-3. Origen de la Onda



La deformacion del efecto onda se origina en el punto de inicio dado y se propaga a lo largo de la malla con frentes de onda circulares, o con frentes de onda rectilíneos paralelos a los ejes X e Y. Esto esta controlado por los botones de seleccion X y Y. Si solo uno de los botones esta presionado es lineal, si lo estan los dos, sera circular ([Figure 21-4](#)).

La onda por si misma es como una perturbacion gaussiana que puede ser, o un pulso unico o una serie de perturbaciones (tren de ondas), segun si el boton *Cycl* esta presionado.

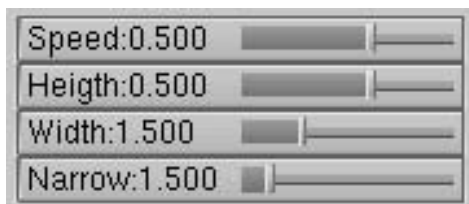
Figure 21-4. Tipo de frente de ondas



La onda esta gobernada por dos series de controles, el primero define la forma de onda, y la segunda la duracion del efecto.

Para lo que concierne a la forma de onda, los controles son *Speed*, *Height*, *Width* y *Narrow* ([Figure 21-5](#)).

Figure 21-5. Controles de frente de onda



El control numerico *Speed* define la velocidad, en unidades por frame, de la perturbacion.

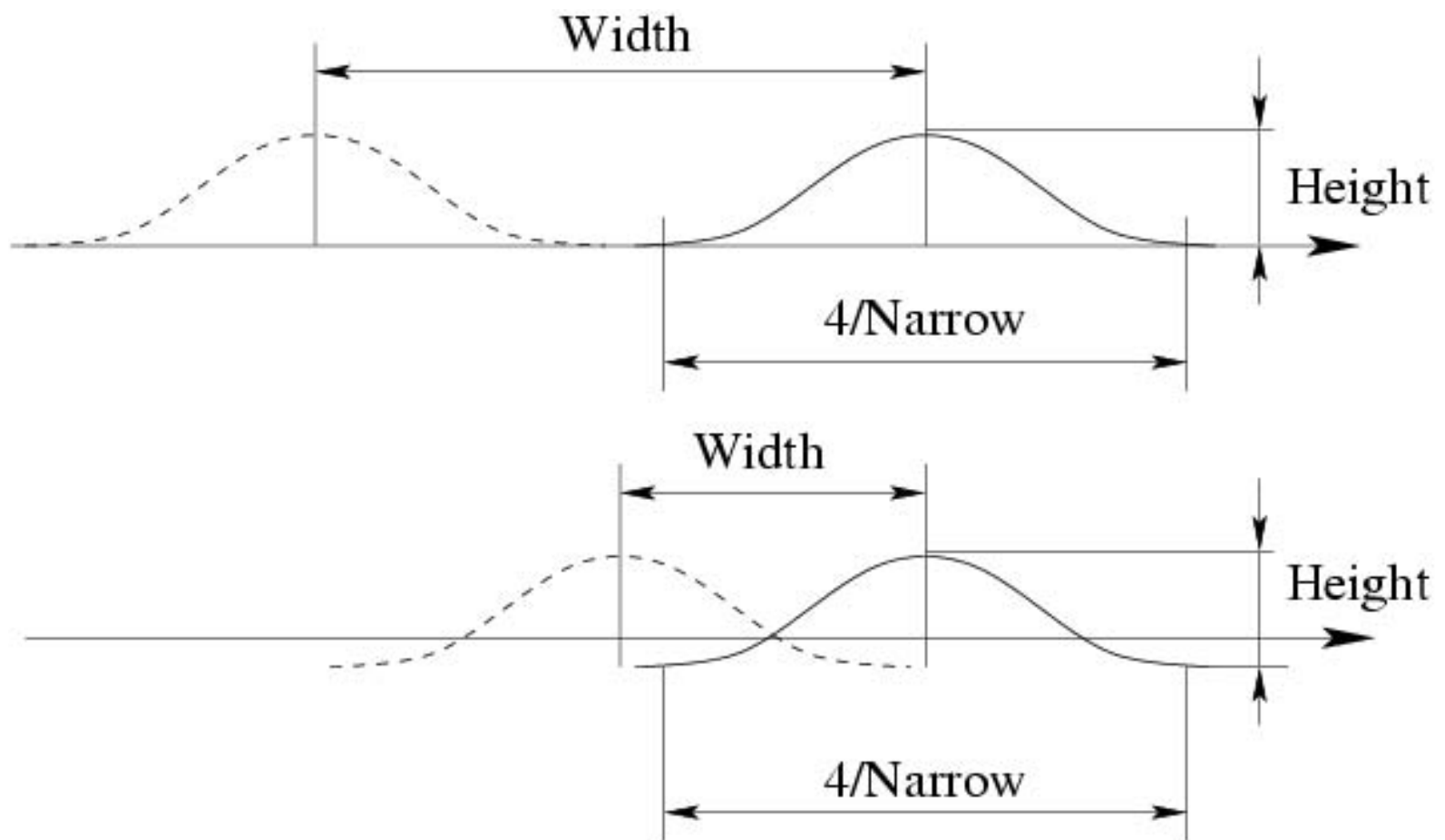
El control numerico *Height* define la altura, en unidades de Blender sobre el eje Z, de la perturbacion ([Figure 21-6](#)).

Si el boton *Cycl* esta presionado, el boton numerico *Width* representa la distancia, en unidades de Blender, entre la mayor distancia de dos perturbaciones seguidas, y el el efecto total de la onda esta dado por la envoltura de todos los pulsos unicos ([Figure 21-6](#)).

Esto tiene un efecto indirecto en la amplitud de la perturbacion. Siendo las perturbaciones de forma Gaussiana, si los pulsos estan muy cerca unos de otros, la envoltura no podria alcanzar el punto $z=0$ en ningun momento. Si esto es el caso, Blender en realidad reduce toda la onda de forma que el minimo sea cero y, consecuentemente, el maximo es menor que el valor de la amplitud esperada, como se ve al final de [Figure 21-6](#) .

Al ancho real de cada pulso de estilo Gaussiano esta controlado por el boton numerico *Narrow*, cuanto mayor sea el valor mas se estrechara el pulso. El ancho real del area en la que un pulso unico sea considerablemente diferente de cero viene dado, en unidades de blender por 4 dividido entre el valor de *Narrow*. Esto es, si *Narrow* es 1, el pulso tiene 4 unidades de ancho, y si *Narrow* es 4 el pulso tiene 1 unidades de ancho.

Figure 21-6. Caracteristicas del frente de onda



Tip Para obtener una onda estilo sinusoidal

Para obtener un bonito efecto de onda similar a las olas del mar y parecido a una onda sinusoidal es necesario que la distancia entre perturbaciones contiguas y el ancho de la perturbacion sea el mismo, esto es que el boton numerico *Width* debe ser igual a 4 entre el valor de *Narrow*.

Los ultimos controles de onda son los controles de tiempo. Los tres botones numericos definen:

Time sta El frame en el que la onda comienza;

Lifetime el numero de frames que dura el efecto;

Damptime es el numero adicional de frames en los que la onda poco a poco se va reduciendo en amplitud hasta llegar a cero. La reduccion se aplica a todas las perturbaciones y empieza en el primer frame despues de llegar al valor de *Lifetime*. Las perturbaciones desaparecen al llegar al valor en frames de *Damptime*.

Figure 21-7. Controles de tiempos de onda

Time sta: 0.00
Lifetime: 0.00
Damptime: 10.00

[Prev](#)

Efecto Build

[Home](#)

[Up](#)

[Next](#)

Tecnicas especiales de modelado

Chapter 22. Técnicas especiales de modelado

Claudio "malefico" Andaur

22.1. Introducción

Relevant to Blender v2.31

Una vez que hemos superado la "fiebre de modelado por extrusión" y que hemos empezado a buscar desafíos de modelado más interesantes, quizás empecemos la búsqueda de métodos alternativos para realizar la tarea. Existe un grupo de técnicas especiales de modelado en Blender que no solo facilitan nuestro trabajo sino que a veces de no ser por ellas no sería *posible* realizarlo.

Estas así llamadas técnicas "especiales" de modelado involucran no solo la manipulación de vértices sino el uso de procedimientos no intuitivos que requieren de una experiencia o conocimiento más profundo que del novato promedio.

En este capítulo describiremos estas técnicas en detalle y explicaremos su utilidad en varios casos de modelado que no podrían haberse resuelto de otra manera.

22.2. DupliVerts

Relevante a Blender v2.31

"DupliVerts" no es una banda de rock ni un apalabra holandesa para algo ilegal (bueno, quizás sí lo sea) sino una contracción para "DUPLIcado en VERTiceS", lo que significa la duplicación de un objeto base en la posición de los vértices de una malla (o incluso un sistema de partículas). En otras palabras, cuando usamos Dupliverts sobre una malla, una instancia del objeto base es ubicada en cada vértice de la malla.

Existen de hecho dos enfoques diferentes para modelar usando DupliVerts. Se los puede usar como una herramienta para ubicar objetos, permitiendonos modelar arreglos geométricos de objetos (por ej: las columnas de un templo griego, los árboles de un jardín, un ejércitos de soldados robots, los pupitres de un aula). El objeto puede ser de cualquier tipo de objeto soportado por Blender.

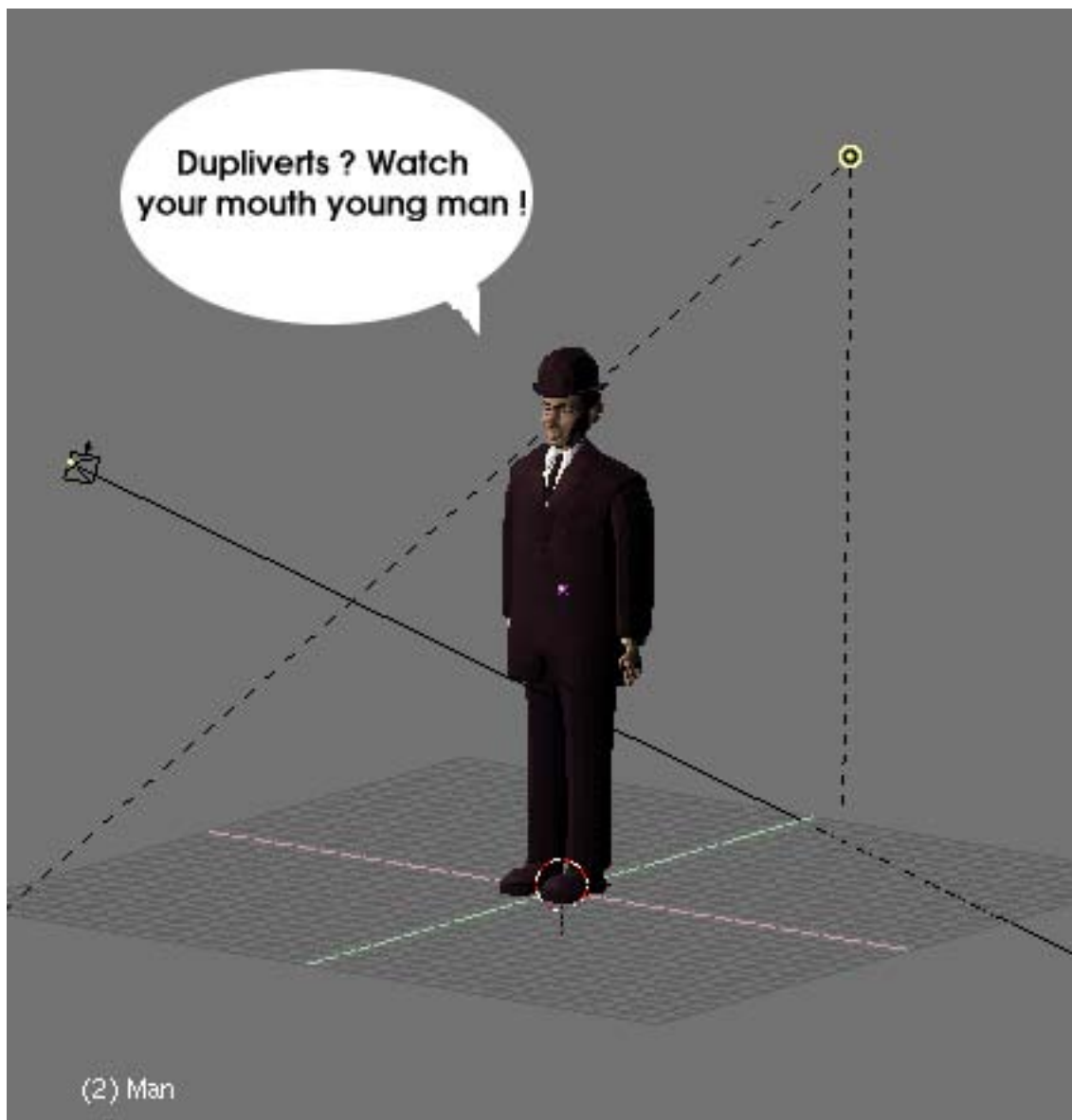
El segundo enfoque se usa para modelar objetos a partir de una parte del mismo (por ej: las púas de un puercoespín, los mosaicos de un piso, los pétalos de una flor).

22.2.1. DupliVerts as an Arranging Tool

Todo lo que necesitaos es un objeto base (ej: el "árbol" o la "columna") y una malla con sus vértices siguiendo el patrón que tenemos en mente.

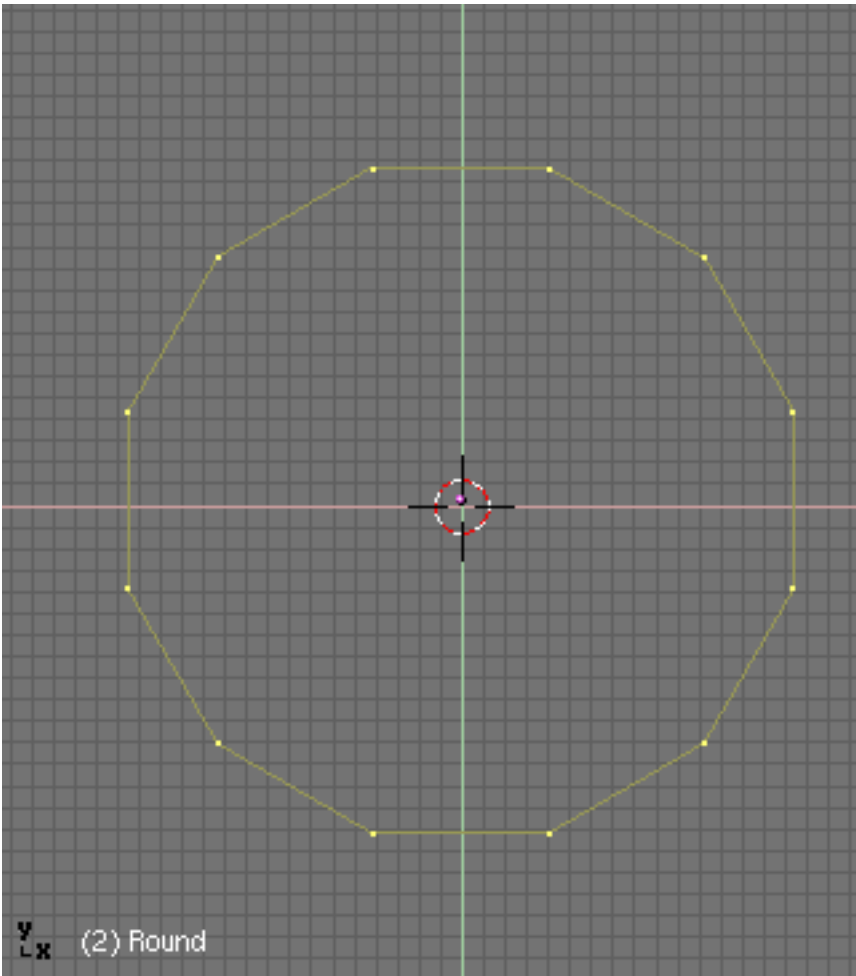
Para lo que sigue usará una escena simple. Consiste en un acámara, unas luces, un plano (para el piso) y un extraño hombrecillo que modelé basado en un personaje famoso de Magritte ([Figure 22-1](#)). Si el surrealismo no te apetece, encontrarás esta parte muy aburrida.

Figure 22-1. A simple scene to play with.



El hombre será mi "Objeto base". Es una buena idea que este se encuentre en el centro de coordenadas y que todas las rotaciones que tenga hayan sido borradas. Moveremos el cursor al centro del objeto base, y desde la vista superior agregaremos una malla Circulo con 12 vértices más o menos. ([Figure 22-2](#)).

Figure 22-2. The parent mesh can be any primitive.



Fuera de Modo Edición, seleccionaremos el objeto base, y añadiremos el círculo a la selección (el orden es importante aquí). Emparentaremos el objeto base al círculo usando **CTRL-P**. Ahora el círculo es el "padre" del personaje ([Figure 22-3](#)). Casi estamos.

Figure 22-3. The man is parented to the circle.

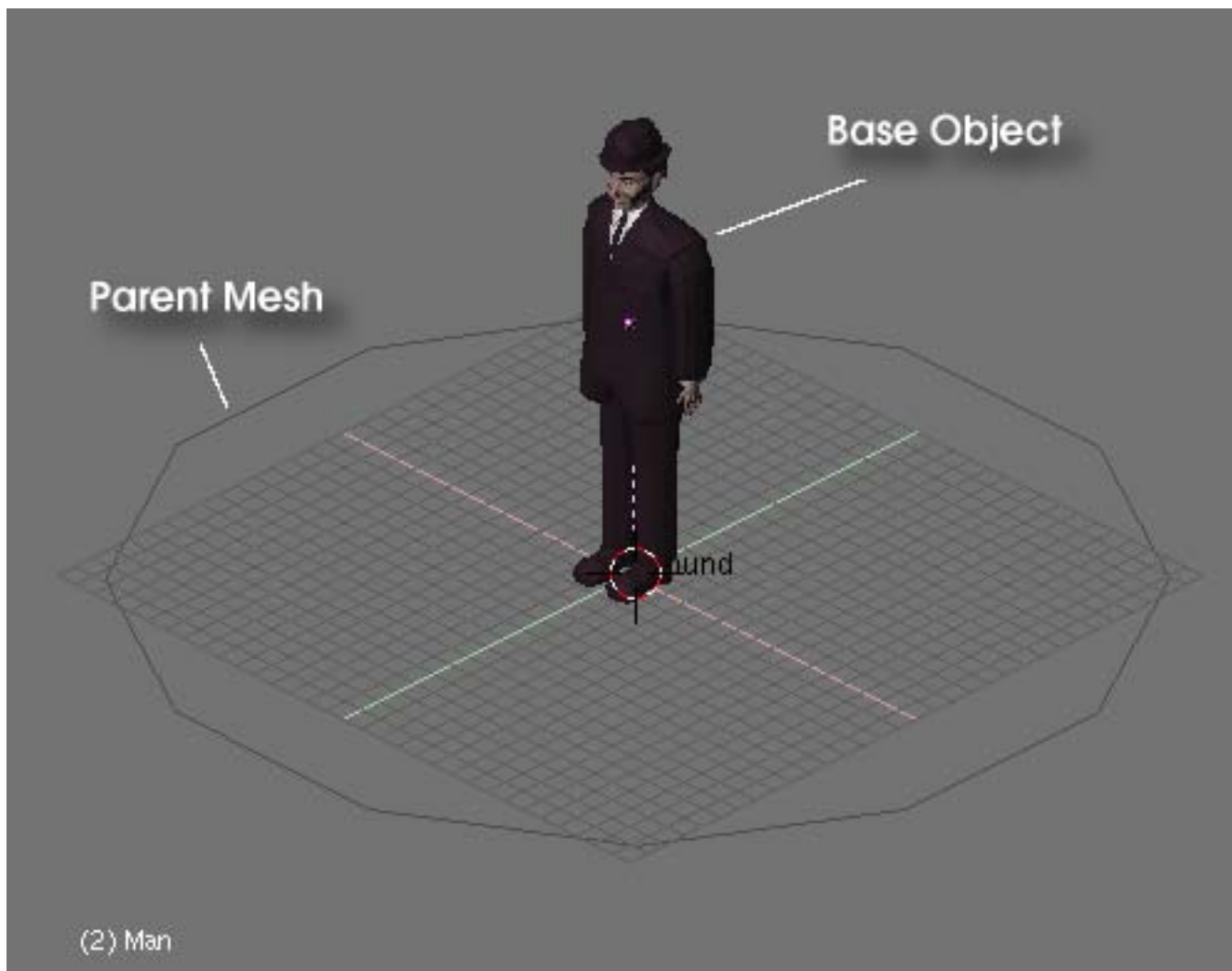
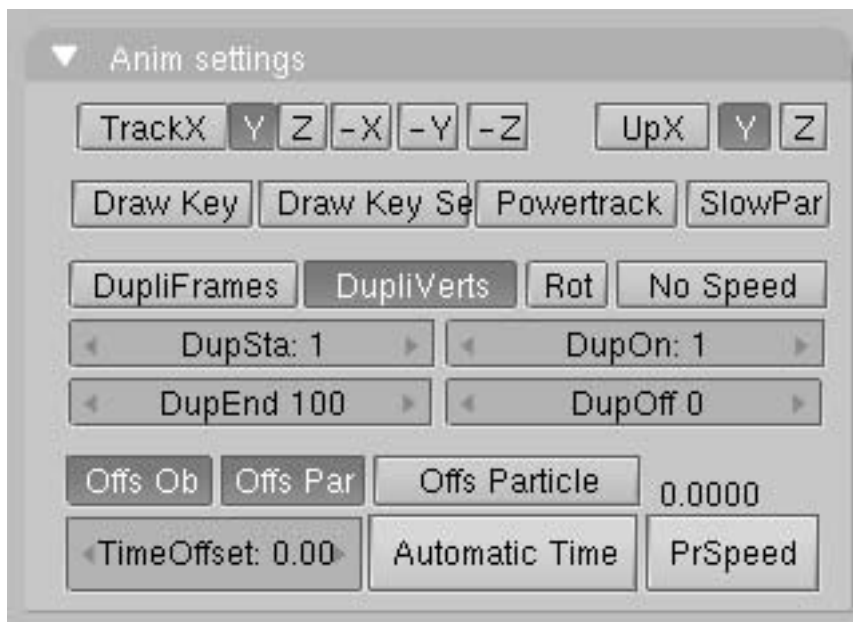


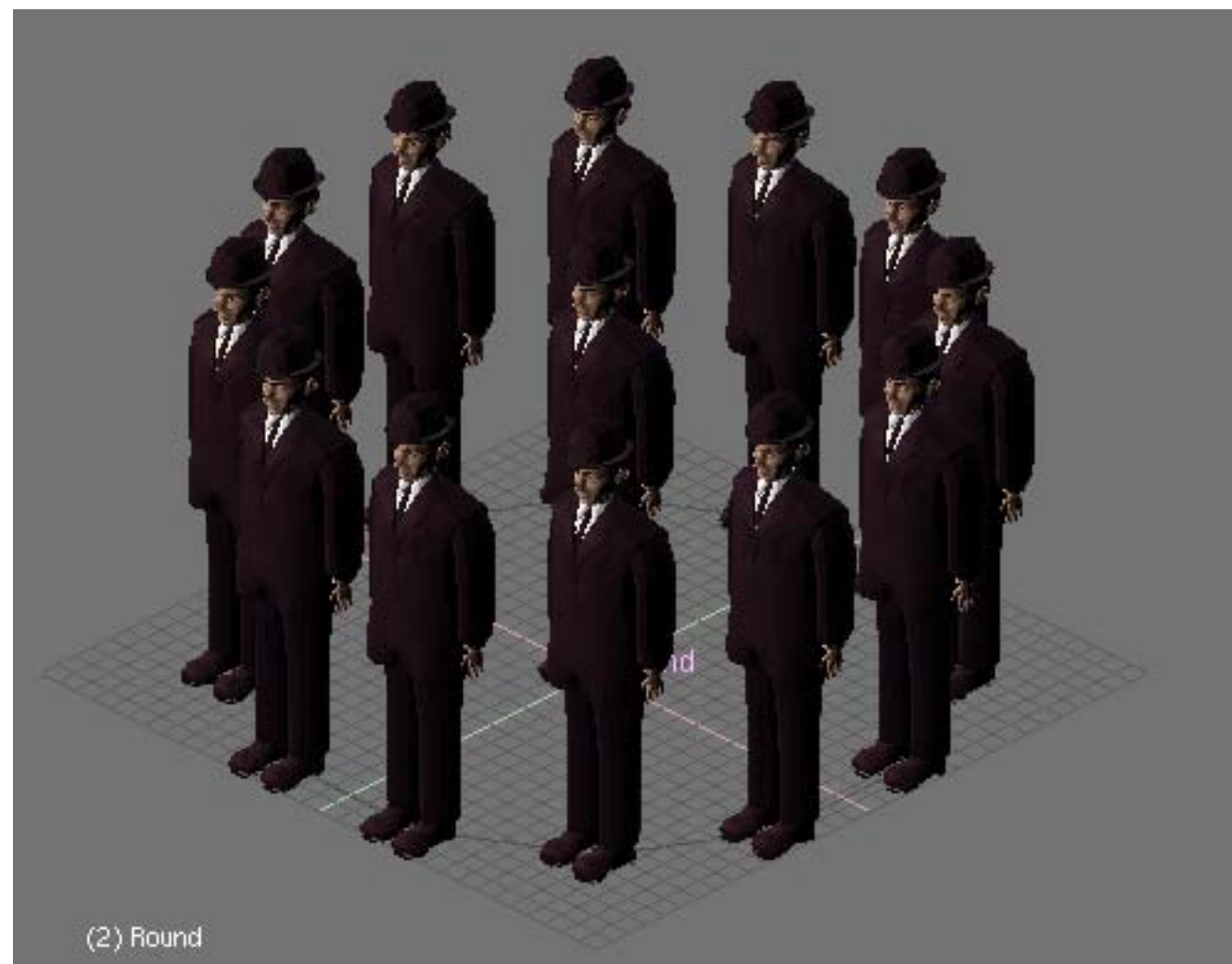
Figure 22-4. The Animation Buttons



Seleccionaremos ahora el círculo, cambiaremos la ventana de botones al Contexto Objeto (via  o

con **F7**) y pulsaremos el botón *DupliVerts* en el panel *Anim Settings* ([Figure 22-4](#)).

Figure 22-5. In every vertex of the circle a man is placed.



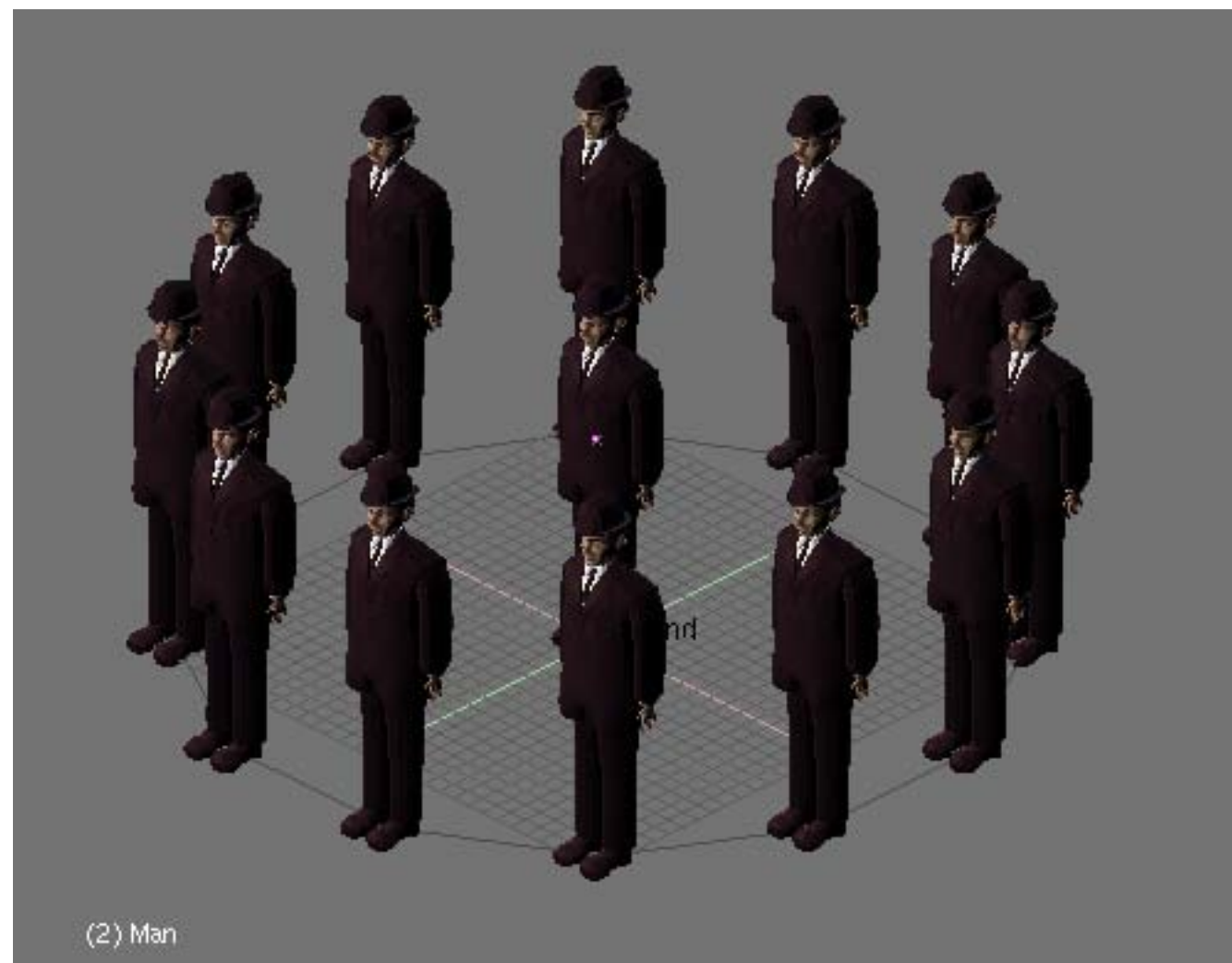
Wow, ¿no es fabuloso ? No te preocupes por el objeto que aparece en el centro del círculo ([Figure 22-5](#)). Aunque aparezca en las vistas 3D *no* será renderizado. Podemos seleccionar ahora el objeto base y modificarlo (escalarlo, rotarlo, en Modo Edición)[[1](#)] y estos cambios se verán en todos los objetos duplicados. Aún más interesante es observar que también podemos editar el círculo padre.

Note El objeto base no será renderizado si aplicamos dupliverts sobre una malla pero *sí* lo será si aplicamos dupliverts sobre un sistema de partículas!

Seleccionemos el círculo y escalémoslo. Podremos ver que los hombres misteriosos son escalados uniformemente con él. Entremos en Modo Edición para el círculo, seleccionemos todos los vértices **AKKEY** y agrandémoslo hasta el triple. Salgamos de Modo Edición y los objetos duplicados se actualizarán ([Figure 22-6](#)). Esta vez mantendrán su tamaño original pero la distancia entre ellos habrá cambiado. No solamente podemos escalar en Modo Edición, también podemos borrar o agregar vertices

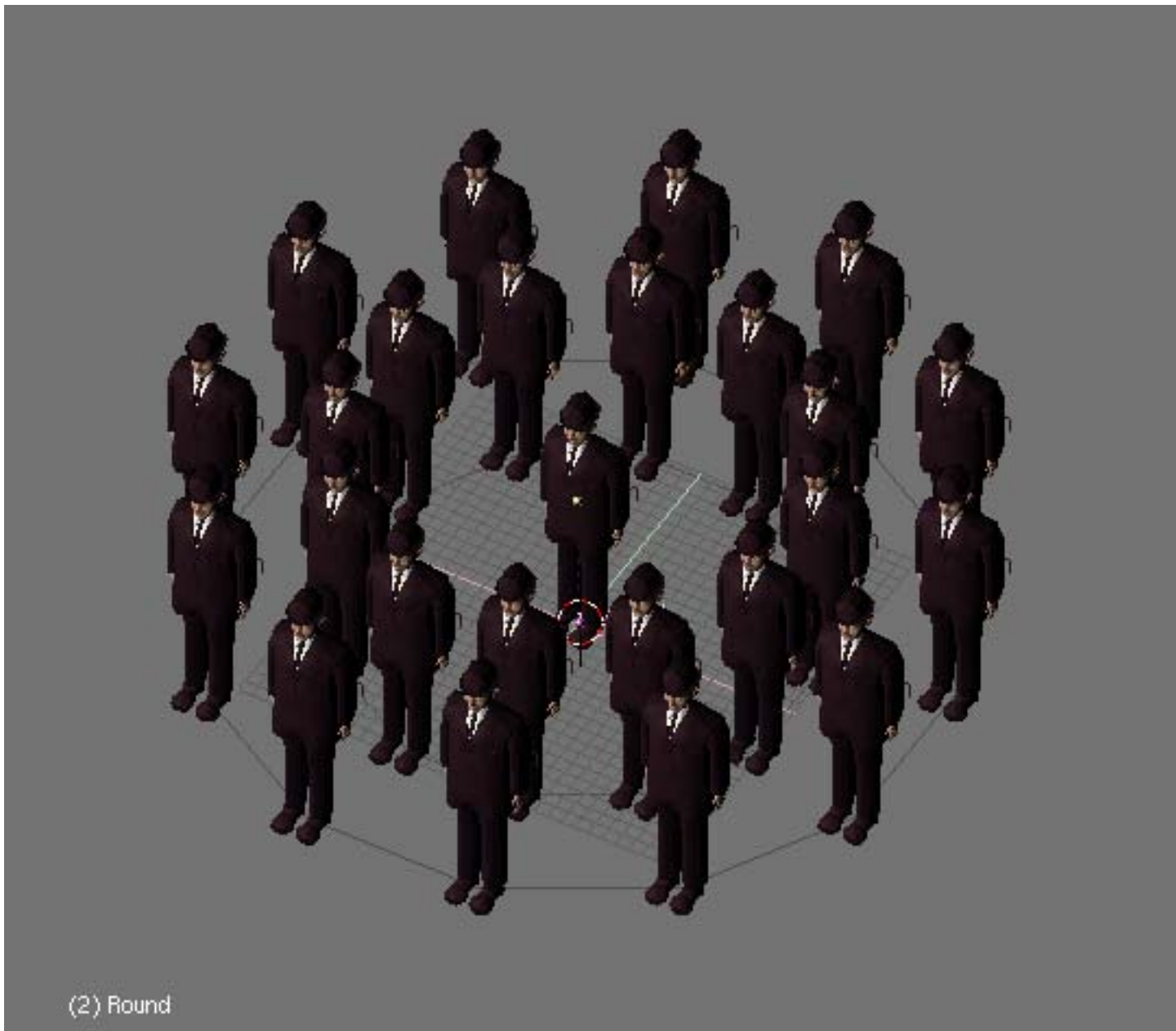
para modificar la disposición de los hombrecitos.

Figure 22-6. Changing the size of the circle in Edit Mode.



Seleccionemos todos los vertices en Modo Edición y dupliquémoslos (**SHIFT-D**). Ahora escalemos los nuevos vertices hacia afuera para obtener un segundo círculo alrededor del original. Salgamos de Modo Edición, y aparecerá un segundo círculo de hombres ([Figure 22-7](#)).

Figure 22-7. A second row of Magritte's men.

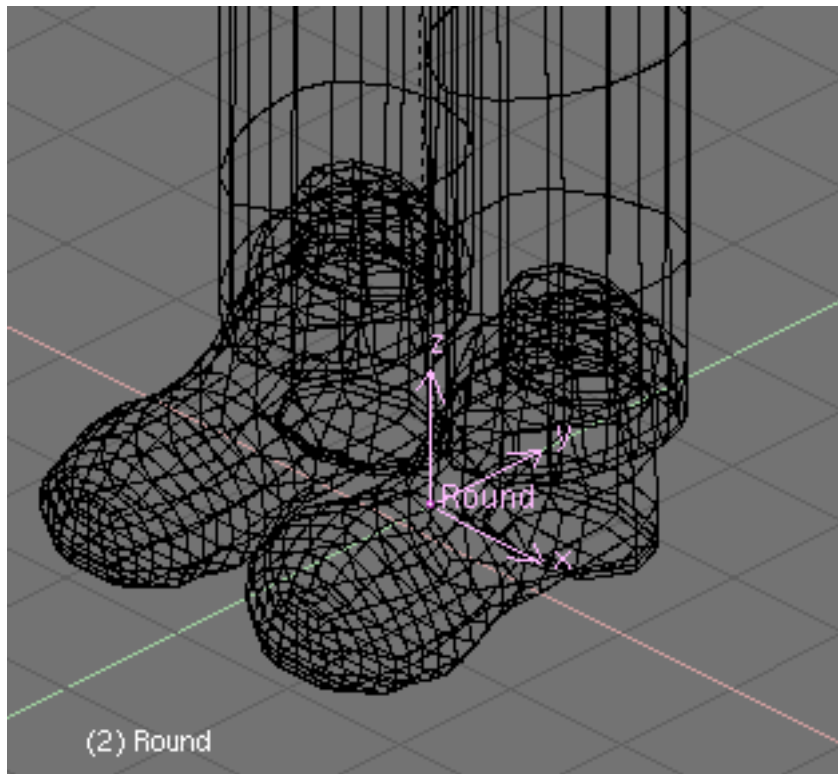


Hasta ahora, todos los hombrecitos Magritte están enfrentados hacia la cámara, ignorándose entre sí. Podemos obtener resultados más interesantes usando el botón *Rotj* junto al botón *DupliVerts* en el panel *Anim Settings*. Con este botón activado, podemos rotar los objetos duplicados de acuerdo a las normales del objeto padre. Más precisamente, los ejes de los objetos duplicados serán alineados con la normal en la posición del vértice

Cual es el eje alineado (X, Y o Z) con la normal de la malla padre, depende de lo indicado en los botones *TrackX*, *Y*, *Z* y *UpX*, *Y*, *Z* en el panel *Anim Settings*. Probando esto con nuestros amigos surreales puede llevarnos a resultados rarísimos dependiendo de la configuración elegida.

La mejor manera de entender qué ocurrirá es primero que nada alinear los ejes de los objetos "padre" y "base" con los ejes del espacio 3D. Esto lo podemos hacer seleccionando ambos objetos y pulsando **CTRL-A**, y haciendo click sobre *Apply Size/Rot?*.

Figure 22-8. Show object's axis to get what you want.



Luego haciendo visibles los ejes del objeto base y los ejes y normales del objeto "padre". ([Figure 22-8](#) - en este caso, siendo un círculo sin caras, se debe deifinir una cara primeramente para que la normal sea visible - de hecho para que la normal exista)

Ahora seleccionaremos el objeto base (nuestro hombrecito Magritte) y jugaremos un poco con los botones de Tracking. Observar los distitnos alineamientos de los ejes con las sucesivas combinaciones de *UpX*, *Y*, *Z* y *TrackX*, *Y*, *Z* ([Figure 22-9](#), [Figure 22-10](#), [Figure 22-11](#), [Figure 22-12](#)).

Figure 22-9. Negative Y Axis is aligned to vertex normal (pointing to the circle's center)

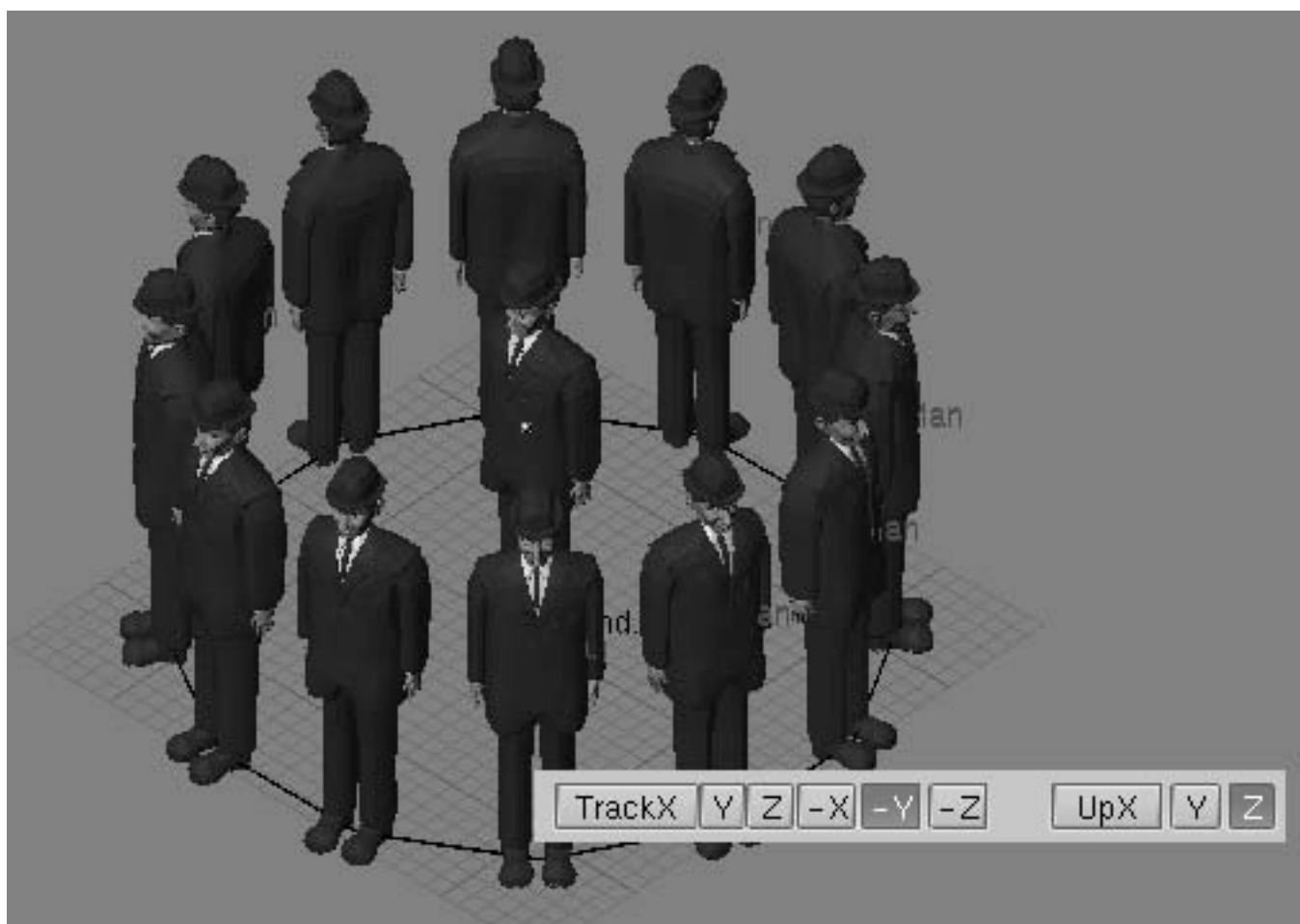


Figure 22-10. Positive Y axis is aligned to normal

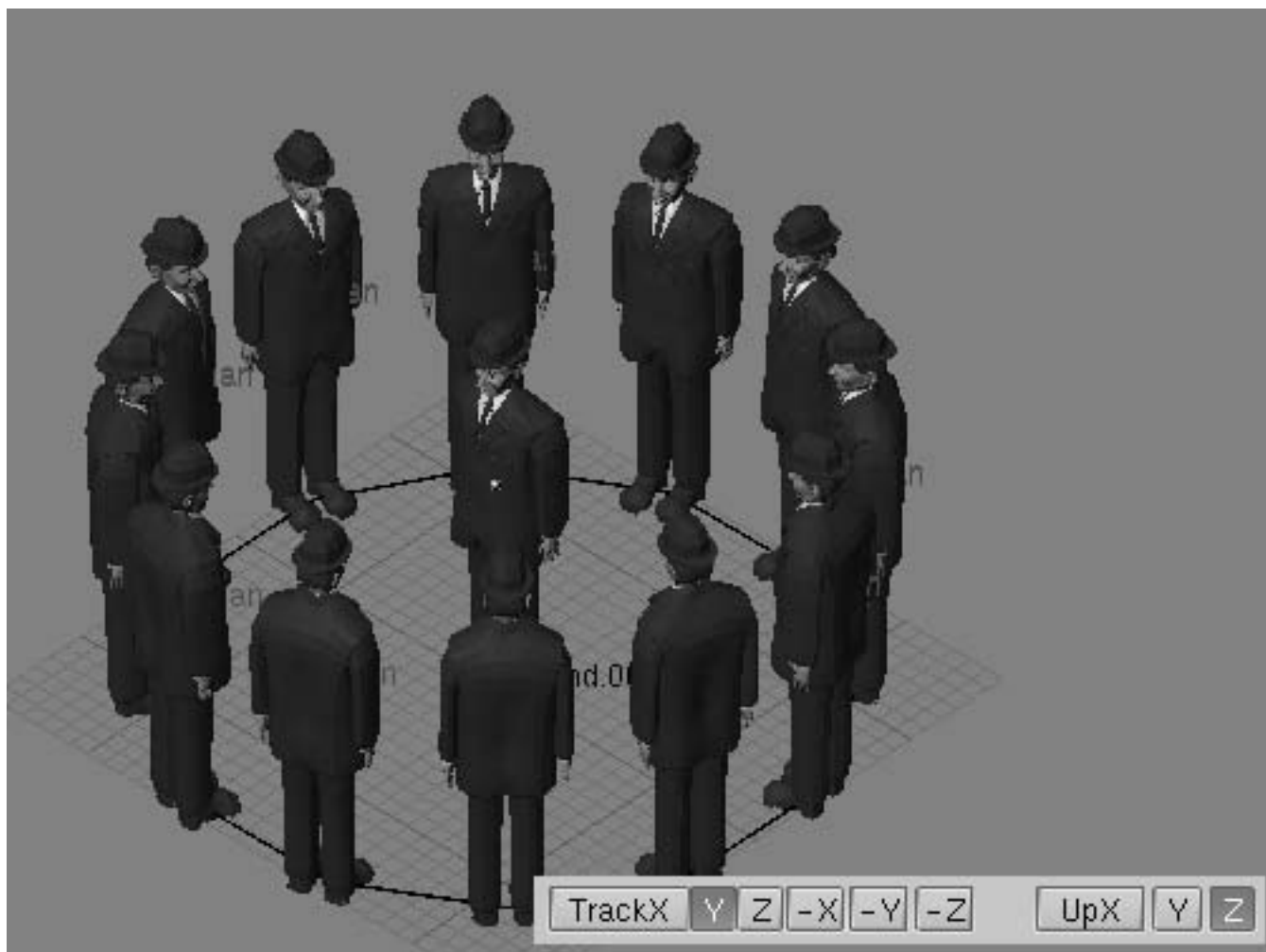


Figure 22-11. Positive X axis is aligned to normal

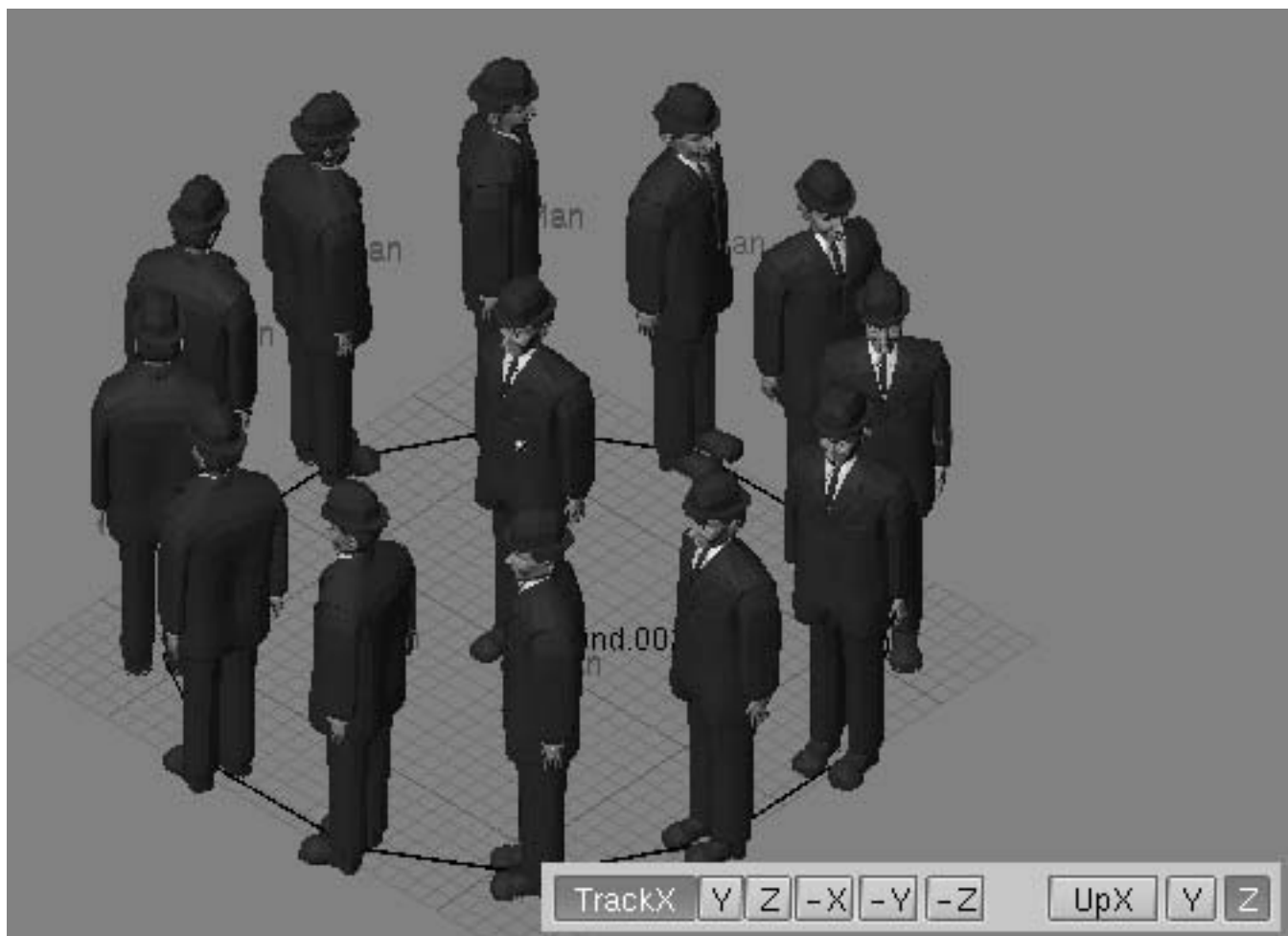
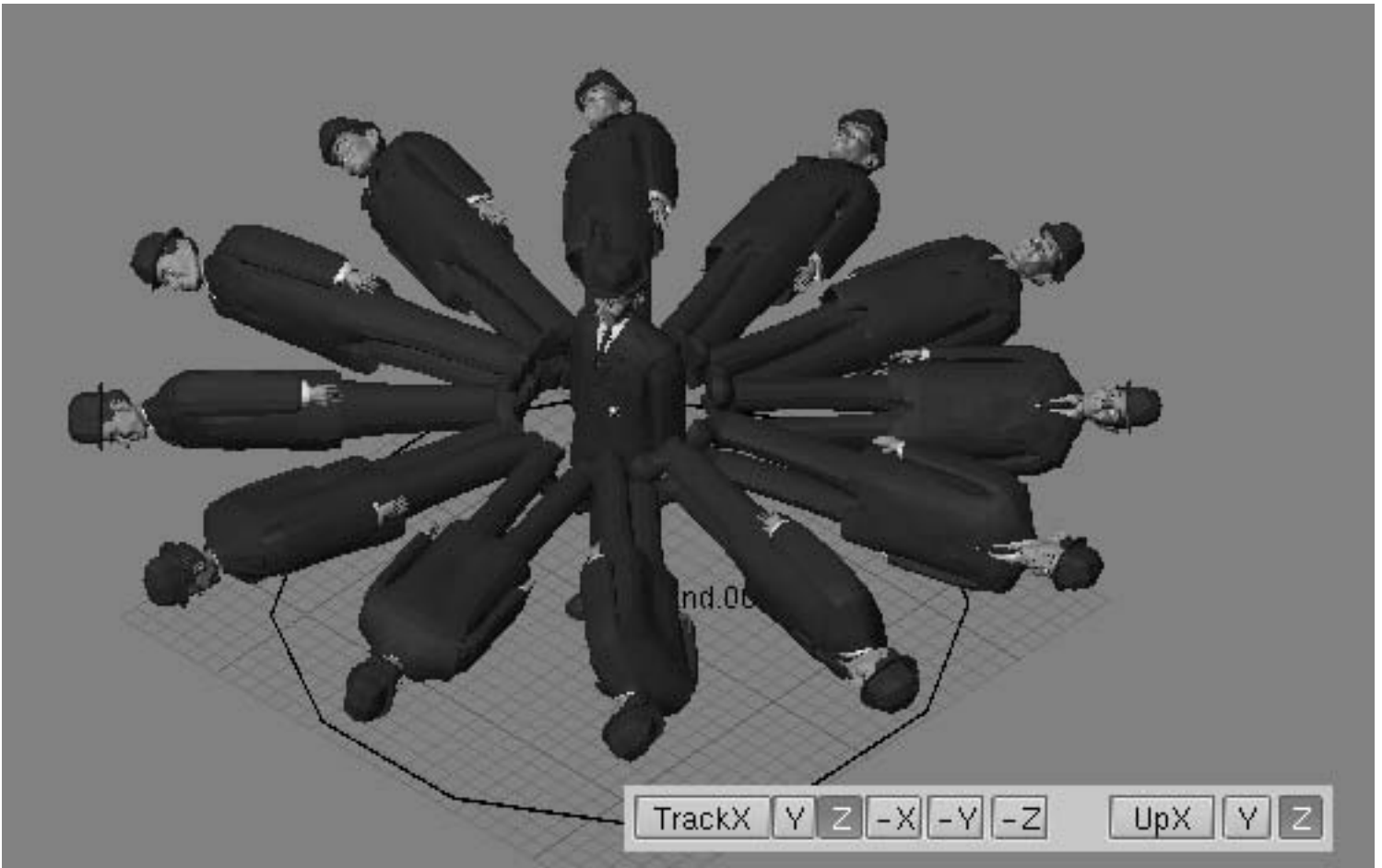


Figure 22-12. Positive Z axis is aligned to normal (weird, huh?)

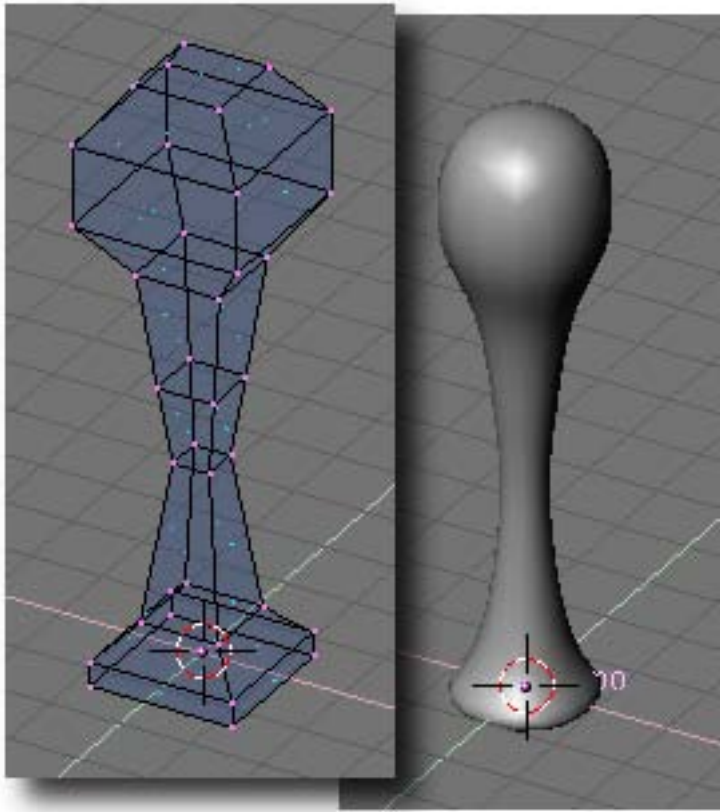


22.2.2. DupliVerts para modelar un objeto

Se pueden modelar objetos muy interesantes usando DupliVerts y una primitiva estándar.

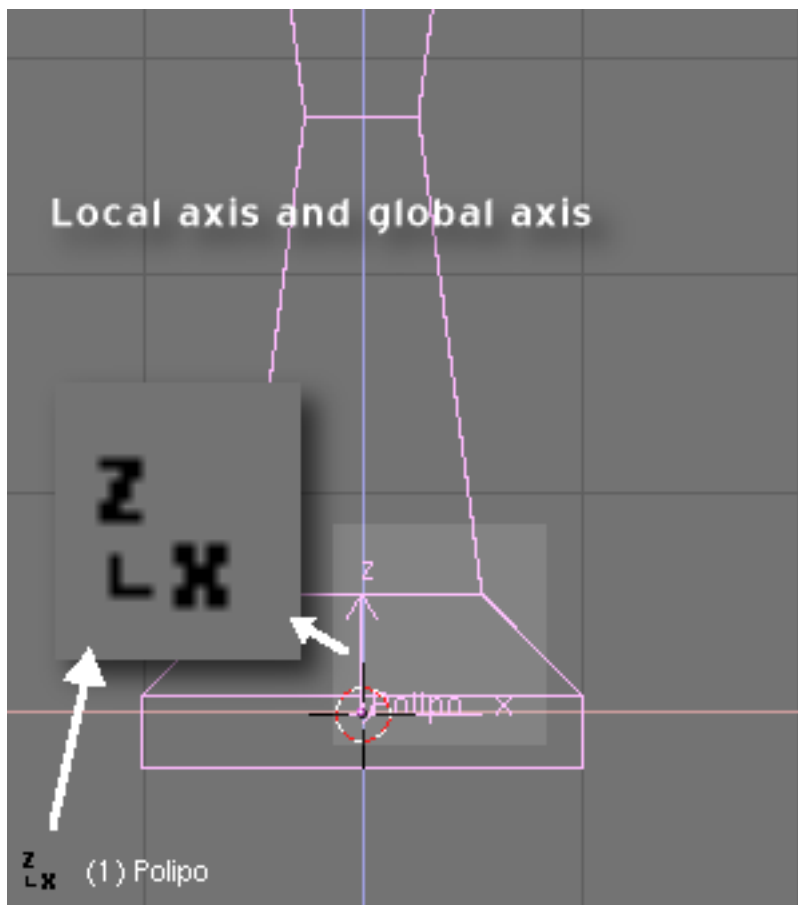
Comenzando con un cubo en la Vista Frontal y extrudando un par de veces he modelado algo que se parece a un tentáculo cuando activamos Subsurf ([Figure 22-13](#)). Después agregué una Icoesfera con 2 subdivisiones.

Figure 22-13. Strange tentacle and SubSurfed version.



Tuve especial cuidado en asegurarme que el tentáculo estuviese en el centro de la esfera y que ambos objetos tuvieran alineados sus ejes con los del espacio 3D como dice líneas arriba ([Figure 22-14](#)).

Figure 22-14. Local reference of the tentacle.



Luego simplemente hago que la icoesfera sea "padre" del tentáculo. Selecciono la icoesfera solamente y activo *DupliVert* en el panel de *Anim Settings* ([Figure 22-15](#)).

Presionemos también el botón *Rot* para hacer girar los tentáculos ([Figure 22-16](#)).

Figure 22-15. DupliVerts not rotated.

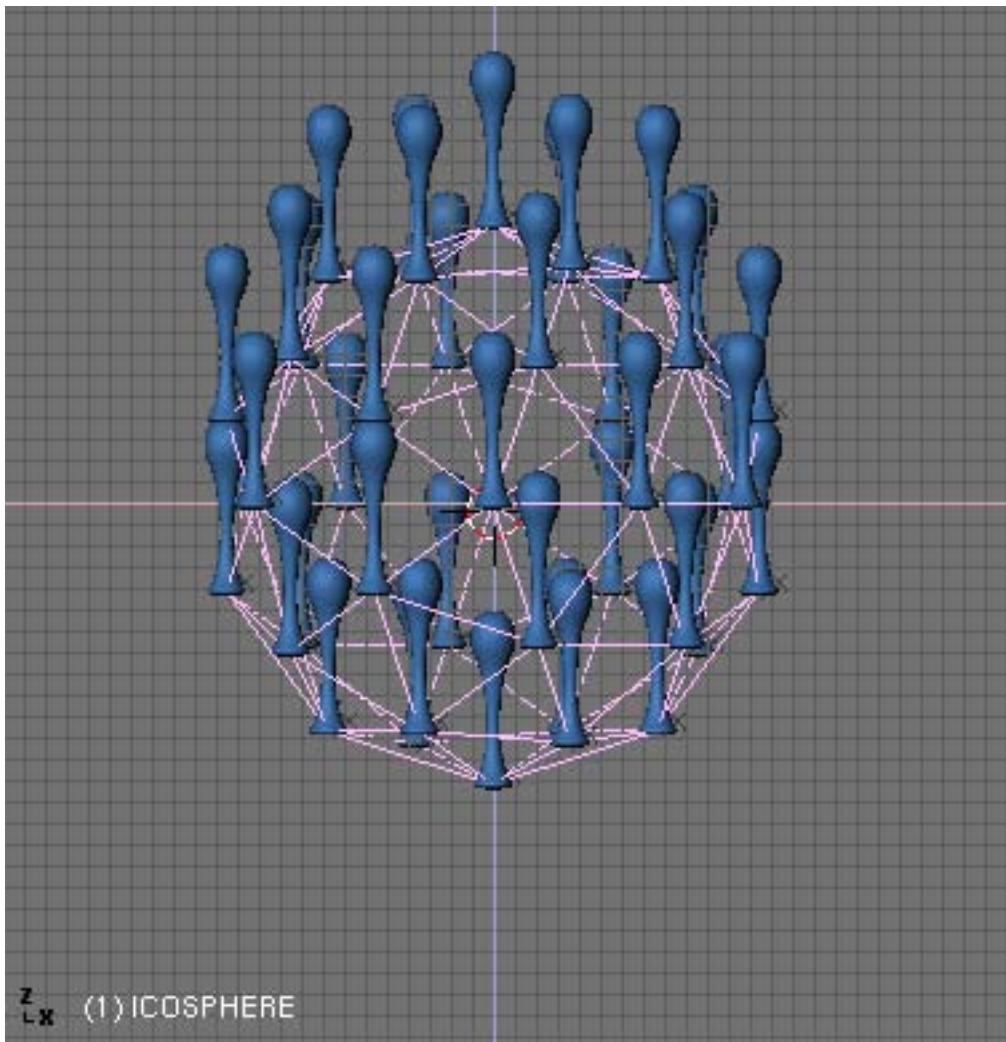
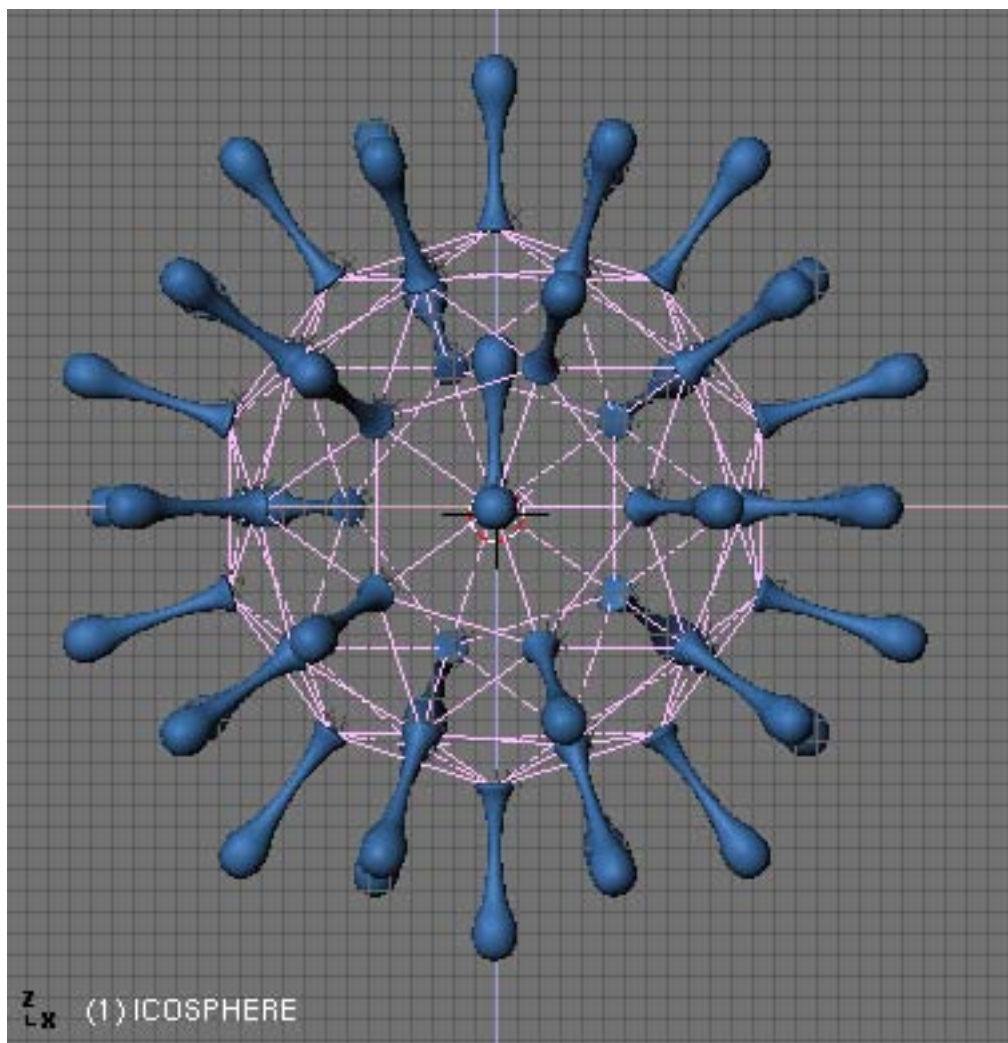


Figure 22-16. DupliVerts rotated.

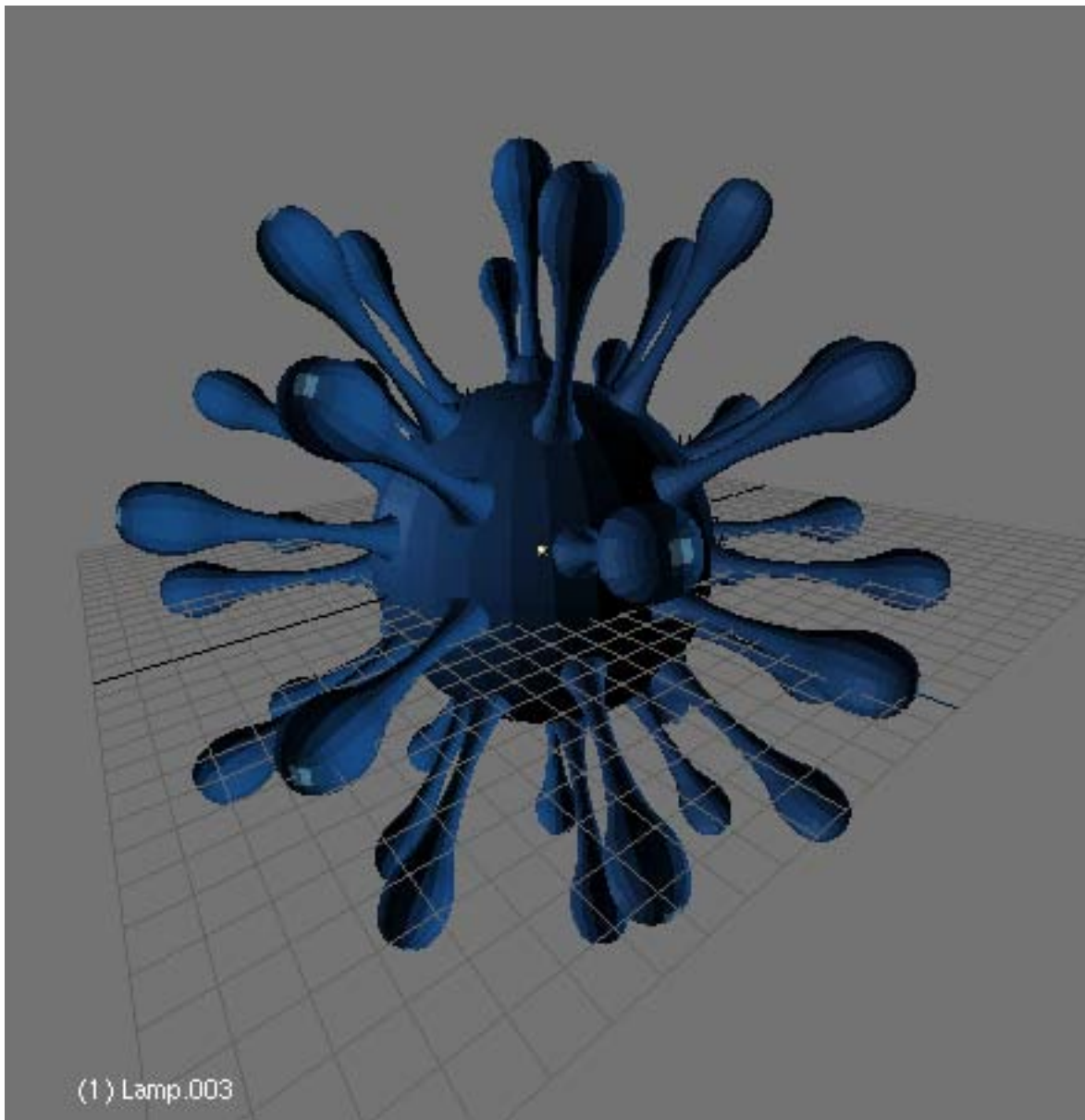


Para hacer que los tentáculos apunten hacia fuera, deberemos una vez más, observar cuidadosamente hacia dónde apuntan los ejes del objeto. Cuando aplicamos *Rot*, Blender intentará alinear uno de los ejes del tentáculo con el vector normal en el vértice de la malla padre.

En el caso de los homrecitos Magritte no le dimos importancia al círculo padre, pero aquí deberemos ocuparnos de la esfera, la que pronto descubrirás que *no* es renderizada. Quizás quieras añadir una esfera extra renderizable para completar el modelo.

Podemos experimentar en Modo Edición con el tentáculo, moviendo sus vertices apartándolos del centro de la esfera, pero los centros de los objetos deberán permanecer siempre en el centro de la esfera para obtener una figura simétrica. Sin embargo, cuidate de no escalar restringido en un eje en Modo Objeto ya que esto llevará a resultados impredecibles en los objetos duplicados cuando activemos el botón *Rot* button.

Figure 22-17. Our model complete.



Quando hayas terminado con el modelo y estés contento con el resultado, puedes seleccionar el tentáculo, presionar **SHIFT-CTRL-A** y hacer click en la opción *Make duplis real ?* para convertir las copias virtuales en masllas reales ([Figure 22-17](#)).

Notes

- [1] y también en Modo Objeto, sin embargo escalar en Modo Objeto puede traer algunos problemas al aplicar Rotación en los DUpliverts como veremos pronto

[Prev](#)

Tecnicas especiales de modelado

[Home](#)[Up](#)[Next](#)

DupliFrames

22.3. DupliFrames

Relevante a Blender v2.31

Podemos pensar en Dupliframes de dos maneras distintas: como una herramienta de modelado o como una herramienta para disponer objetos. En un sentido, los Dupliframes son muy similares a los Dupliverts. La única diferencia es que con Dupliframes, disponemos o ubicamos nuestros objetos haciéndolos seguir una curva en lugar de usar los vértices de una malla.

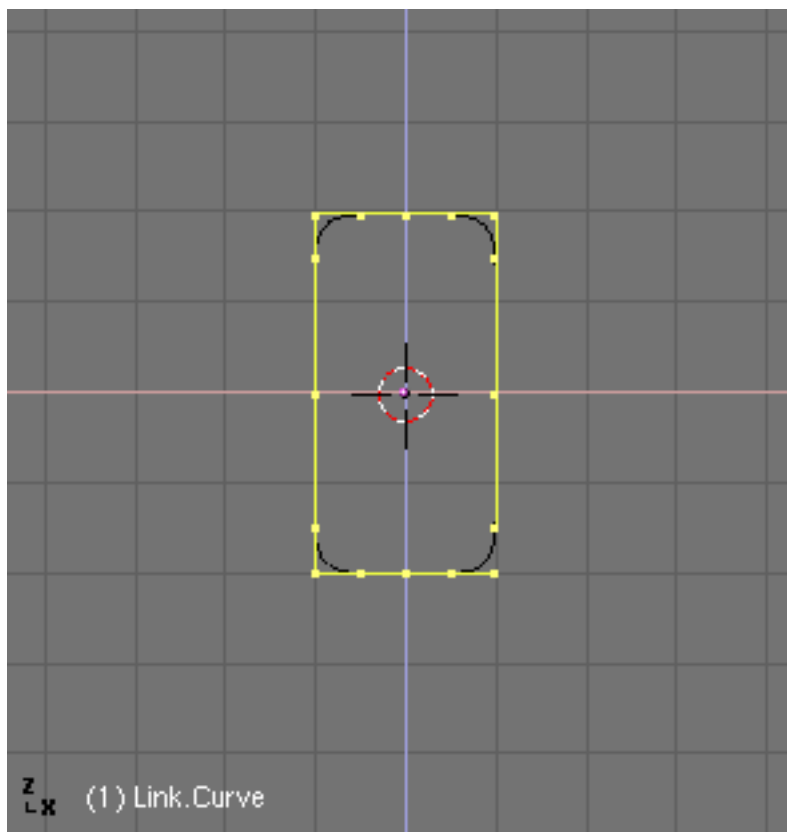
DupliFrames viene de DUPLICado en FRAMES (cuadros) y es una técnica de modelado muy útil para objetos que se repiten a lo largo de un recorrido, como los durmientes de una vía ferroviaria, las tablas de una cerca o los eslabones de una cadena, pero también sirve para modelar objetos curvos complejos como tirabuzones, conchas marinas and espirales.

22.3.1. Modelando con Dupliframes

Vamos a modelar una cadena con sus eslabones usando dupliframes.

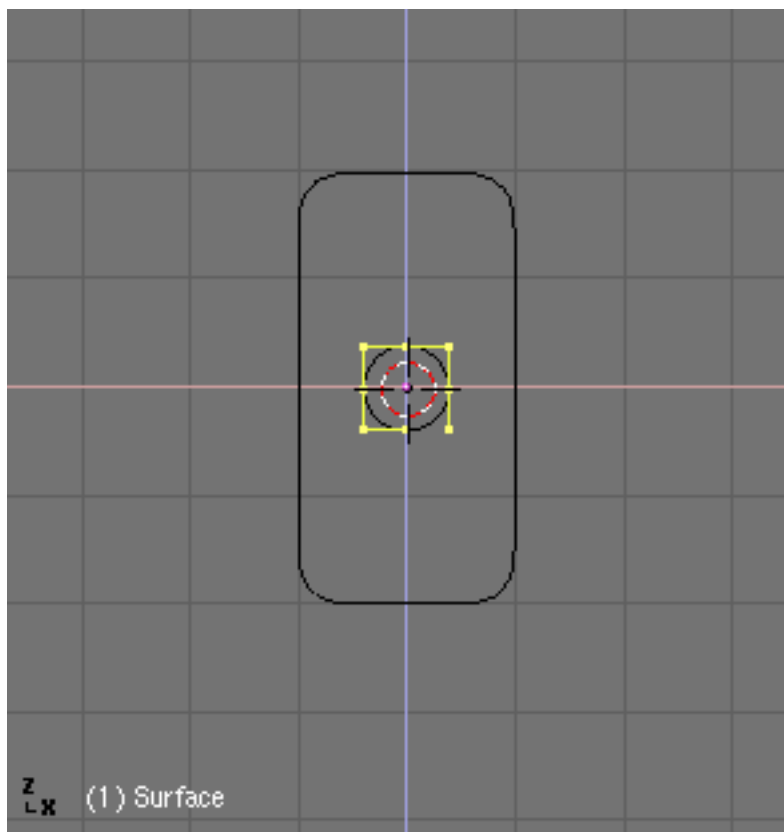
Lo primero viene primero. Para explicar el uso de Dupliframes para modelar, vamos a empezar modelando un eslabón. Para ello crearemos una Curva Círculo en la vista frontal (Bézier o NURBS da lo mismo). En Modo Edición, lo subdividimos una vez y movemos los vértices un poco para que se ajusten al contorno del eslabón ([Figure 22-18](#)).

Figure 22-18. Contorno del eslabón



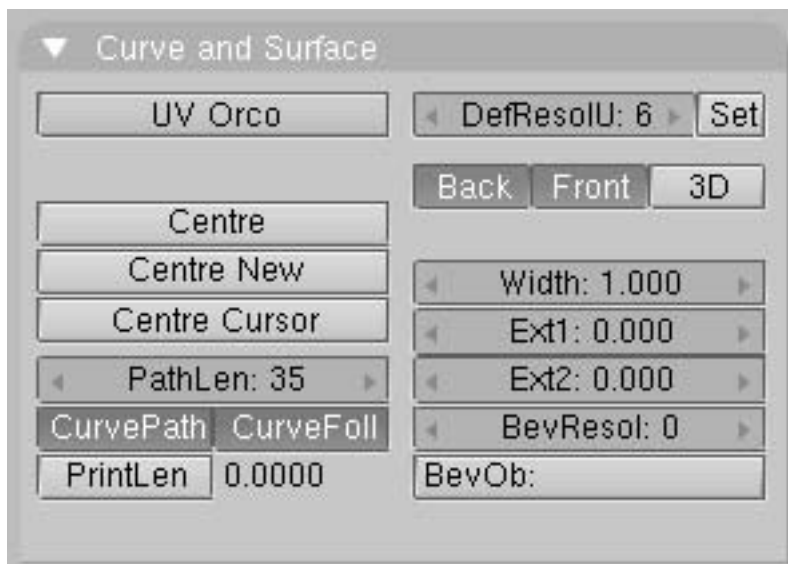
Salimos de Modo Edición y añadimos una Superficie Círculo ([Figure 22-19](#)). Las Superficies NURBS son ideales para esto, porque podemos cambiar la resolución fácilmente después de crearlas, y si lo necesitamos, podemos convertirlas a objetos malla. Es muy importante no confundir Curvas Círculo con Superficies Círculo. Las primeras servirán para dibujar la forma del eslabón, pero no nos permitirá realizar el paso de "skinning" luego. Las segundas servirán para trazar las secciones transversales de nuestro "skinning".

Figure 22-19. Sección transversal del eslabón



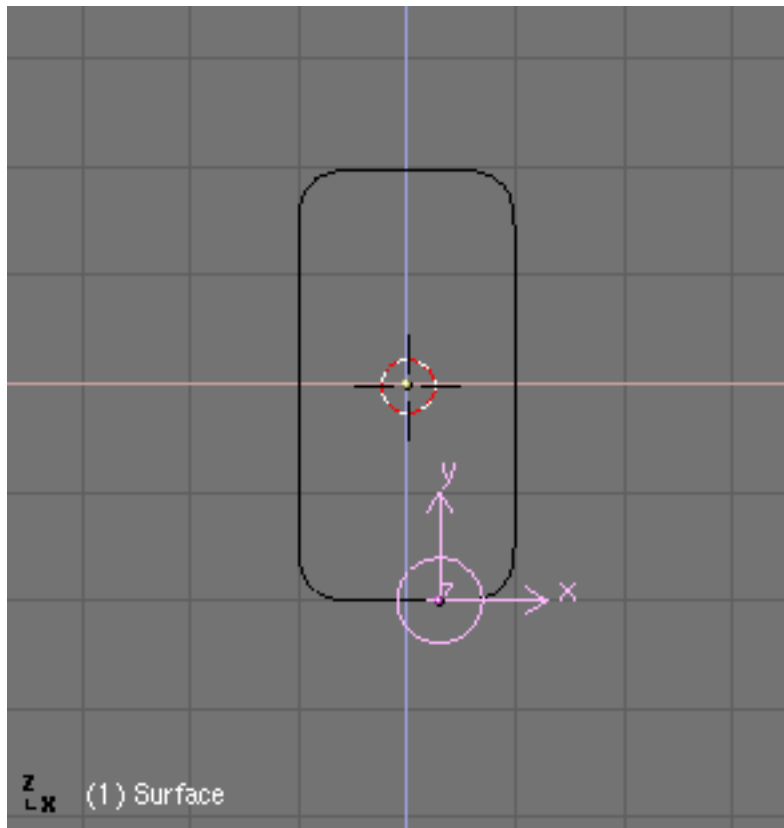
Ahora emparentaremos la superficie círculo a la curva círculo (el contorno del eslabón) usando emparentado *Normal* (no confundir con Curve Follow). Seleccionaremos la curva y en los botones de Object Context y en el panel *Anim Settings* pulsaremos los botones *CurvePath* y *CurveFollow* ([Figure 22-20](#)).

Figure 22-20. Curve's settings: Curve Path y Curve Follow.



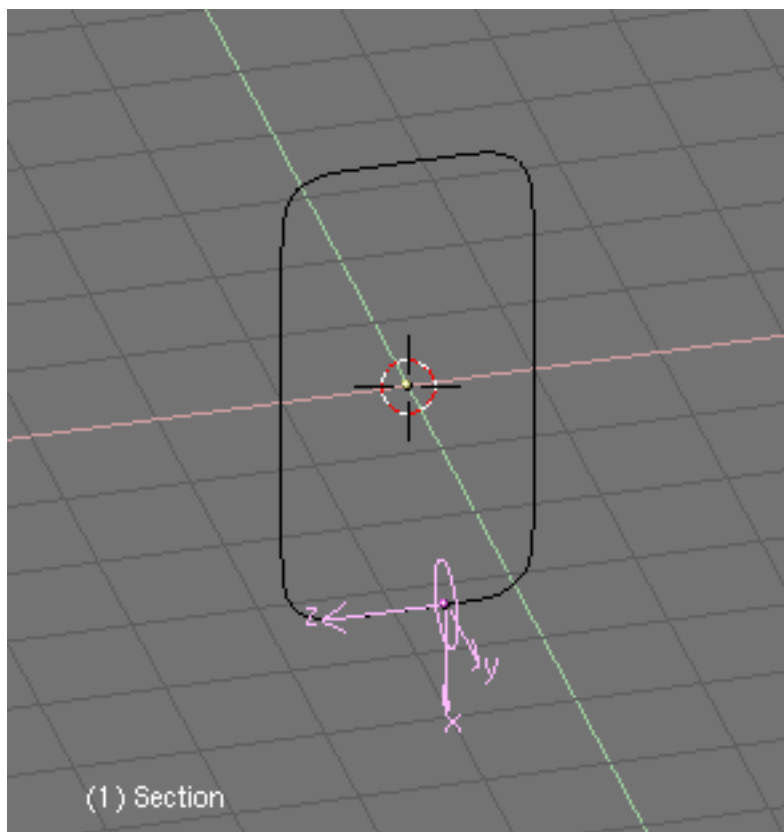
Es probable que la superficie círculo aparezca desplazada. Simplemente la seleccionaremos y pulsaremos **ALT-O** para borrar el origen ([Figure 22-21](#)).

Figure 22-21. Borrando el origen.



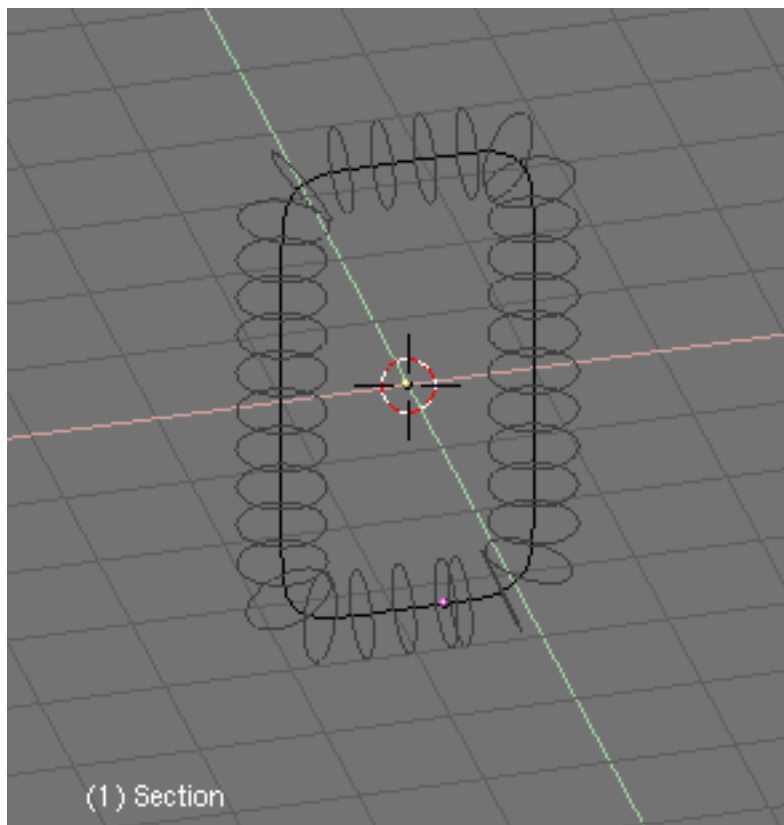
Si pulsas **ALT-A** la superficie avanzará siguiendo la curva. Ahora quizás tengas que ajustar los ejes usando los botones *TrackX*, *Y*, *Z* y *UpX*, *Y*, *Z* en los botones de animación, para que la superficie vaya perpendicular al recorrido curvo. ([Figure 22-22](#)).

Figure 22-22. Siguiendo el eje correcto.



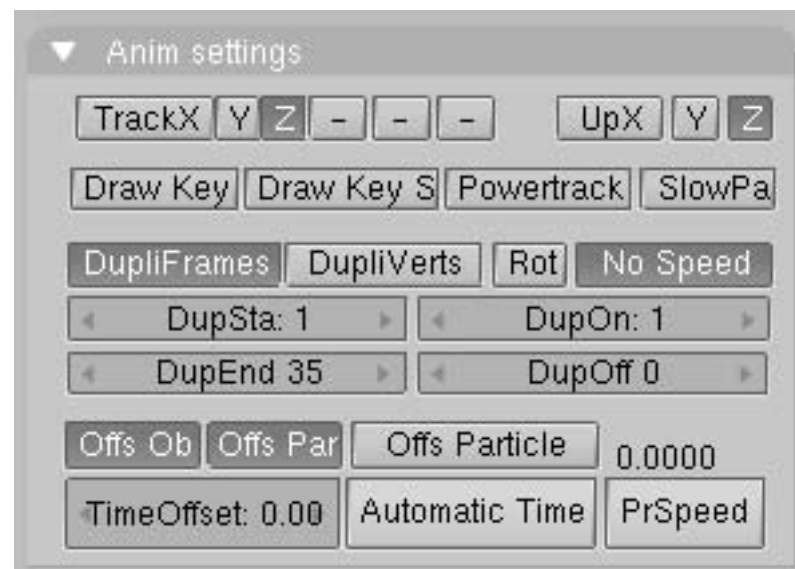
Seleccionemos ahora la superficie y vayamos al panel de *Anim Settings* y pulsemos *DupliFrames*. Un número de instancias de la sección circular aparecerán a lo largo del recorrido curvo. ([Figure 22-23](#)).

Figure 22-23. DupliFrames!



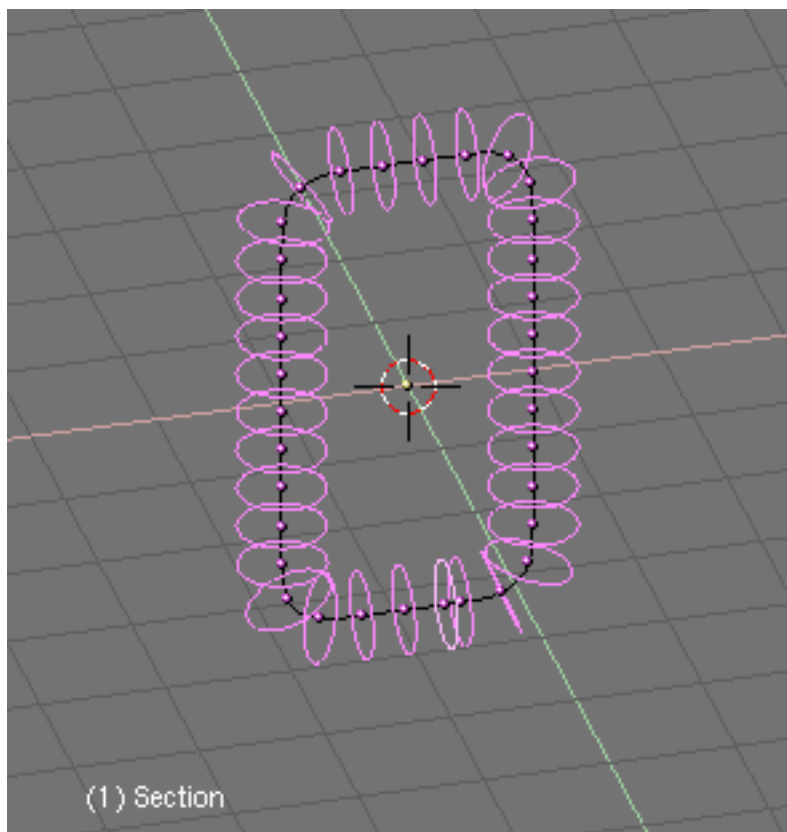
Podemos ajustar el número de círculos que queremos tener con los botones *DupSta*, *DupEnd*, *DupOn* y *DupOff*. Estos botones controlan el Inicio y Final de la duplicación, el número de duplicados por frame y también el Offset o separación entre duplicaciones. Si quieres que el eslabón se vea abierto en lugar de cerrado puedes probar un valor distinto para *DupEnd* ([Figure 22-24](#)).

Figure 22-24. Valores para DupliFrames. Notar que "DupEnd: 35" terminará el eslabón antes del final de la curva.



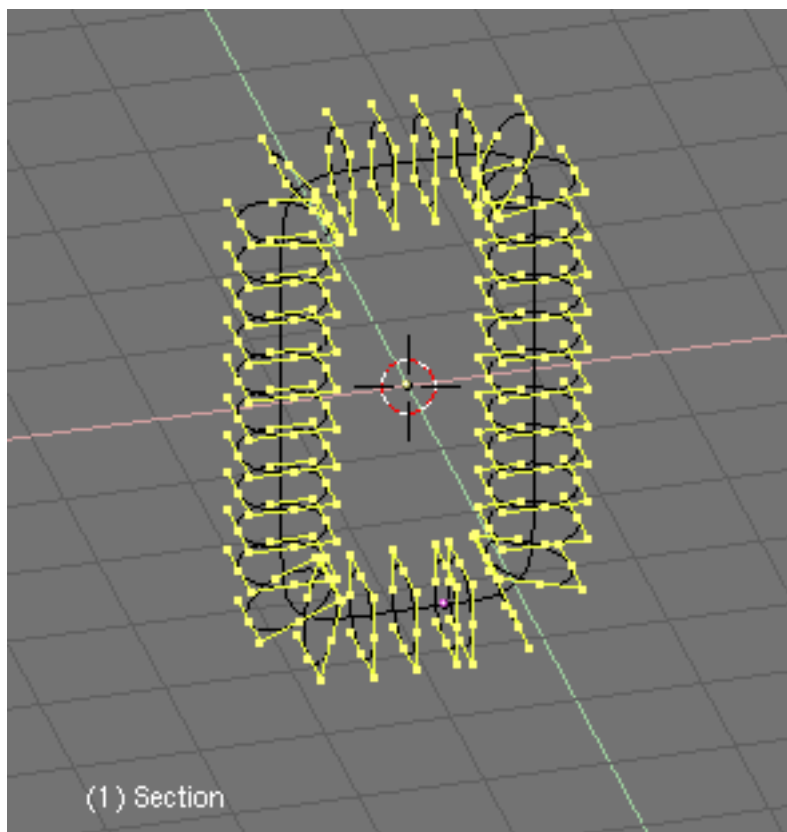
Para convertir la estrucutra en un objeto NURB real, seleccionaremos la Superficie Circulo y pulsaremos **CTRL-SHIFT-A**. Un menu nos preguntará *OK? Make Dupli's Real* ([Figure 22-25](#)).

Figure 22-25. Making Dupli's Real.



No deseccionemos nada. Tenemos una colección de NURBS que forman el contorno de nuestro objeto, pero aún no existe una "piel" que una los contornos (no existe un "skinning"), por lo que no podemos verlo en un render o en un preview sombreado. Para lograr esto necesitamos unir todos los anillos en un solo objeto. Sin deseleccionar ningún anillo pulsa **CTRL-J** y confirma el menu emergente. Ahora entra en Modo Edición para este objeto y pulsa **AKEY** para seleccionar todos los vertices ([Figure 22-26](#)). Estamos listo para generar la superficie. Pulsa **FKEY** y Blender gnerará automaticamente el objeto sólido. Esta operación se conoce como "Skinning" y se describe con más detalle en [Section 9.7](#).

Figure 22-26. Skineando el eslabón.



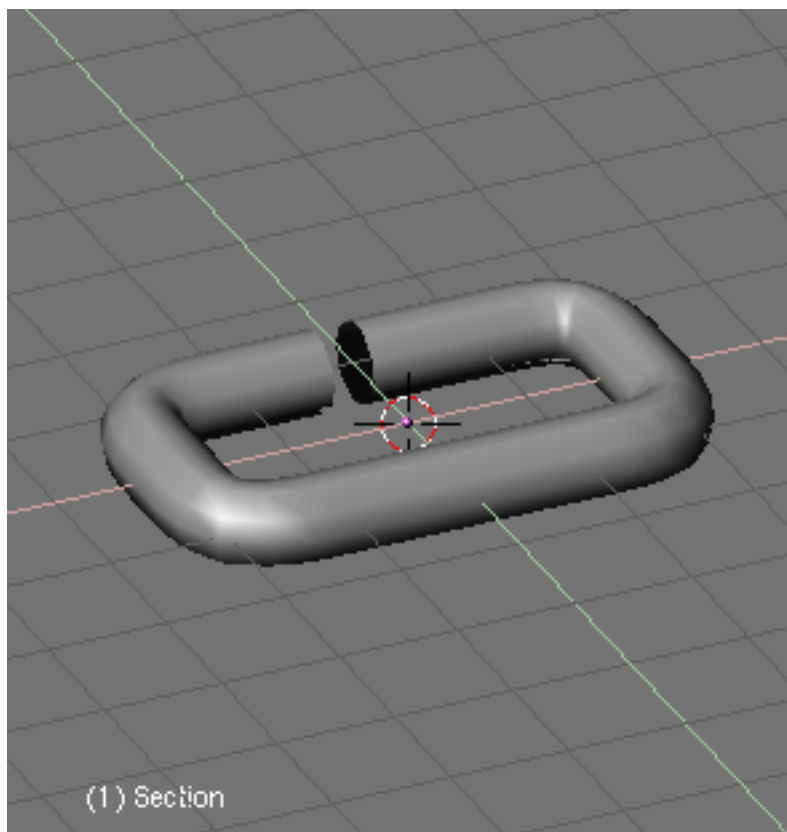
Cuando salgas del Modo Edición, podrás ver el objeto en modo sombreado. Sin embargo se ve muy oscuro. Para corregir esto, entremos otra vez en Modo Edición y seleccionemos todos los vertices, luego pulsemos **WKEY**. Elige *Switch Direction* en el menú y sal de Modo Edición. El objeto será representado correctamente ahora. ([Figure 22-27](#)).

El objeto que hemos creado es un objeto NURBS. Esto significa que aún podremos editarlo. Pero aún más interesante, también podremos controlar la resolución del objeto NURBS por medio de los botones de edición.

Allí podemos establecer la resolución del objeto usando los botones *ResolU* y *ResolV*, así por ejemplo podremos usar una resolución baja mientras trabajamos con el objeto y luego usar una resolución alta al momento de hacer render. Los objetos NURBS también tienen la ventaja de ocupar poco espacio en disco cuando salvamos la escena. Compara el tamaño de una escena NURBS con la misma escena donde las NURBS se han convertido a malla (**ALT-C**).

Para finalizar podemos borrar la curva que usamos para el contorno del eslabón ya que no la necesitaremos más.

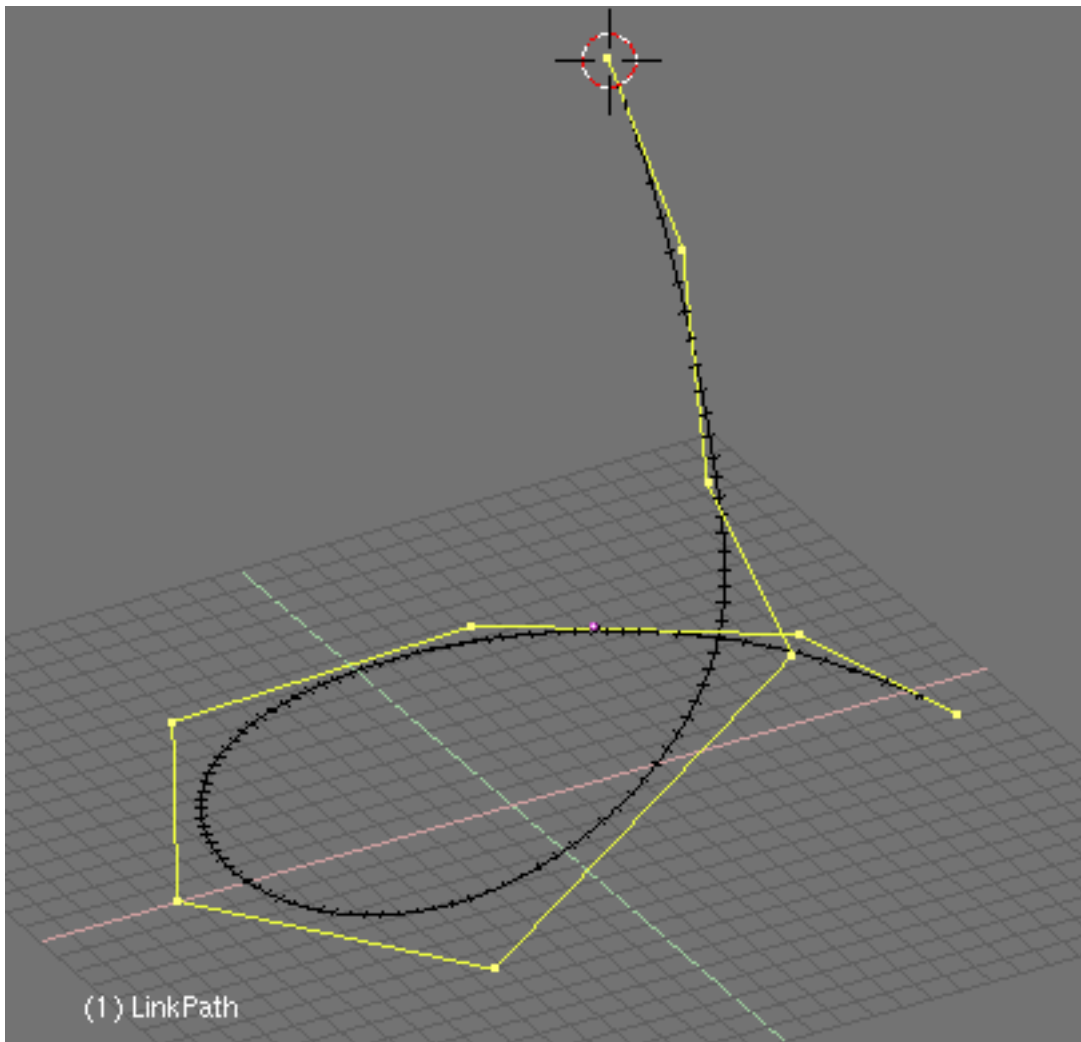
Figure 22-27. Skinned link.



22.3.2. Colocando objetos con DupliFrames

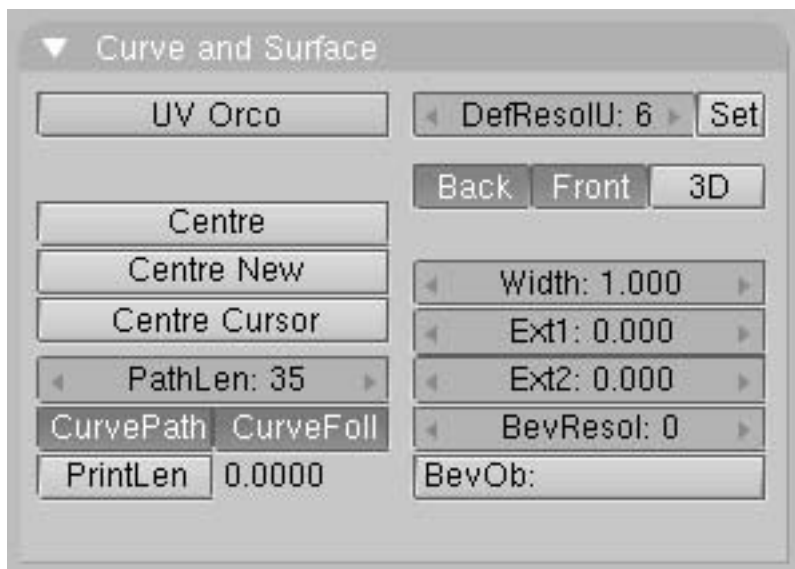
Ahora seguiremos con el modelado de la cadena propiamente dicha. Para esto, añadiremos una Curva Path (Camino) (podemos usar otro tipo de curva pero esta da el mejor resultado). En Modo Edición movemos sus vértices hasta lograr la forma deseada para la cadena ([Figure 22-28](#)). Si no usamos una Curva Path, deberíamos activar el botón *3D* en los botones de edición para que la cadena se despliegue en el espacio 3D.

Figure 22-28. Using a curve path to model the chain.



Selecciona el objeto "Eslabón" que hemos modelado anteriormente y emprantémoslo a la curva "Cadena" usando "Normal Parent". Como estamos usando una Curva Path la opción *CurvePath* en los botones de edición aparecerá activada automáticamente, sin embargo no pasará lo mismo con la opción *CurveFollow* así que deberemos activarla manualmente ([Figure 22-29](#)).

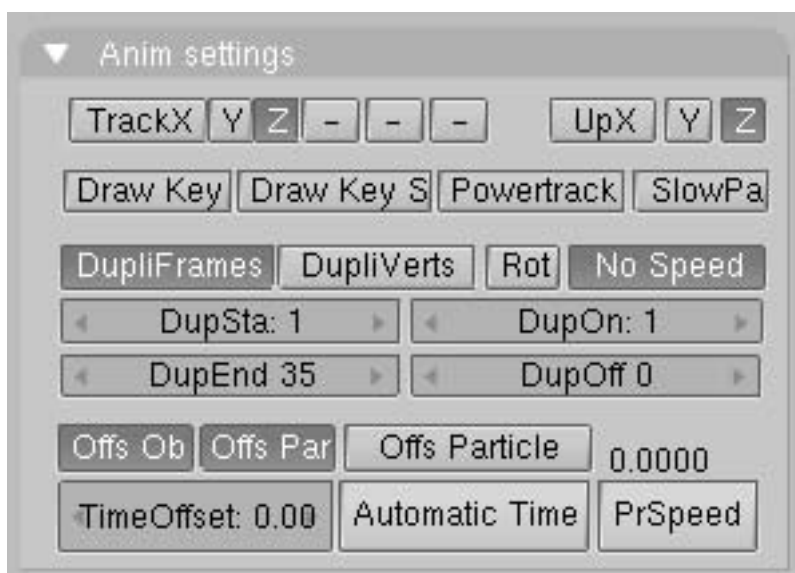
Figure 22-29. Curve settings.



Si el eslabón aparece trasladado, selecciónalo y pulsa **ALT-O** para borrar el origen. Hasta ahora hemos hecho poco más que animar el eslabón a lo largo de la curva. Podemos verificarlo reproduciendo la animación con **ALT-A**.

Ahora, con el eslabón seleccionado, una vez más iremos a Object Context y el panel *Anim settings*. Aquí, activaremos la opción *DupliFrames* como antes. Jugaremos con los botones numéricos *DupSta:*, *DupEnd:* y *DupOf:*. Normalmente usaríamos *DupOf: 0* pero en el caso de una cadena, si usamos *DupOf: 0* los eslabones estarán demasiado cerca uno de otro y deberemos cambiar el valor *PathLen* de la curva camino a un valor menor en el Contexto de Edición y el panel *Curve and Surface* y luego cambiar correspondientemente el valor *DupEnd:* a ese número para el eslabón ([Figure 22-30](#)).

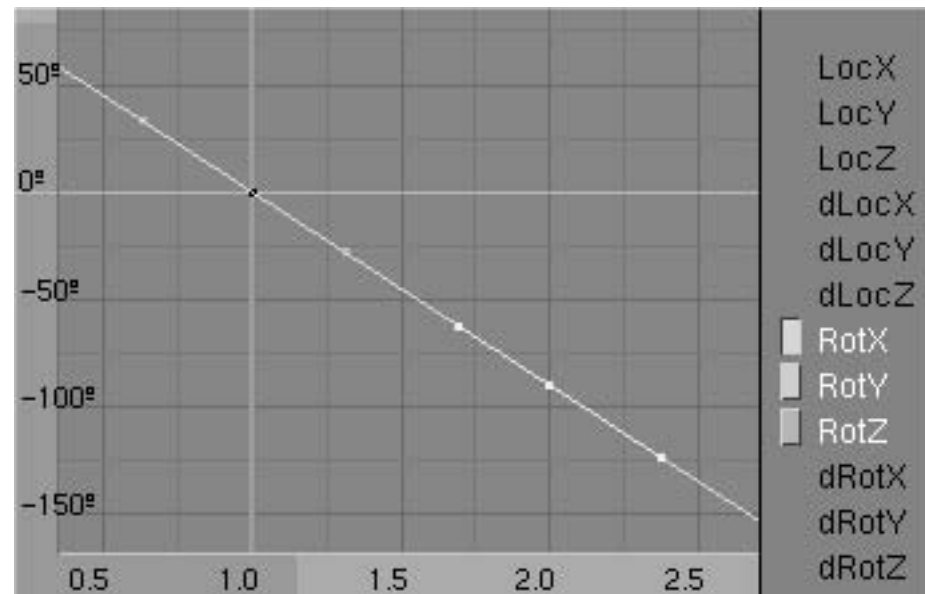
Figure 22-30. Ajustando los DupliFrames.



Necesitamos que los eslabones roten a lo largo de la curva, de forma que queden rotados 90 grados uno respecto del anterior. Para ello, seleccionaremos el eslabón y pulsaremos *Axis* en los botones de edición

para visualizar los ejes del objeto. Insertaremos un keyframe de rotación en el eje paralelo a la curva. Nos moveremos 3 o 4 frames adelante y rotaremos alrededor de este eje pulsando **RKEY** seguido de **XKEY-XKEY** (**XKEY** dos veces), **YKEY-YKEY**, o **ZKEY-ZKEY** para rotar en el eje *local* X, Y o Z ([Figure 22-31](#)).

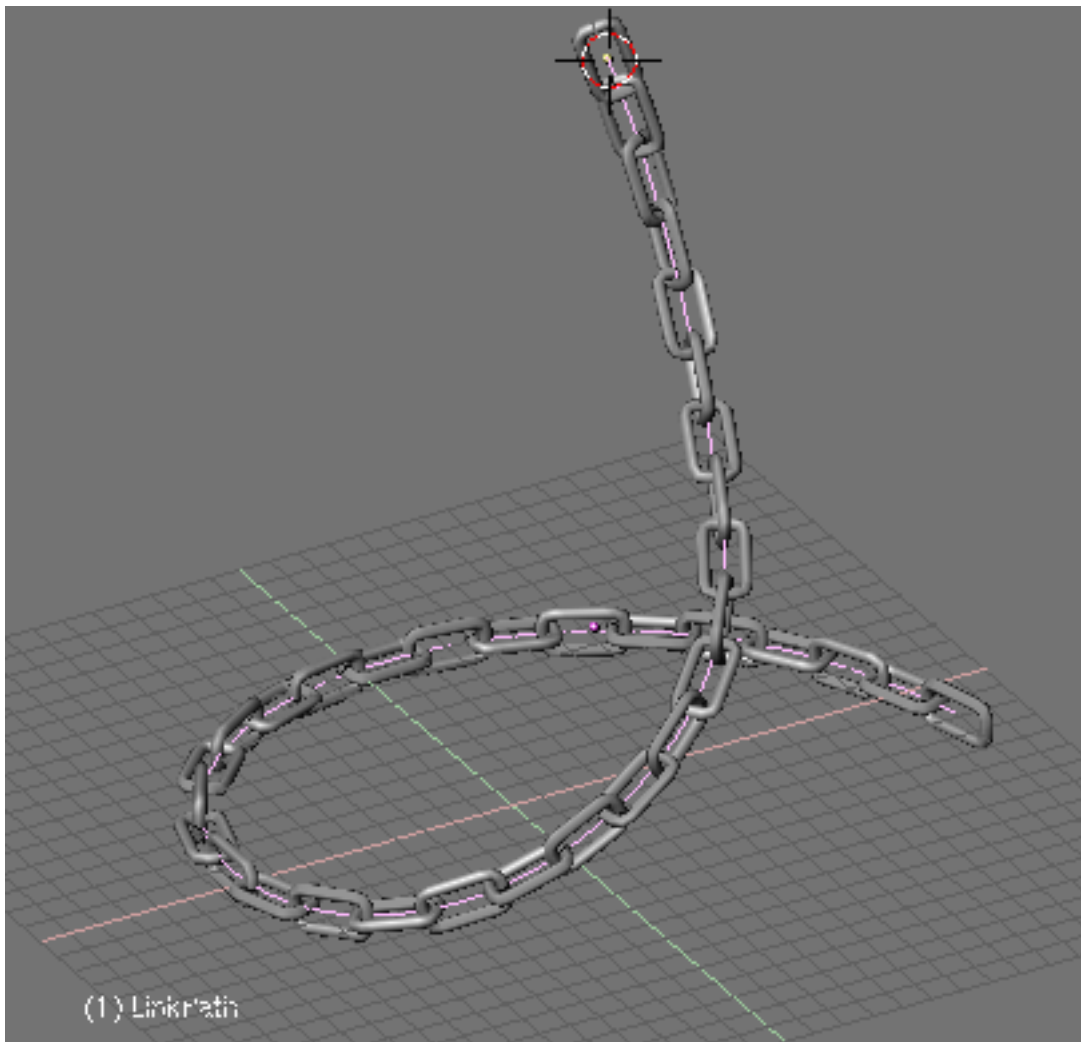
Figure 22-31. Rotando el eslabón.



Abriremos una ventana IPO para editar la rotación del eslabón alrededor del camino. Pulsaremos el *Modo Extrapolación* para que el eslabón rote continuamente hasta terminar el recorrido. Podemos editar la curva IPO de rotación de manera que el eslabón rote exactamente 90 grados cada uno, dos o tres eslabones (cada eslabón corresponde a un cuadro de animación). Usa **NKEY** para ubicar un nodo exactamente en X=2.0 e Y=9.0, que corresponden a 90 grados en un cuadro (desde el cuadro 1 al 2).

Ahora tenemos una bonita cadena ([Figure 22-32](#))!

Figure 22-32. Cadena con Dupliframe.



22.3.3. Más modelado y animación

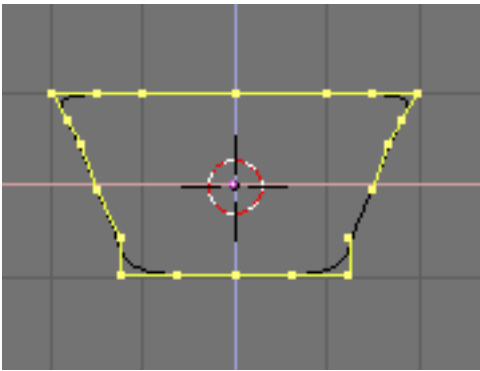
No estás limitado a usar Curvas Path para modelar tus objetos. Los hemos usado por conveniencia, sin embargo en algunos casos no hay necesidad de ellas.

En la vista frontal, agregaremos una superficie Círculo (ya deberías saber por qué a estas alturas [Figure 22-33](#)). Subdividiremos una vez para darle una forma más cuadrada. Moveremos y escalaremos un poco algunos vértices para darle una forma más trapezoidal ([Figure 22-34](#)).

Figure 22-33. A Surface Circle.

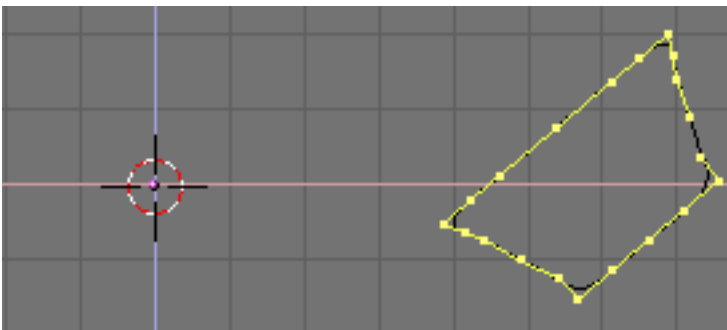


Figure 22-34. Trapezoidal cross-section.



Luego rotaremos todos los vertices algunos grados. Seleccionaremos todos los vértices y los desplazaremos algunas unidades hacia la izquierda o la derecha algunas unidades en X (pero sin cambiar la ubicación en Z). Podemos mantener presionada **CTRL** para mover y escalar con precisión. Salgamos de Modo Edición ([Figure 22-35](#)).

Figure 22-35. Trapezoidal cross section, rotated and translated.

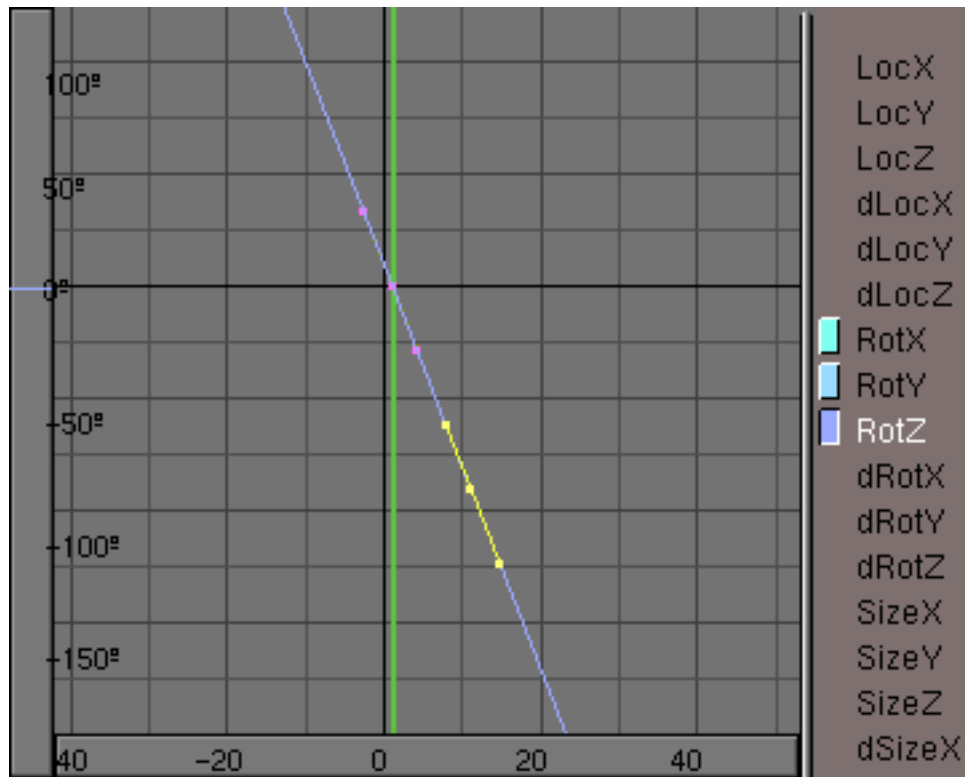


A partir de aquí la única copsa que haremos será editar las curvas IPO de animación. Podríamos llamar a lo que sigue como "Modelando con Animación" si te parece. Ya no entraremos al Modo Edición de la superficie para nada.

Cambiemos a una vista superior. Insertemos un keyframe de rotaicón en el cuadro 1, avancemos 10

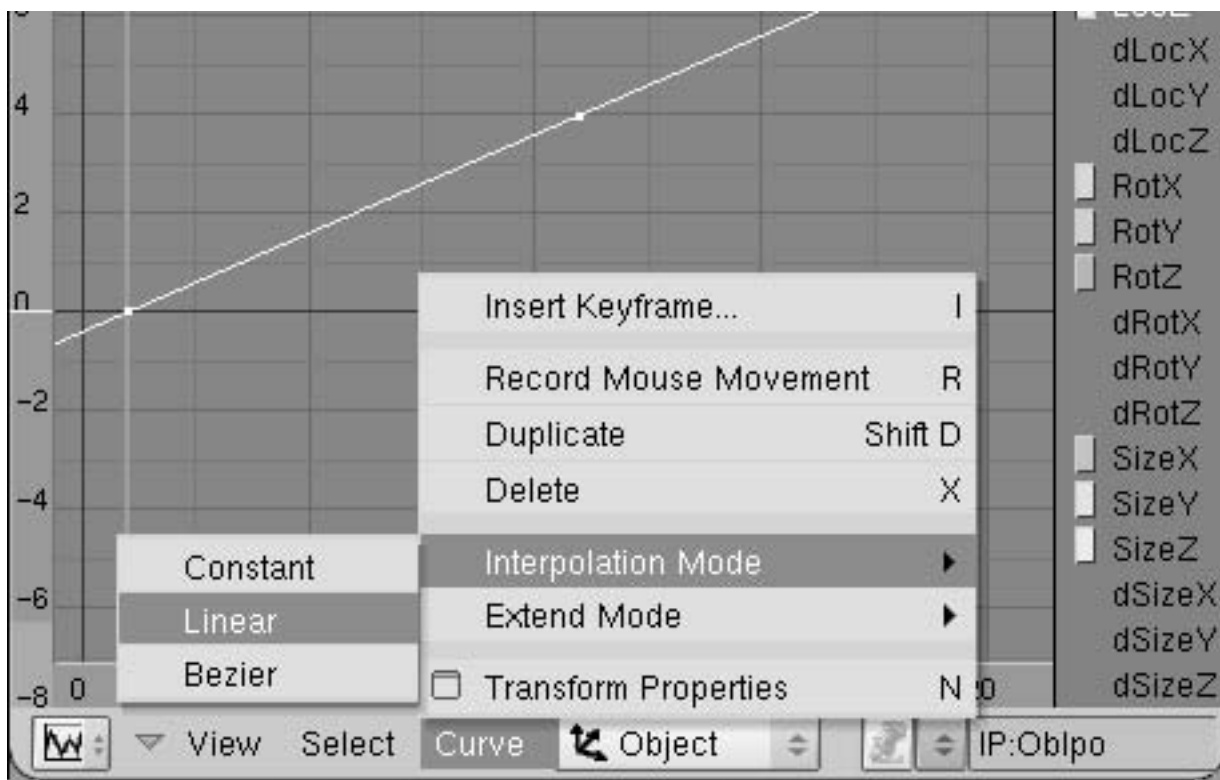
cuadros y rotemos la superficie 90 grados alrededor de su nuevo origen. Insertemos un keyframe más. Abramos una ventana IPO y pongamos la IPO de rotación en Modo Extrapolación. ([Figure 22-36](#)).

Figure 22-36. Rotation IPO for the cross section.



Volvamos al cuadro 1 e insertemos un keyframe de Ubicación (Location). Cambiemos a la vista frontal. Avancemos 10 cuadros (solo pulsa **UPARROW**) y movamos la superficie en Z algunas unidades. Insertemos un nuevo keyframe para Ubicación. En la ventana IPO pondremos la curva *LocZ* en Modo Extrapolación ([Figure 22-37](#)).

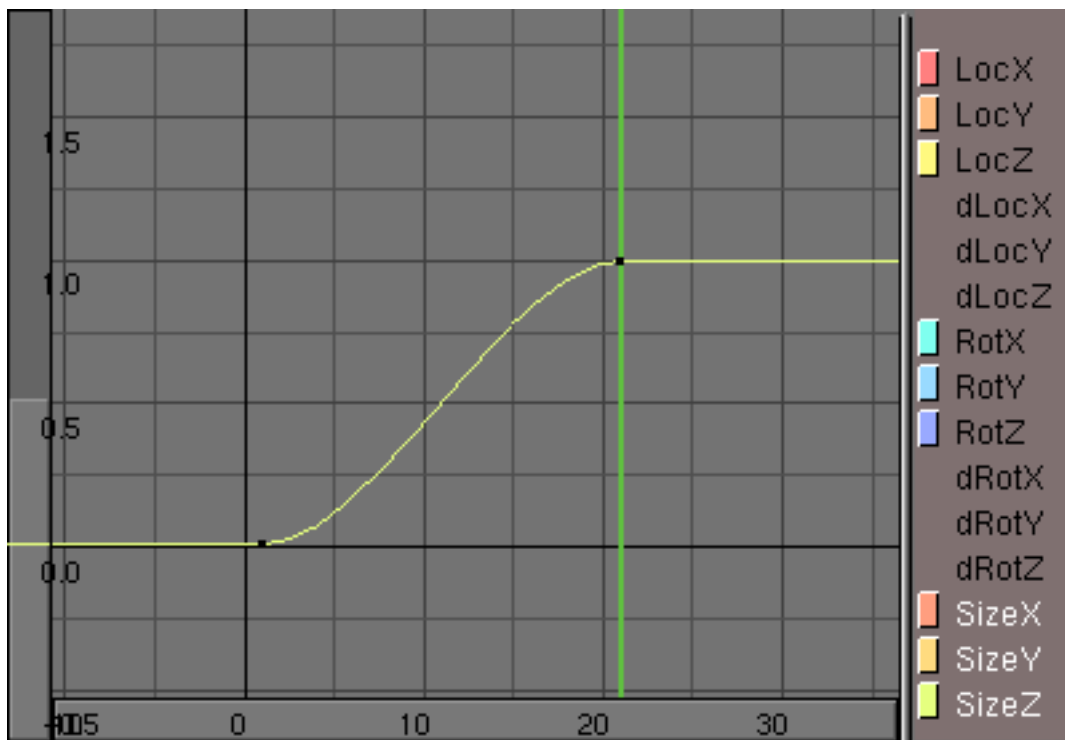
Figure 22-37. Translation IPO for the cross section.



Ahora, por supuesto, iremos a los botones de animación y pulsaremos *DupliFrames*. Podremos ver cómo la superficie va ascendiendo en espiral a través del espacio 3D formando algo parecido a un resorte. Se ve lindo, pero queremos más !. Desactivemos *DupliFrames* para continuar.

En el cuadro 1 escalaremos la superficie casi hasta cero e insertaremos un keyframe para Size. Adelantaremos hasta el cuadro 41 y borraremos el tamaño con **ALT-S**. Insertaremos entonces un nuevo keyframe para el tamaño. Esta IPO en particular no la pondremos en Modo Extrapolación ya que no queremos que se agrande hasta el infinito ¿verdad? ([Figure 22-38](#))?

Figure 22-38. Size IPO for the cross section.



Si ahora activamos DupliFrames verás la silueta de un bonito tirabuzón. ([Figure 22-39](#)). Una vez más los últimos pasos son: Make Duplis Real, unir las superficies, seleccionar todos los vértices y hacer el skinning, invertir la dirección de la normal si es necesario y salir de Modo Edición ([Figure 22-40](#)).

Figure 22-39. Using a curve path to model the chain.

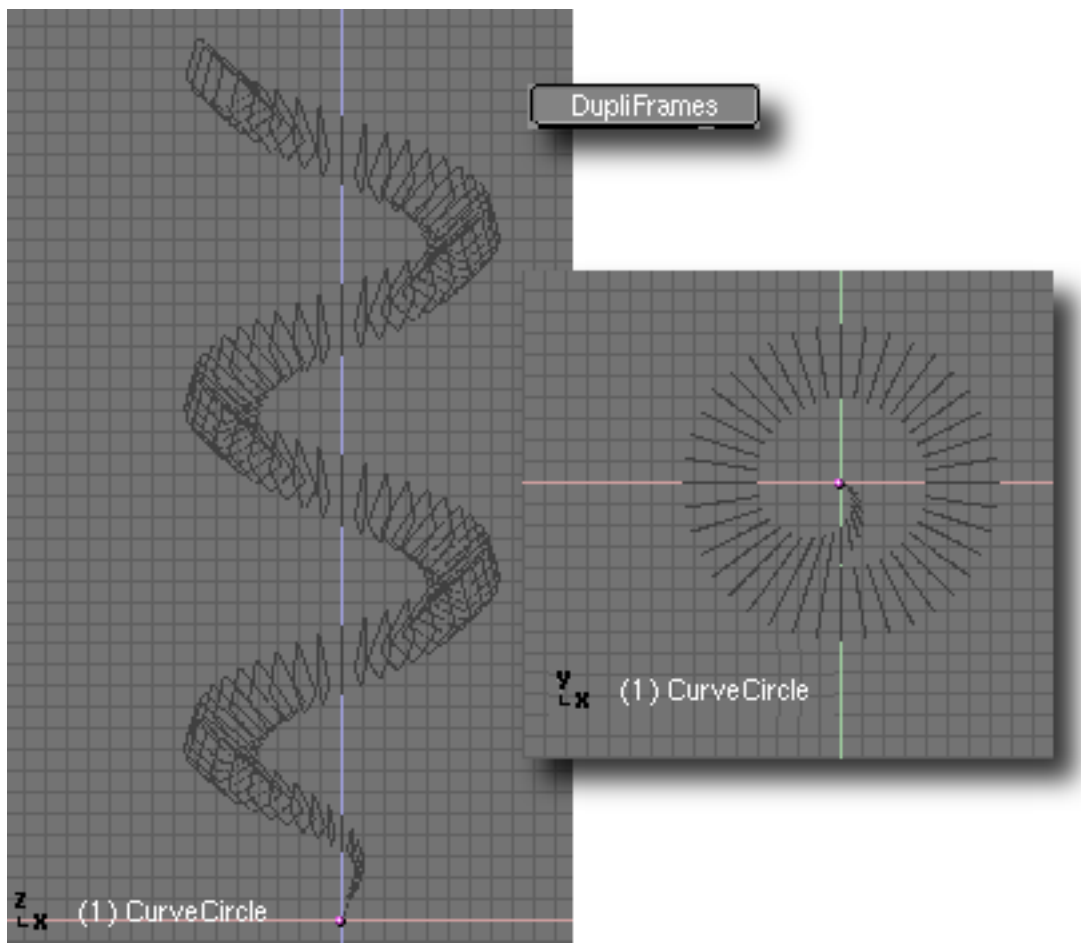
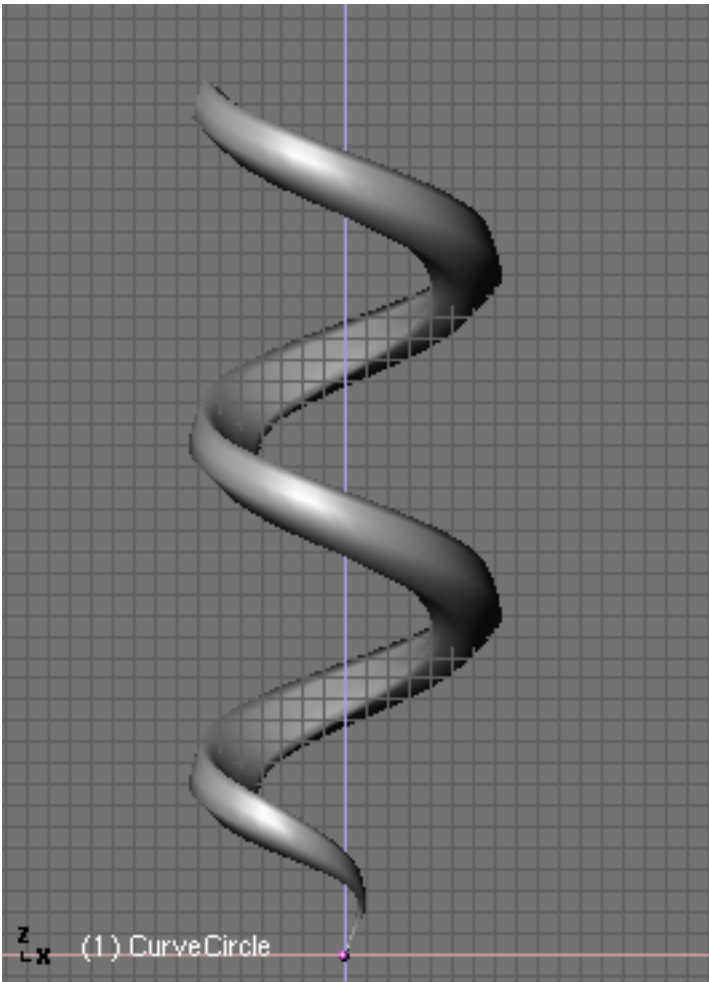


Figure 22-40. Using a curve path to model the chain.



Este fue un ejemplo más o menos sencillo. CON más trabajo de edición de curvas IPO podemos lograr modelos más complejos e interesantes. Solo usa tu imaginación.

[Prev](#)[DupliVerts](#)[Home](#)[Up](#)[Next](#)[Modelando con lattices](#)

22.4. Modelando con lattices

Relevante a Blender v2.31

Una Lattice o Retícula consiste en una rejilla tridimensional de vértices no renderizable. Su principal uso es dar deformaciones adicionales a cualquier objeto "hijo" que posea. Estos objetos pueden ser mallas, superficies e incluso partículas.

¿Por qué usaríamos una Lattice para deformar una malla en lugar de deformar la malla directamente en Modo Edición ?

Hay un buen par de razones para hacerlo:

1. Primero que nada: Es más sencillo. Como una malla podría tener chiquicientos vertices, escalar, mover y rotarlos puede convertirse en una faena monumental. En cambio, si utilizáramos una Lattice simple, el trabajo se reduciría a mover unos pocos vertices.
2. Queda más bonito. Las deformaciones que obtenemos seguro se verán mucho mejores!
3. Es rápido ! Podemos colocar todos nuestros objetos hijo en capas ocultas y deformarlos todos al mismo tiempo
4. Es un buen hábito de modelado. Una Lattice se puede utilizar para obtener distintas versiones de la misma malla con muy poco trabajo adicional y consumo de recursos. Esto lleva a un diseño óptimo de escena, minimizando el trabajo de modelado. Una lattice además no afecta las coordenadas de textura de una malla. Facilita grandemente la aplicación de cambios sutiles en una malla sin tener que modificar la malla propiamente dicha.

22.4.1. ¿ Cómo se hace ?

Una Lattice siempre aparece inicialmente como una rejilla de vertices de 2 x 2 x 2 (que se ve como un cubo normal). Podemos agrandarla o achicarla en Modo Objeto y modificar su resolución a través de los botones *U*, *V*, *W* de Contexto Edición en el panel *Lattice*.

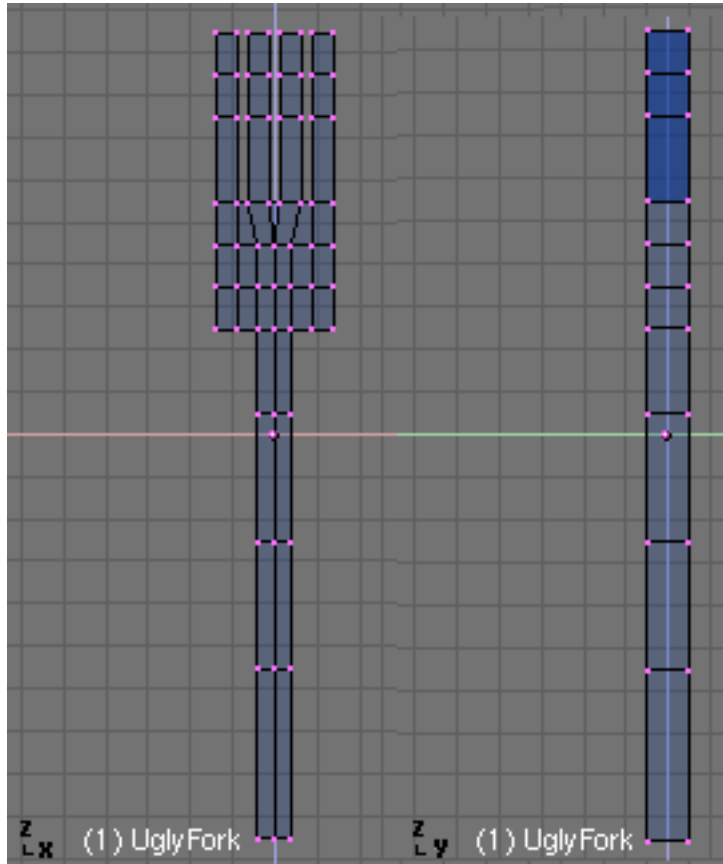
Luego de este paso inicial, podemos deformar la Lattice en Modo Edición. Si existiera un Objeto Hijo, la deformación se mostraría y modificaría continuamente. Si modificamos los valores *U*, *V*, *W* de la Lattice, esta vuelve a su posición y forma iniciales.

Ahroa veremos un caso muy simple en donde utilizando una Lattice simplificamos y aceleramos el

proceso de modelado.

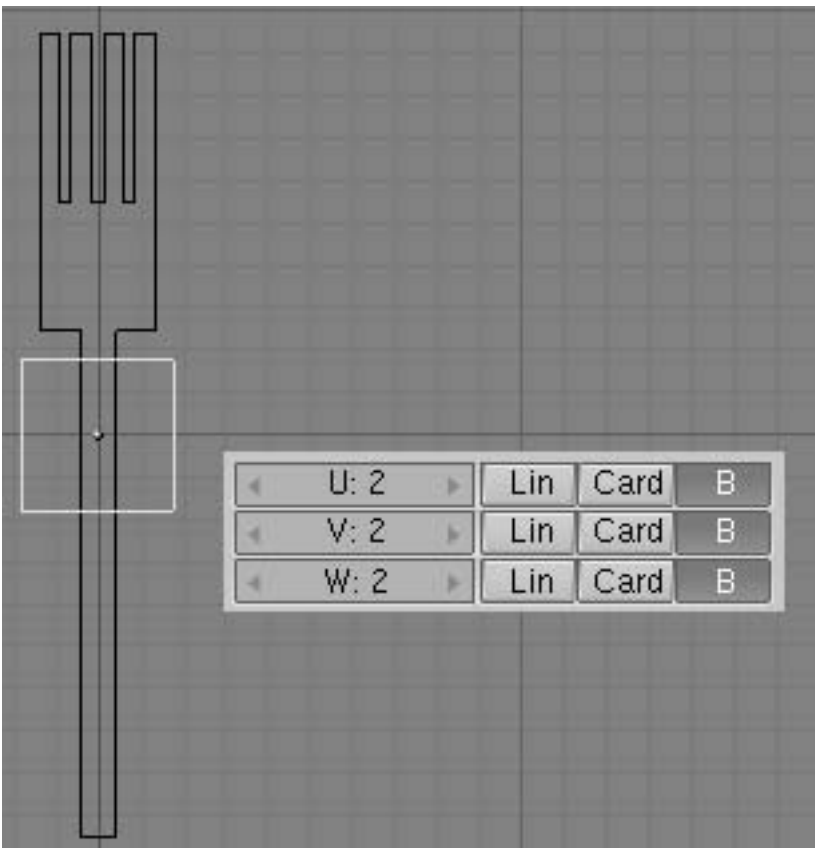
Modelé un tenedor muy elemental usando un plano subdividido un par de veces. Se ve bastante horrible, pero es todo lo que necesitamos. Por supuesto, desde un aVista Lateral se verá completamente plano. Wow, es *realmente* feo ([Figure 22-41](#)). El único detalle importante es que has sido subdividido lo suficiente como para asegurarnos una buena deformación cuando apliquemos la Lattice. ¡ Es que no se puede doblar un segmento de dos vertices!

Figure 22-41. Un tenedor feo.



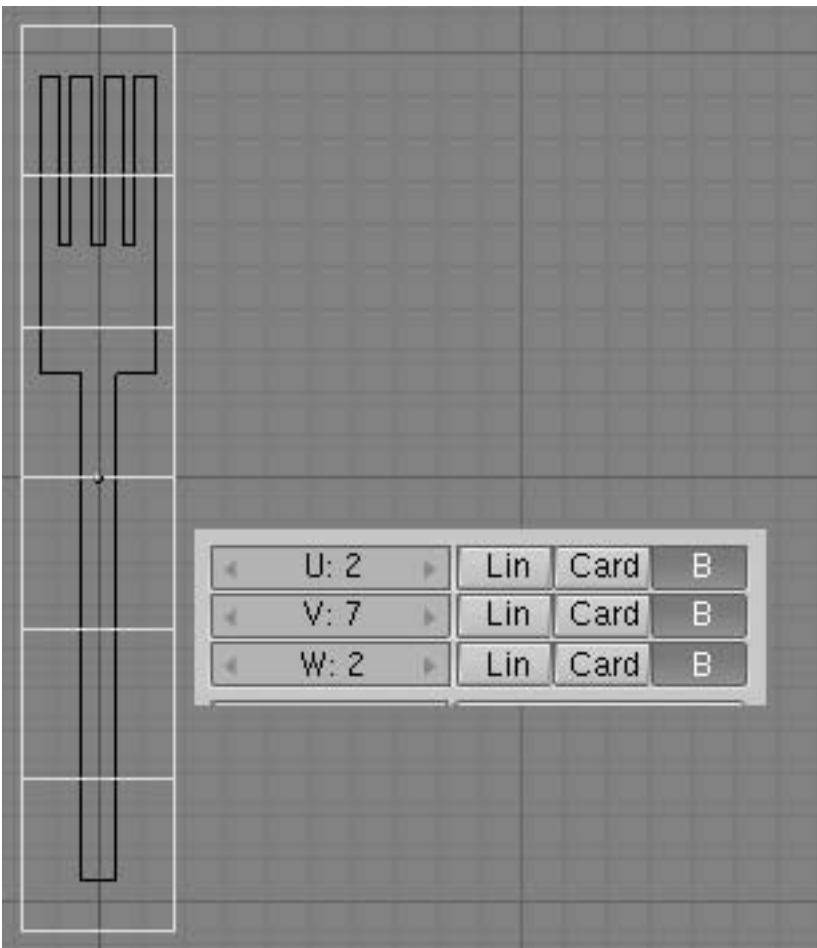
En la Vista Superior, agrego una Lattice. Antes de cambiar su resolución, la escalamos para que abarque completamente el ancho del tenedor ([Figure 22-42](#)). Esto es muy importante. Como quiero mantener la cantidad de vertices de la Lattice lo más baja posible (no tiene sentido que tenga la misma cantidad que la malla ¿verdad?), debo mantener la resolución baja. Por ello es preferible escalar convenientemente.

Figure 22-42. Una lattice de 2x2x2.



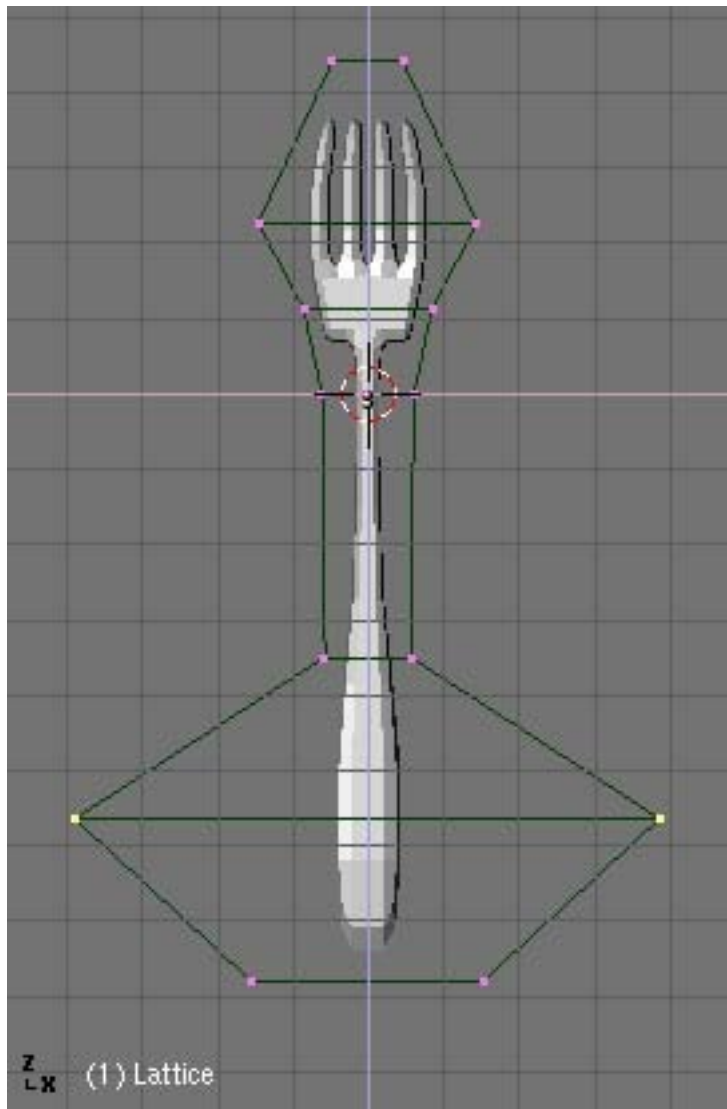
Ajustemos la resolución de la Lattice para abarcar el largo del tenedor ([Figure 22-43](#)).

Figure 22-43. Usa una resolución adecuada pero no exageres.



Ahora estamos listos para la parte divertida. Emparentemos el tenedor a la Lattice seleccionando el tenedor y la Lattice y pulsando **CTRL-P**. Entremos en Modo Edición para la Lattice y empecemos a seleccionar y escalar vertices ([Figure 22-44](#)). Quizás prefieras escalar en los ejes X e Y por separado para tener mayor control sobre la profundidad de la Lattice (para evitar hacer el tenedor más grueso o delgado).

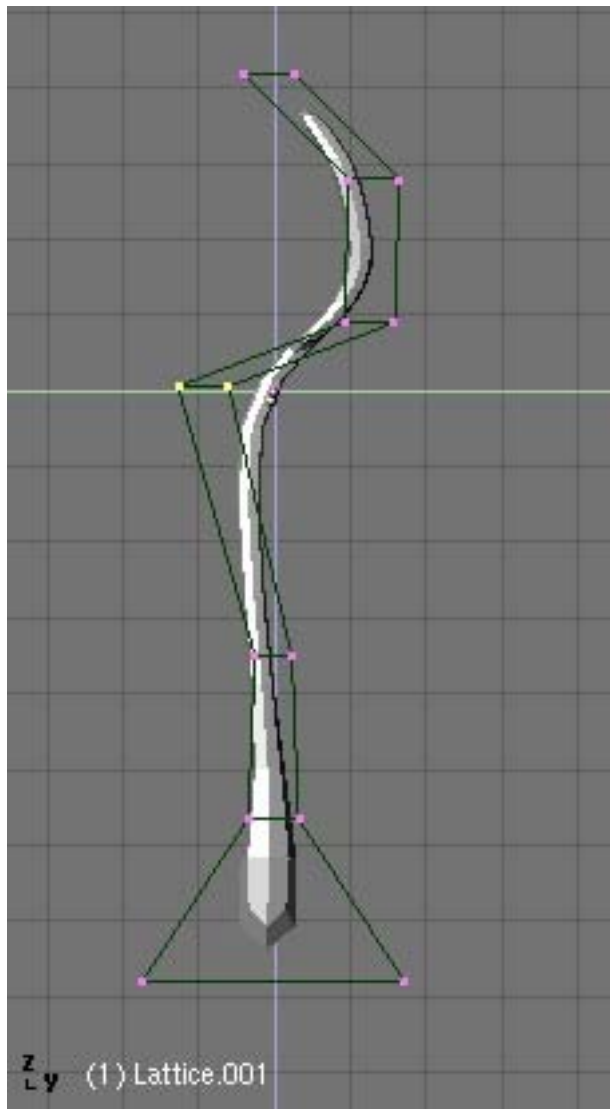
Figure 22-44. Deformar una lattice es un placer !



Observa que si mueves el tenedor arriba y abajo dentro de la Lattice, la deformación se aplicará en partes diferentes de la malla.

Una vez finalizado en Vista Frontal, cambia a la Vista Lateral. Selecciona y mueve diferentes secciones de vertices para darle al tenedor las curvas adecuadas ([Figure 22-45](#)).

Figure 22-45. Doblando cosas.

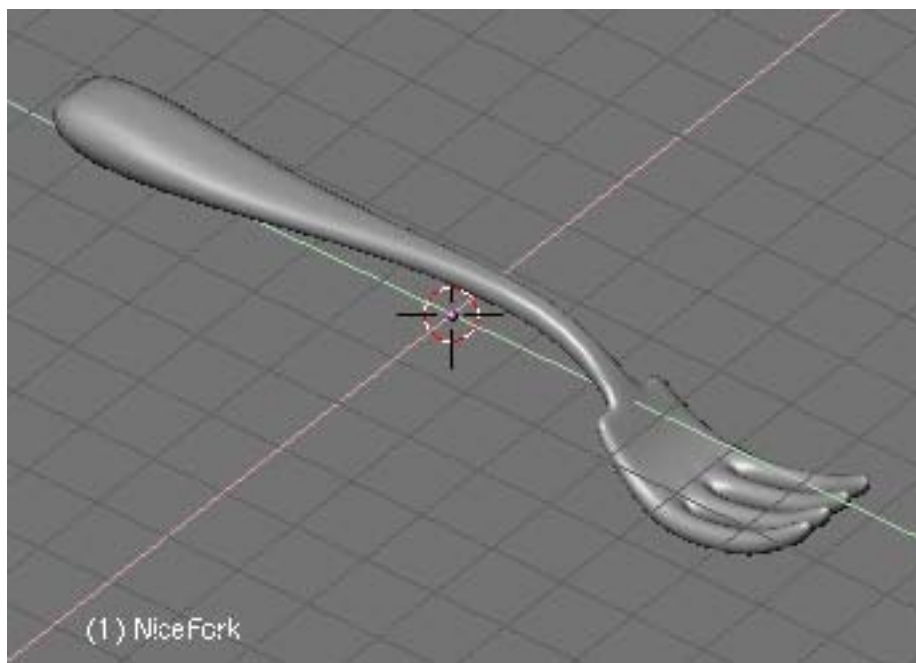


Podemos eliminar la Lattice si no vamos a añadir más objetos hijo. Pero antes de hacerlo, quizás querramos conservar las deformaciones ! Selecciona el tenedor y pulsa **CTRL-SHIFT-A** y luego click en la opción *Apply Lattice Deform?*.

Tip Vertices Locos

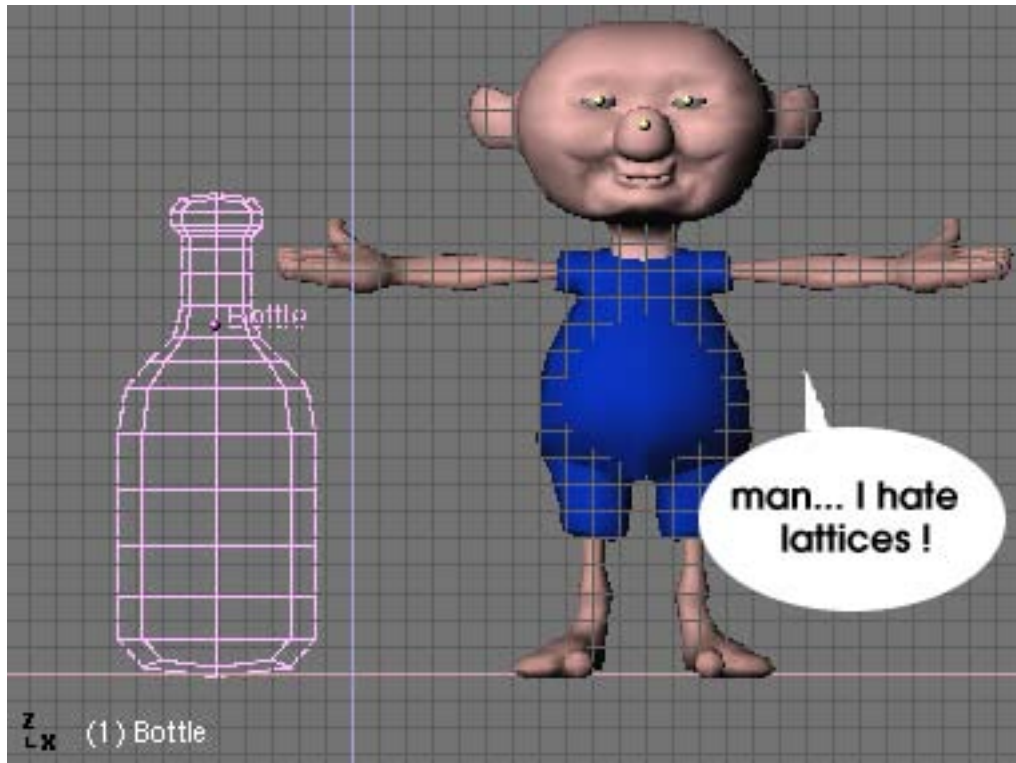
En raras ocasiones, en mallas relativamente complejas, la aplicación de **CTRL-SHIFT-A** hará que veamos la malla *com si* se hubiera arruinado totalmente, Esto es solo una ilusión. Simplemente sal y vuelve a entrar a Modo Edición (**TAB**) y la malla retornará a su deformación correcta.

Figure 22-46. Un lindo tenedor.

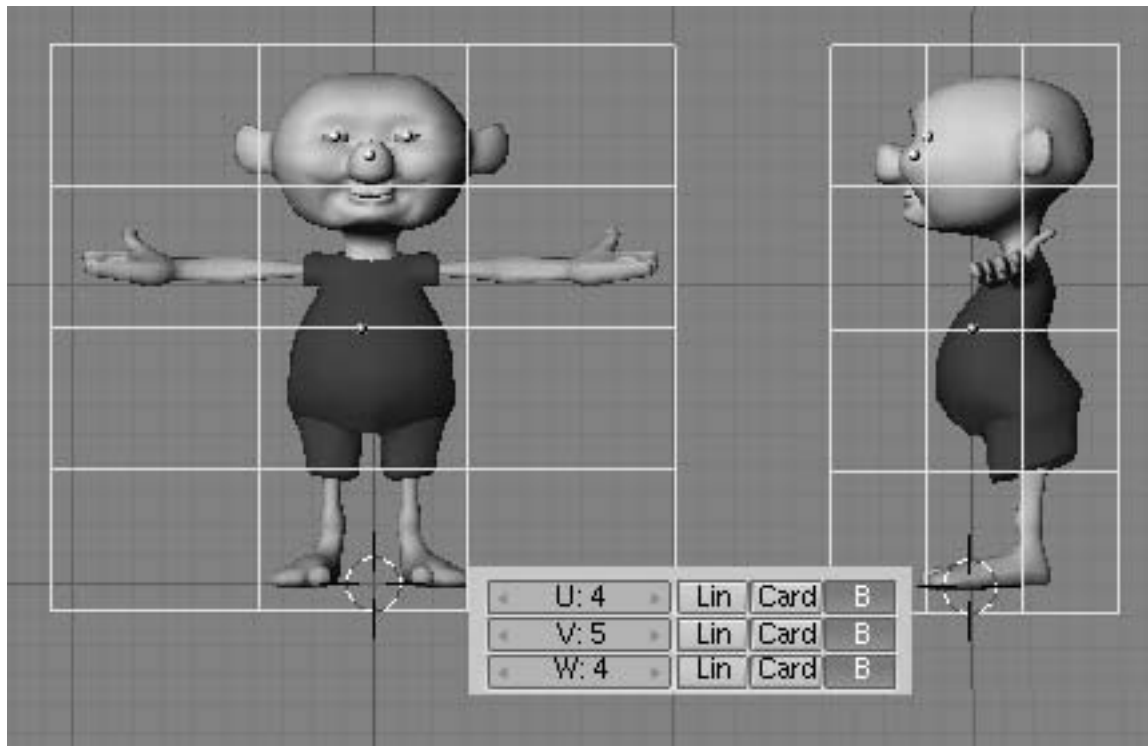


También podemos usar una Lattice para darle a un objeto la forma de otro objeto. Por ejemplo, echémosle un vistazo a la siguiente escena. He modelado una botella y quisiera atrapar al personaje dentro de ella. Se lo merece ([Figure 22-47](#)).

Figure 22-47. Pobre tipo...

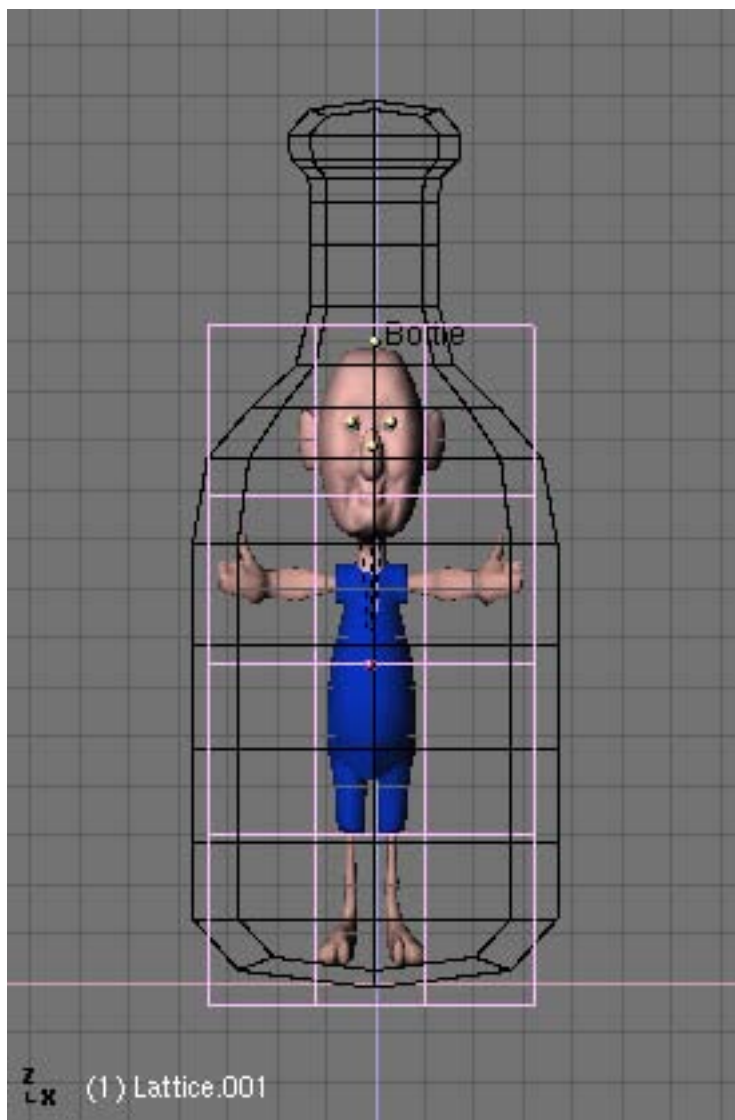


Agrega un Lattice alrededor del personaje. No uses una resolución muy elevada. Escala en X y en Y para envolver completamente al personaje ([Figure 22-48](#)).

Figure 22-48. Doblando cosas.

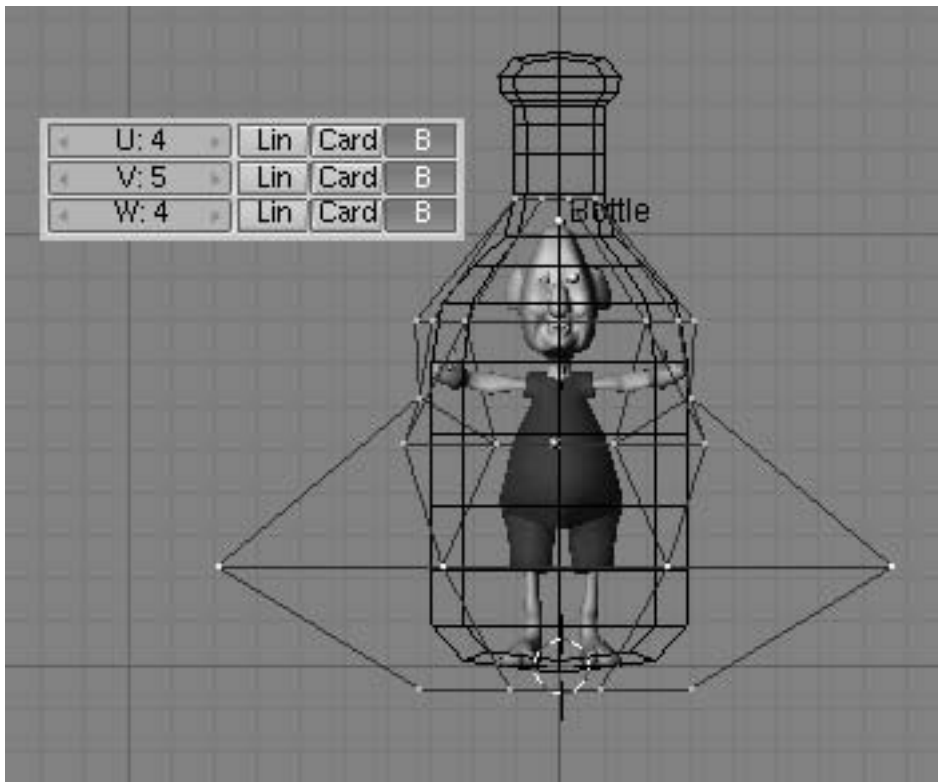
Emparenta el personaje a la Lattice y escalala una vez mas para ajustarla a las dimensiones de la botella ([Figure 22-49](#)).

Figure 22-49. Escala la rejilla para que calce la botella.



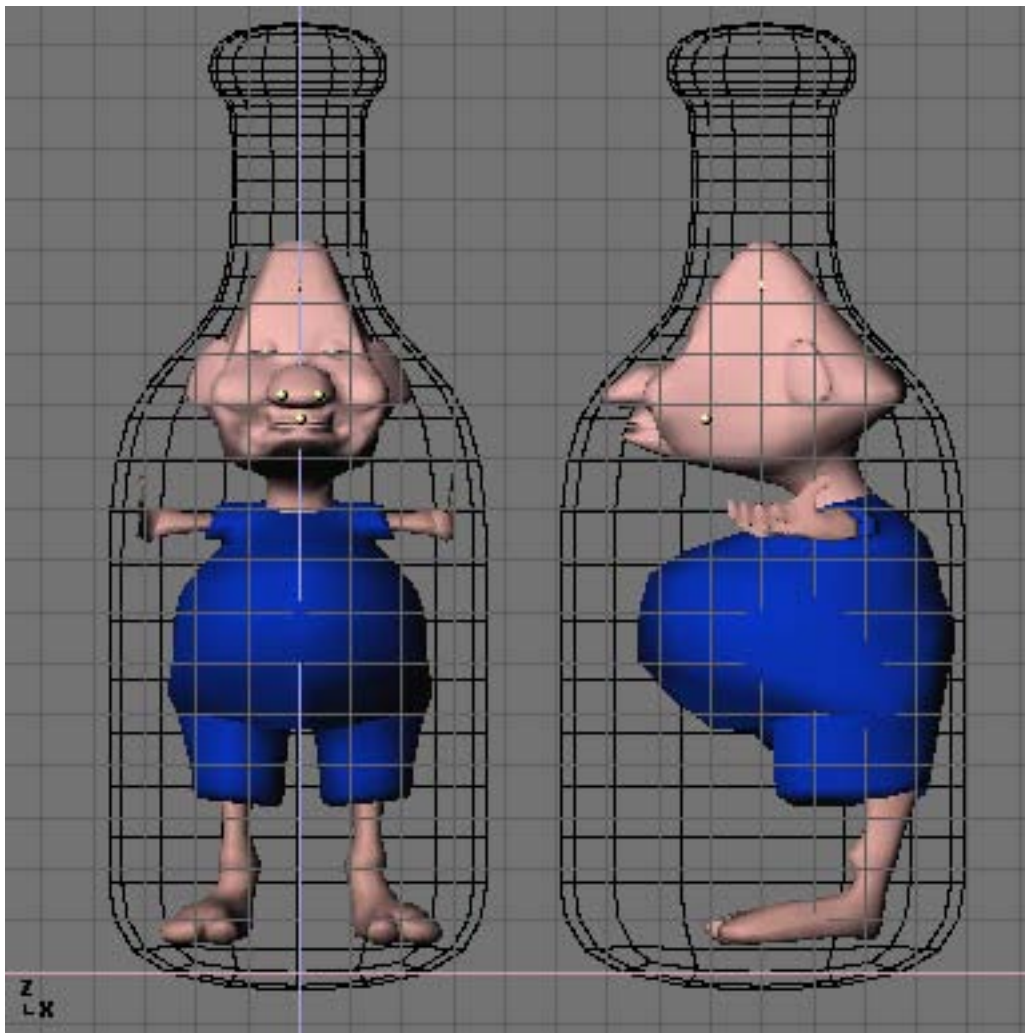
Entremos al Modo Edición para la Lattice. Pulsa el botón *Outside* en el panel *Lattice* Panel en el contexto de Edición, para ocultar los vertices internos de la rejilla. Luego los desocultaremos. Movamos y escalemos los vertices en las vistas lateral y frontal hasta que el personaje se ajuste perfectamente a la forma de la botella. ([Figure 22-50](#)).

Figure 22-50. Edita la Lattice para que el pequeñín entre cómodamente en la botella



Puedes seleccionar la Lattice y modelarla en una ventana 3D usando la Vista Local y ver el resultado en otra ventana usando la Vista Global para hacer la tarea más cómoda. ([Figure 22-51](#)).

Figure 22-51. ¿ Claustrofóbico ?



Si no hiésemos utilizado una lattice, hubiéramos tardado mucho más y habríamos tenido que trabajar mucho más moviendo vertices para deformar el personaje. ([Figure 22-52](#)).

Como las lattices soportan RVK para animación de vertices, se pueden conseguir efectos muy interesantes con esta herramienta.

Figure 22-52. Render Final. Creanme, se lo merecía!



Las Lattices se pueden utilizar en muchas aplicaciones que requieran una deformación de malla "casi líquida". Piensa por ejemplo en un genio saliendo de su lámpara o los ojos de un personaje de tiras cómicas con sus ojos saliendo exageradamente de sus cuencas. Diviertete !

[Prev](#)

DupliFrames

[Home](#)

[Up](#)

[Next](#)

Efectos Volumétricos

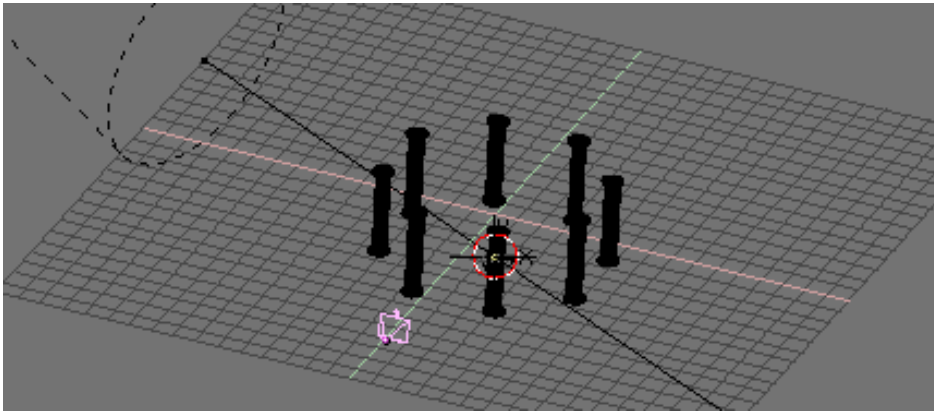
Chapter 23. Efectos Volumétricos

Relativo a Blender v2.31

A pesar de que blender ofrezca una opción bastante buena de niebla en los parámetros Reales (World) para dar a las imágenes algo de profundidad, es posible que desee crear efectos volumétricos reales; niebla, nubes y humo que parezca que realmente ocupan algo de espacio.

[Figure 23-1](#) muestra un escenario con algunas columnas colocadas en anillo, con algunos materiales de su elección para las columnas y la tierra, y un color que defina el cielo global.

Figure 23-1. Columnas sobre un plano.



[Figure 23-2](#) muestra el render relativo, donde [Figure 23-3](#) muestra un render con la niebla incorporada de Blender. Las propiedades de la niebla en este caso particular son: *Linear Mist*, $Sta=1$, $Di=20$, $Hig=5$.

Figure 23-2. El render de un plano.

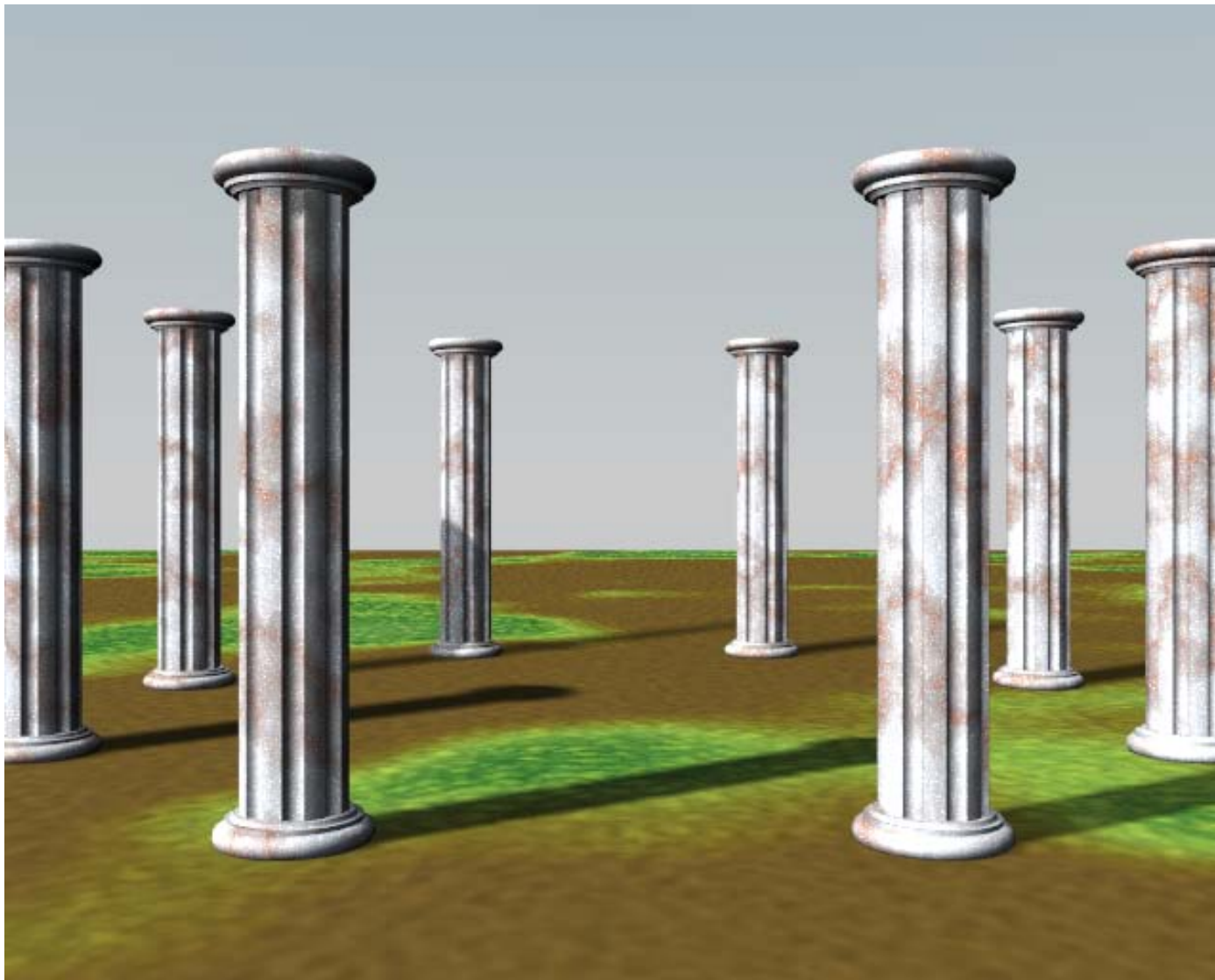
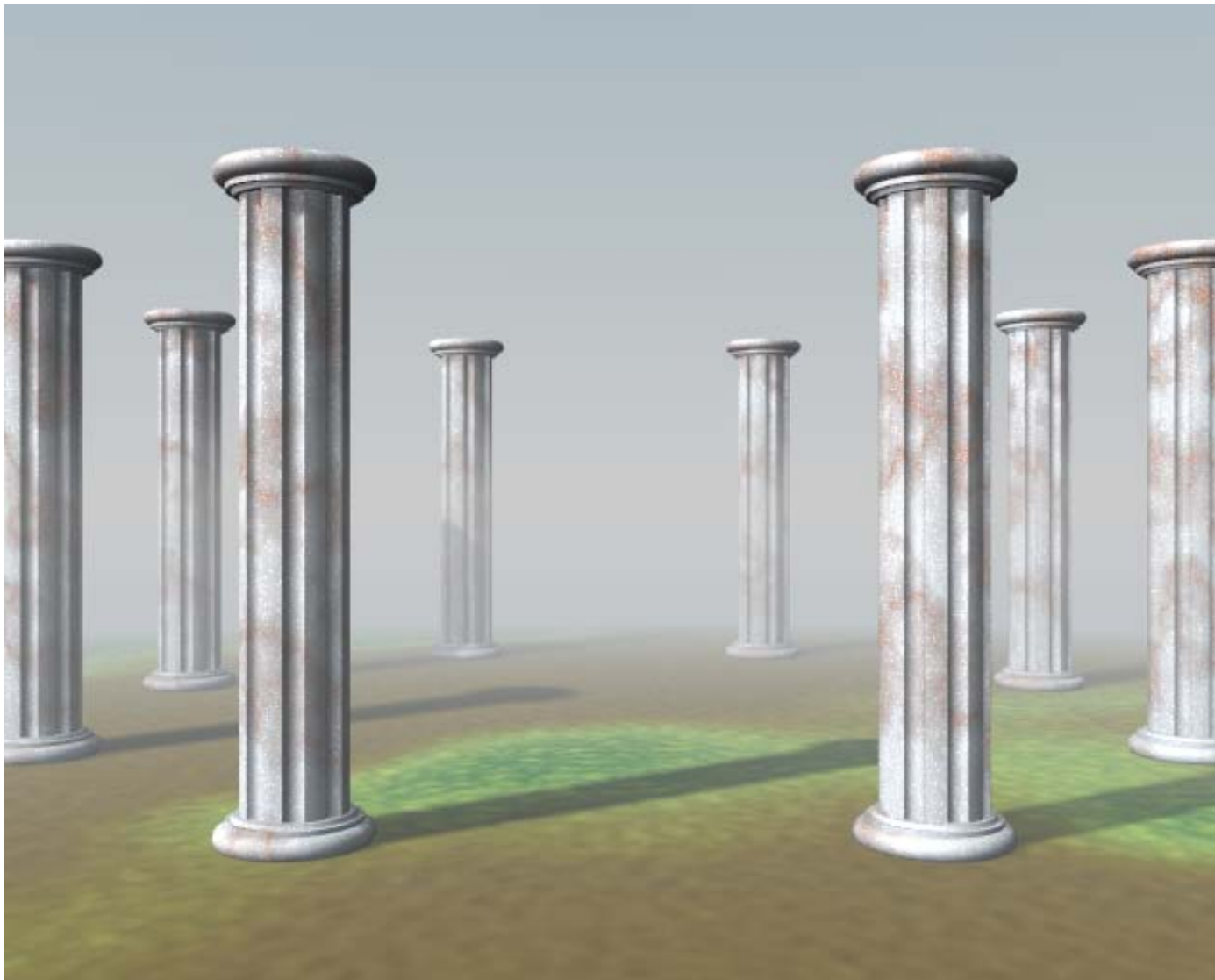
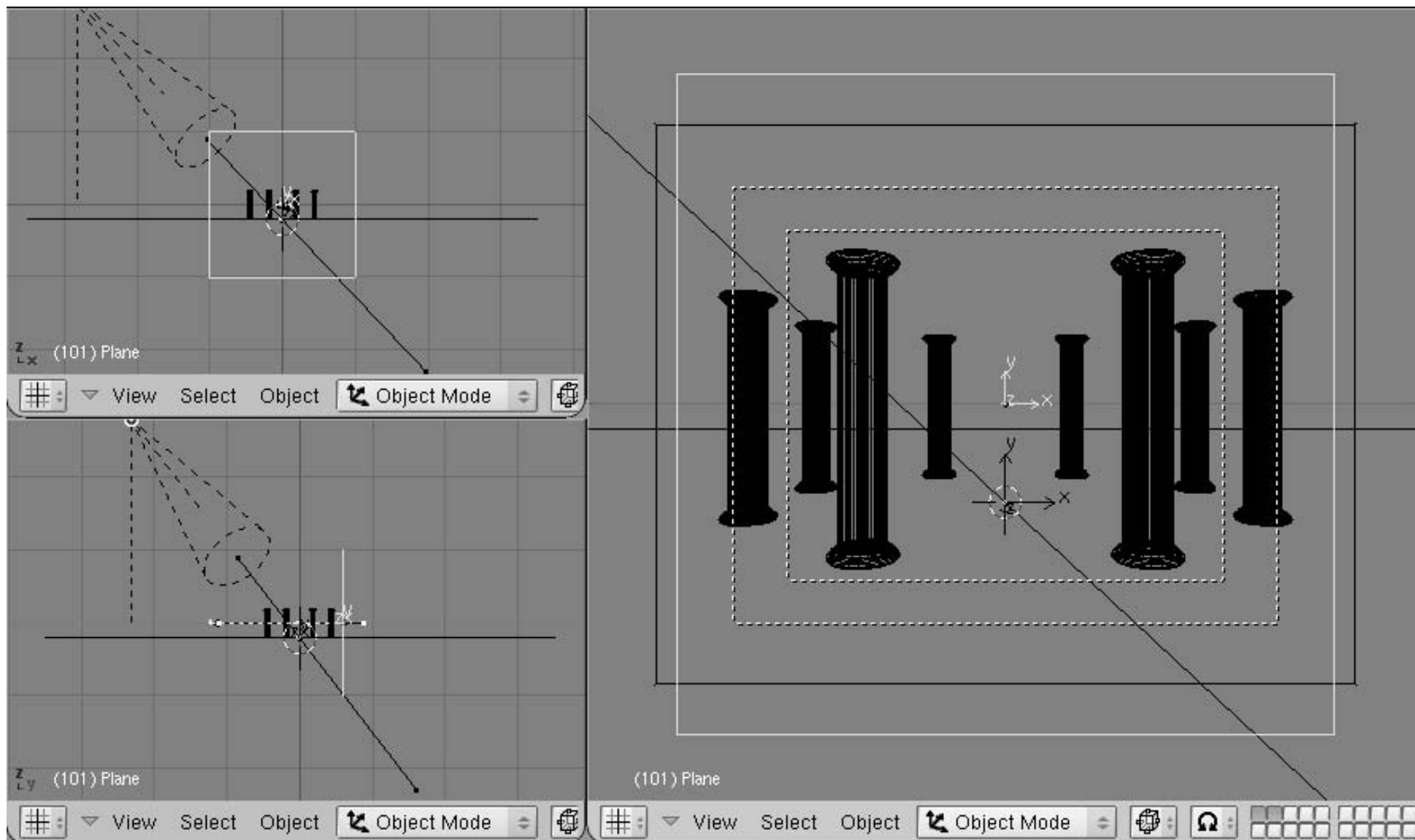


Figure 23-3. El render con la niebla incorporada de Blender.



Pero nosotros queremos crear una niebla mucho mas impactante, con remolinos, y lo mas importante, que no sea uniforme. Las texturas incorporadas de Blender (nubes por ejemplo) son intrinsecamente 3D, pero solo se renderizan cuando estan mapeadas sobre una superficie 2D. Conseguiremos un render "volumetrico" haciendo un muestreo de la textura en una serie de planos paralelos entre si. Cada uno de nuestros planos, exhibira, por lo tanto, una textura de Blender estandar en su superficie 2D, pero el efecto global sera un objeto 3D. Este concepto sera aclarado en el ejemplo siguiente.

Con la camara en $z=0$, mirando hacia delante, cambie a la vista frontal y añada un plano enfrente de la camara, cuyo centro este alineado con el punto de vista de la camara. En la vista lateral mueva el plano hasta donde quiera que acabe el efecto volumetrico. En nuestro caso, cualquier sitio detras de la columna mas alejada. Escale el plano de forma que abarque todo el campo de vision de la camara ([Figure 23-4](#)). Es importante que la camara apunte hacia el eje y , ya que necesitamos que los planos sean ortogonales a la linea de vision. En cualquier caso, podremos moverlos despues.

Figure 23-4. Configuración del plano.

Después de haber comprobado que nos encontramos en el frame 1, es el momento de poner un KeyFrame Loc (**IKEY**). Ahora deberíamos movernos al frame 100, y movemos el plano mucho más cerca de la cámara, e insertamos otro KeyFrame Loc. Ahora pasamos al menú Objeto, panel *Anim Settings (Propiedades de animación)* (**F7**) Presionamos el botón *DupliFrame (Duplicar Frame)*.

La ventana 3D, en la vista lateral, mostrará ahora algo como [Figure 23-5](#). Esto no es buen resultado, ya que los planos son más densos al principio y al final de todo el movimiento. Con el plano aún seleccionado cambie una ventana a una ventana IPO (**SHIFT-F6**). Habrá ahora tres Loc IPO, solo uno no es constante. Selecciónelo, cambie al modo edición (**TAB**) y seleccione ambos puntos de control. Ahora cambielos de suavizado a nitido con (**VKEY**) [Figure 23-6](#).

Figure 23-5. El plano con frame duplicado (Dupliframe).

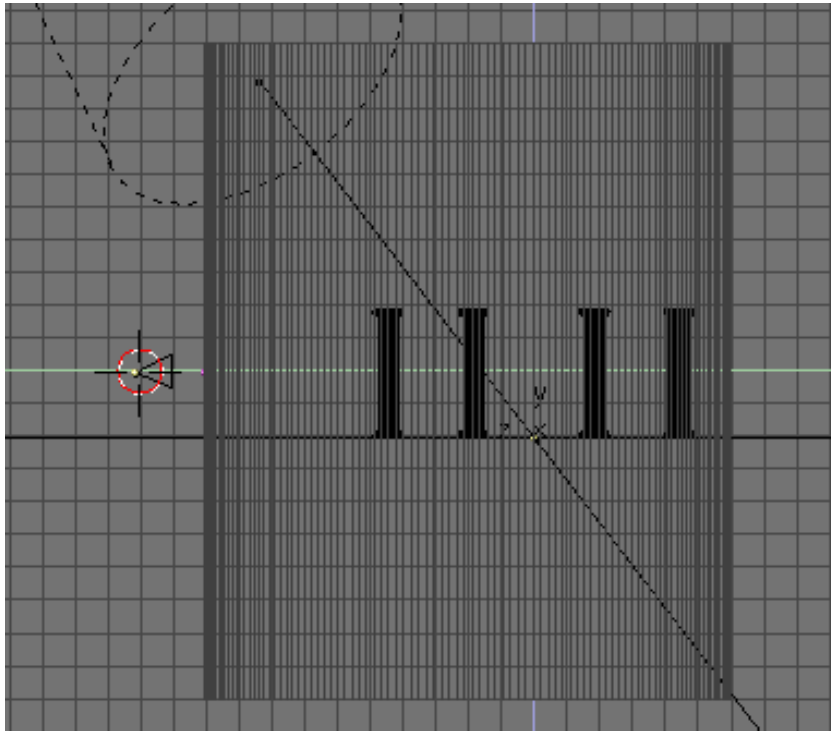
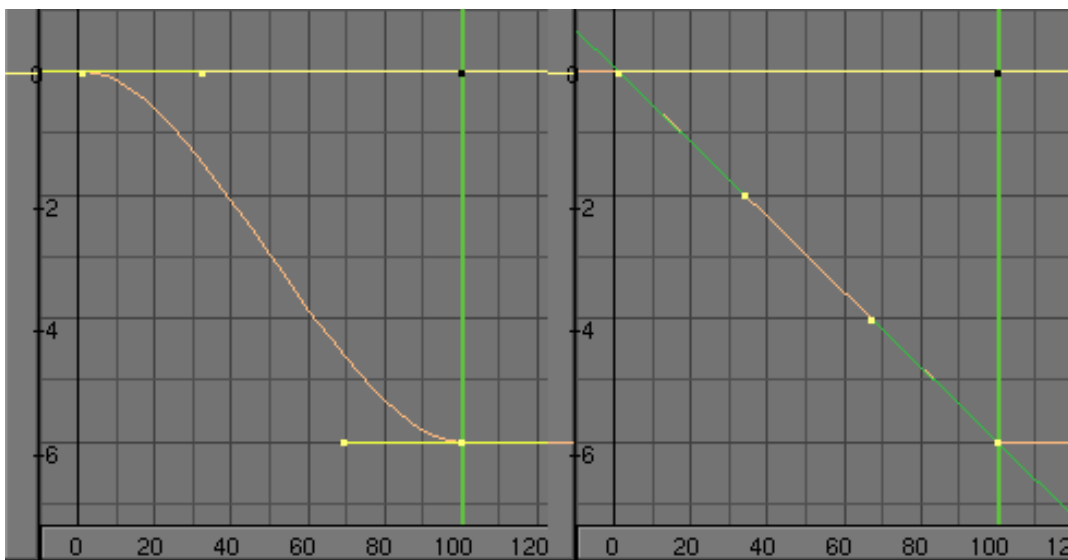


Figure 23-6. Reformando el frame duplicado del plano IPO.



Los planos se verán ahora como en [Figure 23-7](#). Relacione los planos con frame duplicado a la cámara (seleccione el plano, **SHIFT** seleccione la cámara, **CTRL-P**). Ahora tiene una serie de planos que automáticamente siguen la cámara, siempre orientados perpendicularmente a ella. Por ahora podría mover la cámara si lo desea.

Figure 23-7. Reformando el frame duplicado del plano IPO.

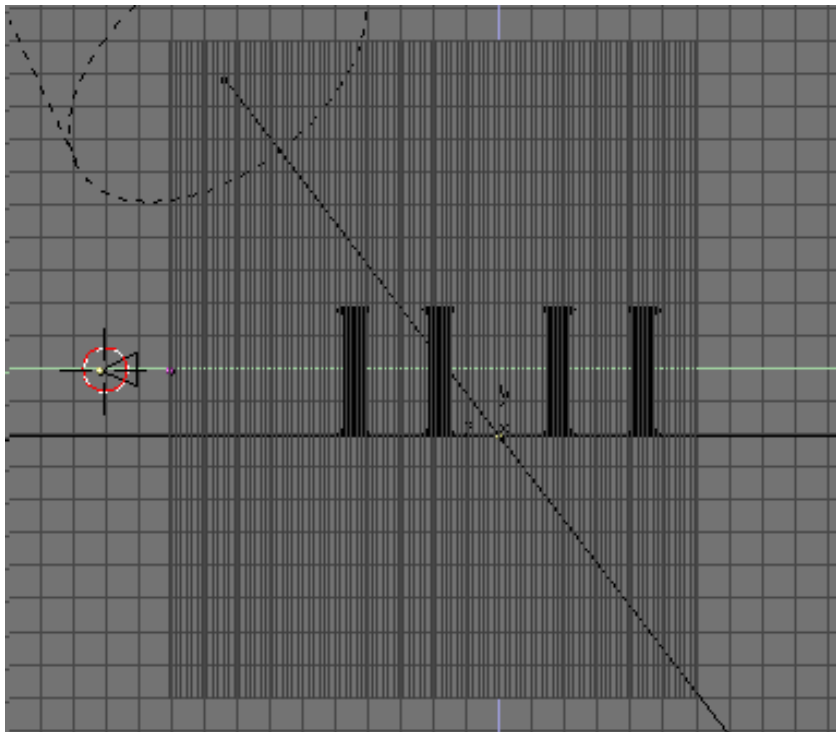
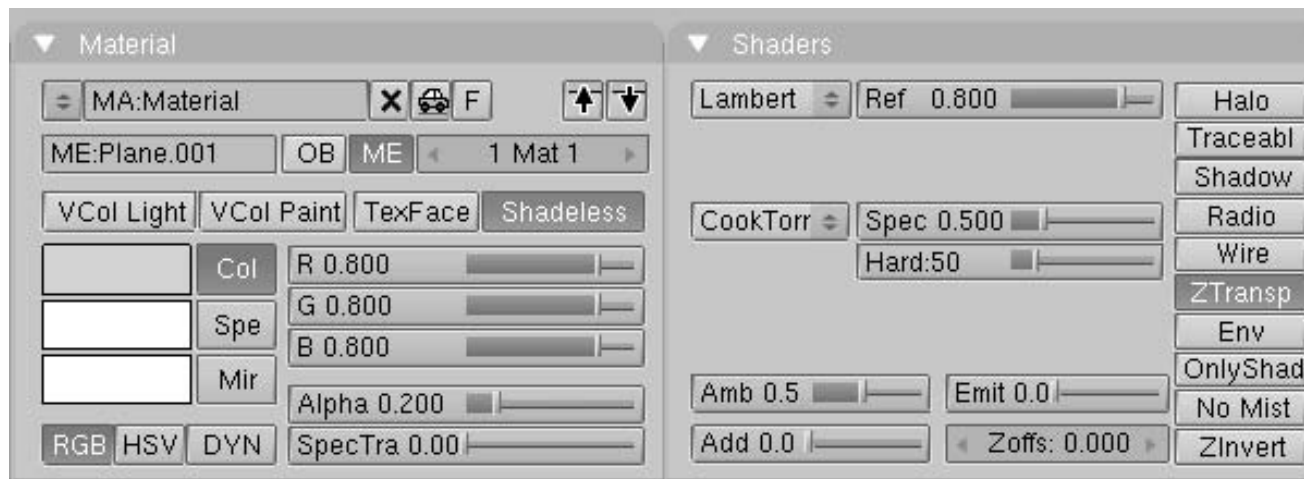


Figure 23-8. Propiedades basicas del material.



Ahora debemos añadir el material de niebla en si mismo. El material debería ser de pocos matices y no proyectar sombras para no llegar a efectos no deseados. Debería tener un pequeño valor alfa ([Figure 23-8](#)). Un material como este actúa básicamente como el incorporado en Blender en niebla, por lo tanto, no deberíamos ver mejora en el resultado de la imagen. El inconveniente es que procesar 100 capas transparentes es costoso para la CPU, especialmente si queremos los mejores resultados en un render unificado.

Tip Previsualización rápida

Puede usar el botón numérico *DupOff*: en el panel *Anim Settings* para desactivar algunos planos, y así conseguir una previsualización más rápida pero de peor calidad de lo que vamos haciendo. Para el render final, deberíamos volver a poner *DupOff* de nuevo a 0.

¡Ponga atención al valor alfa! Cuantos menos planos use, más fina se verá la niebla, de forma que el render final será más 'neblinoso' que las previsualizaciones.

La cualidad realmente interesante podrá verla cuando añada texturas. Necesitaremos al menos dos: Una para limitar la niebla en vertical y mantenerla en el suelo; la segunda para hacerla no uniforme y con algo de variación de tono.

Como primera textura, añada una textura Blend (mezcla) de tipo "lineal", con una línea de color simple, desde blanco puro, *Alpha*=1 en la posición 0.1 a puro blanco, *Alpha*=0 en la posición 0.9 ([Figure 23-9](#)). Añádalo solo en el canal alfa como multiplicador (Botón *Mul*) ([Figure 23-10](#)). Para hacer que la niebla sea coherente con los movimientos de la cámara, y los planos lo sigan, tenemos que activar *Global*. Esto será así también para todas las demás texturas y hará que los planos hagan de una textura volumétrica 3D. Si está preparando una animación verá una niebla estática, respecto a la escena, mientras la cámara se mueve. Cualquiera de las otras propiedades de textura deberían mostrar una niebla que es estática respecto a la cámara, por lo tanto será siempre la misma mientras la cámara se mueve, lo que es bastante irreal.

Figure 23-9. Límite de altura de la textura.

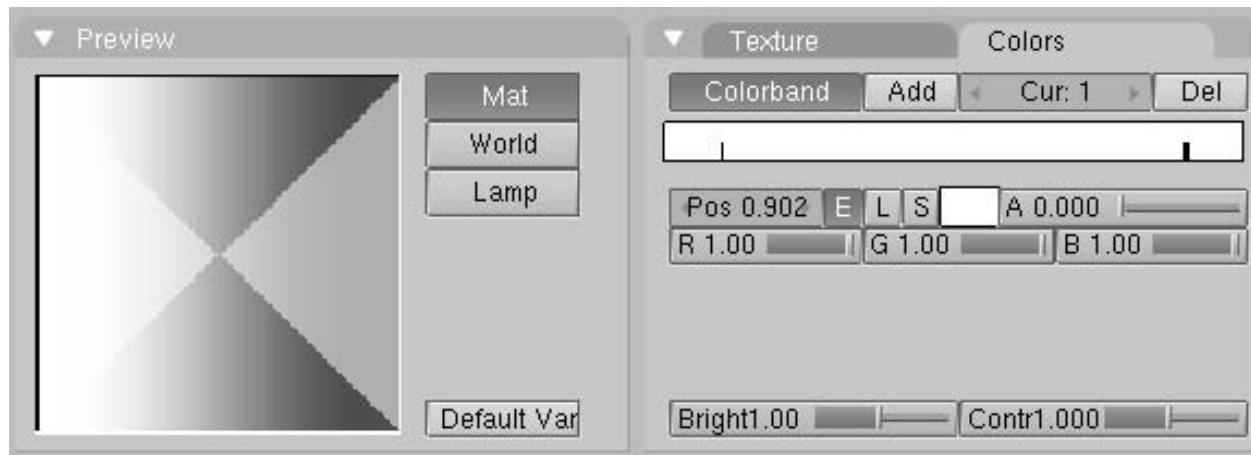
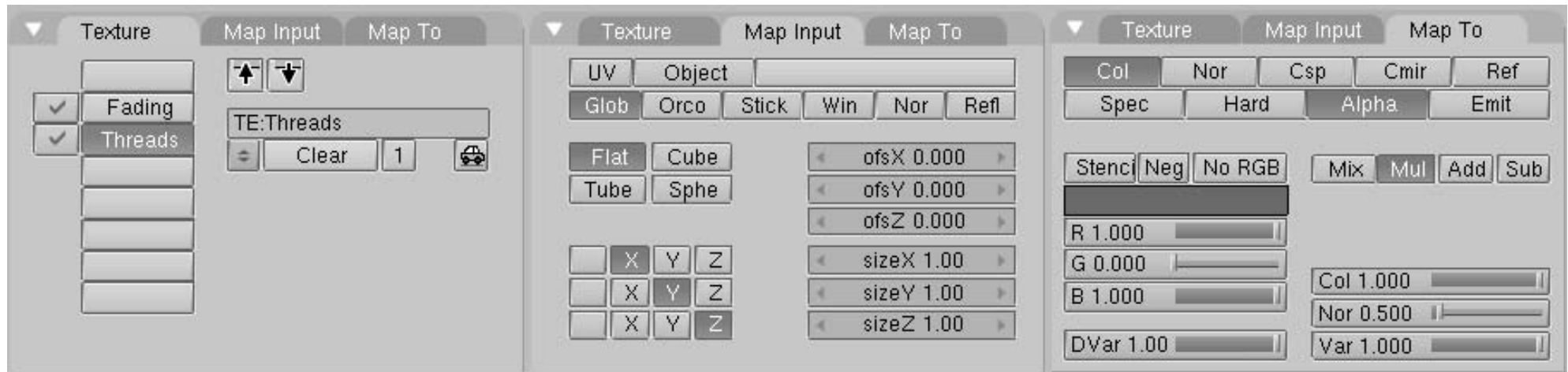


Figure 23-10. Basic Propiedades de material para textura de nubes.



En cualquier caso, si quiere tener un movimiento, rizado, o cambio en la niebla puede hacerlo con la animacion de la textura, como hemos explicado antes.

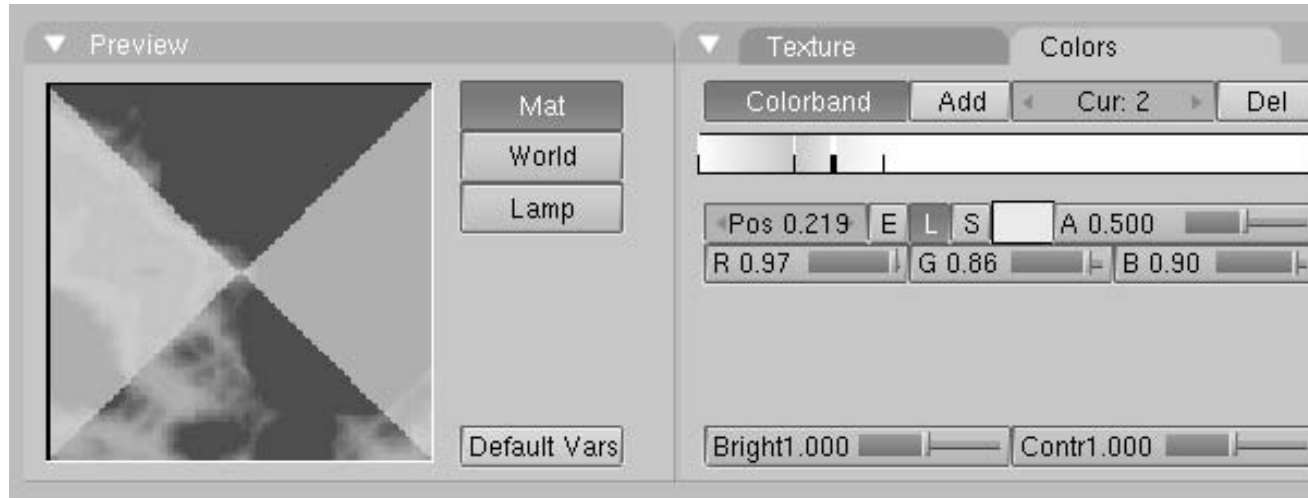
La textura de mezcla trabaja en las direcciones X e Y, asi que si quiere extenderla verticalmente en coordenadas globales, tendra que remapearla ([Figure 23-10](#)). Dese cuenta que la mezcla desde *Alpha*=1 hasta *Alpha*=0 sera desde el global *z*=0 hasta el global *z*=1 a menos que se añadan offsets y escalados. Para nuestros objetivos, los valores estandard son correctos.

Si ahora hace un render, sera indiferente donde esten situados los planos y la camara. La niebla sera densa debajo de *z*=0, inexistente sobre *z*=1 y con un degradado entre medias. Si se lia con esta "complejidad aparente", piensa en que podria conseguir con una textura *Orco* normal, y con planos sin emparentar. Si tiene que mover la camara, especialmente en animaciones, los resultados pueden volverse muy pobres tan pronto como los planos dejan de estar perpendiculares a la camara. Deberia acabar sin niebla al final si la camara deja de estar paralela a los planos.

La segunda textura es la que da la verdadera ventaja sobre la niebla integrada. Añada una textura de nube, haga su valores de *Noise Size*=2, *Noise Depth*=6 y

Hard Noise En ([Figure 23-11](#)). Añada una linea de color tambien, desde blanco puro con *Alpha*=1 en la posicion 0 a un gris azulado palido con *Alpha*=0.8 en una posicion alrededor de 0.15, hasta un tono rosaceo con *Alpha*=0.5 alrededor de la posicion 0.2, terminado con blanco puro en la posicion 0.3 *Alpha*=0, Por supuesto, deberia ir a un amarillo verdoso para nieblas de selva, etc.

Figure 23-11. Propiedades de la textura de nube.



Use esta textura en ambos *Col* y *Alpha* como una textura *Mul*, dejando el resto de propiedades por defecto. Si ahora renderiza la escena las bases de sus columnas estaran ocultas por una bonita niebla ([Figure 23-12](#)). Desde cuenta que el render unificado da muchos mejores resultados aqui.

Figure 23-12. Render final.



Note Si esta trabajando en una animacion y desea que su niebla sea animada como si se moviera por el viento, es en la textura final en la que debe trabajar. Añada una textura de material IPO, asegurese de seleccionar el canal de textura correcto y añada IPO a los valores de *OfsX*, *OfsY* y *OfsZ*.

[Prev](#)

Modelando con lattices

[Home](#)[Up](#)[Next](#)

Cuerpos blandos

Chapter 24. Cuerpos blandos

24.1. Introducción

Relativo a Blender v2.37

El sistema SoftBody (cuerpos blandos) de Blender permite que los vértices se muevan basándose en las leyes de la física. Esto significa que pueden ser configurados para reaccionar a la gravedad y al viento. Los Objetos de Blender pueden ser configurados como cuerpos blandos. En la versión 2.37 únicamente están implementados los objetos *Mesh* y *Lattice*.

El sistema SoftBody está diseñado principalmente para mejorar los sistemas de animación, incluyendo la animación de personajes. Efectos como la piel flexible u ondulante son ahora muy fáciles de lograr.

En los archivos demo de la versión 2.37 (4 MB) pueden encontrarse dos ejemplos de cuerpos blandos, `softbody_basics.blend` y `wind_soft.blend` <http://download.blender.org/demo/test/test237a.zip>.

24.2. El sistema SoftBody de Blender

24.2.1. Las bases

Hay dos métodos principales para controlar el efecto de cuerpo blando:

Goal (objetivo) - *Goal* actúa como una etiqueta en un conjunto de vértices seleccionados, controlando cuanto efecto tiene el cuerpo blando en ellos.

Con *Goal* activo al máximo (1.0), el objeto actuará como cualquier objeto animado normal (sin efecto de cuerpo blando). Ajustando *Goal* a 0.0, el objeto es influido únicamente por las leyes de la física. Ajustando los valores de *Goal* entre 0.0 y 1.0, se puede combinar entre tener un objeto al que sólo afecta el sistema de animación, y tenerlo únicamente afectado por el efecto de cuerpo blando.

Goal sirve también como memoria, para asegurarnos de que los objetos blandos no se deforman demasiado, acabando en la forma animada no blanda.

Utilizando el sistema de pesos de *Vertex Group* (grupo de vértices), puede definirse un peso *Goal* por vértice. Para hacer que esto parezca más natural, pueden definirse fuerzas elásticas para controlar qué distancia pueden alejarse los vértices de su posición original.

Springs (fuerza de retorno a la posición original) - *Edge Spring Stiffness* (rigidez de la arista) define cuánto intentan las aristas mantener su tamaño original. Por ejemplo, añadiendo aristas diagonales dentro de un cubo, lo volveremos más rígido (menos "gelatinoso"). Ajustando el parámetro *E Stiff*, los objetos intentarán, en mayor o menor medida, mantener su forma original, pero continuarán ajustándose a las dinámicas.

Note Cuando activamos el efecto de Cuerpo Blando en un objeto, será siempre simulado con antelación en el tiempo. Al movernos hacia atrás en el tiempo o saltando en pasos superiores a 9 cuadros reiniciará el cuerpo blando a su posición original. Use el play back de la ventana TimeLine (línea de tiempo) para ajustar interactivamente los efectos de Cuerpo Blando.

Una vez esté satisfecho con la simulación, puede cocer (*Bake*) la simulación en un sistema de animación estático. Un cuerpo blando cocido se anima mucho más rápidamente en pantalla, y deja de depender de la simulación.

Note Se recomienda cocer los cuerpos blandos al renderizar animaciones, puesto que la simulación no funciona correctamente para renderizado con Motion Blur (borrosidad de movimiento), o para renderizado por fragmentos a través de un sistema de renderizado por red.

El uso de SoftBody para simulación de tela, especialmente con detección de colisión, está todavía en fase de pruebas. El código de colisión actual necesita mejoras, planeadas para una futura versión. Para un control más preciso sobre la simulación de tela, se añadirán opciones de arista especiales en una próxima versión. Actualmente, todas las aristas tienen un efecto idéntico en el cuerpo blando.

Puesto que los vértices en los Cuerpos Blandos son tratados como partículas, las opciones *Force fields* y *Deflectors* se aplican igualmente. Tenga en cuenta que deflection (colisiones) funciona únicamente en Mallas no deformadas (sin usar hooks, armaduras, lattices, etc). Veanse las notas de la versión para más información.

24.2.2. Interfaz

Se puede acceder a la interfaz de SoftBody mediante el panel *Object* (**F7**), bajo la pestaña *Softbody*. Véase [Figure 24-1](#).

Figure 24-1. Ajustes de Softbody.



Ajustes de Softbody

- **Enable Soft Body** (habilita Cuerpo Blando) - Habilita el sistema softbody en el objeto seleccionado.
- **Bake settings** (opciones de cocido) - Abre las opciones de cocido. Véase [Figure 24-2](#).

- **Friction** (fricción) - Controla la cantidad de fricción que tiene el objeto. Un valor alto significa que las fuerzas, por ejemplo *Gravity* (gravedad) o *Wind* (viento), tienen más facilidad para tomar el objeto y moverlo.
- **Grav** - Gravedad, cantidad de fuerza en la dirección del eje Z negativo. El valor de la gravedad terrestre es aproximadamente 9.8.
- **Error Limit** (límite de error) - Límite *Runge-Kutta*. Define el tamaño del paso durante la simulación. Valores pequeños dan una precisión mayor, pero incrementan el tiempo de cálculo.
- **Mass** (masa) - Valores de masa para los vértices. Mayor masa frena el movimiento, excepto para la gravedad, donde el movimiento es fijo independientemente de la masa.
- **Speed** (velocidad) - Con este valor puede controlarse la temporización interna del sistema softbody.
- **Apply Deform First** (aplicar la deformación primero) - Aplica los cálculos de softbody después de otras deformaciones, tales como las causadas por Lattices o Armaduras.
- **Use Goal** (usar objetivo) - Usa el movimiento de animaciones para la simulación (Ipo, Deform, Parents, etc). "Goal" es la posición final deseada para los vértices basada en esta animación. Cómo tratará el softbody de adquirir este objetivo puede ser definido usando fuerzas de flexibilidad y atenuación de la oscilación.
- **Goal** (objetivo) - El peso objetivo por defecto para todos los vértices cuando no hay grupo de vértices asignado (Vertex Group). Si hay grupo de vértices asignado, este botón muestra el nombre del grupo de vértices objetivo.
- **G Stiff** - Rigidez del objetivo. Un valor pequeño crea una fuerza de retorno muy débil ("agarre" más flexible al objetivo), un valor alto crea una fuerza de retorno fuerte ("agarre" más rígido al objetivo).
- **G Damp** - La atenuación de oscilación del objetivo. Valores altos atenúan el efecto elástico del objetivo en el cuerpo blando.
- **GMin/GMax** - Cuando se pintan los valores en grupos de vértices (usando el modo WeightPaint - pintado de pesos), pueden usarse GMin y Gmax para afinar (fijar) los valores de peso. El menor peso de vértice (azul) será GMin, el valor más alto (rojo) será GMax.
- **Use Edges** (usar aristas) - Las aristas de un objeto Malla (si las hay, comprobar en Editing->Mesh Panel) pueden usarse también como elementos flexibles con fuerza de retorno.
- **Stiff Quads** (rectángulos rígidos) - Para caras rectangulares, las aristas diagonales se usan con fuerza de retorno. Esto evita que las caras se aplasten por completo.
- **E Stiff** - Rigidez de las aristas (cuánto se esturan las aristas). Un valor bajo significa fuerzas de retorno muy débiles (material muy elástico), un valor alto es una gran fuerza de retorno (material más rígido).
- **E Damp** - Atenuación de la oscilación de las aristas. Valores altos atenúan el efecto E Stiff.

Figure 24-2. Ajustes de Bake.



Ajustes de Bake

- **Start/End** (principio/fin) - Configura el rango del cuerpo blando a ser cocido.
- **Interval** (intervalo) - Indica el número de cuadros entre cada "paso" de cocción (La "resolución" del resultado cocido). Las posiciones intermedias serán calculadas usando los pasos como cuadros clave, con interpolación B Spline.
- **Bake** (cocer) - Inicia el proceso de cocción. Dependiendo de la complejidad, esto puede demorar un poco. Puede pulsarse **ESC** para detener la cocción. Una vez cocido, este botón cambia a un botón "Free Bake" (liberar la cocción). Debe liberarse el resultado cocido para modificar los ajustes de cuerpo blando.

24.2.3. Ejemplo

Nuestro ejemplo mostrará un modo de hacer una simple bandera ondeando al viento.

Cree un plano en vista frontal y subdivídalo 3 veces. Vaya a los botones de Editing **F9** y active **Subsurf**. Coloque subsurf level a **2** para mejores resultados. Presione **Set Smooth**.

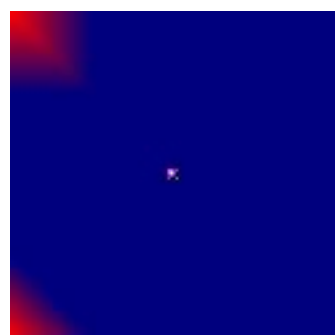
Crearemos ahora dos puntos de sujeción en nuestra bandera en la esquina superior e inferior de nuestro plano.

- Cree un nuevo Vertex Group, y ajuste *Weight* a **0**. Seleccione todos los vértices, y presione **Assign**.
- Ahora, seleccione la esquina superior e inferior de un lado de la bandera. Coloque *Weight* a **1.0**, y presione **Assign** nuevamente. Esto hará que estos vértices se mantengan donde están durante la simulación de softbody. En modo *Weight Paint* debería verse algo como esto [Figure 24-3](#).
- Ahora, salga de modo Edit, y vaya al panel Softbody en los botones Object **F7**. Haga clic en **Enable Soft Body**. Incremente *Grav* a **9.8**. Active el botón *Use Goal*. Haga clic en el pequeño botón junto a *Use Goal* y elija el nombre del *Vertex Group* a usar como objetivo; en este caso, la

única opción debería ser el nombre por defecto *Group*. Ahora ajuste la rigidez de arista *E Stiff* a **0.9**, coloque *Mass* a **0.5**, *Friction* a **0.14** y *Speed* a **2**.

- Ahora, puede presionar **ALT-A** para ver la bandera reaccionando a la gravedad.

Figure 24-3. Ejemplo de configuración de pesos.



Ahora añadiremos algo de viento a la simulación.

- Añada un objeto empty a la escena donde estará situada la fuente de viento. Seleccione la pestaña *Particle Interaction* y active el botón *Wind*. Ajuste *Strength* a **1**. Ahora rote y mueva el empty de tal modo que el eje Z apunte hacia la bandera. Véase [Figure 24-4](#).

Tip Puede incrementar el valor *Strength* temporalmente para poder ver el efecto del viento más fácilmente.

- Ahora, puede pulsar **ALT-A** para ver la bandera reaccionando al viento.
- Añadir una curva IPO para simular la fuerza cambiante del viento dotará a la animación de un mayor realismo. Véase [Figure 24-5](#).

Figure 24-4. Ejemplo de configuración de viento.

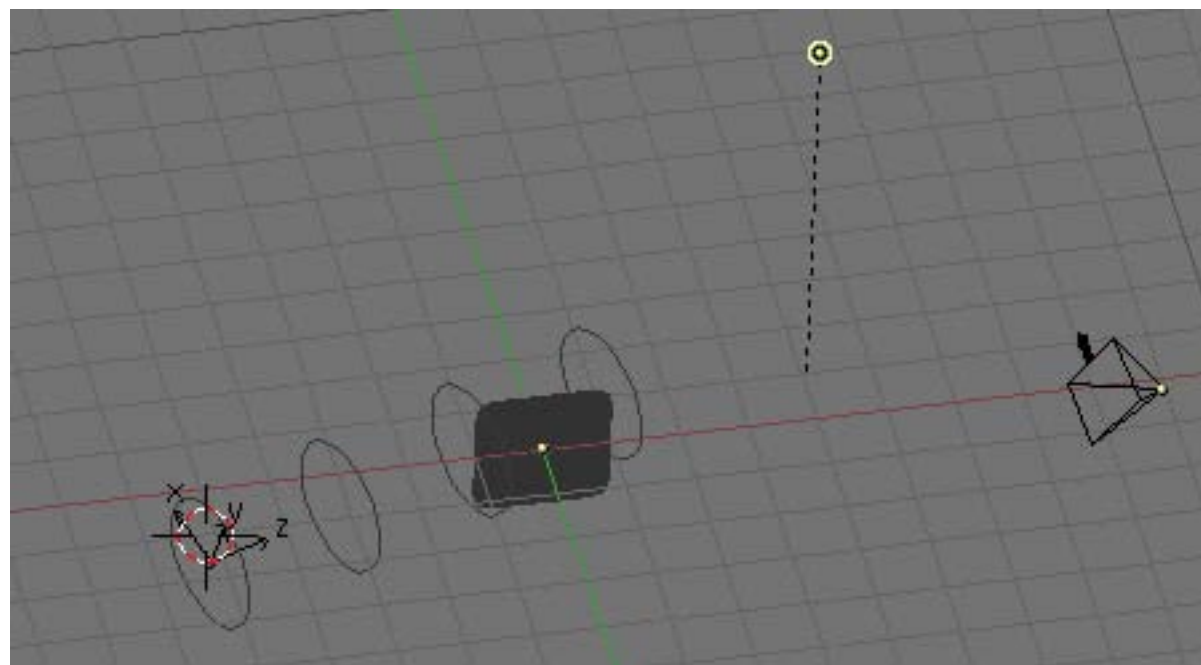


Figure 24-5. Ejemplo de IPO de fuerza del viento.



[Prev](#)

Cuerpos blandos

[Home](#)

[Up](#)

[Next](#)

Editor de secuencia

Chapter 25. Editor de secuencia

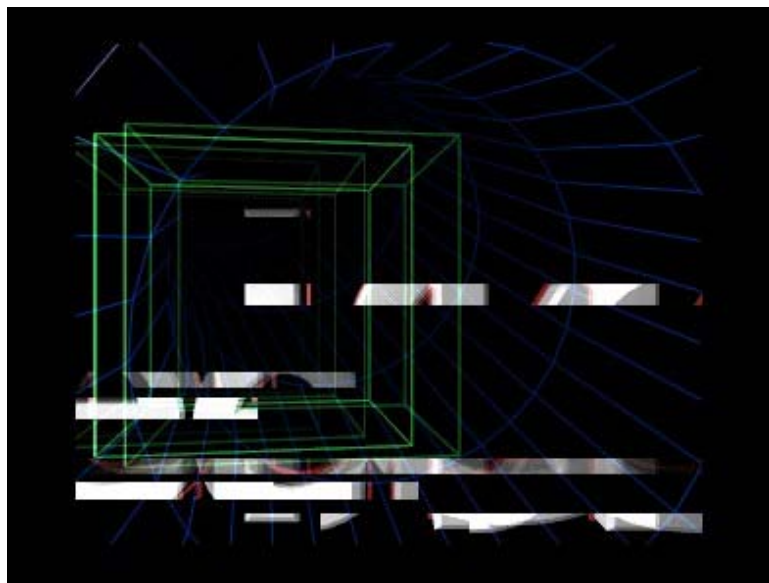
Una de las funciones que quizá sea subestimada en el editor de Blender es el editor de secuencia. Es un editor de vídeo completo que permite combinar múltiples canales de vídeo y efectos. Incluso aunque tenga un numero limitado de operaciones, puede usarlos para crear ediciones de vídeo bastante potentes (especialmente cuando lo combina con el poder de animación de Blender) Y, mas allá, es extensible mediante un sistema de plugin similar al sistema de plugins de Texturas.

25.1. Conociendo el editor de secuencia

Relativo a Blender v2.31

Esta sección muestra un ejemplo de edición de vídeo practico mostrando la mayoría de características del editor de secuencia. Vamos a usar juntas unas cuantas de las animaciones de Blender para obtener algunos impresionantes efectos. Un frame de la animación editada resultante esta en [Figure 25-1](#).

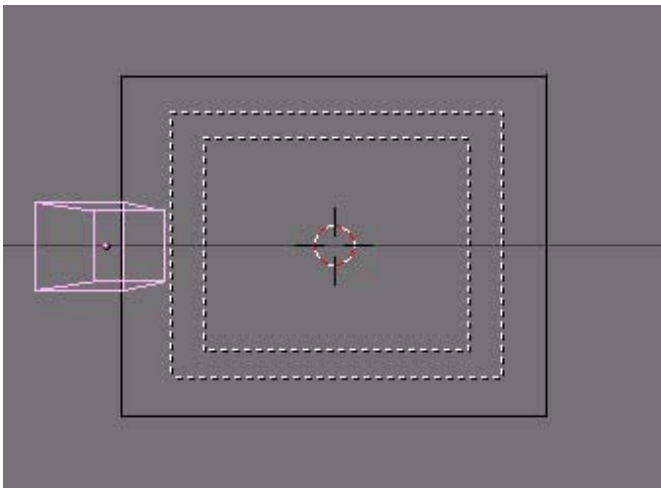
Figure 25-1. Resultado final.



25.1.1. Primera animación: dos cubos

Podemos empezar con algo simple y ver donde nos lleva. Empiece con una escena vacía de Blender y elimine el plano por defecto. Divida la ventana 3D en dos y cambie una de las vistas a la cámara con **NUM 0**. En la vista superior, añada un cubo y muévelo justo fuera del cuadrado punteado que indica la vista de cámara. [Figure 25-2](#).

Figure 25-2. Sacando el cubo de la vista de cámara.



Queremos crear una animación simple del cubo entrando en la vista, rotando, y despues desapareciendo. Defina el final de la animación en 61 (sitúe el valor *End*: en panel del menú Escena *Anim* Botones de render **F10**) e inserte un keyframe LocRot en el frame 1 con **IKEY** y seleccionando *LocRot* del menú que aparece. Esto guardará las dos localizaciones y la rotación del cubo en estos frames.

Vaya al frame 21 (presione **UPARROW (flecha arriba)** dos veces) y mueva el cubo cerca de la cámara. Inserte otro KeyFrame. En el frame 41, mantenga el cubo en la misma posición pero rotelo 180 grados e inserte otro KeyFrame.

Finalmente en el frame 61 mueva el cubo fuera del campo visual, hacia la derecha e inserte el ultimo KeyFrame.

Necesitaremos dos versiones de la animación: una con un material solido y otra con WireFrame (modo alambrico). Para el material, podemos usar un plano blanco iluminado por dos lamparas brillantes - una blanca y otra azul con un valor de energía de dos ([Figure 25-3](#)).

Para el cubo alambrico, ponemos el tipo de material a 'Wire' (alambrico) y cambiamos el color a verde. ([Figure 25-4](#)).

Figure 25-3. Un render del cubo solido.

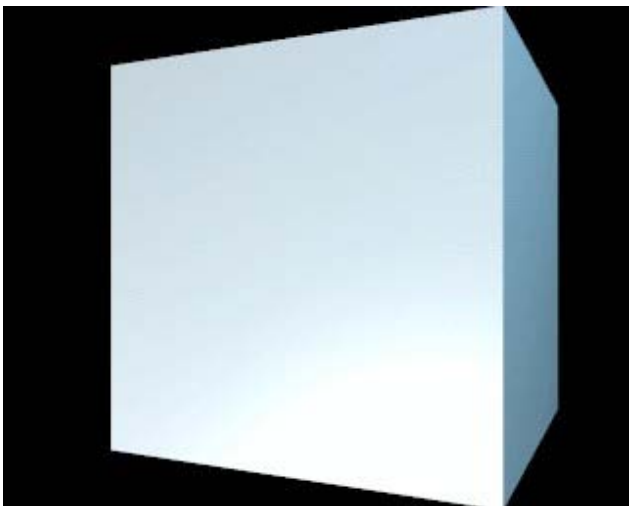
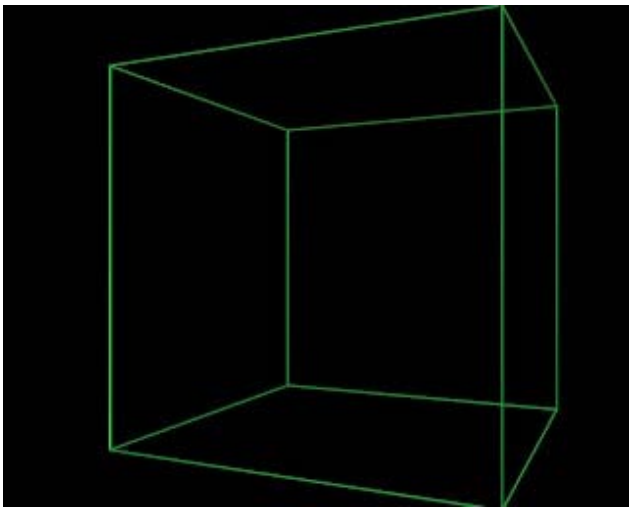
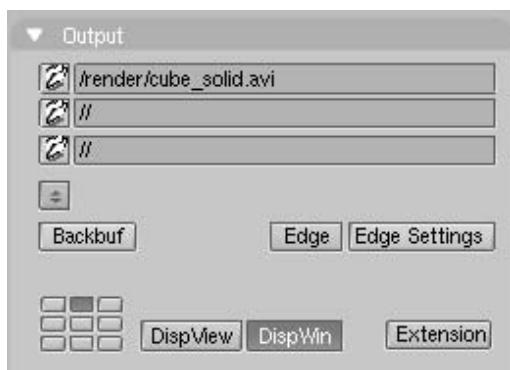


Figure 25-4. Y un render del cubo alamabrico.



Introduzca un nombre de archivo apropiado (por ejemplo 'cubo_solido.avi') en el campo *Pics* (primer cuadro de texto en la parte superior) del menú de escena, submenú render panel *Output* (**F10**) ([Figure 25-5](#)).

Figure 25-5. Establezca un nombre de archivo de salida.



Renderice la animación con el cubo solido blanco. Esto lo guardará en su disco. Guardelo como un archivo AVI. Use AVI Raw si es posible, ya que ofrece mayor calidad - La compresión debería ser el ultimo paso en el proceso de edición - sin embargo si dispone de poco espacio en disco use AVI Jpeg o AVI Codec, la primera menos comprimido y por lo tanto mayor calidad.

Ahora cambie el material al alambrico verde, renderice la animación de nuevo guardando el resultado en cubo_alambrico.avi.

Ahora tiene en su disco duro 'cubo_solido.avi' y 'cubo_alambrico.avi'. Será suficiente para nuestra primera edición de secuencia.

25.1.2. Primera secuencia: alambres retrasados

La primera secuencia usara solo la animación alambrica - ambas - para crear un interesante efecto. Crearemos multiples capas de vídeo, les daremos un pequeño desfase de tiempo y las añadiremos juntas. Esto simulara el efecto de 'cola difuminada' que puede ver en pantallas de radar.

Empiece con un archivo vacío de blender y cambie la ventana 3D a un editor de secuencia presionando **SÍNFITO F8** o seleccionando el icono del editor de secuencia  **Video Sequence Editor** de la cabecera del menú de ventana.

Añada un vídeo a la ventana presionando **SÍNFITO-A** y seleccione *Moví (Vídeo)* ([Figure 25-6](#)) o con la entrada de menú *Dad>>Moví*. En la ventana de selección de archivo que aparece seleccione la animación del cubo alambico que creó anteriormente.

Figure 25-6. Añadiendo un corte de vídeo

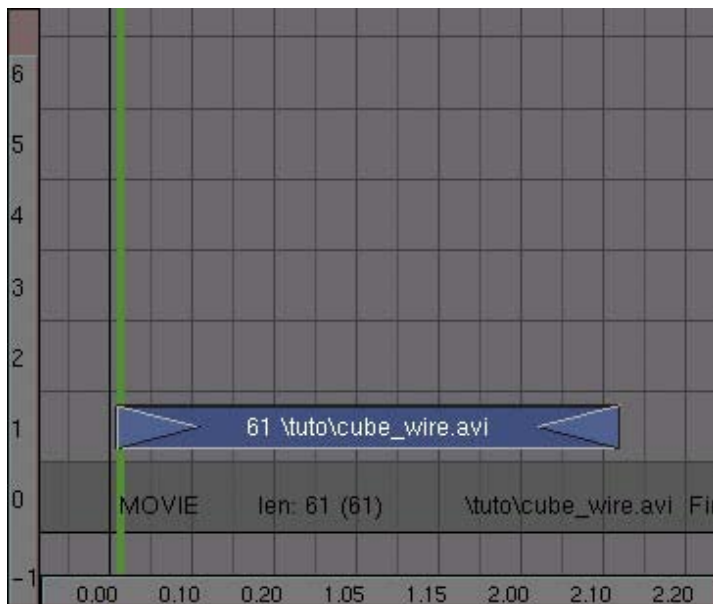


Después de que haya seleccionado y cargado el archivo de vídeo, verá un pequeño corte que la representa. Después de añadir el corte, estará automáticamente en el modo agarrar 'grab' y la franja sigue al ratón. El primer y último frame se muestran ahora en la barra.

Miremos la pantalla del editor de secuencia más de cerca. Horizontalmente verá el tiempo. Verticalmente, verá los canales de vídeo. Cada canal puede contener una imagen, un vídeo o un efecto. Encapando canales diferentes en la parte superior de cada uno y aplicando efectos, podrá mezclar diferentes orígenes juntos. Si selecciona un corte de vídeo, su tipo, longitud y nombre de archivo se mostrarán en la parte inferior de la ventana.

Mueva el corte de vídeo y deje que comience en el frame 1. Coloquelo en el canal 1, este está en la columna del final y presione **LMB** para finalizar ([Figure 25-7](#)).

Figure 25-7. Colocando el corte.



Tip Entradas, salidas y fotogramas

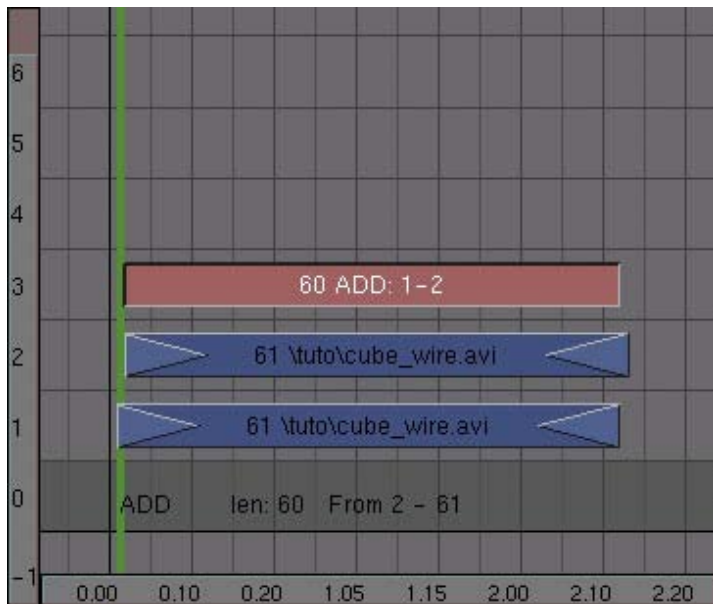
Puede añadir frames *lead-in* (entradas) y *lead-out* (salidas) seleccionando los triángulos al inicio y al final del corte (se volverán color púrpura) y arrastrarlos fuera. En el mismo sentido, puede definir la 'longitud' en frames de una imagen estática.

Duplique el corte de vídeo con **SÍNFITO-D**, sitúe el duplicado en el canal 2 y cambíelo un frame a la derecha. Ahora tenemos dos capas de vídeo una encima de otra, pero solo una se verá. Para mezclar las dos capas necesita aplicarlas un efecto.

Seleccione ambos cortes y presione **SÍNFITO-A**. Seleccione **ADD** del menú emergente ([Figure 25-8](#)). Otra forma sería usar el

Dad>>Effect>>Dad

Figure 25-8. Mezclando dos cortes de vídeo



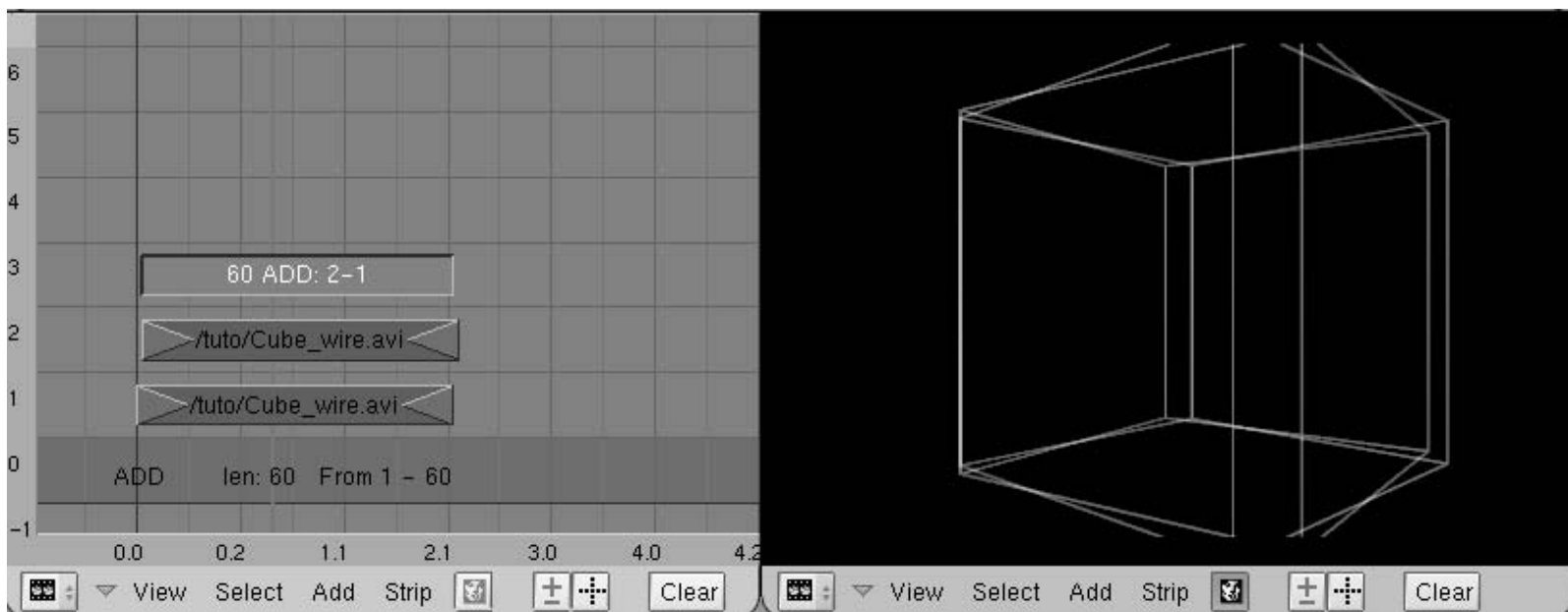
Para ver que ocurre divide la ventana del editor de secuencia y seleccione el botón de imagen en la cabecera ([Figure 25-9](#)). Esto activará la visualización automática ([Figure 25-10](#)). Si selecciona un frame en la ventana del editor de secuencia con los cortes, la previsualización será automáticamente actualizada (con todos los efectos aplicados).

Figure 25-9. Botón previsualizar del editor de secuencia.



Si presiona **ALT-A** en la ventana de visualización, Blender reproducirá la animación. (El render de los efectos la primera vez toma bastante tiempo de proceso, de modo que no espere una previsualización en tiempo real).

Figure 25-10. Añadiendo una ventana de previsualización.

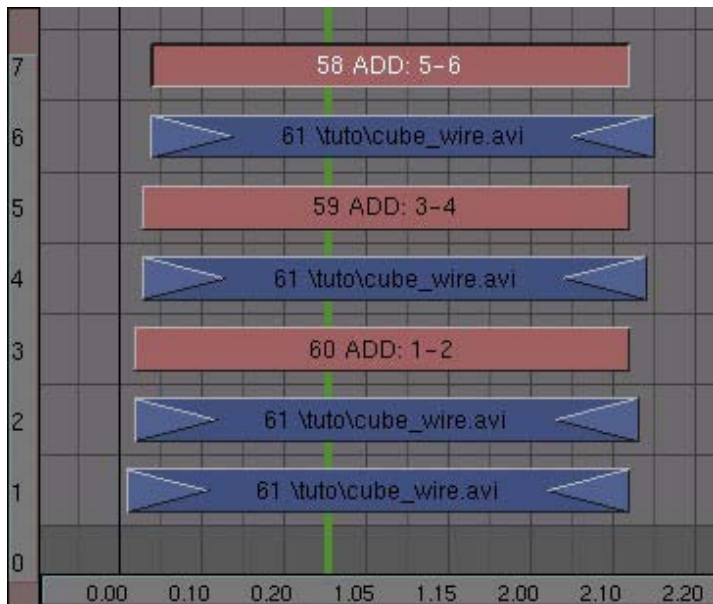


Tip Previsualización sin ventana

Si no le gusta la ventana separada de render, cambie a los botones de render (**F10**) y seleccione 'DispView' en la parte inferior izquierda.

Ahora es el momento de añadir mas caos a la animación. Duplique otra capa de vídeo y situela en el canal 4. Añadala al efecto ADD (añadir) existente en el canal de vídeo 3 con un nuevo efecto ADD (añadir). Repita esto hasta que tenga cuatro cubos alámbricos en la ventana de visualización ([Figure 25-11](#)).

Figure 25-11. Secuencia con 4 cortes de cubos alámbricos añadidos juntos



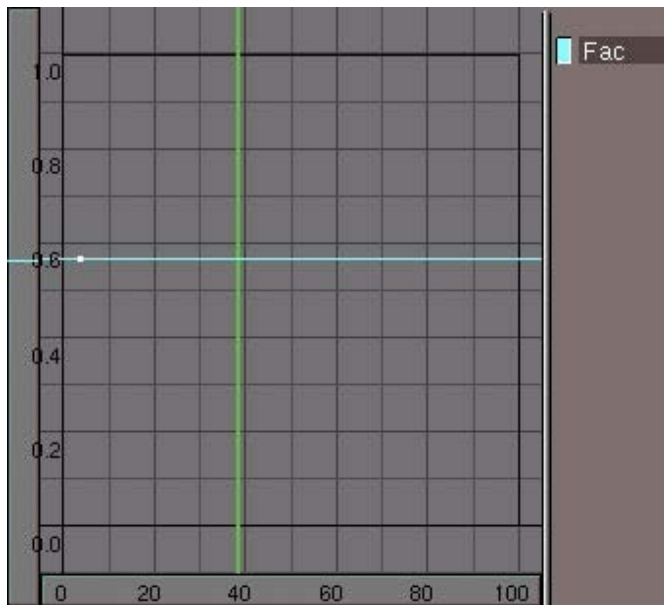
Ahora todos los cubos tienen el mismo resplandor, pero quisiéramos tener una disminución del brillo. Esto es fácil de conseguir: abra una ventana IPO en alguna parte (**F6** y seleccione el icono de secuencia en su Menú IPO de clase ([Figure 25-12](#)).

Figure 25-12. Botón IPO de Secuencia.



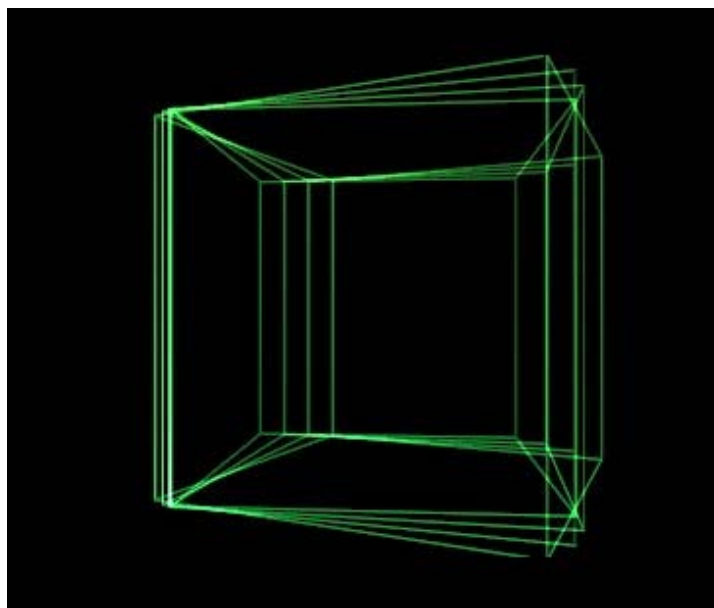
Seleccione el primer corte añadido (el del canal 3), presione **CTRL** y haga click **LMB** en la ventana IPO en un valor de 1. Esto pone el brillo de dicha operación de añadir al máximo. Repita el proceso para los otros dos cortes añadidos, pero disminuya el valor un poco en cada uno, alrededor de 0.6 y 0.3 ([Figure 25-13](#)).

Figure 25-13. Definiendo el brillo de unacapa con un IPO



Dependiendo de los valores ADD (añadir) que haya puesto, su resultado debería ser algo parecido a lo que se ve en [Figure 25-14](#).

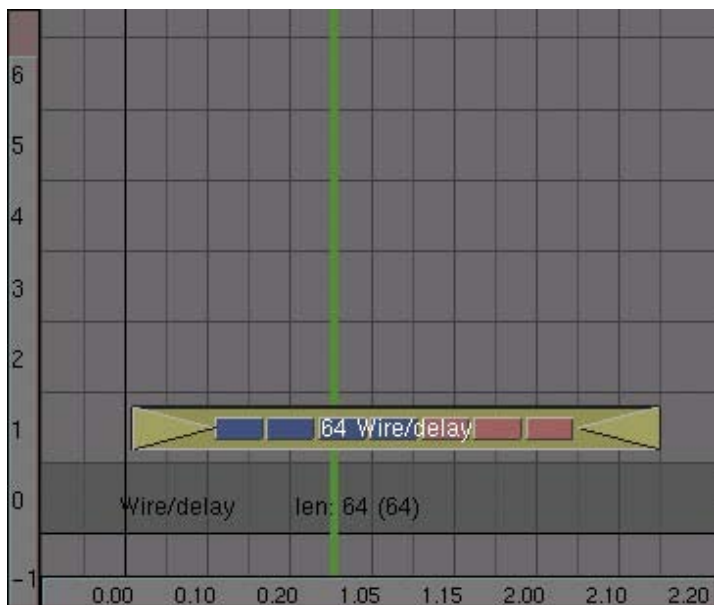
Figure 25-14. Cuatro cubos alámbricos combinados con efectos de fundido.



Ahora tenemos 7 cortes ¡y habíamos empezado solo con nuestra secuencia!. Puede pensar que la pantalla puede volverse saturada. Para hacer nuestro proyecto mas manejable, seleccione todos los cortes (**AKEY** y **BKEY**, presione **MKEY** y **ENTER** o haga click en el popup *Make Meta*. Otra manera que podría usar seria en la entrada de menú *Strip>>Make Meta Strip* Los cortes serán combinados en un meta-corte, y pueden ser copiados como un todo.

Con el meta-corte seleccionado, presione **NKEY** e introduzca un nombre, por ejemplo 'desfase/alambico', para recordar mejor lo que es [Figure 25-15](#).

Figure 25-15. Nombrado el Meta Corte



25.1.3. Segunda animación: un cubo solido desfasado

Es el momento de usar algunas mascaras. Queremos crear dos áreas en las que la animación se reproduzca con un frame de diferencia. Esto crea un efecto visual estilo cristal muy interesante.

Empiece creando una imagen en blanco y negro como [Figure 25-16](#). Puede usar un programa de dibujo o hacerlo en Blender. El camino mas fácil para hacerlo en Blender es crear un material blanco con un valor de emisión de 1 o un material blanco sin sombra en algunos círculos biselados. De esta forma no necesita crear ninguna lampara. Guarde la imagen como mask.tga.

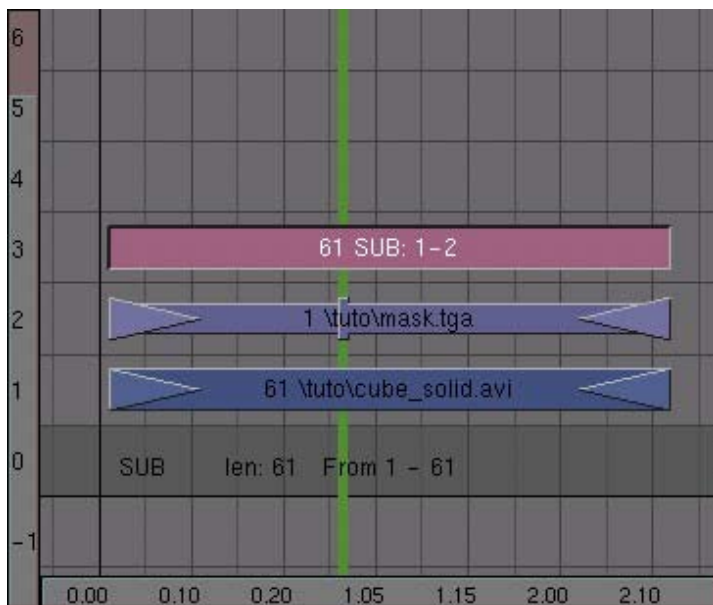
Figure 25-16. Mascara de animación.



Cambie al editor de secuencia y mueva el meta-corte que hemos creado antes, fuera del camino (le reposicionaremos despues). Añada la animación del cubo solido (**SÍNFITO-A** >> *Moví*). Despues, añada la imagen de mascara. Por defecto una imagen estática tiene una longitud de 50 frames en el editor de secuencia. Cambiolo para que coincida con la longitud de la animación del cubo con **RMB** y **GKEY** con las flechas salientes en los laterales del corte de imagen con el botón derecho del ratón.

Ahora seleccione ambos cortes (presione **SÍNFITO**), presione **SÍNFITO-A** y añada un efecto (sustraer) *SUB*. ([Figure 25-17](#)).

Figure 25-17. Substrayendo la mascara del vídeo.



En la ventana de previsualización ahora verá el efecto; las áreas donde la máscara es blanca se habrán eliminado del dibujo ([Figure 25-18](#)).

Figure 25-18. Máscara eliminada.



El efecto ya está preparado; seleccione los tres cortes y conviértalos en un META corte presionando **MKEY**

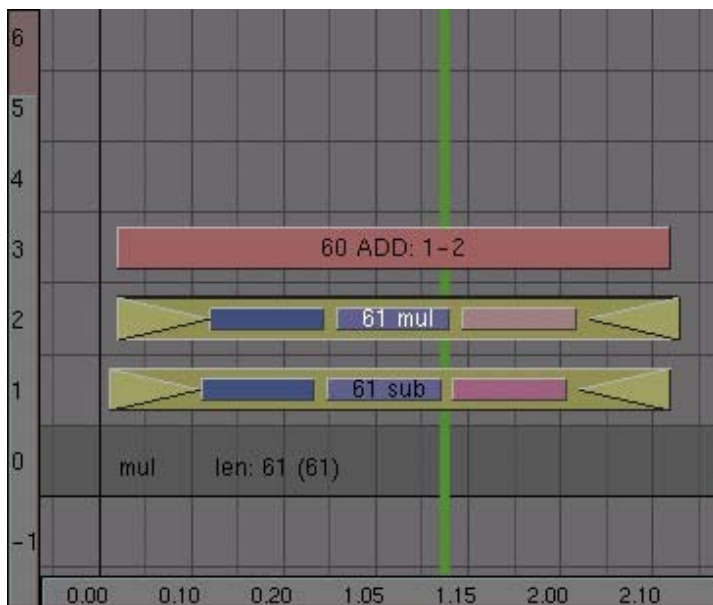
Ahora repita los pasos anteriores, excepto que ahora no usa el efecto *SUB* sino el efecto *MUL* (multiplicar) ([Figure 25-19](#)). Ahora solo verá la imagen original donde la máscara es blanca. Cambie los tres cortes de éste efecto en un meta corte de nuevo.

Figure 25-19. Máscara multiplicada.



Para el último paso tenemos que combinar los dos efectos juntos. Mueva uno de los meta cortes por encima del otro y aplíquele un desfase de un frame. Seleccione ambos cortes y añada un efecto ADD (añadir) ([Figure 25-20](#)).

Figure 25-20. Añadiendo los dos efectos



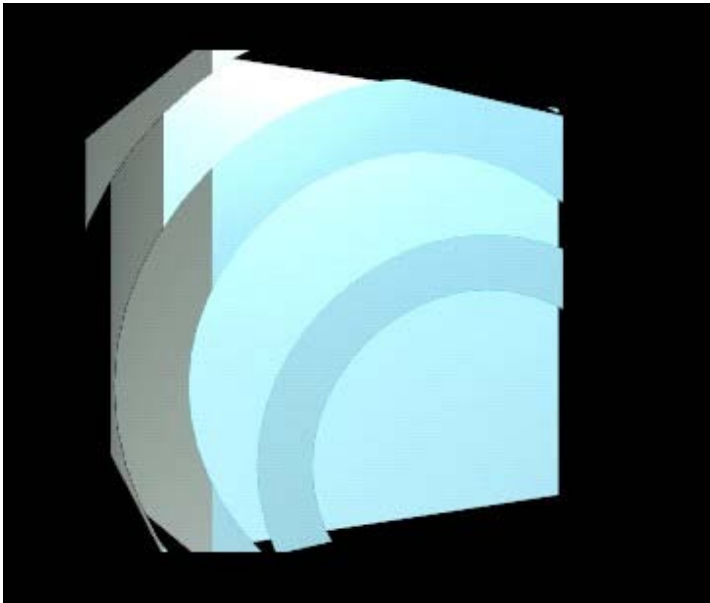
En la ventana de previsualización ahora verá el resultado de la combinación de la animación y la máscara ([Figure 25-21](#)).

Cuando esté listo, seleccione los dos meta cortes y el efecto *ADD* y conviértalos en un nuevo meta corte. (¡Es cierto! Puede tener meta cortes de meta cortes!)

Tip Entrando en un meta corte

Para editar los contenidos de un meta corte, selecciónelo y presione **TAB**. El meta corte 'explotará' para mostrar sus componentes y el fondo se volverá verde amarillento para indicar que está trabajando dentro de un meta corte. Presione **TAB** de nuevo para volver al modo de edición normal.

Figure 25-21. Dos capas cambiadas en el tiempo.

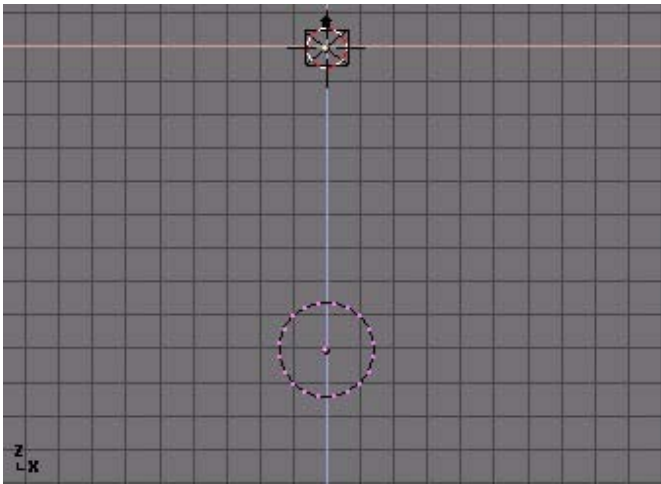


25.1.4. Tercera animación: Un túnel

Ahora queremos un tercer 'efecto' para enriquecer más allá nuestra animación; un 'túnel' 3D para usarlo como efecto de fondo. Es realmente fácil de crear. Primero guarde su trabajo actual - ¡lo necesitará después!

Comience una nueva escena (**CTRL-X**) y elimine el plano por defecto. Cambie a la vista frontal (**NUM 1**). Añada un círculo de 20 vértices aproximadamente 10 unidades por debajo de la línea de $z=0$ (la línea rosa en su pantalla) ([Figure 25-22](#)).

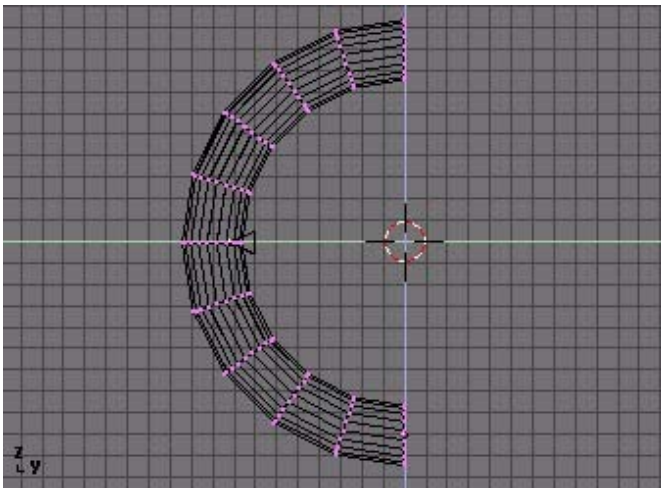
Figure 25-22. Añadiendo un círculo de 20 vértices.



Mientras estamos en el Modo Edición, cambie a la vista lateral (**NUM 3**) y apunte el cursor al origen de forma aproximada al punto $x,y,z=0$ y presionando **SÍNFITO-S**. Seleccione *Curs>>Grid*.

Queremos convertir el círculo en un tubo circular, o toroide. Para ello, usaremos la función Revolución. Vaya al menú edición (**F9**) e introduzca un valor de 180 en el Botón numérico *Degr* e introduzca '10' en el numérico *Steps* en el panel *Mesh Tools* (*herramientas de malla*). Presionando *Spin* ahora rotará los vértices seleccionados alrededor del cursor 180 grados en 10 pasos. ([Figure 25-23](#)).

Figure 25-23. Revolucionando el círculo alrededor del cursor



Salga del modo edición (**TAB**). Con las opciones por defecto, Blender siempre rota y escala alrededor del centro del objeto que se muestra como un pequeño punto. Este punto es amarillo cuando el objeto está sin seleccionar y rosa cuando lo está. Mientras el cursor esté en el origen, presione el botón *Center* en la ventana de Botones de Edición para mover el centro del objeto a la posición actual del cursor. Ahora presione **RKEY** y rote el tubo 180 grados alrededor del cursor.

Ahora es el momento de mover la cámara dentro del túnel. Abra otra ventana 3D y cambíela a la vista de cámara (**NUM0**). Posicione la cámara en la ventana vista lateral para que coincida [Figure 25-24](#), la vista de cámara ahora debería coincidir [Figure 25-25](#).

Tip Aristas perdidas

Si no se muestran todas las aristas del túnel, puede forzar a Blender a dibujarlas seleccionando *All Edges* (todas las aristas) en el panel *Mesh Tools 1* (Herramientas de malla 1) en el menú edición. (**F9**).

Figure 25-24. Cámara dentro del túnel.

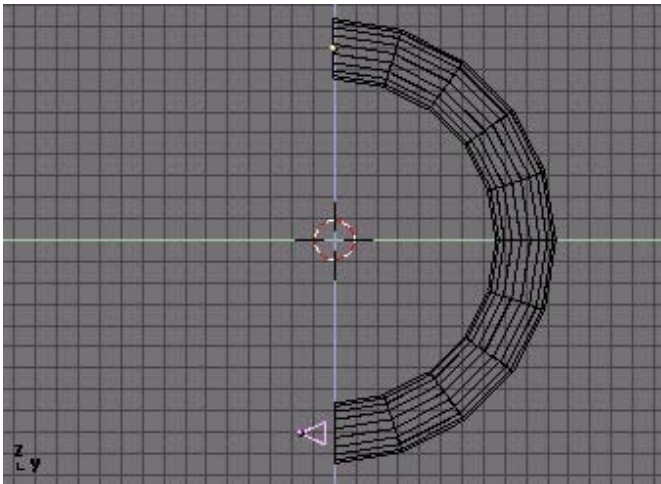
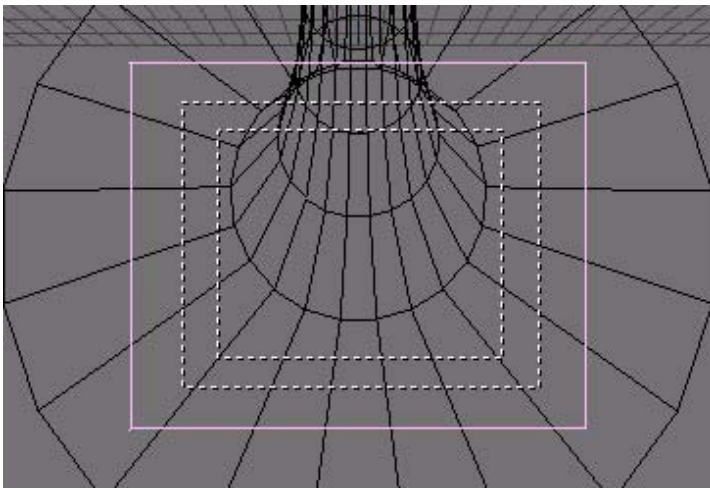


Figure 25-25. Visión de cámara del interior del túnel.



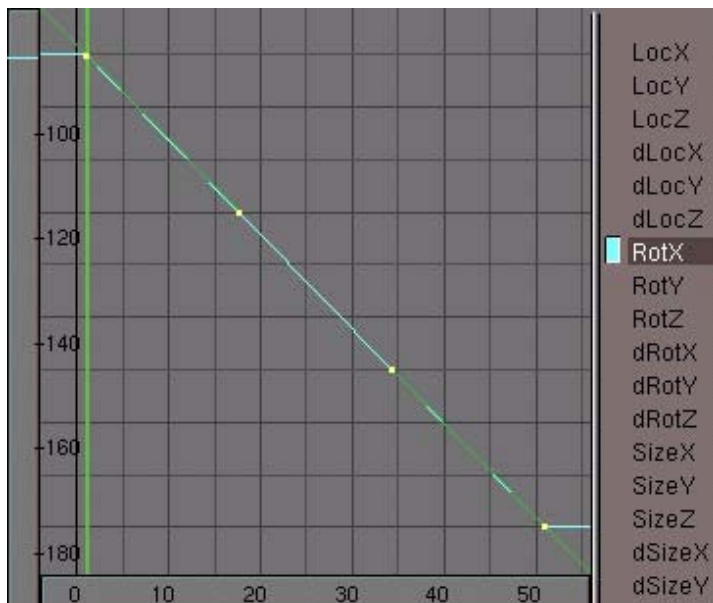
Para quitarnos algún problema, me gustaría renderizar la escena como una animación en bucle. Así, podré añadir tantas copias como quiera a la compilación final de vídeo.

Hay dos cosas que debemos tener en mente cuando creamos animaciones en bucle. Primero, asegúrese de que no hay 'salto' en su animación cuando se repite. Para ello, debe tener cuidado creando los KeyFrames y estableciendo la longitud de la animación. Cree dos KeyFrames: uno con la actual rotación del tubo en el frame 1, y otro con una rotación de 90 grados (presione **CTRL** mientras rota) en el frame 51. En su animación el frame 51 ahora es lo mismo que el frame 1, de modo que cuando renderice necesitará dejar fuera el frame 51 y renderizar desde el 1 hasta el 50.

Dese cuenta de que el número 90 grados no está elegido al azar, debido a que el túnel es periódico, con periodo 18, entonces debe rotarlo por un múltiplo de 18, y 90 lo es, para garantizar que el frame 51 es exactaente el mismo que el 1.

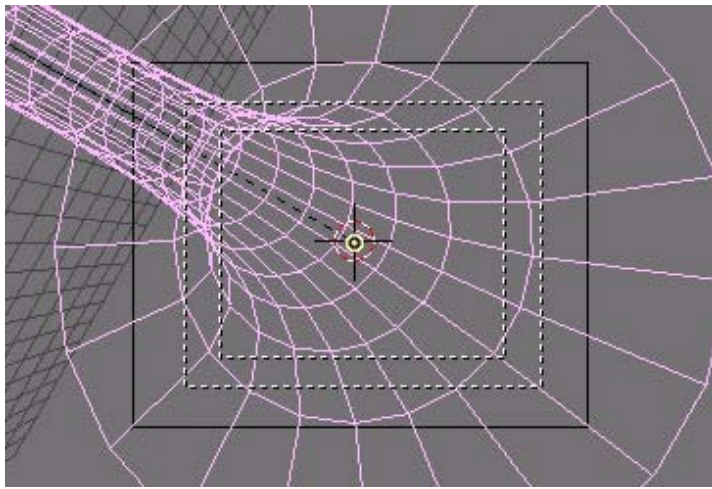
Segundo, para conseguir un movimiento *lineal* necesita eliminar las entradas y salidas de la rotación. Se pueden ver en la Ventana IPO del tubo después de insertar los KeyFrames de rotación. El suavizado IPO empieza y termina como una función coseno. Queremos que sea recto. Para ello seleccionamos la curva de rotación, entramos en el modo edición (**TAB**) y seleccionamos todos los vértices (**AKEY**) y presione **VKEY** ('Vector') para cambiar la curva en una linea. ([Figure 25-26](#)).

Figure 25-26. Rotación IPO del túnel sin entrada y salida.



Para crear un efecto más dramático, seleccione la cámara mientras está en el modo vista de cámara ([Figure 25-27](#)). La misma cámara se muestra como un cuadrado sólido. Presione **RKEY** y rotela un poco. Si ahora reproduce la animación debería repetirse sin vetas.

Figure 25-27. Rotando la cámara para conseguir un efecto más dramático



Para el toque final, añada un material alámbrico azul al tubo y añada una pequeña lámpara en la posición de la cámara. Jugando con los valores *Dist* de la lámpara (distancia de atenuación) puede hacer que el final del tubo desaparezca en la oscuridad sin tener que trabajar con nieblas. ([Figure 25-28](#)).

Cuando esté satisfecho con el resultado, renderice la animación y guárdela como 'tunnel.avi'.

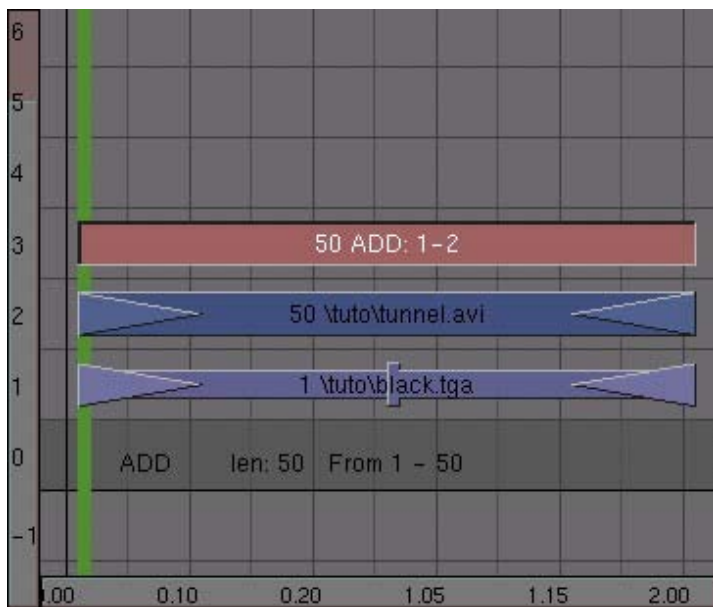
Figure 25-28. Un túnel estupendo.



25.1.5. Segunda secuencia: Usando el túnel como telón de fondo

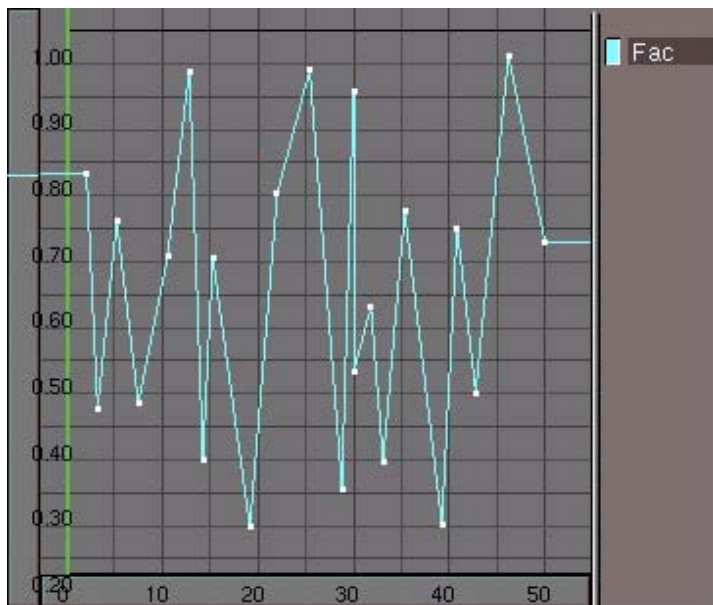
Recargue su archivo de compilación de vídeo de Blender. El túnel que creamos en el paso anterior será usado como telón de fondo para toda la animación. Para hacerlo más interesante he modificado un efecto *ADD* para cambiar el túnel en un telón pulsante. Prepare una imagen completamente negra y llámela 'black.tga' (inténtelo presionando **F12** en un archivo vacío de Blender. Guárdelo con **F3**, pero asegúrese que tiene seleccionado el formato de archivo TGA en la ventana de botones de render). Añada ambos black.tga y la animación del túnel y combínelos con un efecto *ADD*. ([Figure 25-29](#)).

Figure 25-29. Aplicando el efecto de telón.



Ahora, con el efecto **ADD** seleccionado, abra una ventana IPO y seleccione el botón editor de secuencia en la cabecera. Desde los frames 1-50, dibuje una línea irregular manteniendo presionado **CTRL** y con el botón izquierdo. Asegúrese de que los valores están entre 0 y 1 ([Figure 25-30](#)).

Figure 25-30. Añadiendo aleatoriedad con un IPO irregular



Cuando esté listo, eche un vistazo al resultado en una ventana de previsualización y convierta la animación en un meta-corte.

¡Salve su trabajo!

25.1.6. Cuarta animación: un logo saltarín

Vayamos a crear algo de aleatoriedad y caos. Coja un logo (podríamos simplemente añadir un objeto de texto) y haga que salte a través de la pantalla. De nuevo, el modo más fácil de hacer ésto es añadiendo vértices directamente en la ventana IPO (seleccione primero el canal *LocX*, *LocY* o *LocZ*), pero en este momento puede que necesite ser un poco más precavido con los valores mínimos y máximos para cada canal. No se preocupe por el aspecto de cada uno demasiado - el próximo paso hará que sea muy reconocible de cualquier manera. ([Figure 25-31](#)).

Figure 25-31. Logo saltarín

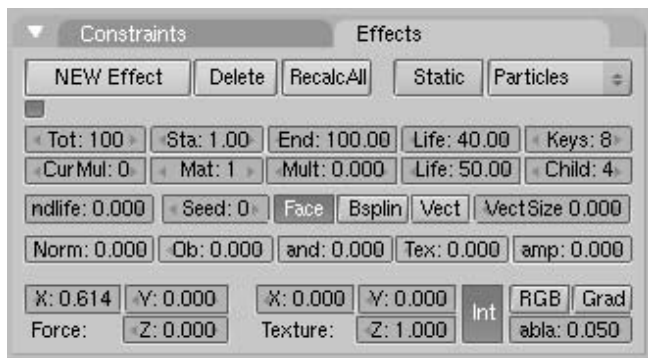


Guarde la animación como 'jumpylogo.avi'.

25.1.7. Quinta animación: Barras de partículas

Nuestro último efecto usará una máscara animada. Combinandolo con el logo del paso anterior, conseguiré un efecto rayado que introduzca el logo en nuestra animación. Esta máscara la haremos usando un sistema de partículas. Para aplicar uno cambie a la vista lateral, añada un plano a su escena y mientras está seleccionado cambie al menú de objeto (**F7**). En la ficha *Effects* del panel *Constraints*. Seleccione *New effect* (*Nuevo efecto*) y cambie el efecto por defecto *build* a *Particles*. Cambie las propiedades del sistema como se indica en [Figure 25-32](#).

Figure 25-32. Propiedades del sistema de partículas.

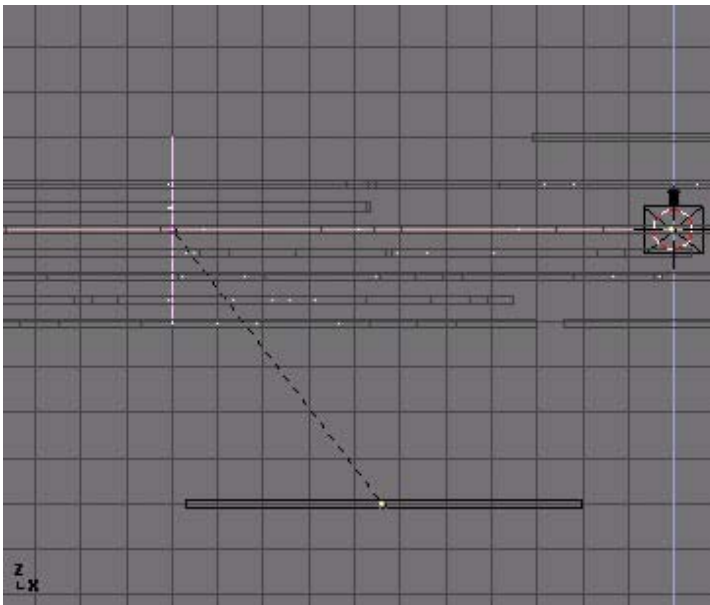


Presione **TAB** para entrar en el modo Edición, seleccione todos los vértices y subdivida el plano dos veces presionando **WKEY** y presionando *Subdivide* del menú emergente.

Lo siguiente será cambiar a la vista frontal y añadir otro plano. Escalo a lo largo del eje X para convertirlo en un rectángulo (presione **SKEY** y mueva su ratón horizontalmente. Entonces haga click **XKEY** o **MMB** para escalarlo sobre el eje indicado solamente). Aplique al rectángulo un material blanco con un valor de emisión de uno.

Ahora necesita cambiar las partículas en rectángulos usando la función de duplicado de vértices (*dupliverts*). Seleccione el rectángulo, después el emisor de partículas y relacionelos. Seleccione solo el plano y en el menú Objeto y panel *Anim Settings* (*Propiedades de animación*) seleccione el botón *DupliVerts*. Ahora cada partícula es remplazada por un rectángulo ([Figure 25-33](#)).

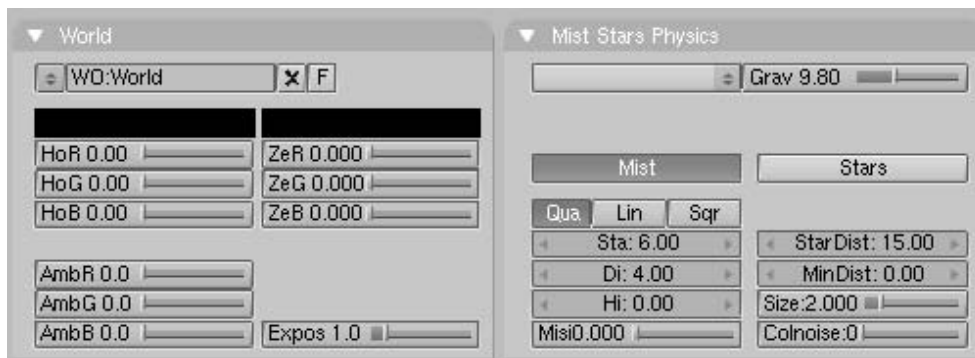
Figure 25-33. Rectángulos con vértices duplicados



Añadiré ahora algo de niebla como un truco rápido para dar a cada rectángulo un sombreado diferente de gris. Ve a la ventana de botones de "World" (Entorno) con **F5** para cambiar al menú de sombreado, y haga click en el botón  y seleccione *Dad New* en el panel *World* (Entorno). Las propiedades de entorno aparecerán.

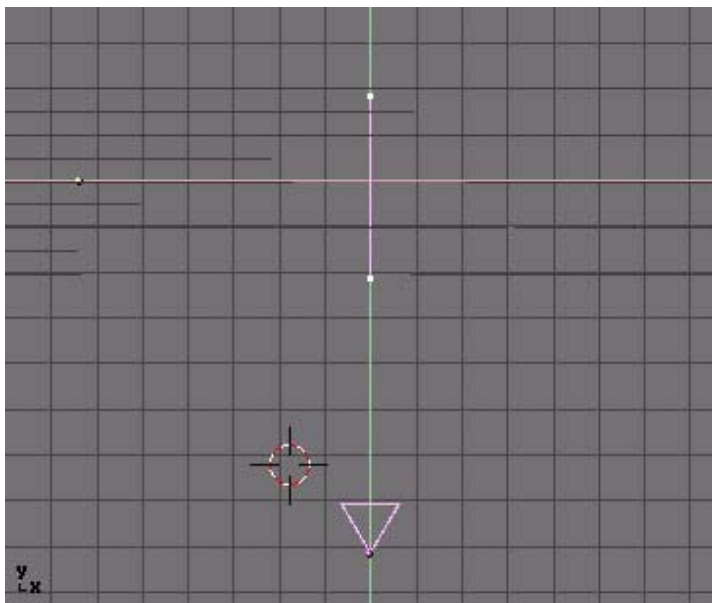
Por defecto, el cielo será renderizado como un gradiente entre azul y negro. Cambie los colores de horizonte (HoR, HoG, HoB) a negro puro ([Figure 25-34](#)).

Figure 25-34. Configurando la niebla.



Para activar el render de la niebla active el botón Mist (Niebla) en el medio de la pantalla. Cuando usamos niebla, tenemos que indicar cual será la distancia de la cámara a la que funcionará. Seleccione la cámara, cambie al menú Edición y active *ShowMist* en el panel *Camera*. Ahora cambie a la vista superior y vuelva al menú de sombreado (**F5**) y los botones de entorno. Retoque los parámetros *Sta:* y *Di:* (Comienzo y distancia, respectivamente) para que la niebla cubra toda la anchura de la cadena de partículas ([Figure 25-34](#) y [Figure 25-35](#)).

Figure 25-35. Configurando los parámetros de niebla



Configure la longitud de la animación en 100 frames y rendericela a disco. Llame al archivo 'particles.avi' ([Figure 25-36](#)).

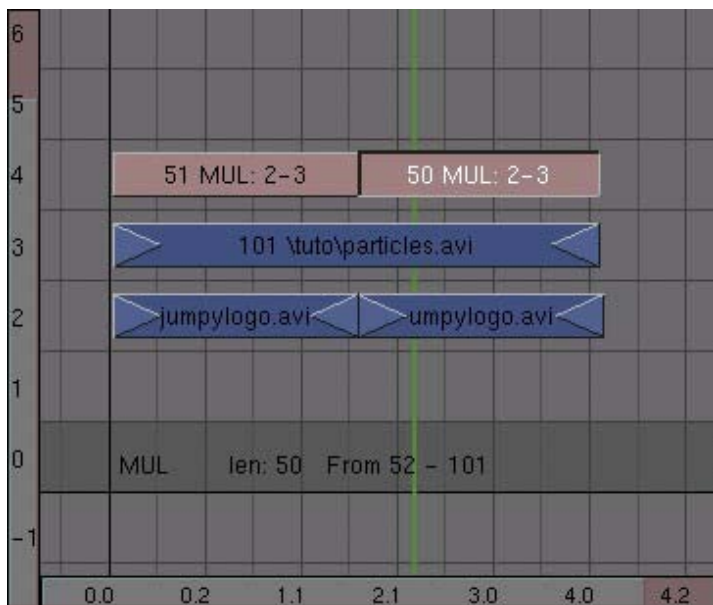
Figure 25-36. Rectángulos de partícula renderizados.



25.1.8. Tercera secuencia: Combinando el logo y las barras de partículas

Ahora sabéis lo que hay que hacer: recargar el archivo de proyecto, cambiar a la ventana del editor de secuencia y añadir tanto 'particles.avi' como 'logo.avi' a vuestro proyecto. Combinelos juntos con un efecto *MUL*. Dado que la animación del logo son 50 frames y la de las partículas son 100 frames, necesitará duplicar la animación del logo una vez y aplicar un segundo efecto *MUL* en él. ([Figure 25-37](#) y [Figure 25-37](#)).

Figure 25-37. Use dos veces el logo de la animación



Combine estos tres cortes en un metacorte. Si se ve con valor, puede hacer algunas copias y darles algo de desfase de tiempo como hicimos con el cubo alámbrico.

Figure 25-38. Animación de partículas combinado con animación de logo

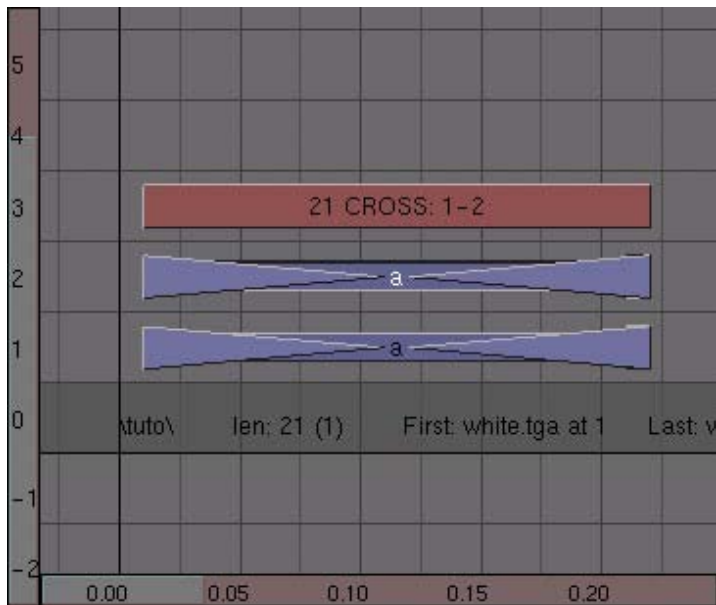


25.1.9. Sexta animación: Haciendo zoom al logo

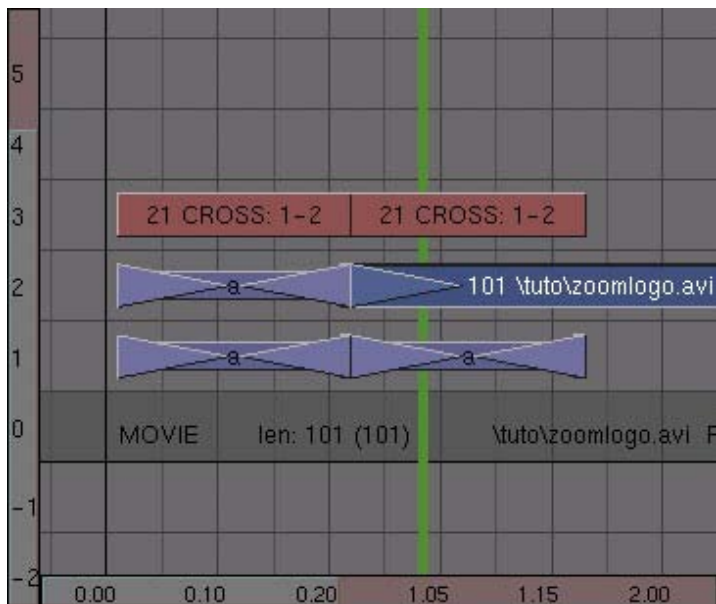
Si quisiera combinar todas sus animaciones, lo más que conseguiría sería una compilación de vídeo realmente revuelta, pero si es la presentación de su compañía desearía presentar el logo de una forma más reconocible. La parte final de nuestro montaje será hacer una animación del logo que haga un zoom muy despacio. Preparelo y guárdelo como 'zoomlogo.avi'. Prepare también una imagen blanca y guardela como 'white.tga'.

Ahora usaremos el efecto *CROSS* para hacer primero una rápida transición de blanco a negro, y después a blanco de nuestra animación de logo. Finalmente, una transición a negro que concluirá el montaje.

Comenzaremos situando black.tga en el canal 1 y white.tga en el canal 2. Haga a los dos de 20 frames de duración. Seleccione ambos y aplique un efecto "cross". El efecto "cross" (cruce) cambiará gradualmente la imagen resultante de la capa 1 a la capa 2. En este caso, el resultado será una transición de negro a blanco ([Figure 25-39](#)).

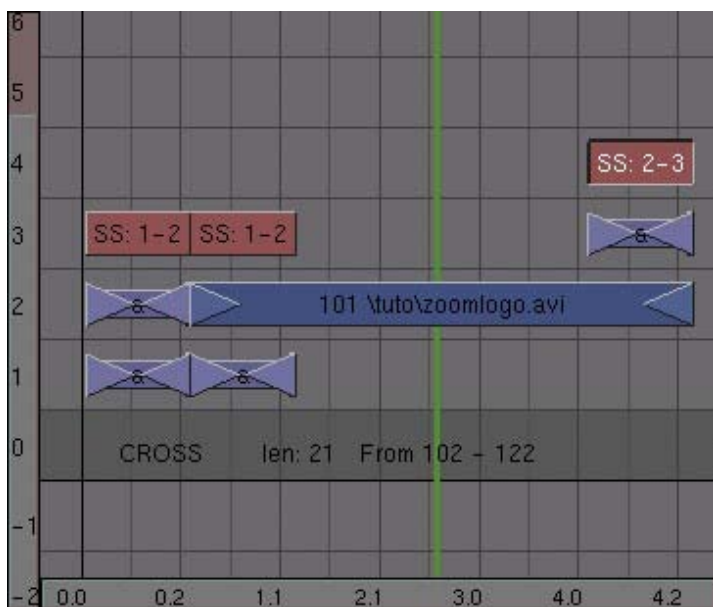
Figure 25-39. Transición negro-blanco.

Ahora, añade un duplicado de white.tga a la capa 1 y sitúela justo a la derecha de black.tga. Hagalo alrededor de la mitad de largo que el original. Situe la animación del zoom de logo en la capa 2 y añade un efecto "cross" entre las dos. En este momento, la animación se ve como un flash blanco seguido de la animación del zoom del logo. ([Figure 25-40](#)).

Figure 25-40. Imagen de transición vídeo blanco

La última cosa que necesita hacer es asegurarse de que la animación tendrá una bonita transición al negro casi al final. Añade un duplicado de black.tga y aplique otro efecto "cross". Cuando esté list, transforme todo en un metacorte strip ([Figure 25-41](#)).

Figure 25-41. Transición vídeo-negro



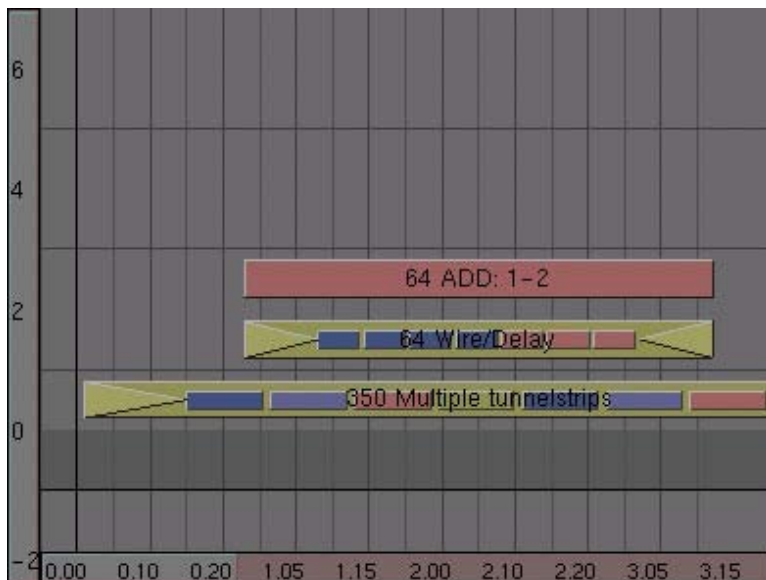
25.1.10. Ensamblándolo todo

¡Ya estamos terminando nuestro trabajo! Es el momento de añadir alguna de nuestras compilaciones que hicimos anteriormente y ver como se ve el trabajo. Lo más importante a recordar mientras crea su compilación final es que cuando renderice su animación, el editor de secuencia solo 've' la capa superior de vídeo. Esto significa que tiene que asegurarse de que sea un corte listo para usarse, o un efecto como *ADD* (*añadir*) que combine algunas capas subyacentes.

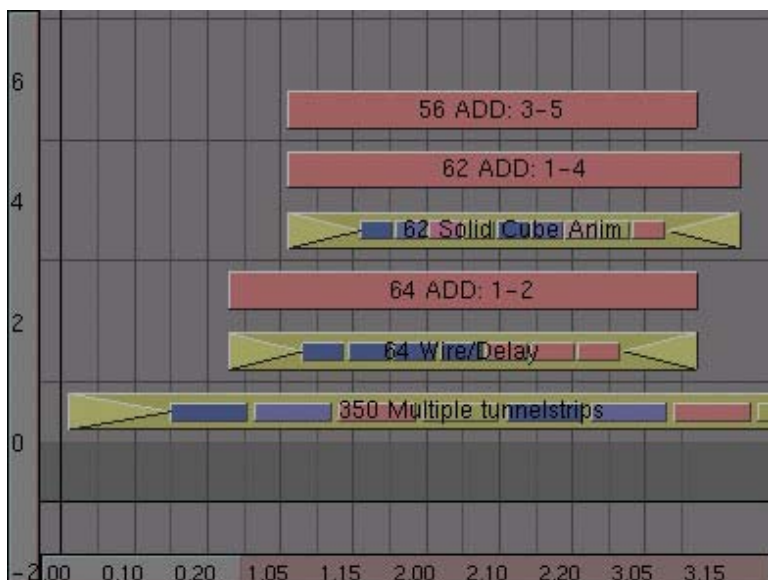
La base de la compilación será el túnel variante. Añada algunos duplicados del meta corte del túnel y sitúelos en el canal uno. Combinelos en un metacorte. No se preocupe por la longitud exacta de la animación aun; siempre podrá duplicar mas cortes de túnel.

Encima de ellas, sitúe el cubo alámbrico desfasado en el canal 2. Añada canal 1 a canal 2 y sitúe el efecto add (añadir) en el canal 3 ([Figure 25-42](#)).

Figure 25-42. Combinando el túnel y el cubo alámbrico

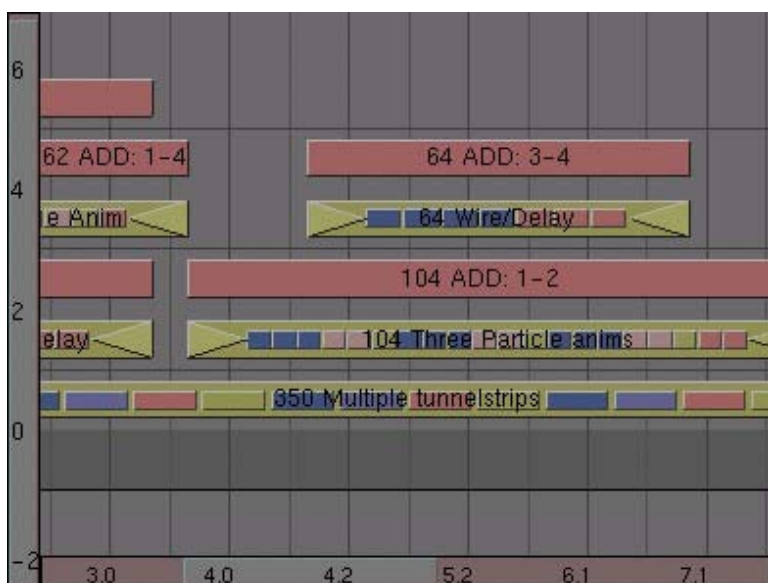


Ahora también queremos añadir el cubo sólido a la animación. Sitúelo en el canal 4, superponiéndolo a la animación alámbrica en el canal 2. Añadalo a la animación del túnel en la capa uno. Aquí es donde las cosas se empiezan a volver algo complicadas; si quisiera dejar algo como esto, la animación en el canal 5 (el cubo sólido junto con el tubo) debería ignorar la animación en el canal 2 (el cubo alámbrico) y el cubo alámbrico se hará visible tan pronto como el cubo sólido se muestre. Para solucionar esto, añada el canal 3 al canal 5 ([Figure 25-43](#)).

Figure 25-43. Combinando el túnel, y los cubos alámbrico y sólido

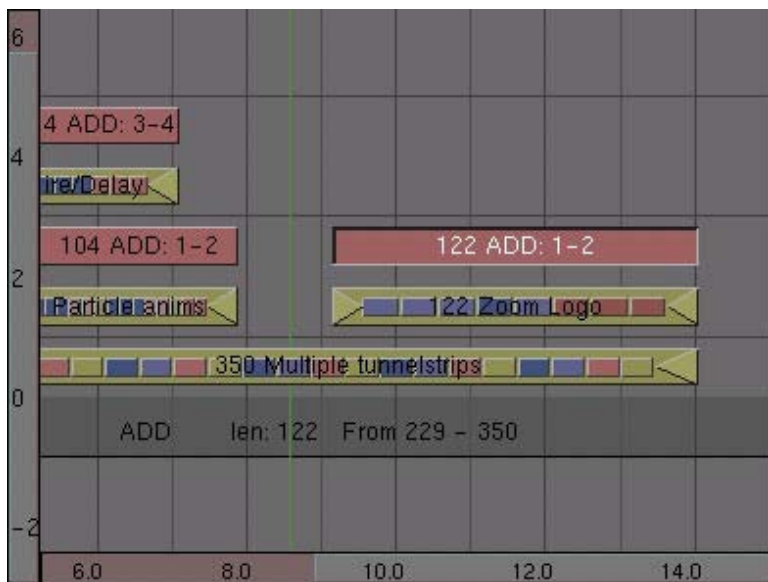
También es posible que necesite aplicar algunas operaciones de "añadir" extra para arreglar algunas partes desaparecidas del vídeo. Esto se verá mucho más después de que haya renderizado la secuencia final.

Deslice la ventana del editor de secuencia un poco a la izquierda y añada el metacorte con la animación del logo con partículas en ella. Coloque este corte en la capa 2 y sitúe el efecto añadir (add) en la capa 3. Para dar algo de variación, duplique la animación alámbrica y combínala con el "add" (añadido) en la capa 3 ([Figure 25-44](#)).

Figure 25-44. Añadiendo la animación del logo con partículas

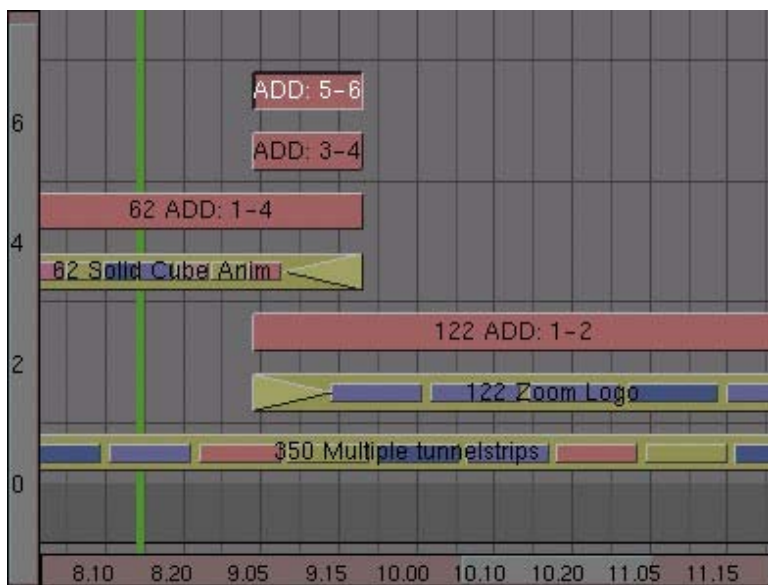
Ahora vaya al final del corte de la animación del túnel. Debería haber suficiente sitio para colocar la animación del logo ampliado al final y después mantener algo de espacio libre ([Figure 25-45](#)). Si no es así, seleccione el corte del túnel, presione **TAB** y añada un duplicado de la animación al final. Presione **TAB** de nuevo para salir del modo edición "meta".

Figure 25-45. Añadiendo la animación del logo con zoom.



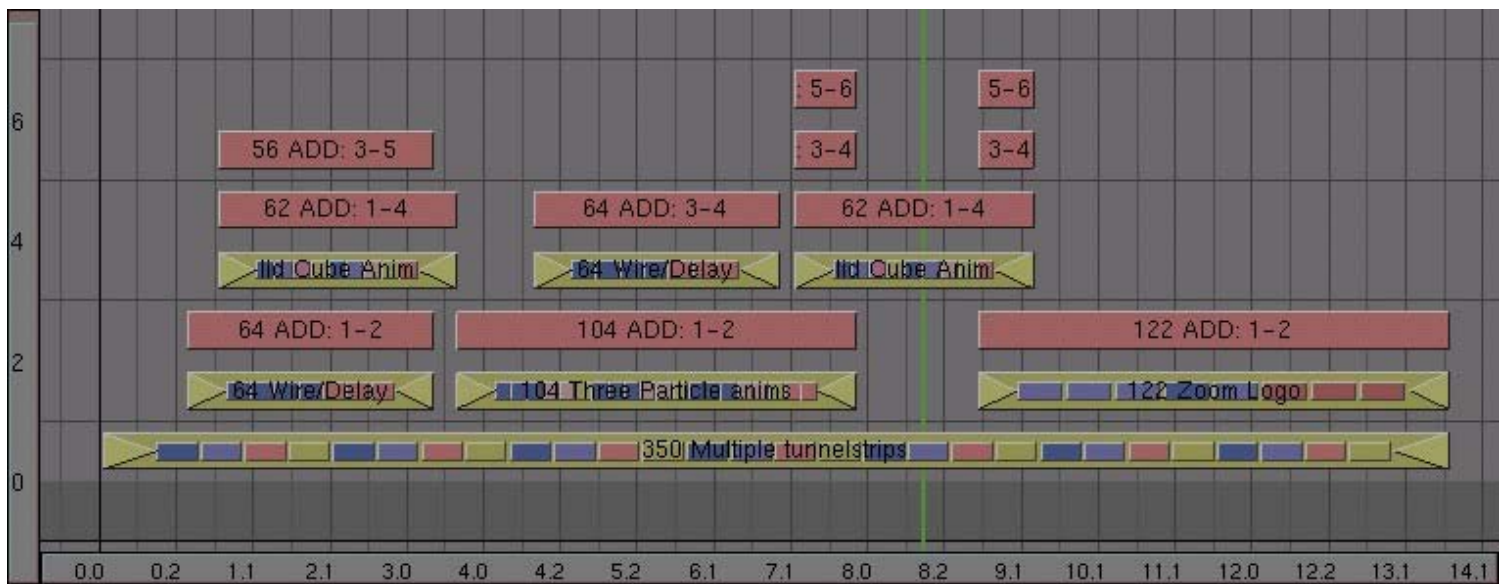
Si sigue habiendo algo de espacio a la derecha, podemos añadir una copia de la animación del cubo sólido. Para conseguir que se muestre correctamente, tendrá que aplicarle dos canales "add" (añadir): uno para combinarla con la animación del logo con partículas y otra para combinarlo con la animación del logo con zoom ([Figure 25-46](#)).

Figure 25-46. El último detalle



[Figure 25-47](#) shows the complete sequence.

Figure 25-47. La secuencia final



25.1.11. Conclusión

Ahora estamos preparados para renderizar la composición final de vídeo. Para decirle a Blender que use la información del editor de Secuencia mientras renderiza, seleccione el botón *Do Sequence* (hacer secuencia) en la ventana de botones de renderizado. Después de ésto, renderice y guarde su trabajo de animación como antes (asegúrese de no sobrescribir ninguna de sus secuencias AVI).

[Prev](#)

El sistema SoftBody de Blender

[Home](#)

[Up](#)

[Next](#)

Editor de Secuencia de sonidos

25.2. Editor de Secuencia de sonidos

Relevante a Blender v2.31

Desde Blender 2.28 hay una herramienta de secuencia de Audio (aun limitada). Puede añadir archivos WAV via el menú **SÍNFITO-A** y seleccionando la entrada *Sound (sonido)* .

Un corte verde de audio será creado. De momento no hay características de mezclado de 'alto nivel'. Puede colocar tantos cortes de audio como desee, y el resultado será la mezcla de todas ellas.

Puede dar a cada corte su propio nombre y ganancia (en dB) via el menú *NKEY* Esto también le permite silenciar o dar balance; -1 totalmente a la izquierda, +1 totalmente a la derecha.

Un IPO de 'Volumen' puede ser añadido al corte en la ventana IPO como hemos hecho para cortes de efectos. El canal *Fac* es el volumen aquí. Los frames 1-100 corresponden con toda la longitud de la muestra, 1.0 es el volumen total, y 0.0 para completo silencio.

Blender todavía no puede mezclar sonido en el resultado final del editor de secuencia. El resultado del editor de secuencia seguirá siendo un archivo de vídeo, si el botón *ANIM* en el panel *Anim* del menú escena está activado, el submenú render es usado como se describe, o un archivo WAV *separado*, conteniendo la secuencia entera de audio, en el mismo directorio que el archivo de vídeo y con el *mismo nombre* pero extensión WAV. El archivo de audio es creado con el botón *MIXDOWN* en el botón *Sequencer (secuenciador)* del menú escena, submenú sonido.

Puede mezclar vídeo y audio después con un programa externo. La ventaja de usar el editor de secuencia de Blender radica en la sencilla sincronización debida a el secuenciamiento de frames y sonido en la misma aplicación.

25.3. Plugins del editor de secuencia

Relevante a Blender v2.31

Como hemos dicho antes, Blender es ampliable mediante un sistema de plugins, y podemos encontrar dos tipos de plugins: De texturas y de secuencia.

Los plugins de secuencia trabajan con cortes de una manera muy similar a las operaciones convencionales *ADD*, *CROSS* etc. Debe tener al menús un corte seleccionado y presionar **SÍNFITO-A**; Plugin o la entrada de menú *Dad Effect Plugin*. Esto abre una ventana de selección de archivos en la cual podrá seleccionar el plugin deseado.

Las funcionalidades de los plugins varían mucho, de modo que no es posible describirlos aquí. Al contrario que los plugins de texturas, los plugins de secuencia no tienen botones en ninguna ventana, así que sus parámetros se suelen configurar mediante **NKEY**.

VI. Extendiendo Blender

Al contrario de muchos programas con los que puedes estar familiarizado, Blender no es monolítico y estático. Puedes extender sus funcionalidades incluso sin tener que modificar las fuentes y recompilarlas.

Hay dos maneras de realizar dicha extensión: Scripts en Python y Plugins, siendo el primero de los dos el preferido y mayormente usado. Esta parte describirá ambas técnicas.

Table of Contents

26. [Python](#)

26.1. [Un ejemplo funcional de Python](#)

26.1.1. [Cabeceras, importación de los módulos y variables globales.](#)

26.1.2. [Dibujando el GUI.](#)

26.1.3. [Manejando eventos.](#)

26.1.4. [Manejando Mallas](#)

26.1.5. [Conclusiones](#)

26.2. [Referencia de Python](#)

26.3. [Scripts de Python](#)

27. [Sistema de plugins de Blender](#)

27.1. [Escribiendo un plugin de Textura](#)

27.2. [Especificación](#)

27.3. [Un plugin de textura genérico](#)

27.4. [Nuestras modificaciones](#)

27.5. [Compilando:](#)

27.6. [Escribiendo un plugin de Secuencia](#)

27.6.1. [Especificación:](#)

27.6.2. [Plugin de Secuencia Genérico:](#)

27.6.3. [Nuestras modificaciones:](#)

27.6.4. [Compilando:](#)

Chapter 26. Python

Relativo a Blender v2.31

Blender tiene una característica muy poderosa que a menudo es pasada por alto. Esta característica es un intérprete de Python totalmente funcional.

Esto le permite a cualquier usuario añadir funcionalidades a Blender escribiendo un simple script de Python. Python es un lenguaje de programación interpretado, interactivo, y orientado a objetos. Incorpora módulos, excepciones, tipeado dinámico, tipo de datos dinámicos de muy alto nivel y clases. Python combina un gran poder con una sintaxis muy sencilla. Está expresamente diseñado para ser usado como una extensión para las aplicaciones que necesiten una interfaz programable, y esto es por lo que Blender lo utiliza.

De las dos maneras que tiene de extender Blender, siendo la otra los plugins binarios, la tratada aquí los script de Python es la más poderosa, versátil, más sencilla de comprender y robusta. Realmente es preferible usar Python que escribir un plugin.

Realmente los script de Python tenían unas funcionalidades limitadas en Blender 2.25, la última de las versiones que liberó NaN. Cuando Blender se hizo código abierto muchos de los desarrolladores que se movían alrededor de la Fundación lo eligieron para trabajar y, junto con el cambio de UI, el API de Python es probablemente la parte de Blender que ha tenido un desarrollo más grande. Una total reorganización de lo que existía fue llevada a cabo y muchos nuevos módulos fueron añadidos.

Dicha evolución está aún en proceso y una mejor integración está por venir en las próximas versiones de Blender.

Blender tiene una ventana texto junto con los otros tipo de ventana la cual es accesible vía el botón



del menú Tipos de Ventana o a través de **SHIFT-F11**.

Una ventana de texto recién abierta es gris y está vacía, y tiene una barra de herramientas muy sencilla ([Figure 26-1](#)). De izquierda a derecha están el botón estándar de selección de ventana y el menú ventana. Luego está el botón de pantalla completa, seguido por un botón que muestra/oculta el número de líneas del texto y el botón Menú.

Figure 26-1. Barra de herramientas de texto.

El botón Menú () permite seleccionar que buffer de texto tiene que ser mostrado, y también le permite crear un nuevo buffer o cargar un fichero de texto.

Si elige cargar un fichero en la ventana texto temporalmente se crea una ventana de selección de ficheros, con las funciones usuales. Una vez que el buffer de texto está en la ventana texto, esta se convierte en un editor de texto muy simple. Escribir en el teclado provoca que aparezca texto en el buffer de texto. Como siempre con presionar **LMB** arrastrar y liberar **LMB** se selecciona el texto. Los siguientes comandos del teclado son aplicables:

- **ALT-C** or **CTRL-C** - Copia el texto marcado al portapapeles;
- **ALT-X** o **CTRL-X** - Corta el texto marcado al portapapeles;
- **ALT-V** o **CTRL-V** - Pega el texto marcado desde portapapeles en la posición del cursor en la Ventana Texto;
- **ALT-S** - Guarda el texto como un fichero de texto, una Ventana de Selección de Ficheros aparecerá;
- **ALT-O** - Carga un texto, una Ventana de Selección de Ficheros aparecerá;
- **ALT-F** - Emergerá la herramienta buscar;
- **SHIFT-ALT-F** or **RMB** - Emergerá el Menú Fichero para la Ventana de Texto;
- **ALT-J** - Emergerá un panel donde puede especificar un número de línea para situar el cursor;
- **ALT-P** - Ejecuta el texto como un script de Python;
- **ALT-U** - Deshacer;
- **ALT-R** - Rehacer;
- **CTRL-R** - Reabrir (recargar) el buffer actual;
- **ALT-M** - Convierte el contenido de la ventana de texto en texto 3D (max. 100 caracteres);

El portapapeles de Blender que corta/copia/pega está *separado* del portapapeles de Windows así que *no podrá* copiar desde o hacia Blender. Para acceder a su portapapeles de Windows utilice **SHIFT-CTRL-C** **SHIFT-CTRL-V**.

Para eliminar un buffer de texto tan sólo presione el botón 'X'. próximo al nombre del buffer, tal y como hace para los materiales, etcétera.

La más importante combinación de teclas es **ALT-P** la cual hace que el contenido del buffer sea pasado al intérprete de Python de Blender.

La siguiente sección le mostrará un ejemplo de script de Python. Antes de que continúe tenga en cuenta que Blender viene con un intérprete de Python muy sencillo con unos pocos módulos totalmente

específicos de Blender, que están descritos en la referencia. .

Tip Otros usos para la ventana de texto

La ventana de texto se utiliza también cuando quieres compartir tus ficheros .blend con la comunidad o con tus amigos. Puedes usarla para escribir un texto aclaratorio de los contenidos de tu fichero. Esto es mucho más interesante que tener dicho fichero en un archivo separado. Asegurate de mantenerlo visible mientras salves tu fichero!.

Si compartes el archivo con la comunidad y quieres hacerlo bajo alguna licencia puedes escribir dicha licencia en una ventana texto.

Para tener acceso a los módulos estándar de Python necesitarás tener instalado Python. Puedes descargarlo de <http://www.python.org>. Asegurate de comprobar en <http://www.blender.org> cual versión *exacta* de Python es con la que fue construida Blender para evitar posibles problemas de compatibilidad.

Blender también debe conocer *donde* se ha realizado dicha instalación. Esto se hace definiendo una variable entorno *PYTHONPATH*.

Configurando PYTHONPATH en Win95,98,Me. Una vez que haya instalado Python en, digamos, C:\PYTHON22 debe editar el fichero C:\AUTOEXEC.BAT con su editor de textos favorito y añadir una línea:

```
SET PYTHONPATH=C:\PYTHON22;C:\PYTHON22\DLLS;C:\PYTHON22\LIB;C:\PYTHON22\LIB\LIB-TK
```

y reiniciar el sistema.

Configurando PYTHONPATH en WinNT,2000,XP. Una vez que haya instalado Python en, digamos, C:\PYTHON22 Vaya al icono "My Computer" en el escritorio, **RMB** y seleccione *Properties* (*Propiedades*). Seleccione la pestaña *Advanced* (*Avanzado*) y presione el botón *Environment Variables* (*Variables de entorno*).

Debajo de la caja de variables del sistema, (la segunda caja), pinche en *New* (*Nuevo*). Si no eres administrador posiblemente no seas capaz de hacer esto. En este caso pinche en *New* en la caja superior.

Ahora, en la caja "Variable Name" (Nombre de variable), escriba PYTHONPATH y en la caja "Variable Value" (Valor de la variable), escriba:

```
C:\PYTHON22;C:\PYTHON22\DLLS;C:\PYTHON22\LIB;C:\PYTHON22\LIB\LIB-TK
```

Pulse OK repetidamente para salir de todos los diálogos. Dependiendo del sistema operativo quizás tengas que reiniciar.

Configurando *PYTHONPATH* en Linux y otros UNIXes. Normalmente tendrás Python ya instalado. Si no fuese así, hazlo. Ahora solo tendrás que descubrir donde está instalado. Esto es sencillo, solo arranque una shell interactiva de Python simplemente abriendo una shell y escribiendo *python* en ella. Escriba los siguientes comandos:

```
>>> import sys
>>> print sys.path
```

y anote lo que se muestre por pantalla que debería ser algo como

```
['', '/usr/local/lib/python2.2', '/usr/local/lib/python2.2 /plat-
linux2', '/usr/local/lib/python2.0/lib-tk', '/usr/lo
cal/lib/python2.0/lib-dynload', '/usr/local/lib/python2.0/
site-packages']
```

Añada esto a su fichero favorito *rc* (fichero de configuración) como una variable de entorno. Por ejemplo, añada en su *.bashrc* la siguiente linea

```
export PYTHONPATH=/usr/local/lib/python2.2:/usr/local/lib/
python2.2/plat-linux2:/usr/local/lib/python2.2/lib-tk:/usr
/local/lib/python2.2/lib-dynload:/usr/local/lib/python2.0/
site-packages
```

todo en una única linea. Abra una nueva shell, o salga e introduzca de nuevo el login.

26.1. Un ejemplo funcional de Python

Relevante a Blender v2.31

Ahora que ha visto que Blender es extensible vía script de Python y que tiene los elementos básicos para el manejo de scripts y sabe como lanzarlos, antes de estrujar su cerebro con la API de Python echemos un vistazo a un pequeño ejemplo.

Realizaremos un pequeño script para producir polígonos. Este script duplica lo que hace **SPACEAdd>>Mesh>>Circle**, pero creará polígonos rellenos, no sólo con el perímetro exterior.

Para hacer el script aún más completo este exhibirá una interfaz gráfica de usuario (GUI) escrita completamente a través de la API de Blender.

26.1.1. Cabeceras, importación de los módulos y variables globales.

Las primeras 32 líneas de código son mostradas en [Example 26-1](#).

Example 26-1. Cabecera del script

```

001 #####
002 #
003 # Script demostración para la Guia de Blender 2.3
004 #
005 #####S68
006 # Este script genera poligonos. Es bastante inútil
007 # ya que puedes hacer lo mismo con ADD->Mesh->Circle
008 # pero es un buen ejemplo, y además los
009 # polígonos están 'rellenos'.
010 #####
011
012 #####
013 # Importación de los módulos
014 #####
015
016 import Blender
017 from Blender import NMesh
018 from Blender.BGL import *
019 from Blender.Draw import *
020
021 import math
022 from math import *
023
024 # Polygon Parameters
025 T_NumberOfSides = Create(3)
026 T_Radius         = Create(1.0)
027
028 # Events
029 EVENT_NOEVENT = 1
030 EVENT_DRAW    = 2
031 EVENT_EXIT    = 3

```

Después de necesarios comentarios que indican lo que hace el script se encuentra la importación de los módulos de Python (lineas 016-022).

Blender es el módulo principal de la API de Python de Blender. *NMesh* es el módulo que proporciona acceso a las mallas de Blender, mientras que *BGL* y *Draw* dan acceso a las funciones y constantes de OpenGL y a la interfaz de ventanas de Blender, respectivamente. El módulo *math* es el módulo matemático de Python, pero dado que ambos módulos 'math' y 'os' están incluidos dentro de Blender no necesita una completa instalación de Python para usarlos!.

Los polígonos se definen a través del número de lados y de su radio. Estos parámetros tienen valores que deben ser definidos por el usuario a través del GUI, lineas (025-026), con lo que se deben crear dos 'botones genéricos', con sus valores por defecto.

Finalmente, los objetos GUI trabajarán con eventos y también los generarán. Los identificadores de los eventos son enteros que serán definidos por el programador. Es una buena práctica utilizar nombres fáciles de recordar para los eventos, como se hace en la lineas (029-031).

26.1.2. Dibujando el GUI.

El código fuente que se encarga del dibujado del GUI está en la función *draw (dibujo)* ([Example 26-2](#)).

Example 26-2. Dibujando el GUI

```
033 #####
034 # Dibujando el GUI
035 #####
036 def draw():
037     global T_NumberOfSides
038     global T_Radius
039     global EVENT_NOEVENT, EVENT_DRAW, EVENT_EXIT
040
041     ##### Titles
042     glClear(GL_COLOR_BUFFER_BIT)
043     glRasterPos2d(8, 103)
044     Text("Demo Polygon Script")
045
046     ##### Parameters GUI Buttons
047     glRasterPos2d(8, 83)
```

```

048      Text("Parameters:")
049      T_NumberOfSides = Number("No. of sides: ", EVENT_NOEVENT, 10,
55, 210, 18,
050                                T_NumberOfSides.val, 3, 20, "Number of sides
of out polygon");
051      T_Radius          = Slider("Radius: ", EVENT_NOEVENT, 10, 35,
210, 18,
052                                T_Radius.val, 0.001, 20.0, 1, "Radius of the
polygon");
053
054      ##### Draw and Exit Buttons
055      Button("Draw",EVENT_DRAW , 10, 10, 80, 18)
056      Button("Exit",EVENT_EXIT , 140, 10, 80, 18)
057

```

Las líneas (037-039) proporcionan acceso a los datos globales. El verdadero meollo del asunto comienza a partir de las líneas (042-044). La ventana de OpenGL es inicializada, y su actual posición configurada a $x=8$, $y=103$. El origen de esta referencia es la esquina inferior izquierda de la ventana de scripts. Entonces el título *Demo Polygon Script* es mostrado.

Una cadena de caracteres es escrita en las líneas (047-048), entonces se crean los botones para introducir los parámetros. El primero (líneas 049-050) es un botón numérico, exactamente como esos que hay en la ventana de botones de Blender. Para comprender todos los parámetros por favor diríjase a la referencia de la API. Básicamente está el botón etiqueta, el evento generado por el botón, su localización (x,y) y sus dimensiones (anchura, altura), su valor, que es un dato perteneciente al objeto mismo, los valores máximos y mínimos permitibles y un texto que aparecerá como ayuda mientras sitúa el cursor sobre el botón, como una etiqueta de ayuda.

Las líneas (051-052) definen un botón numérico con un deslizador, con una sintaxis muy similar. Las líneas (055-056) finalmente crean el botón *Draw* (*Dibujar*) el cual creará los polígonos y un botón *Exit* (*Salir*).

26.1.3. Manejando eventos.

El GUI no se ha dibujado, y no trabajará, hasta que se escriba un manejador de eventos apropiado y se registre ([Example 26-3](#)).

Example 26-3. Manejando eventos

```

058 def event(evt, val):

```

```

059     if (evt == QKEY and not val):
060         Exit()
061
062 def bevent(evt):
063     global T_NumberOfSides
064     global T_Radius
065     global EVENT_NOEVENT, EVENT_DRAW, EVENT_EXIT
066
067     ##### Manages GUI events
068     if (evt == EVENT_EXIT):
069         Exit()
070     elif (evt== EVENT_DRAW):
071         Polygon(T_NumberOfSides.val, T_Radius.val)
072         Blender.Redraw()
073
074 Register(draw, event, bevent)
075

```

Las líneas (058-060) definen el manejador de eventos del teclado, en este caso respondiendo a la **Tecla Q** con una sencilla llamada a *Exit()*.

Hay más interesantes líneas en (062-072) donde se manejan los eventos del GUI. Cada vez que un botón del GUI es usado esta función es llamada, con el número de evento definido en el los parámetros del botón. El núcleo de esta función es una estructura de "selección" que ejecuta diferentes códigos dependiendo del número de evento.

Como última llamada, se invoca a la función *Register*. Esto efectivamente dibuja el GUI y comienza el bucle que captura los eventos.

26.1.4. Manejando Mallas

Finalmente, [Example 26-4](#) muestra la función principal, la que crea el polígono. Es una simple edición de la malla, pero muestra muchos puntos importantes de la estructura interna de datos de Blender.

Example 26-4. Cabecera del script

```

076 #####
077 # Cuerpo principal
078 #####
079 def Polygon(NumberOfSides, Radius):

```

```

080
081 ##### Creates a new mesh
082 poly = NMesh.GetRaw()
083
084 #####Populates it of vertices
085 for i in range(0,NumberOfSides):
086     phi = 3.141592653589 * 2 * i / NumberOfSides
087     x = Radius * cos(phi)
088     y = Radius * sin(phi)
089     z = 0
090
091     v = NMesh.Vert(x,y,z)
092     poly.verts.append(v)
093
094 #####Adds a new vertex to the center
095 v = NMesh.Vert(0.,0.,0.)
096 poly.verts.append(v)
097
098 #####Connects the vertices to form faces
099 for i in range(0,NumberOfSides):
100     f = NMesh.Face()
101     f.v.append(poly.verts[i])
102     f.v.append(poly.verts[(i+1)%NumberOfSides])
103     f.v.append(poly.verts[NumberOfSides])
104     poly.faces.append(f)
105
106 #####Creates a new Object with the new Mesh
107 polyObj = NMesh.PutRaw(poly)
108
109 Blender.Redraw()

```

La primera línea importante aquí es la número (082). Aquí hay un nuevo objeto malla, donde se crea el *polígono*. El objeto malla está constituido de una lista de vértices y de una lista de caras, además de otras interesantes cosas. Para nuestros propósitos los vértices y la lista de caras será lo que necesitaremos.

Por supuesto la malla recientemente creada ésta vacía. El primer bucle (lines 085-092) calculará la localización x,y,z de los vértices (*NumberOfSides*) necesarios para definir el polígono. Para un plano su coordenada 'z' será cero para todos los vértices.

La línea (091) llama al método *Vert* de *NMesh* para crear un nuevo objeto vértice de coordenadas (x,y,z). Tal objeto es entonces añadido (línea 096) en la lista de *vértices* de la malla *polígono*.

Finalmente en las líneas (095-096) un último vértice es añadido al centro.

Las líneas (099-104) conectan los vértices para crear las caras. No será necesario crear todos los vértices primero y luego las caras. Podrás crear una nueva cara tan pronto como todos los vértices estén disponibles.

La línea (100) crea una nueva cara. Un objeto cara tiene su propia lista de vértices v (hasta 4) definidos. Las líneas (101-103) añaden 3 vértices a la lista $f.v$ originalmente vacía. Los vértices son dos vértices adyacentes del polígono y el vértice central. Estos vértices deben de ser tomados de la lista de *vértices* de la malla. Finalmente la línea (104) añade la cara recién creada a la lista de *caras* de nuestra malla *polígono*.

26.1.5. Conclusiones

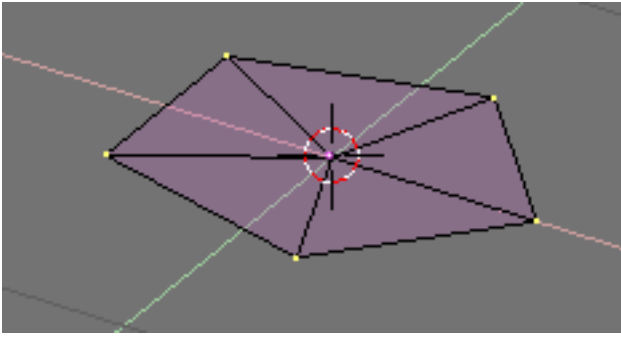
Si crea su fichero *polygon.py* conteniendo el código anteriormente descrito y lo carga dentro de una ventana de texto de Blender, como ha aprendido en la sección anterior y presiona **ALT-P** en dicha ventana para lanzarlo verá como la ventana desaparece y se vuelve gris. En la esquina inferior izquierda el GUI se dibujará ([Figure 26-2](#)).

Figure 26-2. El GUI de nuestro ejemplo.



Seleccionando, por ejemplo, 5 vértices y un radio de 0.5, y presionando el botón *Draw* aparecerá un pentágono en el plano xy de la ventana 3D ([Figure 26-3](#)).

Figure 26-3. El resultado de nuestro script de ejemplo.



[Prev](#)

Extendiendo Blender

[Home](#)

[Up](#)

[Next](#)

Referencia de Python

26.2. Referencia de Python

Relevante a Blender v2.31

La documentación de referencia del lenguaje de programación Python para Blender es un libro en si mismo. Por razones de espacio no se incluye en este documento.

[Aquí está :\)](#)

26.3. Scripts de Python

Relativo a Blender v2.31

Hay más de 100 scripts para Blender disponibles en Internet. Como los plugins, los scripts son muy dinámicos, cambiando su interfaz, sus funcionalidades y su localización en Internet muy rápidamente, así que para una lista actualizada de ellos por favor dirijase a uno de estos dos sitios, www.blender.org o www.elysiun.com.

Chapter 27. Sistema de plugins de Blender

por Kent Mein

Esta sección muestra una referencia detallada de como escribir plugins de Secuencia y de Textura para Blender.

27.1. Escribiendo un plugin de Textura

Relevante a Blender v2.31

En esta sección escribiremos un plugin de textura básico y después enseñaremos a usarlo. Los elementos básicos detrás de un plugin de textura consisten en que tu estás dando algunos valores de entrada, tales como posición, etc y alguna información más. Entonces dependiendo del tipo de plugin de textura se te devolverá información sobre la intensidad, el color y/o las normales.

Tanto los archivos necesarios para desarrollar los plugins como unos pocos plugins se pueden encontrar en el directorio blender/plugins. Alternativamente puede obtener más plugins de <http://www.cs.umn.edu/~mein/blender/plugins>

Los plugins son utilizados (cargados/llamados) en Blender a través de dlopen(). Para aquellos que no están familiarizados con el sistema dlopen diremos que este permite a un programa (Blender) usar un objeto compilado como si este fuese parte del programa mismo, similar a cuando se enlazan dinamicamente librerías, exceptuando que el objeto a cargar se determina cuando se está ejecutando el programa.

La ventaja de usar el sistema dlopen para los plugins consiste en que es muy rápido acceder a una función, y además no hay sobrecarga en la interfaz del plugin, lo cual es crítico cuando (como en el caso de los plugins de textura) el plugin se puede llamar varios millones de veces en un único render.

La desventaja de este sistema es que el plugin funciona como si realmente fuese parte de Blender, con lo que si el plugin se muere, Blender muere.

Los ficheros de cabecera que se encuentran en el subdirectorio plugin/include/ de la instalación de Blender están documentados mostrando las funcionalidades que se proporcionan a los plugins. Dichas

funcionalidades incluyen las funciones de la librería Imbuf para cargar y trabajar con imágenes y buffers de imagen, y funciones de ruido y turbulencia para un texturizado consistente.

[Prev](#)

Scripts de Python

[Home](#)

[Up](#)

[Next](#)

Especificación

27.2. Especificación

Relevante a Blender v2.31

- **#include <plugin.h>**

Todos los plugins de Blender deben incluir este fichero cabecera, que contiene todas las estructuras y definiciones necesarias para trabajar correctamente con Blender.

- **char name[]="Tiles";**

Un string (cadena de caracteres) conteniendo el nombre del plugin, este valor se usará para indicar el título de la textura en la ventana de botones de textura

- **#define NR_TYPES 2 char stnames[NR_TYPES][16]= {"Square", "Deformed"};**

A los Plugins se les permite tener subtipos separados para variaciones minimas en algoritmos - por ejemplo la textura "clouds (nubes)" que está por defecto en Blender tiene los subtipos "Default" y "Color".

NR_STYPES debería ser definido con el numero de subtipos que el plugin requiera, y además a cada subtipo se le tiene que dar un nombre. Cada plugin debe tener al menos 1 subtipo y el nombre de ese subtipo.

- **VarStruct varstr[]={...};**

"varstr" contiene toda la información que Blender necesita para mostrar los botones del plugin. Los botones para los plugin pueden ser de tipo numerico para introducir datos, o de tipo texto para introducir comentarios u otra información. Se pueden utilizar como máximo 32 variables en los plugins.

Cada estructura VarStruct consta de un tipo, un nombre, un rango de información y una nota emergente sobre la herramienta.

El tipo define el tipo de dato para cada botón, y la manera en la que se muestra dicho botón. Para botones numericos este valor debería ser una combinación (ORed) de INT o FLO para el formato del numero y NUM, NUMSLI o TOG para el tipo de botón. Los botones de texto son de tipo LABEL.

El **nombre** es lo que se mostrará en el botón y está limitado a 15 caracteres.

El **rango de información** consiste en 3 numeros reales que definen los valores maximo, minimo y "por defecto" para el botón. Para los botones TOG el minimo es configurado cuando el boton se haya presionado y el maximo cuando está sin presionar.

La **nota emergente (tool tip)** es una cadena de caracteres que se mostrará cuando el puntero esté encima del botón (si el usuario tiene los tool tips activos). Tiene un límite de 80 caracteres y se tiene que poner a NULL (") si no se utiliza.

- **typedef struct Cast {...};**

La estructura Cast se usa en la llamada a la función doit, y simplifica el acceso de cada plugin a los valores de los datos.

La estructura Cast debería contener, en este orden, un entero o numero real por cada botón definido en varstr, incluidos los botones de texto. Tipicamente deberian tener el mismo nombre que el botón para permitir una busqueda o una referencia sencilla.

- **float result[8];**

El vector result se usa para enviar y recibir información al plugin. Los valores de result estan definidos de esta manera:

Result Indice	Significado	Rango
result[0]	Valor de la intensidad	0.0 to 1.0
result[1]	Valor del color rojo	0.0 to 1.0
result[2]	Valor del color verde	0.0 to 1.0
result[3]	Valor del color azul	0.0 to 1.0
result[4]	Valor del color transparente (alpha)	0.0 to 1.0
result[5]	Valor del desplazamiento normal en X	-1.0 to 1.0
result[6]	Valor del desplazamiento normal en Y	-1.0 to 1.0
result[7]	Valor del desplazamiento normal en Z	-1.0 to 1.0

El plugin siempre devuelve el valor de la intensidad. Devolver el color (RGB) o la normal es opcional, y se debería indicar con el "return" de la funcion doit(), siendo "1" para (RGB) y "2" para (Normal).

Antes de que se llame al plugin, Blender asigna el resultado de renderizar normalmente en result [5], result[6] y result[7].

- **float cfra**

El valor cfra actualizado por Blender a su valor actual antes de que todos los renders se hagan. Este valor es el numero de frame +/- .5 dependiendo de las configuraciones del campo.

- **Prototipo de plugin_tex_doit**

La funcion plugin_tex_doit deberia ser prototipada para que pueda ser usada por la funcion getinfo. No necesita cambiar esta linea.

- **plugin_tex_getversion**

Esta funcion deberia estar en cada plugin para que pueda ser cargado correctamente. No debería cambiar esta función.

- **plugin_but_changed**

Esta función se usa para pasar información sobre que botones cambia el usuario en la interface. La mayoría de los plugins no necesitan esta función, solo cuando el interface permita al usuario modificar algunas variables que fuerze al plugin a recalcularse (una tabla hash aleatoria por ejemplo).

- **plugin_init**

Si fuese necesario los plugins pueden utilizar esta funcion para inicializar datos internos. NOTA: Esta función de iniciacion se puede llamar varias veces si el mismo plugin de textura se copia. En esta función, no inicialice datos globales especificos a una unica instancia de un plugin.

- **plugin_getinfo**

Esta función se usa para comunicar información a Blender. Nunca deberías cambiarla.

- **plugin_tex_doit**

La función doit es la responsable de devolver información a Blender sobre pixeles requeridos.

Los argumentos

- **int stype**

Este es el número de subtipos seleccionados, mira las entradas **NR_TYPES** y **char stypes** anteriores.

- **Cast *cast**

La estructura Cast que contiene los datos del plugin, mira la entrada **Cast** anterior.

- **float *texvec**

Esto es un puntero a 3 floats, los cuales son las coordenadas para las cuales un valor de la textura debe retornarse.

- o **float *dxt float *dyt**

Si los punteros son no-NULL apuntarán a 2 vectores (2 arrays de 3 floats) que definen el tamaño del valor textura requerida en el espacio pixel Apuntan solo a no-NULL cuando el OSA esté activado, y se usan para calcular adecuadamente el antialiasing.

La función doit debería rellenar el array result y devolver 0, 1, 2 o 3 dependiendo de que valores hayan rellenado el array. La función doit debería *siempre* rellenarse con el valor intensidad. Si la función asigna el valor "color" debería devolver 1, si asigna el valor "normal" debería devolver 2 y si asigna todo debería devolver 3.

Interacción Texturas/Materiales

Blender es algo diferente de la mayoría de paquetes 3D en lo que se refiere a la separación lógica entre materiales y textura En Blender las texturas son objetos que devuelven ciertos valores, son generadores de señales de hecho. Los materiales controlan el mapeado de texturas en los objetos, lo que está afectado, en que cantidad, en que manera, etc. Plugins adecuadamente diseñados deberían incluir únicamente variables que afecten a la señal devuelta no al mapeado de ella. Es mejor incluir botones para controlar el escalado, el rango, los ejes, etc únicamente cuando hagan a la textura más fácil de usar (en caso del botón tamaño en el plugin Tiles) o aceleran el calculo (los subtipos Intensity/Color/Bump en el plugin Clouds2). De otra manera los botones del material hacen a esos botones redundantes, y la interfaz incrementa innecesariamente su complejidad.

[Prev](#)[Sistema de plugins de Blender](#)[Home](#)[Up](#)[Next](#)[Un plugin de textura genérico](#)

27.3. Un plugin de textura genérico

Relevante a Blender v2.31

```
#include "plugin.h"

/* Nombre de la textura */
char name[24]= "";

#define NR_TYPES 3
char stnames[NR_TYPES][16]= {"Intens","Color", "Bump"};
/* Estructura para los botones,
 * butcode name default min max 0
 */

VarStruct varstr[]= {
    {NUM|FLO, "Const 1", 1.7, -1.0, 1.0, ""},
};

typedef struct Cast {
    float a;
} Cast;

float result[8];
float cfra;
int plugin_tex_doit(int, Cast*, float*, float*, float*);

/* Funciones fijas */
int plugin_tex_getversion(void) {
    return B_PLUGIN_VERSION;
}

void plugin_but_changed(int but) { }

void plugin_init(void) { }
```



```
void plugin_getinfo(PluginInfo *info) {
    info->name= name;
    info->stypes= NR_TYPES;
    info->nvars= sizeof(varstr)/sizeof(VarStruct);

    info->snames= stnames[0];
    info->result= result;
    info->cfra= &cfra;
    info->varstr= varstr;

    info->init= plugin_init;
    info->tex_doit= (TexDoit) plugin_tex_doit;
    info->callback= plugin_but_changed;
}

int plugin_tex_doit(int stype, Cast *cast, float *texvec, float *dxt,
    float *dyt) {

    if (stype == 1) {
        return 1;
    } if (stype == 2) {
        return 2;
    }
    return 0;
}
```

[Prev](#)

Especificación

[Home](#)

[Up](#)

[Next](#)

Nuestras modificaciones

27.4. Nuestras modificaciones

Relevante a Blender v2.31

El primer paso es tener un plan. Para qué sirve este plugin y como los usuarios interactuarán con él. Para este ejemplo crearemos una textura sencilla que crea un simple patrón de ladrillos.

Ahora copiaremos nuestro plugin generico al archivo cube.c y rellenaremos los espacios vacíos.

Es siempre una buena idea añadir algunos comentarios. Lo primero de todo es decir a los usuarios qué hace el plugin, de donde pueden obtener una copia, con quién deberían contactar para mejorarlo o para informar de fallos y cual es la licencia del código. Cuando uses comentarios asegurate de usar `/* */`. Los plugins están en C y algunos compiladores de C no aceptan los comentarios con el estilo de C++ (`//`).

```
/*
    Descripción: Este plugin es un ejemplo de plugin de textura
que crea
    un simple patrón de ladrillos.

    Toma dos valores, el tamaño brick y el tamaño mortar.
    El tamaño brick es el tamaño de cada ladrillo.
    El tamaño mortar es el tamaño que hay entre los ladrillos
(cemento).

    Autor: Kent Mein (mein@cs.umn.edu)
    Sitio web: http://www.cs.umn.edu/~mein/bender/plugins
    Licencia: Dominio Público
    Última modificación: Tue Oct 21 0:57:13 CDT 2003
*/
```

Lo siguiente que necesitamos será rellenar el nombre, realmente debería ser el mismo que tu fichero .c, preferiblemente descriptivo, con menos de 23 caracteres, sin espacios y todas las letras en minúsculas.

```
char name[24]= "cube.c";
```

Vamos a mantener este plugin simple, solo tiene un tipo que está relacionado con la intensidad. Así que

necesitamos lo siguiente:

```
#define NR_TYPES          1
char stnames[NR_TYPES][16]= {"Default"};
```

Para nuestro interfaz vamos a permitir a la gente cambiar el tamaño del ladrillo, el espaciado entre ellos y los valores de intensidad devueltos por el ladrillo y el espaciado entre ladrillos. Para ello necesitamos modificar varstr y Cast. Cast debería tener una variable por cada entrada que haya en varstr.

```
/* Estructura para los botones,
 * butcode      name                default  min   max   Tool tip
 */
VarStruct varstr[]= {
    {NUM|FLO,    "Brick",            .8,      0.1,   1.0, "Size of Cell"},
    {NUM|FLO,    "Mortar",          .1,      0.0,   0.4, "Size of boarder in
cell"},
    {NUM|FLO,    "Brick Int",        1,       0.0,   1.0, "Color of Brick"},
    {NUM|FLO,    "Mortar Int",       0,       0.0,   1.0, "Color of Mortar"},
};

typedef struct Cast {
    float brick,mortar, bricki, mortari;
} Cast;
```

Ahora rellenaremos la función plugin_tex_doit. Basicamente queremos romper nuestra textura en "bloques" que estarán formados de un ladrillo y del espaciado a lo largo de las aristas del ladrillo (cemento) y entonces determinar si estamos en el ladrillo o estamos en el cemento. El código siguiente hace esto.

```
int plugin_tex_doit(int stype, Cast *cast, float *texvec, float *dxt,
    float *dyt) {
    int c[3];
    float pos[3], cube;

    /* configura el tamaño de nuestro bloque */
    cube = cast->brick + cast->mortar;

    /* necesitamos determinar donde estamos dentro del ladrillo
actual. */
    c[0] = (int)(texvec[0] / cube);
    c[1] = (int)(texvec[1] / cube);
    c[2] = (int)(texvec[2] / cube);
```

```
pos[0] = ABS(texvec[0] - (c[0] * cube));
pos[1] = ABS(texvec[1] - (c[1] * cube));
pos[2] = ABS(texvec[2] - (c[2] * cube));

/* Descubrir si estamos en una posición de cemento o no. */
if ((pos[0] <= cast->mortar) || (pos[1] <= cast->mortar) ||
    (pos[2] <= cast->mortar)) {
    result[0] = cast->mortari;
} else {
    result[0] = cast->bricki;
}
return 0;
}
```

Una cosa que destacar, la función ABS está definida en una cabecera que está en plugins/include. En ese lugar hay otras funciones de uso común por lo que sería aconsejable que echara un vistazo allí.

[Prev](#)

Un plugin de textura genérico

[Home](#)[Up](#)[Next](#)

Compilando:

27.5. Compilando:

Pertinente a Blender v2.31

bmake es una utilidad simple (shell script) para ayudar a la compilación y desarrollo de plugins, y puede encontrarse en el subdirectorio plugins del directorio de instalación de Blender. Se invoca mediante: bmake (nombre_plugin.c) e intentará enlazar las librerías adecuadas y compilar adecuadamente para su sistema el archivo C especificado. Si usted está intentando desarrollar plugins en una máquina windows, bmake puede no funcionar en este caso; debería usar lcc. Puede usar lo siguiente para compilar un plugin con lcc, asumiendo que usted tiene sus plugins en c:\blender\plugins: Aquí tiene un ejemplo de cómo se compilaría el plugin de textura sinus.c. Abra una ventana DOS y haga lo siguiente:

(Nota: Asegúrese de que el directorio lcc\bin está en su path)

```
cd c:\blender\plugins\texture\sinus
lcc -Ic:\blender\plugins\include sinus.c
lcclnk -DLL sinus.obj c:\blender\plugins\include\tex.def
implib sinus.dll
```

27.6. Escribiendo un plugin de Secuencia

Pertinente a Blender v2.31

En esta sección escribiremos un plugin de secuencia básico y luego seguiremos los pasos para usar un plugin de secuencia. Las bases de un plugin de secuencia son: se reciben unas entradas (1-3 buffers de imagen de entrada, así como alguna otra información), y se produce como resultado un buffer de imagen de salida.

Todos los archivos necesarios para el desarrollo de plugins así como unos pocos plugins de ejemplo pueden encontrarse en el directorio blender/plugins. Puede conseguir adicionalmente un puñado de plugins en <http://www.cs.umn.edu/~mein/blender/plugins>

27.6.1. Especificación:

Pertinente a Blender v2.31

- **#include <plugin.h>**

Todos los plugins de Blender deben incluir este fichero cabecera, que contiene todas las estructuras y definiciones necesarias para trabajar correctamente con Blender.

- **char name[]="Blur";**

Un string (cadena de caracteres) conteniendo el nombre del plugin, este valor se usará para indicar el título de la textura en la ventana de botones de textura.

- **VarStruct varstr[]={...};**

"varstr" contiene toda la información que Blender necesita para mostrar los botones del plugin. Los botones para los plugin pueden ser de tipo numerico para introducir datos, o de tipo texto para introducir comentarios u otra información. Se pueden utilizar como máximo 32 variables en los plugins.

Cada estructura VarStruct consta de un tipo, un nombre, un rango de información y una nota emergente sobre la herramienta.

El tipo define el tipo de dato para cada botón, y la manera en la que se muestra dicho botón. Para botones numericos este valor debería ser una combinación (ORed) de INT o FLO para el formato del numero y NUM, NUMSLI o TOG para el tipo de botón. Los botones de texto son de tipo LABEL.

El nombre es lo que se mostrará en el botón y está limitado a 15 caracteres.

El rango de información consiste en 3 numeros reales que definen los valores maximo, minimo y "por defecto" para el botón. Para los botones TOG el minimo es configurado cuando el boton se haya presionado y el maximo cuando está sin presionar.

La nota emergente (tool tip) es una cadena de caracteres que se mostrará cuando el puntero esté encima del botón (si el usuario tiene los tool tips activos). Tiene un límite de 80 caracteres y se tiene que poner a NULL (") si no se utiliza.

- **typedef struct Cast {...};**

La estructura Cast se usa en la llamada a la función doit, y simplifica el acceso de cada plugin a los valores de los datos.

La estructura Cast debería contener, en este orden, un entero o numero real por cada botón definido en varstr, incluidos los botones de texto. Tipicamente deberian tener el mismo nombre que el botón para permitir una busqueda o una referencia sencilla.

- **float cfra**

El valor cfra actualizado por Blender a su valor actual antes de que todos los renders se hagan. Este valor es el numero de frame +/- .5 dependiendo de las configuraciones del campo.

- **Prototipo de plugin_seq_doit**

La funcion plugin_seq_doit deberia ser prototipada para que pueda ser usada por la funcion getinfo. No necesita cambiar esta linea.

- **plugin_seq_getversion**

Esta funcion deberia estar en cada plugin para que pueda ser cargado correctamente. No debería cambiar esta función.

- **plugin_but_changed**

Esta función se usa para pasar información sobre que botones cambia el usuario en la interface. La mayoría de los plugins no necesitan esta función, solo cuando el interface permita al usuario modificar algunas variables que furze al plugin a recalcularse (una tabla hash aleatoria por ejemplo).

- **plugin_init**

Si fuese necesario los plugins pueden utilizar esta función para inicializar datos internos. NOTA: Esta función de iniciación se puede llamar varias veces si el mismo plugin de textura se copia. En esta función, no inicialice datos globales específicos a una única instancia de un plugin.

- **plugin_getinfo**

Esta función se usa para comunicar información a Blender. Nunca deberías cambiarla.

- **plugin_seq_doit**

La función de secuencia doit es responsable de aplicar el efecto del plugin y de copiar los datos finales en el buffer de salida.

Los argumentos

- **Cast *cast**

La estructura Cast, que contiene los datos del plugin; véase la entrada **Cast** más arriba.

- **float facf0**

El valor de la curva IPO del plugin para el offset del primer campo. Si el usuario no ha hecho una curva IPO, su rango va entre 0 y 1 para la duración del plugin.

- **float facf1**

El valor de la curva IPO del plugin para el offset del segundo campo. Si el usuario no ha hecho una curva IPO, su rango va entre 0 y 1 para la duración del plugin.

- **int x int y**

La anchura y altura de los buffers de imagen, respectivamente.

- **Imbuf *ibuf1**

Un apuntador al primer buffer de imagen al que el plugin está enlazado. Será siempre un buffer de imagen válido.

- **Imbuf *ibuf2**

Un apuntador al segundo buffer de imagen al que el plugin está enlazado. Los plugins que usan este buffer deberían comprobar si es un buffer NULL, pues el usuario puede no haber enlazado el plugin a dos buffers.

- **Imbuf *out**

El buffer de imagen de la salida del plugin.

- **Imbuf *use**

Un apuntador al tercer buffer de imagen al que el plugin está enlazado. Los plugins que

usan este buffer deberían comprobar si es un buffer NULL, pues el usuario puede no haber enlazado el plugin a tres buffers.

Estructura de imagen ImBuf

La estructura ImBuf contiene siempre 32 bits de datos de píxel ABGR.

Las estructuras ImBuf son siempre iguales en tamaño, indicado por los valores de **x** e **y**.

Interacción con el usuario

Blender no tiene ninguna manera de saber cuántas entradas espera el plugin, por lo que le es posible a un usuario adjuntar sólo una entrada a un plugin que espera dos. Por esta razón es importante comprobar siempre los buffers que usa el plugin para asegurarnos de que son todos válidos. Los plugins de secuencia deberían incluir también una etiqueta de texto describiendo el número de entradas requeridas en la interfaz de botones.

27.6.2. Plugin de Secuencia Genérico:

```
#include "plugin.h"
char name[24]= "";

/* estructura para los botones,
 * butcode name default min max 0
 */

VarStruct varstr[]= {
    { LABEL, "In: X strips", 0.0, 0.0, 0.0, "},
};

/* La estructura cast es para entrada en la función doit principal
   Varstr y Cast deben tener las mismas variables en el mismo orden */

typedef struct Cast {
    int dummy; /* debido al botón 'etiqueta' */
} Cast;

/* cfra: el cuadro actual */

float cfra;
```

```

void plugin_seq_doit(Cast *, float, float, int, int,
    ImBuf *, ImBuf *, ImBuf *, ImBuf *);

int plugin_seq_getversion(void) {
    return B_PLUGIN_VERSION;
}

void plugin_but_changed(int but) {
}

void plugin_init() {
}

void plugin_getinfo(PluginInfo *info) {
    info->name= name;
    info->nvars= sizeof(varstr)/sizeof(VarStruct);
    info->cfra= &cfra;

    info->varstr= varstr;

    info->init= plugin_init;
    info->seq_doit= (SeqDoit) plugin_seq_doit;
    info->callback= plugin_but_changed;
}

void plugin_seq_doit(Cast *cast, float facf0, float facf1, int xo,
int yo,
    ImBuf *ibuf1, ImBuf *ibuf2, ImBuf *outbuf, ImBuf *use) {
    char *in1= (char *)ibuf1->rect;
    char *out=(char *)outbuf->rect;
}

```

27.6.3. Nuestras modificaciones:

El primer paso es diseñar un plan de juego. Qué hará este plugin, cómo interactuarán con él los usuarios. Para este ejemplo crearemos un filtro simple que tendrá un botón de deslizamiento para una intensidad de 0-255. Si alguno de los componentes R, G o B de un píxel en la imagen fuente es menos de la intensidad seleccionada, devolverá negro y alfa; de lo contrario devolverá lo que haya en la imagen. Ahora copiaremos nuestro plugin genérico a simpfilt.c y rellenaremos los vacíos.

Siempre es una buena idea añadir algunos comentarios. Primero, explicar a los usuarios lo que hace el

plugin, dónde pueden obtener una copia, a quién deberían contactar para fallos/mejoras, y cualquier restricción de licencia sobre el código. Al usar comentarios, debemos asegurarnos de usar comentarios del estilo `/* */`. Los plugins están en `c`, y algunos compiladores de `c` no aceptan comentarios del estilo `//`.

```
/*
Descripción: Este plugin es un ejemplo de plugin de secuencia que
filtra píxeles
de baja intensidad. Funciona con un Strip como entrada.
Autor: Kent Mein (mein@cs.umn.edu)
Sitio web: http://www.cs.umn.edu/~mein/blender/plugins
Licencia: Dominio Público
Última modificación: Sun Sep  7 23:41:35 CDT 2003
*/
```

Después necesitamos rellenar el Nombre, debería ser el mismo que el archivo `.c`. Preferiblemente descriptivo, menos de 23 caracteres, sin espacios, y todo minúsculas.

```
char name[24]= "simpfilt.c";
```

Cast y varstr necesitan estar sincronizados. Queremos una barra de desplazamiento por lo que haremos lo siguiente:

```
varStruct varstr[]= {
    { LABEL,      "In: 1 strips", 0.0, 0.0, 0.0, "},
    { NUM|INT,    "Intensity", 10.0, 0.0, 255.0, "Our threshold
value"},
};

typedef struct Cast {
    int dummy;                                /* debido al botón 'etiqueta'
*/
    int intensity;
} Cast;
```

Ahora necesitamos rellenar `plugin_seq_doit`. Básicamente queremos pasar por cada píxel y si RGB es menor que la intensidad seleccionada, coloca el píxel de salida a: 0,0,0,255; si no, devuelve el mismo valor que el píxel de entrada para esa posición.

```
int x,y;
```

```

for(y=0;y < cast->intensity) &
    (in1[1] > cast->intensity) &
    (in1[2] > cast->intensity)) {
    out[0] = out[1] = out[2] = 0;
    out[3] = 255;
} else {
    out[0] = in1[0];
    out[1] = in1[1];
    out[2] = in1[2];
    out[3] = in1[3];
}
}
}

```

Terminamos pues con `simpfilt.c`.

27.6.4. Compilando:

`bmake` es una utilidad simple (shell script) para ayudar a la compilación y desarrollo de plugins, y puede encontrarse en el subdirectorio `plugins` del directorio de instalación de Blender. Se invoca mediante: `bmake (nombre_plugin.c)` e intentará enlazar las librerías adecuadas y compilar el archivo C especificado adecuadamente para su sistema. Si usted está intentando desarrollar plugins en una máquina windows, `bmake` puede no funcionar en este caso; debería usar `lcc`. Puede usar lo siguiente para compilar un plugin con `lcc`, asumiendo que usted tiene sus plugins en `c:\blender\plugins`: Aquí tiene un ejemplo de cómo se compilaría el plugin de secuencia `sweep.c`. Abra una ventana DOS y haga lo siguiente: (Nota: Asegúrese de que el directorio `lcc\bin` está en su path)

```

cd c:\blender\plugins\sequence\sweep
lcc -Ic:\blender\plugins\include sweep.c
lcclnk -DLL sweep.obj c:\blender\plugins\include\seq.def
implib sweep.dll

```

[Prev](#)

Compilando:

[Home](#)

[Up](#)

[Next](#)

Más allá de Blender

VII. Más alla de Blender

Al ser Blender un programa extendible, una serie de scripts para exportar modelos a otros motores de renderizado han sido escritos por usuarios entusiastas.

En esta parte nos centraremos unicamente en un script, Yable, asi como en uno de esos motores externos de renderizado, YafRay. YafRay ha sido elegido sobre otros porque es nuevo, muy prometedor , y está siendo desarrollado muy activamente por un grupo de 'históricos' usuarios de Blender.

Table of Contents

- 28. [Yafray como motor de render integrado en Blender](#)
 - 28.1. [Parte 1](#)
 - 28.1.1. [La interfaz](#)
 - 28.1.2. [Paso-1](#)
 - 28.1.3. [Paso-2](#)
 - 28.1.4. [Paso-3](#)
 - 28.2. [Parte 2](#)
 - 28.2.1. [Y la luz se hizo... y hubo YafRay en todos los lados](#)
 - 28.3. [Parte 3 \(Aún por traducir\)](#)
 - 28.3.1. [GI and other features](#)
 - 28.3.2. [The available *Methods* of GI](#)
 - 28.3.3. [Features for Full method of GI](#)
 - 28.3.4. [Other features of Yafray](#)
 - 28.3.5. [The Depth of Field](#)
 - 28.3.6. [HDRI or High Dynamic Range Illumination](#)
 - 28.3.7. [Closing comments](#)
 - 28.4. [Glosario para los geeks \(Aún por traducir\)](#)
[Glosario para los geeks \(Aún por traducir\)](#)
- 29. [De Blender a Yafray usando YableX](#)
 - 29.1. [¿Qué es Yable?](#)
 - 29.1.1. [¿Qué Yable?](#)
 - 29.1.2. [¿Dónde conseguir YableX?](#)
 - 29.2. [Instalando el script](#)
 - 29.3. [La interfaz](#)

29.3.1. [Filosofía del flujo de trabajo.](#)

29.3.2. [Parámetros Globales](#)

29.3.3. [Ajustes de Material](#)

29.3.4. [Ajustes de luces](#)

29.4. [Ejemplo de utilización de Yable](#)

30. [YafRay](#)

30.1. [Introducción](#)

30.2. [Instalación](#)

30.2.1. [YafRay para Windows](#)

30.2.2. [YafRay para Mac OSX](#)

30.2.3. [YafRay en Linux](#)

30.3. [Repaso del Lenguaje de Descripción de Escenas](#)

30.4. [Sombreadores](#)

30.4.1. [Sombreadores Base](#)

30.4.2. [Procedural](#)

30.4.3. [Meta Sombreadores](#)

30.5. [Objetos renderizables](#)

30.5.1. [Etiqueta Transform](#)

30.5.2. [Etiqueta Object](#)

30.5.3. [La etiqueta Mesh:](#)

30.6. [Luces](#)

30.6.1. [Luz Spot](#)

30.6.2. [Luz Point](#)

30.6.3. [Luz Sun](#)

30.6.4. [Luz Soft](#)

30.6.5. [Luz Area](#)

30.6.6. [Luz Path](#)

30.6.7. [Luz Hemi](#)

30.6.8. [Luz Photon](#)

30.7. [Fondo](#)

30.7.1. [Fondo de imagen normal](#)

30.7.2. [Fondo HDRI](#)

30.7.3. [Fondo constante](#)

30.7.4. [Fondo de sol/cielo](#)

30.8. [Cámara](#)

30.9. [Renderizado](#)

30.10. [Filtros](#)

30.10.1. [Filtro anti ruido](#)

30.10.2. [Profundidad de Filtro de Campo](#)

[Prev](#)

Escribiendo un plugin de
Secuencia

[Home](#)

[Next](#)

Yafray como motor de render
integrado en Blender

Chapter 28. YafRay como motor de render integrado en Blender

Relevante a Blender v2.34

por Gaurav Nawani

28.1. Parte 1

La integración de YafRay es una de las mejores funcionalidades añadidas a Blender. La actual versión de Blender 2.35 tiene una sencilla y efectiva integración de Blender y YafRay, además de ser bastante estable. Desafortunadamente, el uso de YafRay está limitado para aquellos que ya manejan con soltura la interfaz de Blender. La primera parte de este tutorial corresponde a los pasos básicos necesarios para renderizar con YafRay, y las partes siguientes sirven como guía de uso para el resto de características disponibles en YafRay.

28.1.1. La interfaz

Blender tiene, por sí mismo, dos motores de render internos: su propio raytracer y el antiguo motor de render por scan-line. YafRay es, al contrario que los anteriores, un raytracer independiente. A través de la interfaz de Blender se puede acceder a sus funcionalidades, exportando los parámetros de la escena a un formato entendible por YafRay. Esto se puede realizar de dos maneras. Para la primera de ellas Blender posee el soporte interno que permite usarlo como un plug-in, virtualmente como si YafRay fuese uno de los motores de render internos de Blender. La otra opción es que Blender exporte los datos de la escena a un fichero en el formato XML usado por YafRay, llamado *YBtest.xml*, y después YafRay lo renderice como programa independiente (de manera externa a Blender). Ambos métodos necesitan que YafRay se encuentre instalado en el sistema, de hecho este tutorial asume que tienes instalados tanto Blender 2.34 como YafRay 0.0.7.

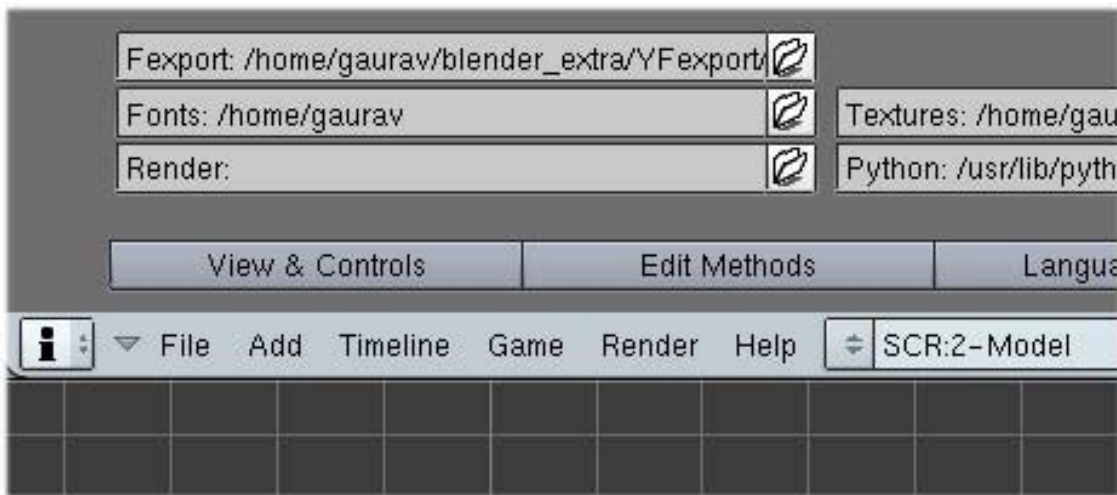
La integración de YafRay en Blender se divide, en términos generales, en dos partes. Una relacionada con el interfaz para configurar la iluminación (luces / lamps). La otra se refiere al núcleo de las características de render de YafRay.

28.1.2. Paso-1

Hay que tener en cuenta que este paso podría no ser necesario en Windows. Antes de proceder, necesitamos configurar primero las preferencias de YafRay. Arrastra hacia abajo la barra de menu superior para mostrar las *preferencias de usuario* de Blender. Selecciona el botón *File Paths* para abrir el menú de los Path (rutas). En la parte superior izquierda hay una caja de texto, *YFExport*. Introduce el directorio donde quieres que se grabe el fichero XML que contiene la exportación de la escena de Blender en formato YafRay. Esto se usa sólo cuando se renderiza utilizando la segunda opción. También es necesario si deseas salvar el fichero exportado para editarlo

manualmente.

Figure 28-1. Selección del Path en el menú de preferencias de usuario.



28.1.3. Paso-2

Como queda dicho, Blender permite elegir entre su propio raytracer interno y YafRay, por lo que para usarlo primero tendremos que indicárselo a Blender. Para ello, pulsa **F10** para mostrar la ventana Rendering Options, y ve al panel *Render* ([Figure 28-2](#)). En dicho panel, selecciona *YafRay* en la lista desplegable ([Figure 28-3](#)), por defecto está seleccionado *Blender internal*.

Note No te confundas con el botón Ray. Activarlo no sirve para nada cuando se usa YafRay. No tiene ningún efecto sobre el resultado si YafRay está seleccionado como raytracer en la lista desplegable.

Figure 28-2. Las opciones por defecto en el panel Render.



Fíjate que al seleccionar YafRay aparecen dos nuevas solapas detrás de la de *Render* ([Figure 28-3](#)). Ambas solapas, *YafRay* y *YafRay GI*, se utilizan para controlar los parámetros de render y demás características de YafRay.

Figure 28-3. Selección de YafRay como raytracer y las dos solapas extra.



Ahora selecciona la solapa *YafRay* de entre las dos nuevas, aquí veremos algunas funciones nuevas. Por ahora sólo nos interesa el botón llamado *XML* (activado por defecto) ([Figure 28-4](#)). Púlsalo para tenerlo desactivado por ahora. Explicaremos por qué, como se ha dicho anteriormente, Blender tiene dos opciones para renderizar con YafRay, o bien vía plug-in o bien llamando a YafRay de manera externa, como un programa de línea de comandos.

Figure 28-4. Solapa de YafRay por defecto con el fichero de exportación activado. (botón XML).

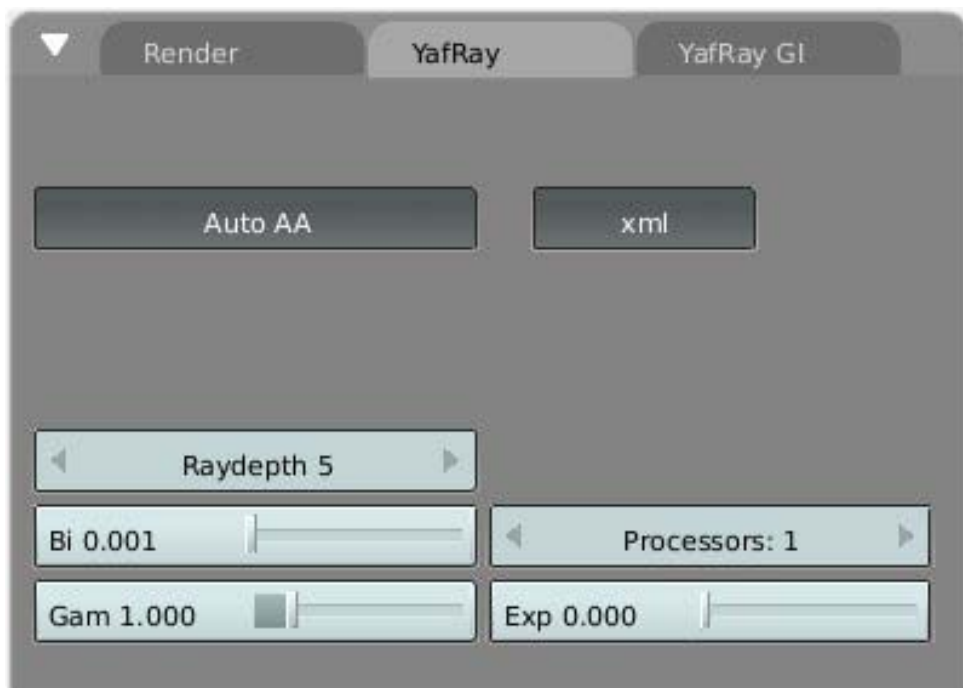
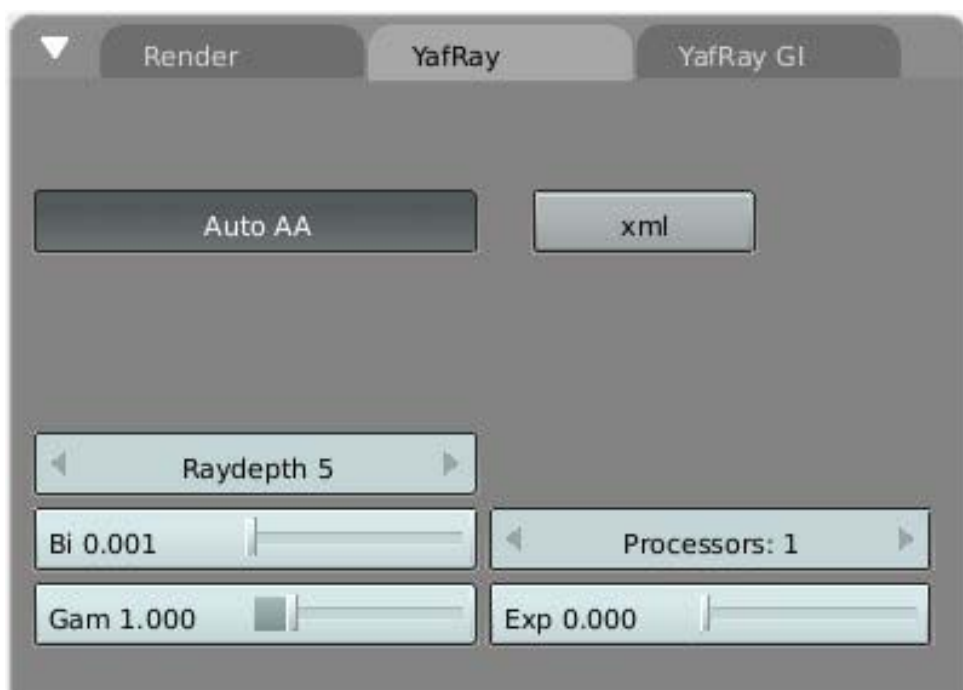
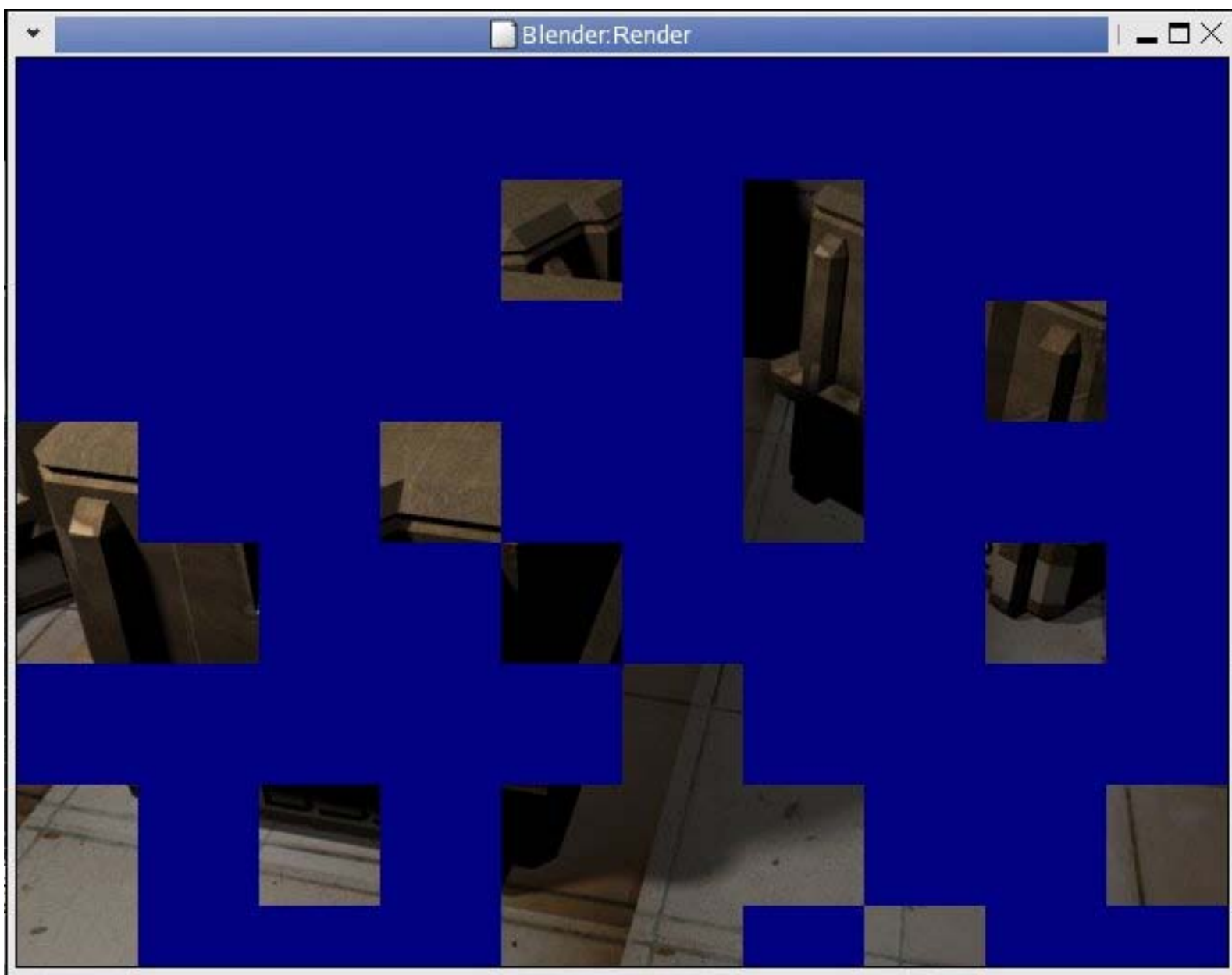


Figure 28-5. Exportación a fichero de YafRay desactivada.



Al escoger la primera opción de render, vía plug-in (botón *XML* desactivado), podremos ver el progreso del *render* en la ventana *Render* ([Figure 28-6](#)), prácticamente igual que con el propio motor de Blender, y esta es una de las razones por las que YafRay requiere y utiliza algo más de memoria que antes. Usando la segunda opción, la cual es activada por defecto (botón *XML* pulsado), primero ha de exportar la escena al directorio que definimos con *YFExport* en las preferencias, como vimos en [Section 28.1.2](#). Después se llama a YafRay como programa de línea de comandos, y empieza el proceso del fichero XML exportado. Sólo al final, cuando el renderizado ha acabado, se muestra el resultado en la ventana de *Render*. Por lo tanto, no se produce ninguna interacción, salvo por la información en modo texto mostrada en la ventana de terminal de Blender.

Figure 28-6. El render en progreso con XML desactivado.



28.1.4. Paso-3

Tras seguir los dos primeros pasos ya tienes todo lo que se necesita para realizar un render con YafRay. En este punto tienes tu escena con las luces que has elegido, lo mejor para seguir desde aquí será pulsar **F12** para empezar el render. Y aquí es donde los *problemas* surgen para casi todo el mundo. Me explico... Dependiendo de la configuración de luces de tu escena, puedes ver una pantalla en negro o ver unos contornos débiles de los objetos de la escena. En otro caso si tus luces tienen una intensidad (valor) de luz muy alta verás la escena correctamente. Otra opción, en un caso extremo, es que YafRay y Blender den un error fatal.

En el caso de obtengas un render negro hay grandes posibilidades de que puedas renderizar con YafRay; todo lo que necesitas es ajustar los parámetros de las luces y tendrás tu propio render con YafRay en minutos.

28.1.4.1. Puntos para evaluar problemas

- El primer problema de los renders con YafRay es que prácticamente todas las veces la intensidad de luz es insuficiente para la escena. En realidad, esto no es un problema de YafRay sino que se debe a motivos de implementación de la integración con Blender. Por lo que, en este caso, necesitas incrementar la intensidad de luz, o su valor, de cada una de las fuentes de luz de la escena (más información en [Section 28.2](#)).
- Si todo falla, necesitarás comprobar los valores de la distancia de las luces, ya que la atenuación de la luz es bastante más pronunciada en los gráficos por computador (para reducir cálculos). Los objetos en la escena podrían aparecer negros porque su distancia de ray casting (algo así como la distancia máxima de colisión del rayo con el objeto. N. del T.), podría no alcanzar a ninguno de los objetos de la escena. Puedes solucionar esto ajustando el valor de la distancia en el panel *Lamps*.
- En caso de error fatal, por favor busca posibles soluciones en los foros de elysium (Normalmente los errores de este tipo (crash) están más relacionados con YafRay que con Blender, así que lo mejor es preguntar en el foco del problema: los foros de YafRay. N. del T.)

[Prev](#)

Más alla de Blender

[Home](#)

[Up](#)

[Next](#)

Parte 2

28.2. Parte 2

En lugar de enseñarte a modificar las configuraciones por aquí y por allá, creo que podría ser mejor explicar los tipos de luz de YafRay uno a uno. Esto te permitirá tomar decisiones con conocimiento de causa.

28.2.1. Y la luz se hizo... y hubo YafRay en todos los lados

Hay cinco tipos de luz utilizadas en Blender y sólo cuatro de ellas están directamente soportadas por YafRay. Al activar YafRay como raytracer en el panel de *Render*, inicialmente verás seis tipos de luz (en la ventanas de opciones de luz, **F5**), de las cuales YafRay utiliza Lamp, Area, Spot, y Sun. Volveré más tarde sobre las otras dos luces de la lista, Hemi y Photons.

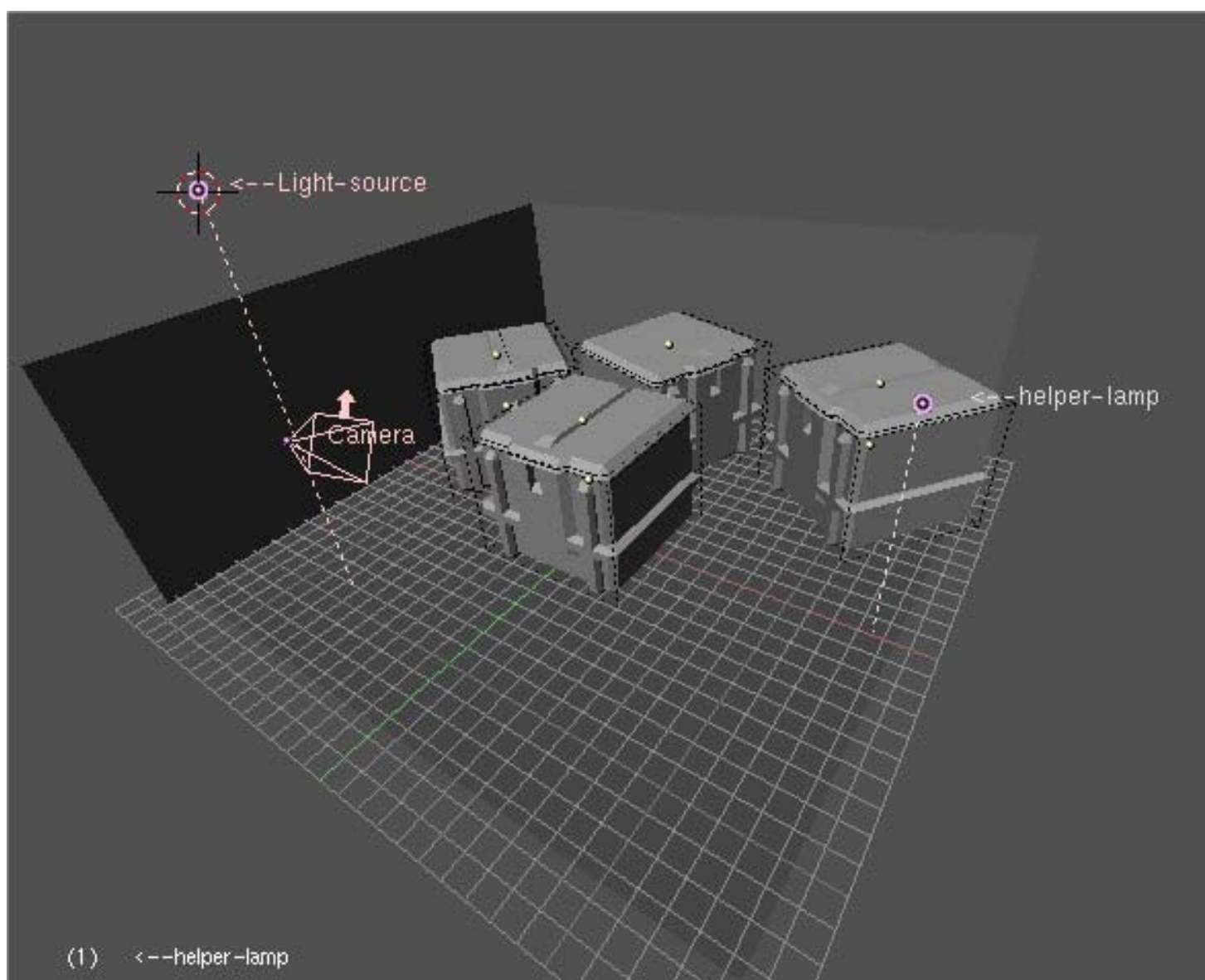
Figure 28-7. La escena.





Descripción de la escena: La escena en cuestión tiene dos fuentes de luz. Una es una luz de apoyo (*Sun*, sin valor de sombra, shadow, y con valor .200). Esta luz proporciona luz no direccional que es utilizada para potenciar la atmósfera de la escena. La otra fuente de luz es una plantilla que será cada una de los tipos de luz a explicar. Las posición de dicha fuente es estática y tiene activada el *Ray Shadow* (shadow casting) para todos los tipos de luz que utilizaremos (ver [Figure 28-8](#)).

Figure 28-8. La escena como se ve en el Viewport.



Note Los usos sugeridos para los tipos de luz definidos aquí son generales, como ilustración para su uso. No han de ser tomados como la verdad absoluta para la iluminación de tus escenas. La iluminación varía dependiendo de la visión del artista sobre la misma. Para obtener mejores resultados, experimenta con las luces de tu escena hasta encontrar el mejor resultado para tí.

**Importante**

Los tiempos de render de las distintas imágenes se han redondeado, ya que son relativos y sólo los utilizamos para dar una idea de la variación de los tiempos de render para la misma escena, con diferentes tipos de luz y configuración.

28.2.1.1. Lamp

Esta es la fuente de luz más simple de entre todas las disponibles. En algunos casos, es como una luz solar, en el sentido de que es omnidireccional (es decir, emite luz en todas las direcciones desde el punto origen) y esférica por naturaleza. Aunque lanza rayos de luz en todas las direcciones, dichos rayos pueden perderse con lo que van a desperdiciar parte del proceso de cálculo. Aunque los raytracers utilizan una serie de trucos para eliminar cálculos indeseados, sigue siendo un poco más lenta que otras fuentes de luz direccionales, en el caso de que la mayor parte de la luz recaiga dentro de la parte visible de la escena.

El uso más frecuente de la fuente de luz Lamp es la *iluminación interior*, como en habitaciones y pasillos, pero no está limitada a este uso ya que también se utiliza como *luz de ayuda o de relleno*.

Values: When a single lamp is used for the scene in Yafray (through-out the tutorial when I say Yafray I mean Blender-Yafray), it will generally not provide sufficient lighting to light up the scene unless it is kept at a higher value. For the sample scene, the value of Lamp was set at 10 (maximum) but still there seems to be the need for additional lighting (but we are sticking to it for the sake of understanding). See [Figure 28-9](#).

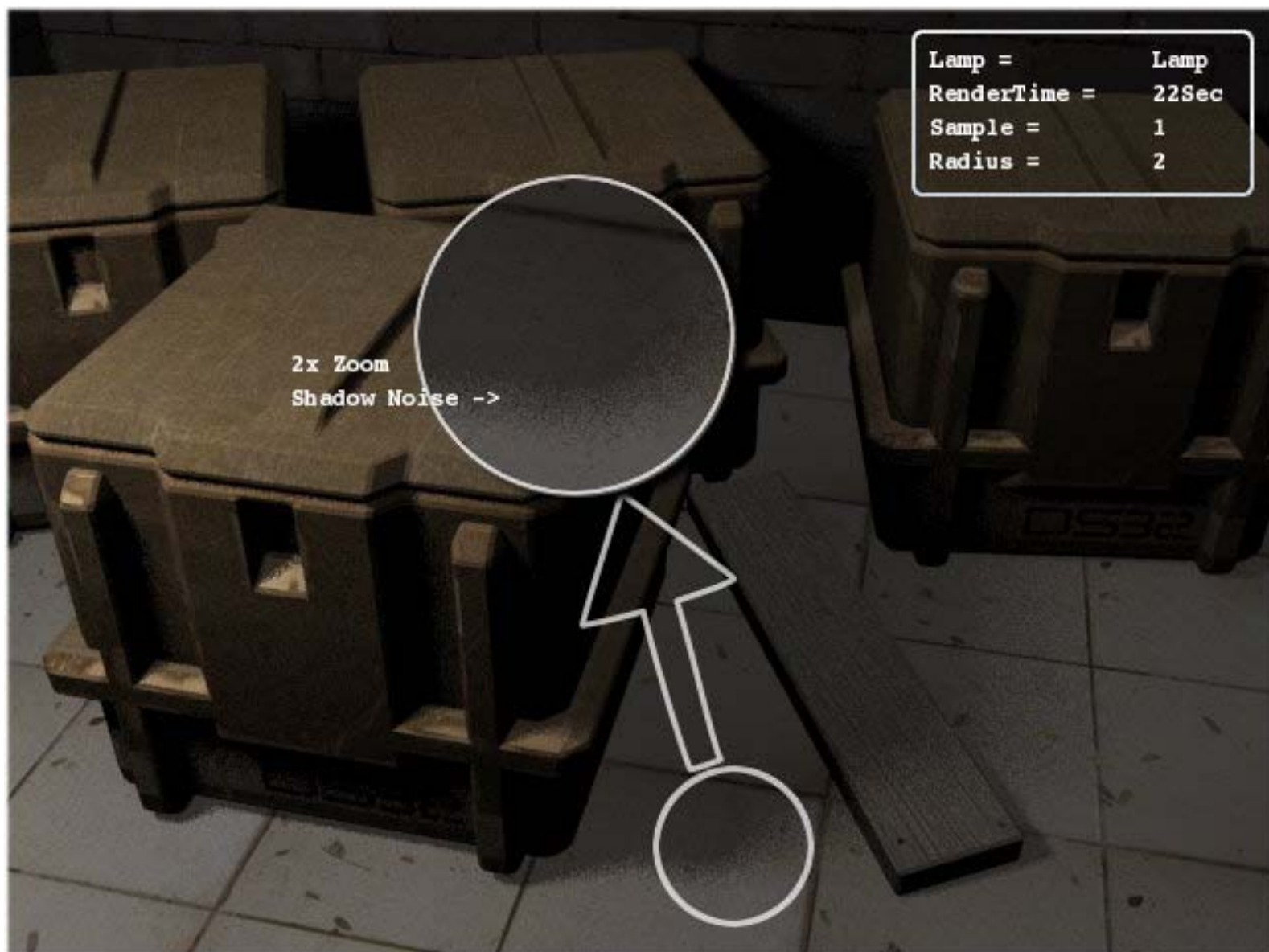
Figure 28-9. Rendered with Lamp at default settings.



The Lamp will cast sharper shadows at default (zero) radius. The radius setting in Lamp is used to increase the area size for shooting the light. If the size of Lamp is bigger than the objects, then the light shot from some parts of the lamp could directly reach to the parts of the scene where the other part of the lamp casts a shadow. This intersection results in the shadow being diluted for that part, thus getting blurred (also called *partial shadows*). You can observe this phenomenon in real world (sun light).

LINK: for more info on *shadow dilution* or Partial shadows go to [Ditto head's Light tutorial](#).

Figure 28-10. The noise comes when sampling values is low.



The scene above ([Figure 28-10](#)) was rendered at radius 2 with a sampling value at 1, as can be seen the scene has grainy partial shadows. The use of *samples* button is only to reduce the grain in shadows, the sampling buttons control the number of samples used in shadow calculations, increasing the sampling results in smoother shadows. The sampling button is only available in Lamp and Area Light and functions the same for both.

Figure 28-11. The increase in sampling to 5 remove the noise.



28.2.1.2. Area

The *Area Light* is a directional source. The shape of Area Light can be varied from square to rectangular from the drop down list in the *Lamp* tab. Below the shape selection there is also an option to increase the size of the *Area Light*.

Figure 28-12. The basic render from Area Light at default setting and value of 4.



Values: For Area Lamp you will need lower light intensity values for rendering a scene with Yafaray. The sample scene is lit by light value at 4, it is brighter than the one rendered by Lamp at value 10. This is because it is shooting all the light towards a direction from a plane while the Lamp shoots the light in every direction distributing energy where it may not be required.

Figure 28-13. Although the scene looks similar to [Figure 28-11](#) but is brighter and have better shadows.



Note One important point to remember is that in the official Build of Blender 2.34 using two or more Area Lights results in error in the renderings, it is a recognized bug and has been patched in some more recent builds.

28.2.1.3. Spot

The *Spot* is also a directional light source and as the name suggests it provides a circular spot of light or more appropriately a cone of light. The size of cone can be controlled by the *SpotSi* or angle of spotlight beam. The higher the angle the more nearer it behaves like an area light, but with one major difference. It cannot cast partial shadows. Spot light has a parameter to control the *SpotBi* or Spot's edge smoothness only. It can be seen only if the Spot light is inside the scene view, otherwise you will not notice any difference in Spot or other lights.

Figure 28-14. The Spot lamp render with default settings.



Figure 28-15. Spot lamp with spherical light attenuation.



Notice the increased brightness at the portion nearer to the lamp in the [Figure 28-15](#) while using the Spherical light attenuation (Quad is not supported in Yafaray). Increase in the *distance* value the spherical attenuation also increases in radius, for example the [Figure 28-15](#) has a default *distance* value of 20 while [Figure 28-16](#) has the *distance* value at 30. The effect looks like that of typical household lamps where the attenuation is sharper.

Note The light attenuation in Yafaray has only quadratic falloff, while Blender can have linear, cubic and the mix of these also. That explains relatively sharper light intensity falloff in Yafaray renders.

Figure 28-16. Spot lamp with Spherical attenuation enabled.



28.2.1.4. Sun

This is also a directional light source. It tries to emulate sunlight by shooting light of the same intensity everywhere in the scene without attenuation ([Figure 28-17](#)). This results in the environment being lit up with constant ambience. Its value must be kept lower. The sample scene uses a light intensity value of .100. It is obviously good for out door lights, especially for a sun.

In outdoor scenes, if you do not want sharper shadows you can disable shadow casting for Sun, and use other lamps for shadow generation. But make sure they have sufficient light intensity to cast a shadow.

Figure 28-17. The Sun light at value 2. No wonder why this lamp is called Sun.



28.2.1.5. More lamps?

Hemi - Yafray does not support Blender's implementation of Hemi light internally; right now it just uses Yafray's implementation of Sun light instead.

Photon - Last of all, the Photon light source or *photons lamp* button is not to be confused with any light source. It does not cast any shadow or light, it is only used in Caustics* calculations and requires being placed or directed where you need the caustic calculation. Photon lamp shoots photons in an area, the photons are used by Yafray as the specialized ray elements only to calculate the caustics on passing through the objects like glass or mirrors, which have the property of bending light when passed through them, known as *total internal reflection*. Basically the placement of Photon lamp is to allow the user to optimize the rendering. Caustics is one of the most computation intensive jobs in Ray tracing.

Note This photon lamp and photons are not to be confused with the photon option in the GI method, which is explained in the [Section 28.3](#).

[Prev](#)

Yafray como motor de render
integrado en Blender

[Home](#)

[Up](#)

[Next](#)

Parte 3 (Aún por traducir)

28.3. Parte 3 (Aún por traducir)

28.3.1. GI and other features

Yafray supports *Global Illumination**. We have to go back to the *Render* tab to learn more about Yafray's implementation of GI. The first look at the *Yafray GI* tab and you might think 'Oh! It is so easy.' You are right, the features available are simplified for your use. The available options are *Method* and *Quality*, let's have a look at them one by one. The quality is the same for each and is explained later.

Figure 28-18. The Yafray GI tab.



28.3.2. The available *Methods* of GI

28.3.2.1. Skydome

While Skydome is simply a method to have lighting from the sky or more appropriately the atmosphere, it does not provides full GI in the true sense, as it does not takes into account the indirect light bounces on the surfaces of objects. Rather it affects the light in the scene by the colour of the atmosphere and also the diffusion of light in the atmosphere is controlled by it. For example in the default scene the colour of atmosphere was set at reddish (chosen just to clearly identify the effect). See [Figure 28-19](#) for how the diffusion affects the scene.

Figure 28-19. The left part of image have the power of diffusion set at 2 while the right one at 4.



The colour of the background or atmosphere can be changed in the *Word Panel* (**F8**). You have to enable the world for Skydome to work fully. The Skydome offers no other features. The Skydome is faster method of GI, while its results may not be physically more accurate, its results are not too bad to ignore for the speed advantage it offers.

28.3.2.2. Full

On the other hand, the Full option takes into consideration the reflected and/or refracted light bounces (or indirect light) on neighboring surfaces. This method is a very close simulation of actual lighting in the real world, and that is why the GI produces more photo-realistic images than any other method used in CG, that is also why it requires higher computations and is slower.

Figure 28-20. Using Full GI increases the time as well as realism.



Notice the difference in realism between the two parts in [Figure 28-20](#). The light bounces provide enough diffusion in the second part of the image, which lights up the shadow portion in the scene according to the distance of the surfaces in contact. While the first part has approximately similar shadow depth through out.

Figure 28-21. The Yafaray GI tab. Options for Full method are visible.



28.3.3. Features for Full method of GI

28.3.3.1. Depth:

This refers to the total number of light bounces for one reflected or diffuse ray. Generally *Depth* 3 works best for normal scenes. Increasing the *Depth* results in higher amounts of calculation per ray.

28.3.3.2. Cdepth:

The *Cdepth* is nothing but the bounce depth for caustics, or for the transparent materials like the glass, gems, liquids etc. Higher computations are required for caustics calculations, that is why the GI in CG keeps separate the normal lights with photons, even though in the real world the photons are the actual light particles. For better quality of caustics you will need to have as much as 3-5 *Cdepth* or more for detail.

28.3.3.3. Photons:

This *Photons* button should not be confused with the photon lamp. This is only used as a helper in global illumination for YafRay and bears no relation with the caustic photon lamps or caustic photons.

28.3.3.4. Count:

The *Count* refers to the total number of photons to be made available in the scene for helper in GI. The number of photons will vary for the scene, but higher numbers in the scene will provide smoother results, again it's at your disposal to find the time/quality limit.

28.3.3.5. Radius:

This refers to the distance within which the photons calculated have higher precision value to effect the GI, outside which the photons do not effect the scene much. So the idea is to optimize the radius to the size of the area where better GI calculation is required; keep the radius half the size of area.

28.3.3.6. MixCount:

This allows you to choose the number of photons which should be kept inside the Radius. The *Radius* and *MixCount* make up your *photon map*. The photon-map is nothing but the optimized area for photons used in GI calculation and Yafray sufficiently processes the sampling and photon gathering on the photons inside the photon-map and leaves the photons outside the photon-map.

28.3.3.7. Tune Photons:

This allows the re-use of successful gathering of photons and positions from cache, which helped in GI calculation on previous render, and only re-compute the other photons for faster result. It is may not necessarily faster in every rendering but can provide good speed improvements generally. Due to speed improvement it is good to use it during test renders once you come close to what you wanted. You can disable it after you are satisfied with the test renders and want to create the final version of the image to do the full computation again.

28.3.3.8. Quality:

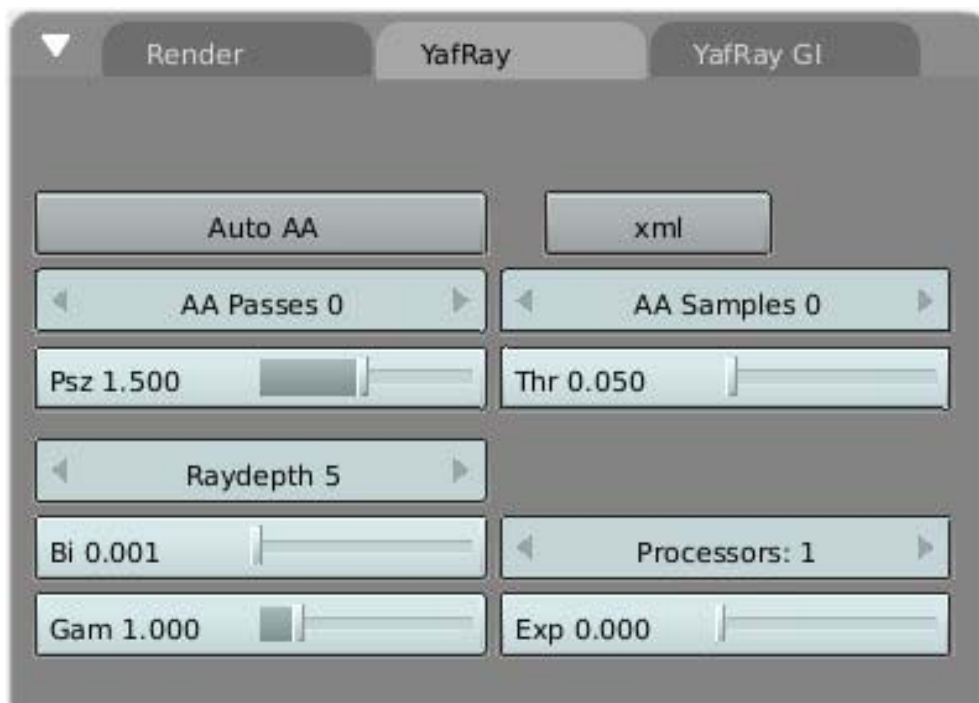
The quality dropdown list allows you to choose the quality level of GI for both the Skydome and Full GI methods. The *Quality* setting allows control over the number of samples used to test in GI. Since its use is automatic by default, it will try to used maximum value for *MultiPasses* and *samples per pass* depending upon the quality level. If you want, you can disable the *AutomaticAA* manually and can instruct Yafray accordingly.

28.3.4. Other features of Yafray

28.3.4.1. Anti-Aliasing:

The *Yafray* tab in *Rendering options* window lists the options for manual control over anti-aliasing. Press the *AutoAA* button to un-hide the manual settings.

Figure 28-22. The Yafray tab.



- **AA Passes:** Selects the number of passes to be used for Anti-Aliasing. The higher the passes, the better the results, longer render times also. While using passes of more than one, every pass samples the total number of samples chosen in *AA Samples* for every pass, thereby achieving better AA of pixels in noisier areas. If the results are noisier in a single pass you can increase the number of passes for better control and also a higher number of samples per pass. Single pass anti-aliasing with higher sampling, is not necessarily better than lesser samples with multipass.
- **AA Samples:** The number of samples per pass. The total AA sampling can be calculated by multiplying *AA Passes* with *AA Samples*. In our case for the the final blurred image for the DoF used below, we have used $4 \times 4 = 16$ anti-aliasing samples.
- **Psz:** The *Psz* is the *<Pixel filter size>* that is used during the aliasing calculations. It indicates the number of neighbouring pixels used in the calculations.
- **Threshold:** This sets the threshold or the maximum brightness difference with the neighbouring pixel, when this is above the chosen level, extra samples are taken until the result is below the threshold limit or maximum samples/passes are reached. Lower threshold means more pixels will be antialiased, so at 0 all pixels in the render will be anti-aliased, when 1 no anti-aliasing is done.

Raydepth: This effects the maximum number of *bounces* a reflected/refracted ray can make. This is only important for glass and mirrors, higher ray depth will improve the quality of the reflection/refraction in the glasses and or mirror.

28.3.5. The Depth of Field

YafRay supports the real camera-like focus effect, Depth of Field. The DoF setting can be accessed easily. First select camera in view port then press **F9**. You will see a tab along with the *Camera* tab named *YafRay DoF*. In the *Camera* tab enable *Show limits*, doing this will allow you to see a yellow cross on the *Show limits* line of the camera in Viewport. The yellow cross is what we need to put at or near the objects we want to have sharp focus on. That distance can be set by the *DoFDist* numeric button. The *Aperture* setting is the value you need to change to get the required DoF you want. Normally a setting in-between .100 to .500 will suffice for most scenes.

Figure 28-23. Notice the noise on the wooden boards.



How to remove the artifacts in DoF rendered scenes. DoF filter requires more samplings per scene to get the right amount of blurring. To do that you need to disable *Automatic AA* in *Yafray* tab in Render panel. Here try to adjust the *AA Passes* and *AA Sample* accordingly. If you have large aperture settings then you need to have multiple passes plus higher samplings per pass. Increasing anti-aliasing will also increase the render times.

Figure 28-24. Increasing the sampling manually, and aperture size solves the problem.



28.3.6. HDRI or High Dynamic Range Illumination

Yafray has HDRI support. To use this go to World buttons **F8**. Go to *Texture and Input* tab and add a new texture. Then go to texture selection **F6** select *Texture type* as image, press *Load image* button, locate the HDR file, select and press **Enter**. Blender will not show the HDRI image in the *World* tab, however it is loaded automatically during rendering. You can also increase or reduce the exposure of the HDRI from the *Texture brighteners* button in the *colour* tab in *texture* window (**F6**). The possible exposure settings are -1,0,1, for brightness' sliders at 0,1,2. This is because HDRI exposure can be modified in integer values only. For the final step, use any one of the GI methods. Generally Skydome will work fine.

Figure 28-25. HDRI render using Uffizi probe. It need more quality level.



28.3.7. Closing comments

I would like to thank eeshlo, to an large extent I was motivated to learn about Yafray because of him, and thus was able to write this tutorial, and also for his help in solving my queries. Thanks also goes to Dreamsgate for helping in the editing and support for this document.

This tutorial is not finished, as there are several aspects which I have not tried myself, like caustics, and GI techniques for certain types of renderings. So I call this tutorial version 1. Later improvements will not necessarily come at regular intervals. If you have any queries or have some point that I overlooked or misunderstood in the matter written above, I would be glad to hear from you. Also welcome are the suggestions for improvements. My mail-id

[Prev](#)

Parte 2

[Home](#)

[Up](#)

[Next](#)

Glosario para los geeks (Aún por traducir)

28.4. Glosario para los geeks (Aún por traducir)

Global Illumination: It is a method (algorithm) of computation for light calculation in the scene which takes into account the light bounces from the neighboring surfaces, along with the normal illumination of direct lights. In other words GI calculates the Indirect light also, thus it makes the renders more photo-realistic. Examples of GI methods are Radiosity and Ambient Occlusion in Blender and on a general scale Radiosity, Ray tracing and Caustics all use different GI algorithms.

Ray tracing: A method in CG which uses an algorithm to calculate the effect of lights on the surface of objects in the scene. In CG the ray-tracer works by calculating the light effect on the scene by *tracing* the light photons back to the point of origin, from the scene or the camera. It uses the reverse of what is in real life, the sun shoots photons and we receive them through reflection/refraction from the objects, the photon energy is also modified by the objects by absorption or adsorption to form a particular texture or colour of the object.

The reverse way of ray-tracing is done so as to reduce the amount of calculations, as it is faster to take just the photons or lights which reach the scene or our eye or the camera, than calculating everything what is outside the view.

Photon: The photon is also referred as Ray of light. And it is the smallest unit of light energy.

Caustics: The caustics are referred to as the refraction pattern formed by highly transparent objects such as a glass of fluids which have a certain degree of total internal reflection, for example the light falling on a glass filled with wine will form some strange patterns of different colours and intensity which are referred to as light caustic. The computer method for Caustic calculation is also referred to as photon mapping. The photon lamp in Blender/Yafray is for this purpose only.

HDRI: Or High Dynamic Range Illumination. This method is relatively new in GI. This uses the actual light probe value of a real scene in the real world taken through with special equipment to produce a 360° view of a scene, and stores the information of the light from all areas in the scene in a spherically mapped image called HDR. The renderer uses that information to shoot light from and provides even more photo-realistic rendering.

Anti-aliasing: Refers to a method to reduce the brightness levels between two neighboring pixels by overlapping the colours to average the level between neighboring pixels. This make the images appear smoother.

Photonmap: An assumed area or bound with-in which more density of photons are kept for calculations of indirect illumination.

[Prev](#)

Parte 3 (Aún por traducir)

[Home](#)

[Up](#)

[Next](#)

Glosario para los geeks (Aún por traducir)

Glosario para los geeks (Aún por traducir)

Global Illumination

It is a method (algorithm) of computation for light calculation in the scene which, takes in to account the light bounces from the neighboring surfaces, along with the normal illumination of direct lights. In Other words GI calculates the Indirect light also, thus it makes the renders more photo-realistic. Examples of GI methods are Radiosity and Ambient Occlusion in Blender and on a general scale Radiosity, Ray tracing and Caustics all use different GI algorithms.

Ray tracing

A method in CG which uses an algorithm to calculate the effect of lights on the surface of objects in the scene. In CG the ray-tracer works by calculating the light effect on the scene by *tracing* the light photons back to the point of origin, from the scene or the camera. It uses the reverse of what is in real life, the sun shoots photons and we receive them through reflection/refraction from the objects, the photon energy is also modified by the objects by absorption or adsorption to form a particular texture or colour of the object.

The reverse way of ray-tracing is done so as to reduce the amount of calculations, as it is faster to take just the photons or lights which reach the scene or our eye or the camera, than calculating everything what is outside the view.

Photon

The photon is also referred to as a Ray of Light, and it is the smallest unit of light energy.

Caustics

The caustics are referred to as the refraction pattern formed by highly transparent objects such as a glass of fluids which have a certain degree of Total internal reflection, for example the light falling on a glass filled with wine will form some strange patterns of different colours and intensity which are referred to as light caustic. The computer method for Caustic calculation is also referred to as photon mapping. The photon lamp in Blender/Yafray is for this purpose only.

HDRI

See: [High Dynamic Range Illumination](#)

High Dynamic Range Illumination

Or *HDRI*. This method is relatively new in GI. This uses the actual light probe value of a real scene in the real world taken through with special equipment to produce a 360 view of a scene, and stores the information of the light from all areas in the scene in a spherically mapped image called HDR. The renderer uses that information to shoot light from and provides even more photo-realistic rendering.

See Also: [HDRI](#).

Anti aliasing

It is refers to a method to reduce the brightness levels between two neighboring pixels by overlapping the colours in the difference level to neighboring pixels. This make the images appear smoother.

Photonmap

An assumed area or bound with-in which more density of photons are kept for calculations of indirect illumination.

[Prev](#)

Glosario para los geeks (Aún por traducir)

[Home](#)

[Up](#)

[Next](#)

De Blender a Yafaray usando YableX

Chapter 29. De Blender a YafRay usando YableX

Pertinente a Blender v2.31

por Manuel Bastioni

29.1. ¿Qué es Yable?

Yable es un script Python, diseñado originalmente por Andrea Carbone, que permite exportar la escena de Blender al formato XML de YafRay, para que pueda explotarse dicho motor para renderizados altamente fotorrealistas. Sin embargo, Yable no es un mero "conversor de formato", sino un verdadero laboratorio procesador de escenas que permite asignar y cambiar las luces, materiales y ajustes ambientales, aprovechando plenamente las capacidades de YafRay. Desde otro punto de vista podríamos considerar el script Yable como una GUI (interfaz gráfica), capaz de visualizar y manejar de forma simple la gran cantidad de parámetros utilizados por YafRay.

29.1.1. ¿Qué Yable?

Las primeras versiones de Yable (finales de 2002) han sido completamente realizadas por Andrea y, posteriormente, como resultado del éxito del script entre los usuarios, se han realizado muchas versiones diferentes, para corregir fallos y añadir nuevas funciones. Es importante mencionar la contribución de Alejandro Conty Estévez, Alfredo "Eeshlo" de Greef, Christoffer Green, Leope, Johnny "guitargeek" Matthews y Jean-Michel "jms" Soler.

Después de la publicación oficial de Yable 0.30, se han hecho muchos parches no-oficiales, generalmente llamados YableX, y publicados en el foro de Yable en www.Kino3d.com. Para el propósito de este capítulo hemos examinado todas esas versiones que aparecen como una nueva versión "oficial", y hemos decidido basarlo en la última publicación de YableX que, gracias a las modificaciones llevadas a cabo por Jms, funciona con la última versión de Blender.

29.1.2. ¿Dónde conseguir YableX?

Los dos sitios principales de Blender, que debería ya conocer, poseen enlaces a él; de todos modos la última versión ha sido realizada por Jms y puede descargarse desde su sitio Zoo-Blender (<http://www.zoo-blender.com>).

zoo-logique.org/3D.Blender). Recomendando, no obstante, que eche un vistazo al foro oficial de YafRay, en la sección de exportadores: <http://www.YafRay.org>

[Prev](#)

Glosario para los geeks (Aún por traducir)

[Home](#)

[Up](#)

[Next](#)

Instalando el script

29.2. Instalando el script

Pertinente a Blender v2.31

Yable es un script, es decir un simple fichero de texto que puede cargarse en Blender, por lo que no podemos hablar propiamente de una instalación. De hecho es suficiente con cargarlo en la ventana de texto de Blender y pulsar **ALT-P** para ejecutarlo. Antes de hecer esto, de todos modos, debemos prestar atención a dos puntos fundamentales:

- Es necesaria una instalación completa de Python (la adecuada a su versión de Blender), descargable en www.python.org.
- Debe editarse la línea 81 del script Yable.

Lo primero es una condición necesaria para que el script pueda hallar los módulos Python requeridos, mientras que lo segundo es necesario para configurar el directorio en el que se guardarán los ajustes y el XML generado por Yable. Modificando la línea 81 a:

```
YABLEROOT = "C:/"
```

Implica que cada vez que usted exporte una escena desde un archivo *foo.blend*, una nueva carpeta denominada *foo* será creada en *C:/* conteniendo todos los elementos de la escena. Así, incluso si usted cierra Blender, y posteriormente reabre el archivo y reinicia el script, Yable podrá encontrar los ajustes anteriores, puesto que los buscará en una carpeta con el mismo nombre que el archivo *.blend* actual.

Note Esta denominación automática es muy útil, aunque desafortunadamente implica también que si quiere guardar el archivo *.blend* con un nombre distinto, debe reexportarlo desde Yable como mínimo una vez, para que los ajustes sean reescritos de nuevo. Alternativamente puede copiar el contenido de la carpeta vieja en la nueva.

Si lo desea, puede configurar un visualizador externo que se inicie automáticamente para mostrar el resultado de la renderización: Si lo desea, puede editar la línea 90, asignando a la variable *VIEWERAPP* la ruta a la aplicación escogida.

Como ya se ha dicho, todos los datos se guardarán en el directorio definido como *YABLEROOT*; todos *excepto* las texturas. Es importante que todas las imágenes utilizadas para la escena se hallen en el mismo directorio, puesto que tal directorio debe ser adecuadamente indicado a Yable, aunque esto se

realiza en tiempo de ejecución en el botón de texto que aparece tan pronto como se lanza el script ([Figure 29-1](#)). Se requiere la ruta *completa* de la textura.

Figure 29-1. Parámetro Texture Dir y botón de inicio.



[Prev](#)

De Blender a Yafray usando
YableX

[Home](#)

[Up](#)

[Next](#)

La interfaz

29.3. La interfaz

Pertinente a Blender v2.31

Las funciones de Yable han sido divididas en tres pantallas principales, accesibles al presionar los tres botones superiores que se pueden ver en [Figure 29-2](#).

Figure 29-2. Botones de cabecera de Yable.



29.3.1. Filosofía del flujo de trabajo.

Una vez hemos creado una escena de Blender con objetos, materiales y luces, podemos cargar e iniciar el script, posiblemente dividiendo el panel de visualización principal en dos, y asignándole a una de las dos mitades una ventana de tipo Texto. Así podremos ver al mismo tiempo la escena y la GUI (interfaz gráfica) de Yable.

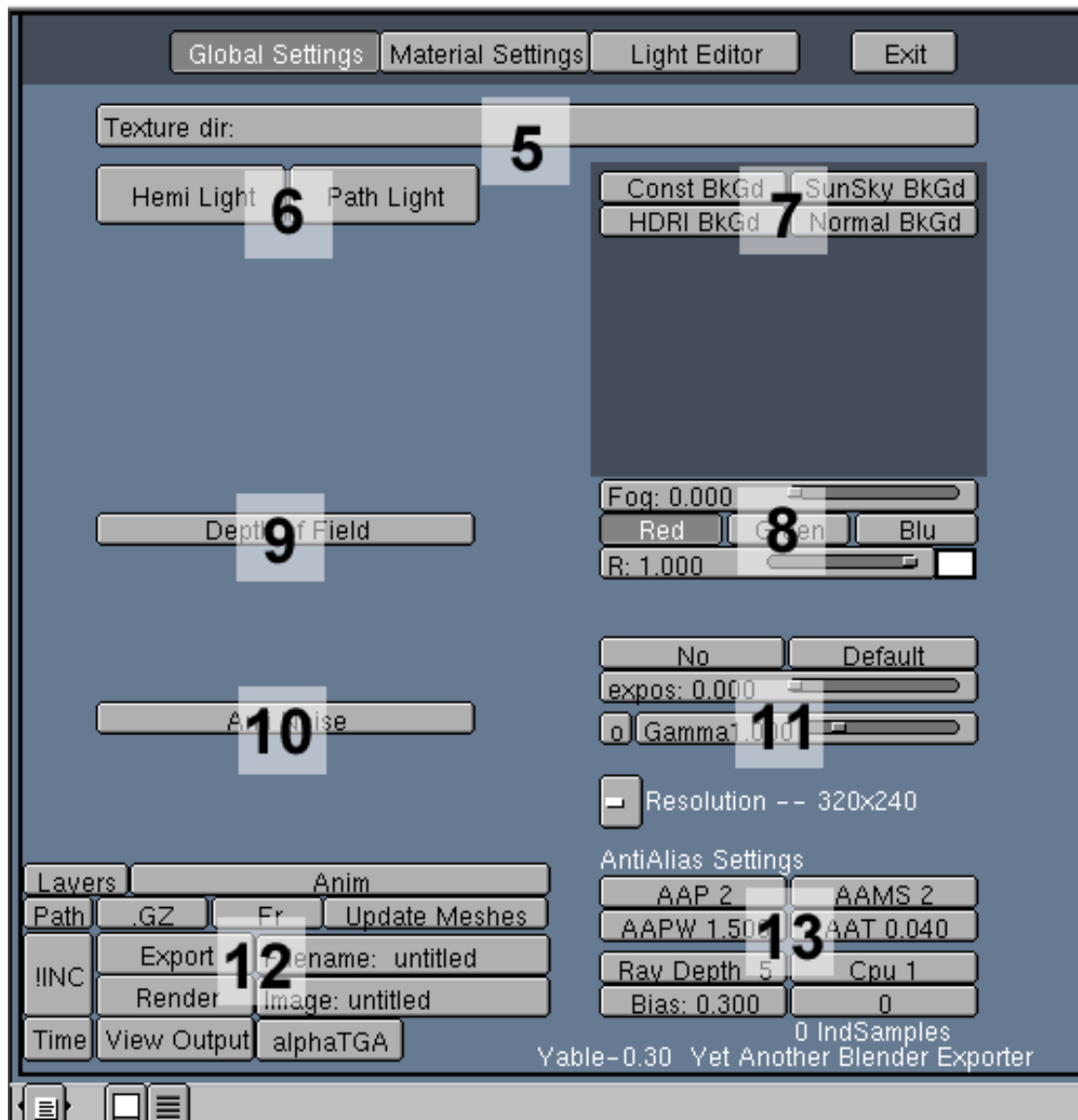
El flujo de trabajo de Yable es como sigue:

- Seleccione un objeto de la escena;
- Vaya a la parte Material o Light (luces) de la interfaz, dependiendo de qué es lo que esté definiendo, y pulse el botón *Get Selected*. Así Yable toma los parámetros del objeto (si hay alguno seleccionado);
- Edite los atributos. ¡Son completamente independientes de los de Blender!
- Pulse el botón *Assign* para asignar al objeto los parámetros que definió. ¡No olvide este paso! Un botón *Assign All* puede usarse para asignar los datos a todos los objetos seleccionados.

29.3.2. Parámetros Globales

Esta parte de la GUI nos permite acceder a las funciones de configuración de la escena general [Figure 29-3](#).

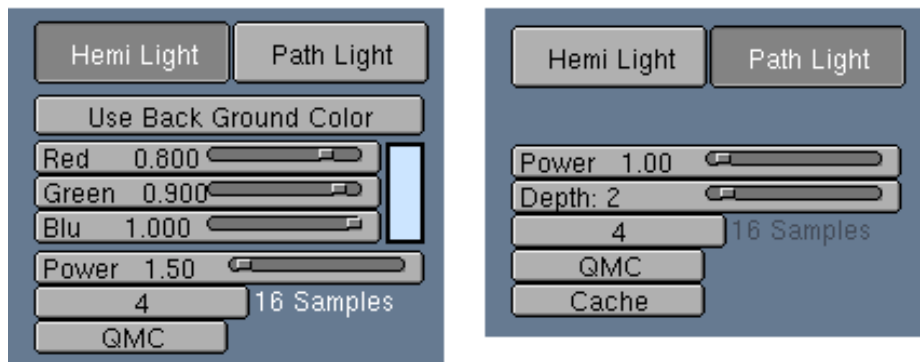
Figure 29-3. Botones de configuración headerGeneral de Yable.



- *Texture path* (Ruta de textura, [Figure 29-3#5](#)) - La ruta de la textura se puede redefinir en cualquier momento.
- *Global Illumination* (Iluminación Global, [Figure 29-3#6](#)) - Añade a la escena la iluminación global, que es la simulación de la luz difusa, originada en su naturaleza como un efecto de las infinitas reflexiones y refracciones mutuas entre objetos. Su efecto se añade al de las posibles luces directas.

Las luces Path y Hemi están diseñadas para producir el mismo efecto, pero usando algoritmos diferentes, con sus ventajas y desventajas, para la descripción de los cuales debemos referirnos al capítulo específico de YafRay.

Figure 29-4. Configuración de Iluminación Global de Yable.



[Figure 29-4](#) muestra las diferentes opciones para las luces Hemi o Path. En el primer caso podemos definir el color usando los botones numéricos *Red* (rojo), *Green* (verde) y *Blue* (azul), o tomar el color del fondo con el botón *Use Background color* (usar color del fondo). Es posible usar este último recurso cuando utilizamos fondos basados en imágenes (incluso si el botón *Use Background color* desaparece), colocando los tres deslizadores RGB a 0.

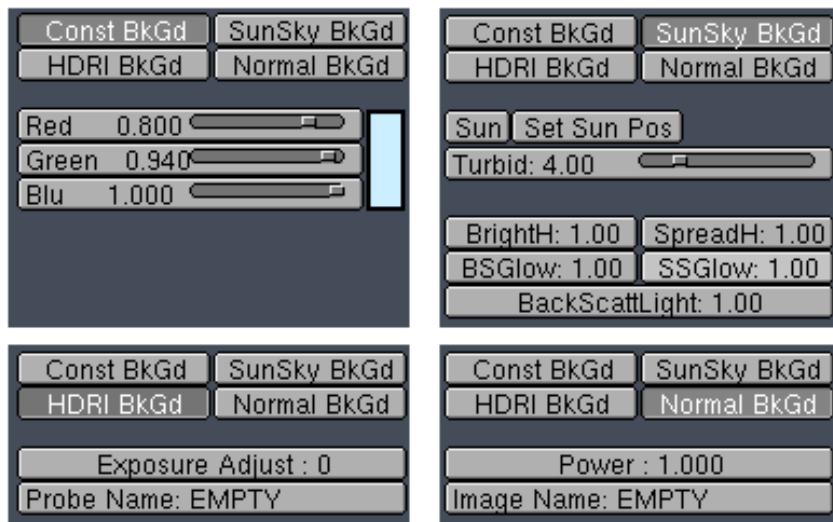
En el caso de la luz Path, por otro lado, la imagen de fondo se usa por defecto. Otra diferencia entre la luz Hemi y la luz Path es el parámetro *Depth*, referido al número de rebotes a ser considerados en el cálculo del intercambio de reflexiones entre los objetos. Para conseguir un mínimo efecto de radiosidad es necesario tener como mínimo dos intercambios de luz.

Puesto que el cómputo de una luz Path es bastante complejo, se incluye una opción de Cache, permitiendo optimizar y disminuir los tiempos de renderizado. Actúa básicamente como un preproceso usado para determinar las zonas de la imagen que necesitan más muestreados; como ejemplo, en una gran superficie plana podemos asumir que la luz difusa es bastante uniforme, y por lo tanto necesitará menos cálculos.

Los parámetros comunes para ambos tipos de luces son Power (fuerza), QMC, y Samples (muestras). Power indica la intensidad de la emisión luminosa, mientras Samples indica la precisión del muestreado durante el render: valores altos mejoran la claridad de la luz (la luz hemi tiene tendencia a presentar granulosidad) pero incrementan mucho los tiempos requeridos para el cálculo. QMC se refiere al uso del método Quasi MonteCarlo para la determinación de las zonas a computar: está basado en secuencias de números casi aleatorios, y acelera el renderizado, aunque algunas veces genera patrones granulosos en la imagen.

- *Background (Fondo, [Figure 29-3#7](#))* - Hay cuatro opciones. Dependiendo en la elección hecha, pueden aparecer algunos botones adicionales, casi todos ellos de comprensión inmediata ([Figure 29-5](#)).

Figure 29-5. Ajustes Background.



Const BkGd es el más fácil de usar ([Figure 29-5](#) arriba izquierda): es un color homogéneo definido por sus valores RGB.

Normal BkGd ([Figure 29-5](#) abajo derecha) nos permite usar una imagen (la última versión de YafRay soporta JPG y TGA); el único parámetro es *Power*, que indica el brillo de la imagen.

HDRI BkGd ([Figure 29-5](#) abajo izquierda) es quizá el que permite el máximo realismo. Las imágenes HDR (High Dynamic Range, Rango Dinámico Alto), almacenando colores de píxel como números de punto flotante, contienen muchos más datos que otros formatos. Además, normalmente están disponibles como *sondas*, esto es, fondos completos de 360° horizontales, y 180° verticales. Después de haber obtenido las imágenes apropiadas, es necesario ponerlas en la misma carpeta que las texturas, escribir el nombre en el botón *Probe Name* (nombre de la sonda), y ajustar la exposición que deseamos usar (positivo significa más brillante).

Finalmente *SunSky BkGd* ([Figure 29-5](#) arriba derecha) utiliza un sofisticado algoritmo para la simulación de las condiciones de la luz solar. La posición del sol puede ser definida seleccionando un objeto en la escena de Blender (normalmente vacío) y pulsando el botón *Set Sun Pos* y confirmando. Es importante notar que la dimensión del sol dependerá también de la distancia entre el objeto seleccionado y la cámara. Un parámetro particularmente importante para la construcción de la escena es *Turbid* (turbio), que permite regular el valor de la densidad de las capas atmosféricas que envuelven el planeta: capas densas permiten el paso únicamente a determinadas longitudes de onda de la luz solar, por lo que cambian tanto el color como la intensidad de la luz. Los otros botones controlan el halo y el alcance de los deslumbramientos. El fondo SunSky, si es usado con luz Path o con luz Hemi, es capaz de emitir luz en formas extremadamente realistas. En el caso en el que no queremos usar Iluminación Global, podemos presionar el botón Sun y Yable añadirá una luz de tipo Sun en la escena exportada, que simulará, en modo más impreciso pero más rápido, el efecto de la luz solar.

- *Fog* (Niebla, [Figure 29-3#8](#)) - Con el botón de desplazamiento Fog seleccionamos la cantidad de niebla presente en la escena (cero por defecto), mientras que el color es definido seleccionando uno de los tres botones *Red*, *Green* y *Blue*, y usando el único botón numérico bajo ellos ([Figure 29-6](#)).

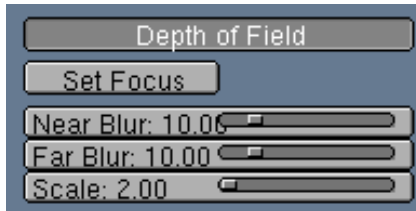
Figure 29-6. Ajustes de Fog.



- *Depth of field* (Profundidad de campo, [Figure 29-3#9](#)) - Necesario para imitar el foco de una cámara real. Esta característica la renderizará YafRay mucho más rápidamente que otros motores de render, ya que se procesa como una operación de postprocesado - sin embargo, esto significa que tiene algunas limitaciones en el cálculo preciso de reflexiones.

La regulación se realiza seleccionando cualquier objeto en la escena de Blender y pulsando el botón *Set Focus* (Ajustar foco, [Figure 29-7](#)). El punto escogido de este modo será perfectamente nítido. Con los otros dos botones numéricos podemos regular la amplitud de la profundidad de campo: *Near Blur* (desenfoque cercano) regula cuánto serán desenfocados los objetos que están entre la cámara y el punto de enfoque, mientras que *Far Blur* (desenfoque lejano) afecta a los objetos más alejados del foco, desde la cámara.

Figure 29-7. Ajusted de Depth of Field.



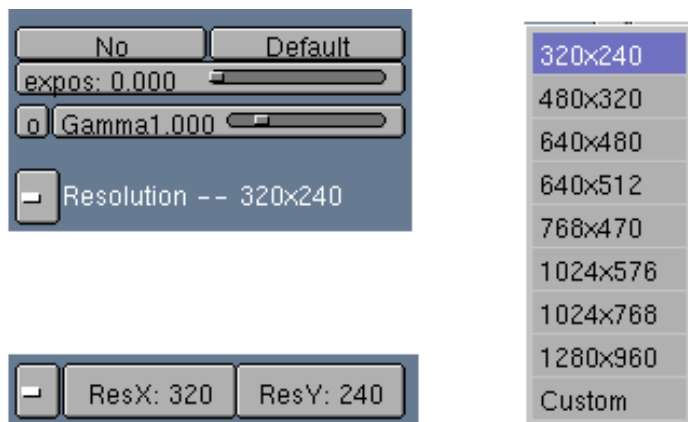
- *Anti-noise Filter* (Filtro anti ruido, [Figure 29-3#10](#)) - Este es también un filtro aplicado en postprocesado. Funciona de modo iterativo tomando algunos puntos dentro de un área circular y asignándoles el mismo color si sus colores difieren por encima de un umbral determinado.

Figure 29-8. Ajuste del filtro Anti Noise.



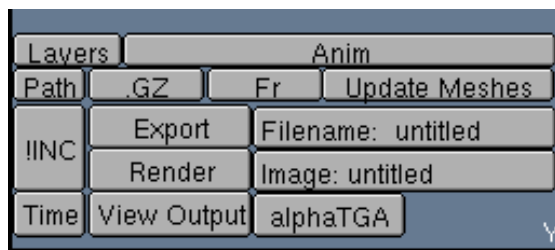
La amplitud del área circular está determinada por el parámetro *Radius* (radio, [Figure 29-8](#)), mientras que el umbral viene dado por *Max Delta*. Es un filtro muy útil pero debe usarse con precaución, porque conlleva también un efecto desenfocante que puede comprometer la calidad del resultado. Valores altos de delta tienden a unificarlo todo.

- *Gamma correction, exposure, resolution* (Corrección de gamma, exposición, resolución [Figure 29-3#11](#)) - Un simple grupo de botones que permiten ajustar el brillo y gamma del render completo. Los botones *No*, excluyen completamente ambos postprocesados. El botón *Default* (defecto) devuelve el botón numérico *expos* a su valor por defecto. El botón numérico *Gamma* permite la regulación de la gamma ([Figure 29-9](#)).

Figure 29-9. Ajustes de Resolution.

El botón de menú Resolution permite escoger las dimensiones de la imagen renderizada. Presionándolo, podemos escoger entre los formatos más comunes: 320x240, 480x320, 640x480, 640x512, 768x470, 1024x576, 1024x768 y 1280x960. Escogiendo la opción *Custom* (a medida), se visualizan dos nuevos botones, que permiten ajustar cualquier resolución.

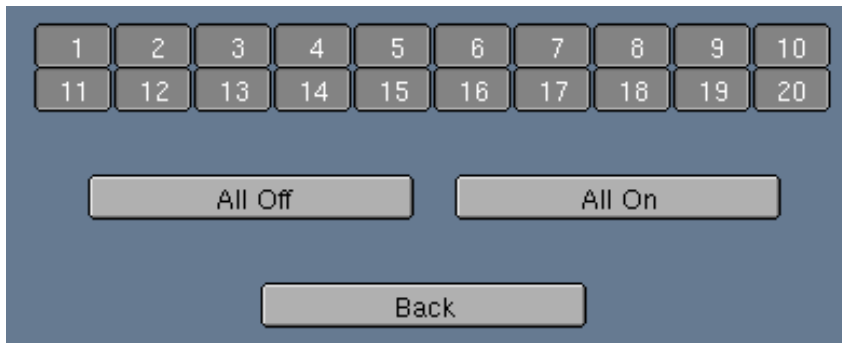
- *Ajustes Rendering* (renderizado, [Figure 29-3#12](#)) - Este grupo de botones permite ajustar detalles del fichero exportado y permite ejecutar YafRay directamente desde Blender ([Figure 29-10](#)).

Figure 29-10. Ajustes Rendering.

Las teclas fundamentales son *Export* (exportar), *Render*, *Filename* (nombre de archivo) y *Image* (imagen). Las dos últimas son necesarias para escoger el nombre que tendrá el archivo XML y la imagen renderizada. *Export* produce únicamente, dentro del directorio *YABLEROOT*, todos los archivos XML necesarios, mientras que si se usa junto con *Render*, ejecutará también YafRay y producirá la imagen final; finalmente, si el botón View Output está también seleccionado, al final del render Yable ejecutará también la aplicación especificada en *VIEWERAPP*, para visualizar el resultado. Nótese que debe tener YafRay en su Path para poder ejecutarlo desde Blender.

El botón Layers (capas) abre un nuevo panel para la selección de la capa (o capas) a exportar ([Figure 29-11](#)).

Figure 29-11. Selección de capas.



El botón *Path* fuerza la descripción de la escena a ser exportada utilizando archivos separados (un archivo principal, y subarchivos guardados en las subcarpetas adecuadas, subdividido por materiales y mallas). La ubicación de tales archivos será indicada a YafRay por el uso de una ruta completa.

Por el contrario, el botón *!INC* forzará a Yable a producir un archivo "monolítico". Si en el momento de la exportación ni *Path* ni *!INC* están seleccionados, Yable usará automáticamente la opción *Path*.

Tip Problemas de renderizado

Algunas veces, al utilizar archivos diferentes, Yable incurre en algunos problemas y puede mezclar los objetos creados en renderizados anteriores. En tal caso merece la pena usar el archivo único, que es seguramente sobrescrito cada vez, o borrar los XMLs antiguos.

El botón *Anim* fuerza a que un archivo XML *diferente* se exporte para *cada* cuadro de la escena de Blender. Una vez los cuadros son renderizados, pueden componerse en una animación usando el Editor de Secuencias de Blender (Sequence Editor). El botón *Anim* implica que el botón *Fr* (frame, cuadro) esté también presionado, el cual añade al nombre escogido para archivos de imágenes y XML un sufijo numérico que indica el cuadro renderizado. Todos los XMLs son guardados en un subdirectorio a parte denominado como el archivo de Blender, con el sufijo *_MOVIE*.

Tip Un ejemplo

Supongamos que nuestro *YABLEROOT* es *C:/bar/* y estamos trabajando con el archivo *robot.blend*; cuando presionamos *Export*, Yable creará, en primer lugar, una carpeta *C:/bar/robot/*; luego, si hemos especificado la opción *Path*, será creado dentro de este directorio el archivo XML principal (denominado *robot.xml*), así como dos carpetas: *Materials* y *Meshes*, de los cuales YafRay, leyendo las rutas (paths) en *robot.xml*, obtendrá los datos para los materiales y objetos.

En el caso que escojamos usar el botón *!INC*, no se crearán carpetas dentro de *C:/bar/robot/*, sino únicamente un solo archivo *robot.xml*, conteniendo todos los datos.

Finalmente, si el botón *Anim* ha sido usado, se creará otra carpeta, *C:/foo/robot/robot_MOVIE/* conteniendo tantos XMLs *robot.001.xml*, *robot.002.xml*, *robot.003.xml*,... como cuadros haya en la animación.

Tenga en cuenta que si los archivos de la animación son obtenidos con la opción *Path*, será necesario copiar las carpetas *Meshes* y *Materials* en la carpeta *robot_MOVIE*. Si la animación no incluye transformaciones de morphing (como ejemplo RVK), es seguro dejar el botón *Update Mesh*

(actualiza malla) desactivado. Si no, para cada cuadro exportado cada malla de la escena será exportada también.

Los tres últimos botones son *GZ*, *Time* (tiempo), y *alphaTGA*: el primero habilita la creación de archivos gzip, el segundo hace que el tiempo empleado en el render aparezca en la consola de YafRay y el tercero modifica el XML para que YafRay guarde imágenes TGA con el canal alfa.

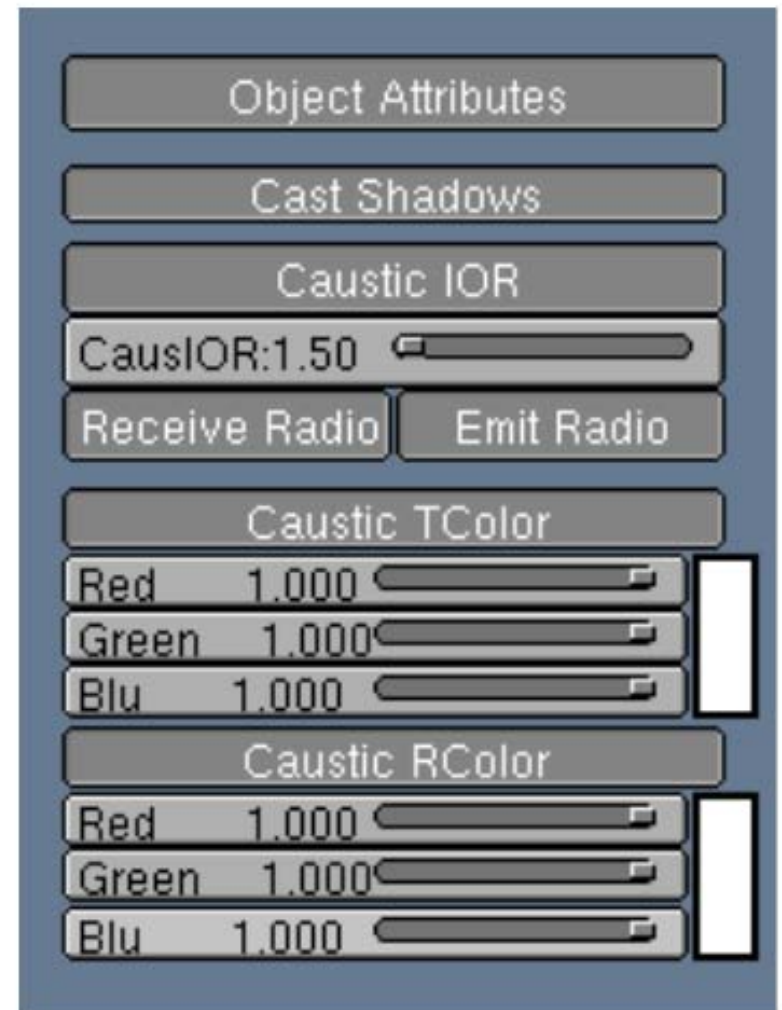
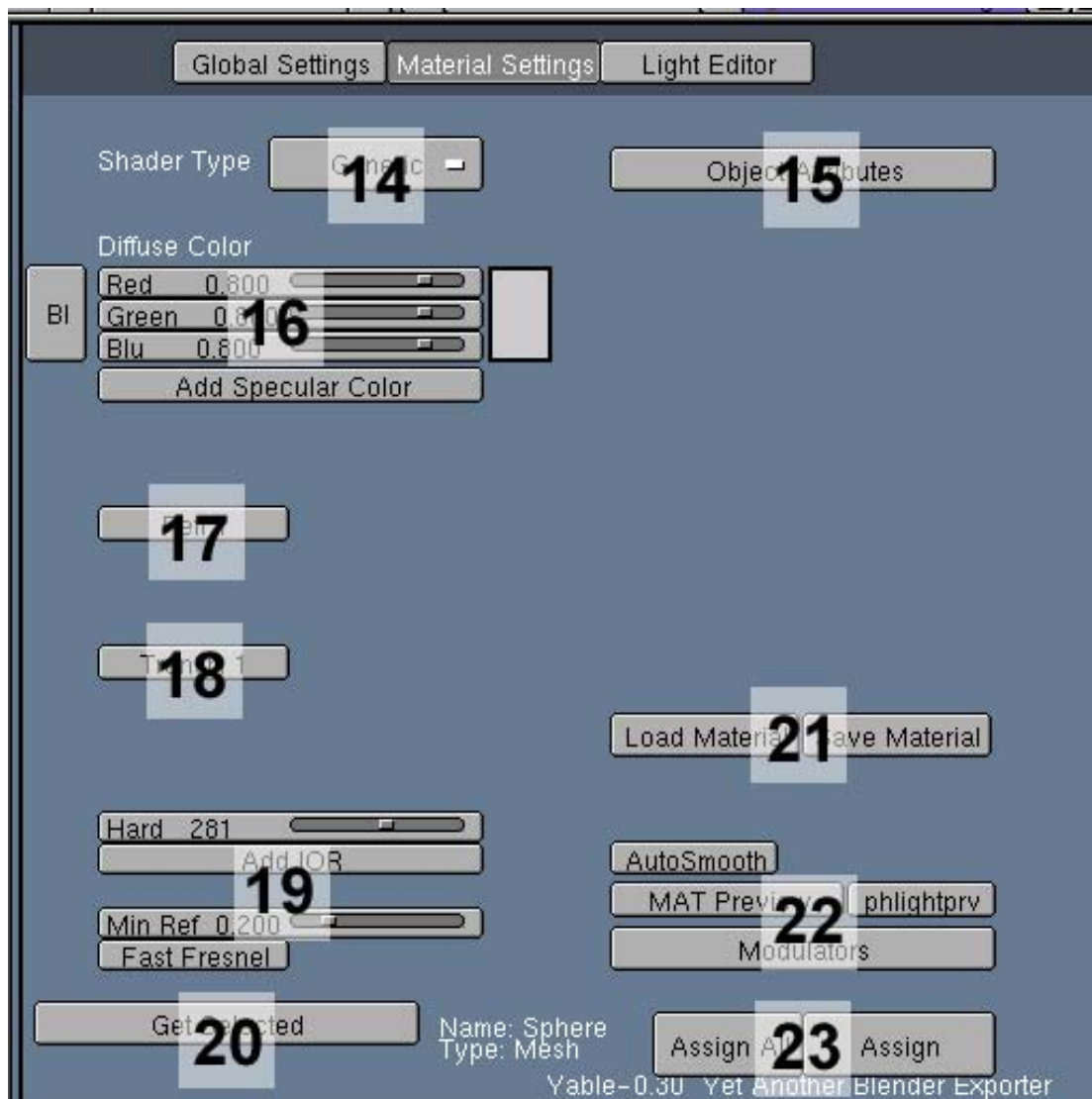
- *Ajustes de Antialiasing* ([Figure 29-3#13](#)) - El botón numérico *AAP* indica el número de pasadas de antialiasing; poniéndolo a cero indica antialias desactivado. *AAMS* regula el número de muestreos a utilizar para cada pasada AAS. *AAPW* regula el parámetro de anchura de píxel, o sea el solapamiento de píxeles; el rango varía entre 0 y 2, y usando valores altos, podemos obtener mejor suavidad, aunque algunas veces demasiado acentuada. *AAT* establece el valor de umbral más allá del cual el píxel será procesado con antialiasing: el valor puede variar entre 0 (todos los puntos serán procesados) y 1 (ningún píxel es procesado). *CPU* indica, en caso que haya múltiples CPUs presentes, cuántas deberían usarse para el render.

29.3.3. Ajustes de Material

El segundo panel GUI contiene los ajustes de Material. Aquí es posible asignar a cada objeto el material que se utilizará en el renderizado de YafRay. Los Materiales asignados con Yable y los materiales de Blender son dos cosas diferentes: el script toma de la escena únicamente ciertos valores, como las coordenadas UV y el color difuso. Este último únicamente bajo demanda del usuario. El resto es todo independiente; desde este punto de vista Yable es una especie de laboratorio: no sólo exporta la escena pasivamente, sino que nos permite estudiar y aplicar nuevos materiales.

Una vez un objeto es seleccionado, debemos presionar el botón *Get Selected*: nuevos botones aparecen, conteniendo parámetros de YafRay si el objeto los tiene ya, o vacíos si no ([Figure 29-12](#)).

Figure 29-12. Botones de Material.



- *Shader Type* (Tipo de sombreador, [Figure 29-12#14](#)) - Este botón permite escoger el tipo de material sombreador a aplicar.

El sombreador *Constant* (constante) es el más simple, caracterizado únicamente por los botones numéricos *Red* (rojo), *Green* (verde) y *Blue* (azul). *Crafter* es un caso distinto: es una interfaz utilizada para cargar el sombreador Crafter, que es un programa autónomo para la composición visual de materiales. *Generic* (genérico) es un material más versátil, e incluye también las características de otros materiales, por lo que lo describiremos en profundidad, usándolo como un paradigma para comprensión general.

- *Object Attributes* (atributos de objeto, [Figure 29-12#14](#)) - Presionando este botón aparecen ciertos botones nuevos ([Figure 29-12](#) a la derecha). Son características muy importantes, vinculadas al *Objeto* y no al *Material* en sí. *Cast Shadow* (arrojar sombras) conmuta si el objeto proyecta sombras o no. El botón *Caustic IOR* habilita el cálculo de las cáusticas para los haces de luz que pasan a través del objeto; éstos serán desviados de acuerdo al valor del

índice de refracción indicado en el botón numérico *CausIOR*. Valores altos de IOR producen cáusticas más finas (piense, como idea, en las lentes que concentran los haces de luz en un punto). Los botones *Receive Radio* y *Emit Radio*, si están presionados, forzarán al objeto a participar en el cálculo de la iluminación global, recibiendo y reemitiendo energía. Los botones numéricos *Caustic Tcolor* permiten especificar el color transmitido, es decir, el color que asume la luz que pasa a través del objeto. Los botones numéricos *Caustic Rcolor*, por otro lado, se refieren a la luz reflejada por el objeto.

Note Tenga en cuenta que incluso si configuramos el material correctamente, participará en los efectos de cáusticas y radiosidad sólo si es iluminado con luces adecuadas, como la luz Path, o la luz Photon.

- *Diffuse Colour* (color difuso, [Figure 29-12#16](#)) - Es el color básico, y corresponde al Diffuse Colour en los materiales de Blender. El botón Bl al lado se usa para tomar los valores RGB directamente del Material de Blender. Presionando el botón *Add Specular Color* (añadir color especular), aparecen nuevos botones, similares a los ya vistos, utilizados para definir el color especular. También en este caso el significado de este color es el mismo que en Blender.
- *Reflection and Transmission Colours* (colores de reflexión y de transmisión, [Figure 29-12#17](#) y [#18](#)) - Es posible ajustar los colores de Reflexión y de Transparencia del material. Tenga en cuenta que la transparencia también se ajusta mediante botones numéricos RGB para definir un color, y no simplemente a través de un valor "alfa".

El parámetro Transmit no decide el grado de transparencia, sino solamente qué color de la luz pasa a través del (o es bloqueado por el) material. Para ver un ejemplo, usando el color negro imponemos que ningún color pase a través del material; usando uno rojo, significaría que el objeto es transparente únicamente para la componente roja de la luz; usando el blanco permitimos pasar toda la luz. Presionando además los botones *Refl2* y *Transm2* que aparecen (y se crean nuevos botones numéricos RGB), podemos definir un comportamiento diferente del material con la incidencia de la luz que roza.

- *Hardness, Index of Refraction* (dureza, índice de refracción, [Figure 29-12#19](#)) - El parámetro *Hard* gobierna la nitidez de las luces resaltadas especulares exactamente como en Blender. El índice refractivo *IOR* es fundamental en objetos transparentes, y se utiliza para calcular la desviación de los haces de luz que cruzan el material. Como resultado de este efecto, los cuerpos inmersos en un medio transparente nos aparecen distorsionados (piense en un remo inmerso en agua clara). [Table 29-1](#) muestra algunos IOR (índices de refracción) de materiales comunes:

Table 29-1. IORs de ejemplo

Material	IOR
Vacío	1.0
Aire	1.00029
Hielo	1.31
Agua (a 20°C)	1.33
Alcohol etílico	1.36
Glicerina	1.473
Cristal	1.52
Zafiro	1.77
Diamante	2.417

- *Get Selected* (tomar seleccionados, [Figure 29-12#20](#)) - *Get Selected*, debe presionarse cada vez *después* de haber seleccionado el objeto y *antes* de empezar a modificar el material.

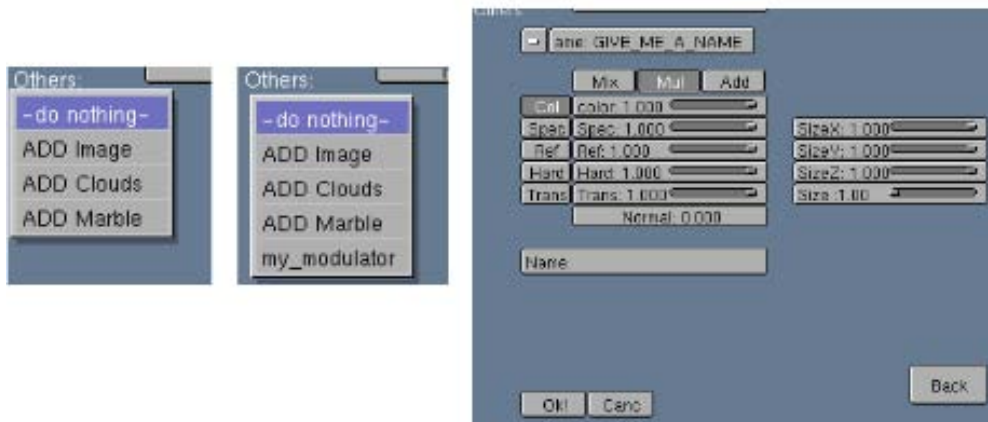
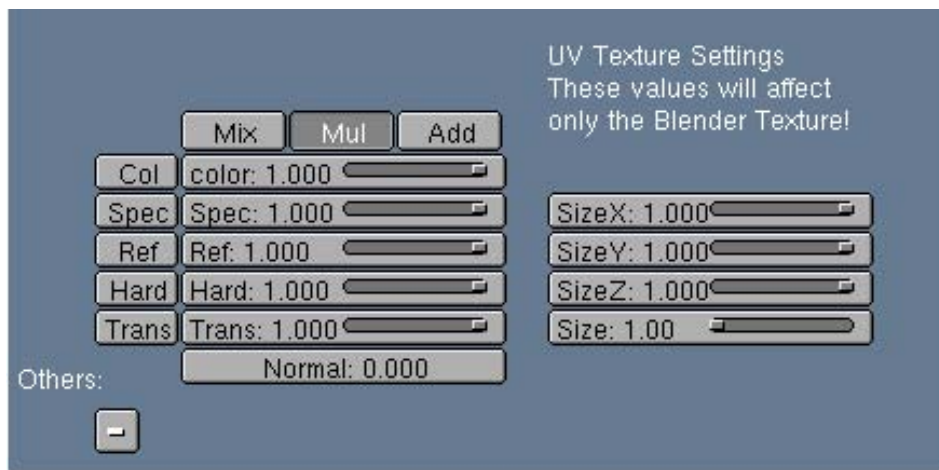
- *Get Selected, cargando y salvando los materiales* ([Figure 29-12#21](#)) - Los botones *Load Material* (cargar material) y *Save Material* (salvar material) permiten guardar las configuraciones de materiales y recuperarlas rápidamente. Puede darse un nombre a cada material.
- *Previsualización del material, autosuavizado y moduladores* ([Figure 29-12#22](#)) - El botón *AutoSmooth* se usa para regular la apariencia de las superficies. Presionándolo aparece un botón numérico adicional que regula el ángulo por debajo del cual el ángulo formado por las caras se considera suave, exactamente como en Blender.

El botón *Mat preview* crea una previsualización del material usando una escena de ejemplo. El TGA se guarda en el directorio actual (por ejemplo, en Windows, se guarda en la carpeta donde está ubicado el ejecutable de Blender). El botón *phlightprv* indica "Photon Light Preview" (previsualización de luz Photon) y se usa para poner luz fotónica durante la previsualización de materiales.

El botón *Modulators* (moduladores) permite acceder a un panel a parte para la composición de sombreadores avanzados formados por capas superpuestas. Cada capa puede ser una imagen o una textura de procedimiento. Obviamente es posible ajustar la modalidad con la que las capas deben ser mezcladas y también el porcentaje de transparencia.

El panel del principio se refiere a los moduladores por defecto, y se utiliza automáticamente si el objeto tiene coordenadas UV y es mapeado con una imagen en Blender. Desgraciadamente, debido a un fallo de programa, estas fórmulas no están mantenidas por Yable, que continúa usando los ajustes por defecto. Sin embargo todas las capas sucesivas de moduladores que son añadidas funcionan correctamente. Para añadir un nuevo componente es suficiente con pinchar el botón *Others*, y seleccionar el tipo de modulador que deseamos. En [Figure 29-13](#) vemos (empezando por arriba a la izquierda, en el sentido de las agujas de reloj) el panel principal, el panel con nuevos miembros añadidos, y el menú *Others*, antes y después la adición de un nuevo miembro:

Figure 29-13. Moduladores de Materiales.



Es posible añadir una capa *Image* (imagen), una capa *Clouds* (nubes) y una *Marble* (mármol). Lo primero es asignar un nombre al modulador creado; para ello será suficiente escribir algo significativo en lugar de *GIVE_ME_A_NAME*.

Si el modulador es una imagen es necesario insertar el nombre de la misma imagen: únicamente el nombre y la extensión, sin la ruta, ¡que ha sido definida una vez para todo anteriormente!

Los parámetros se refieren, por defecto, a mapeado bump, que es el efecto relieve que se dará al objeto: usando valores positivos levanta las zonas más claras y rebaja las más oscuras. Los distintos botones *Size* (tamaño) se refieren al escalado de las coordenadas UV (los tres ejes X, Y, Z pueden ser escalados independientemente, o todos juntos) mientras los distintos botones *Col*, *Spec*, *Ref*, *Hard* y *Trans* tienen los valores de costumbre e indican qué característica, y por qué cantidad, se aplica la modificación. Al final de los ajustes, presione el botón *Ok!*, para regresar al panel principal de materiales. Al final del procedimiento el nuevo modulador añadido estará en el Menú y puede ser seleccionado y cancelado en cualquier momento, usando el botón *Del* que aparece junto a los botones *Ok!*, *Canc* y *Back*.

Note El único tipo de mapeado soportado por ahora es el tipo UV. Requiere que el objeto posea esas coordenadas, para una descripción de las cuales, le remitimos a la sección adecuada de esta Guía. Si todo es ejecutado correctamente, Yable exportará automáticamente (sin añadir un modulador de tipo imagen) tanto las coordenadas UV como la imagen utilizada. En cuanto a esta imagen, debemos prestar atención a lo siguiente: Yable no utiliza la imagen cargada en la textura de los materiales de Blender, sino la que está cargada en la ventana Image (en la que se hacen los cálculos para posicionamiento UV). Obviamente, las imágenes deben estar todas en la carpeta usual, especificada al comienzo del script.

La inserción de los moduladores Clouds y Marble es similar a la del tipo Image, excepto que estos paneles tienen unos pocos parámetros específicos adicionales, autoaclaratorios, para estos dos tipos de textura de procedimiento.

- *Assign* (asignar, [Figure 29-12#23](#)) - El botón *Assign* finaliza el material y lo asigna al objeto. ¡No lo olvide! El botón *Selected All* permite asignar los ajustes a más de un objeto seleccionado.

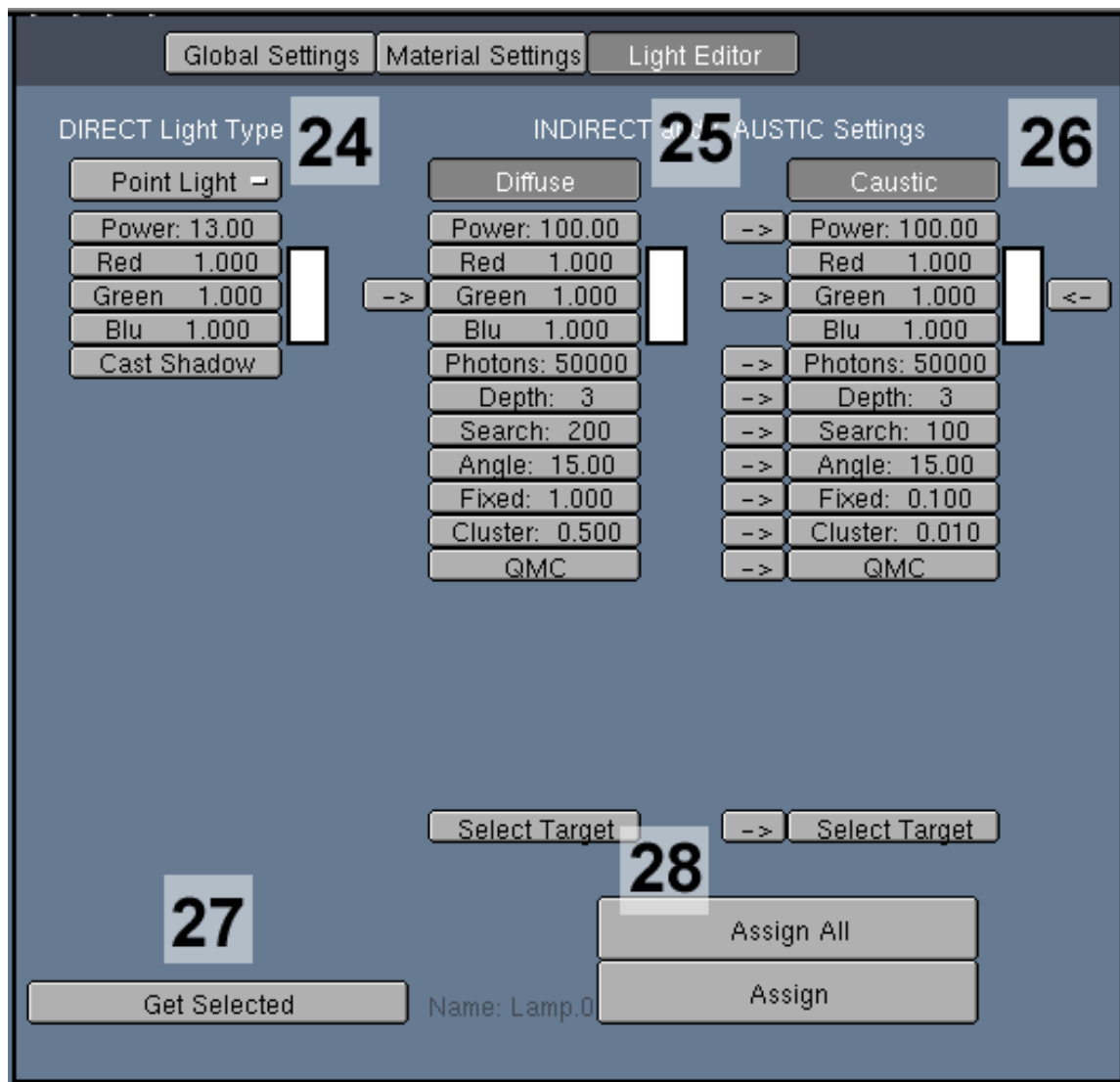
Note Salvar un material y asignar un material son dos acciones separadas. Si lo asigna, el objeto adquiere ese material, si lo salva, estará disponible más tarde.

29.3.4. Ajustes de luces

Los ajustes de luz en Yable se realizan con la misma modalidad de asignación que los materiales: se selecciona una luz en Blender, se presiona *Get Selected* y las características que queremos exportar a YafRay son ajustadas y asignadas definitivamente mediante *Assign*.

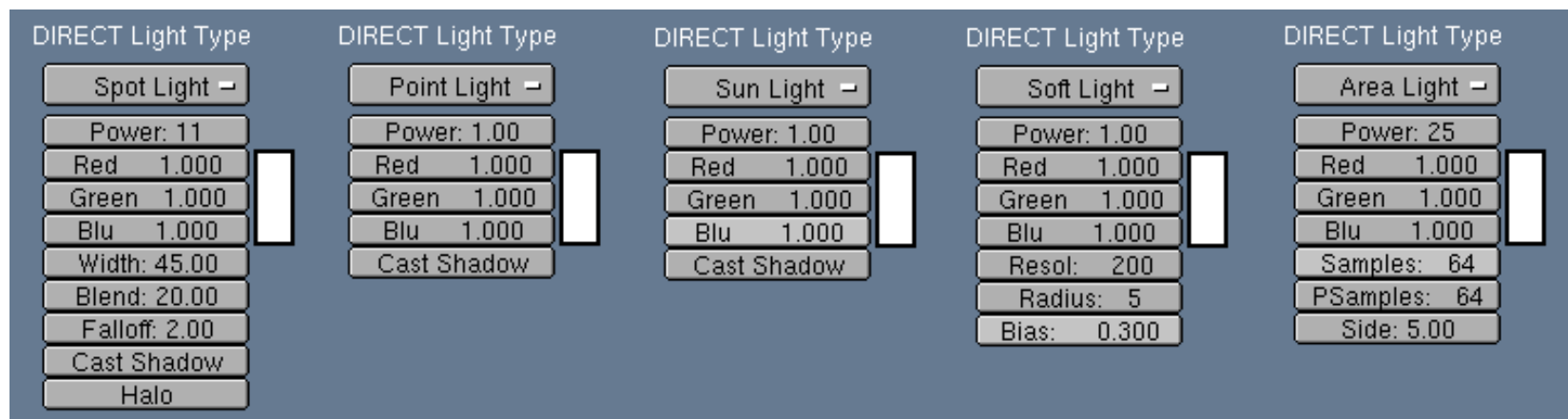
El tipo de luz usado en la escena de Blender *no* guarda ninguna relación con el tipo de luz que será exportada: los únicos parámetros que serán conservados con seguridad son las coordenadas posicionales de la luz; todo el resto, incluida la dirección donde apunta la luz, puede ser asignado independientemente con Yable. En [Figure 29-14](#) tenemos representada una luz Point con los botones Diffuse y Caustic activados, para tener un ejemplo que incluya la mayor parte de las opciones disponibles.

Figure 29-14. Panel GUI Light.



- *Tipos de luces* ([Figure 29-12#24](#)) - El menú nos permite escoger entre varios tipos de luz directa: *Point Light*, *Spot Light*, *Soft light*, *Area Light* y *Photon Light*. De acuerdo al tipo de luz, las opciones inmediatamente por debajo varían. [Figure 29-15](#) las muestra todas.

Figure 29-15. Opciones de Direct Light.



La luz Point es una fuente puntual que emite luz en todas direcciones. La fuerza de la luz es seleccionada con el botón numérico *Power*, mientras su color es seleccionado a través de los botones numéricos RGB y es posible definir si debe o no proyectar sombras con el botón *Cast Shadow*.

La luz Spot es muy similar a la spot de Blender: los parámetros *Blend* y *Falloff* tienen el mismo significado, mientras que *Width* representa la anchura angular del cono de luz.

Halo indica la presencia de halo (luz volumétrica); presionándolo añadimos algunos botones nuevos: tres botones numéricos para el color del halo, *Res* = resolución del mapa de sombras, *Density* = cantidad de niebla contenida en el halo, *Blur* = desenfoque aplicado al mapa de sombras, *Samples* = número de muestras usadas en el renderizado.

La dirección del Spot se define mediante un objetivo. Debemos presionar el botón *Select Target* (seleccionar objetivo), seleccionar un Objeto que actúe como objetivo de la luz spot en la escena de Blender, y finalmente presionar el botón *Confirm* (confirmar) para completar la operación.

Tanto la luz Point como la luz Spot decrecen siguiendo la ley física del inverso del cuadrado de la distancia.

La luz Soft es similar a la luz Point, con la diferencia que produce sombras suaves. Las sombras que aparecen demasiado "nítidas" son una de las desventajas de los motores de trazado de rayos, pero este tipo de luz, usando un mapa de sombras, resuelve el problema de forma segura. A parte de los parámetros usuales, el parámetro *Radius* (radio) define la anchura de la transición entre sombra y luz. El parámetro *Bias* controla la proximidad de la sombra al objeto que la produce, mientras el parámetro *Resol* controla la resolución del mapa de sombras: cuanto más alto es este parámetro, mejor es la precisión de la sombra.

La luz Sun simula las características de la luz solar; parece no decaer con la distancia (es una impresión debida a la enorme energía del Sol). Es, por lo tanto, una luz mucho más simple, en la cual podemos ajustar el color únicamente, y cuya intensidad permanece constante.

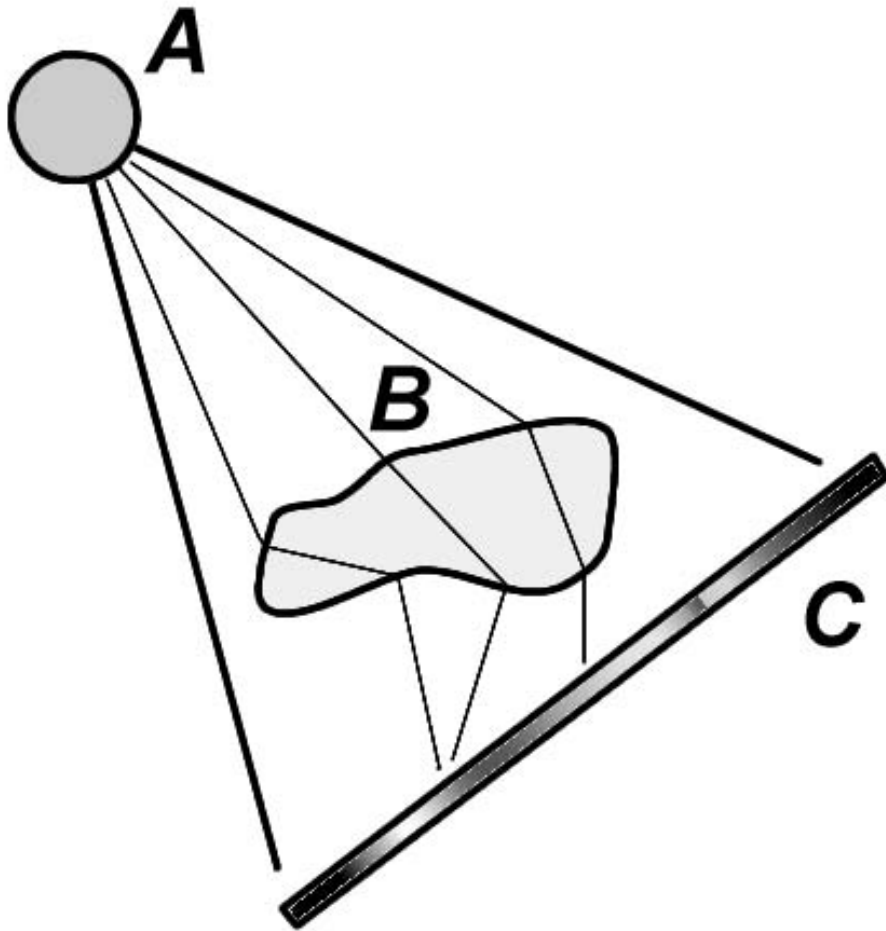
La luz Area es una fuente luminosa extendida. Mientras todas las que hemos visto hasta ahora emiten luz desde un punto (en realidad corresponde, como ejemplo, al pequeño filamento de una bombilla), la luz de área es emitida desde una superficie entera. YafRay admite también superficies rectangulares pero Yable está limitado a usar únicamente cuadrados. Los parámetros específicos son *Samples* (muestras), *PSamples* y *Side* (lado); son el número de muestras generales, el número de muestras en la zona de penumbra, y la longitud del lado del cuadrado emisor de luz.

- *Luz Photon (Diffuse y Caustic)* ([Figure 29-12](#)#25 and #26) - La luz Photon es un tipo de lámpara muy peculiar: se comporta de forma más realista ajustándose a la teoría de que la luz está compuesta por un haz de fotones. Estos fotones deben ser "disparados" literalmente hacia los objetos, para calcular su comportamiento cuando viajan a través de cuerpos transparentes o son difundidos por otros opacos. Es un cálculo muy complejo, por lo que no es siempre deseable que se ejecute sobre todos los elementos de la escena; por lo tanto podemos especificarlo en base a "material a material", y "objeto a objeto", a través de los parámetros Receive (recibir) y Emit (emitir).

Luz Diffuse (difusa): Habilitando este botón, una luz Photon de tipo Diffuse será añadida a la escena exportada, solapando la correspondiente luz Directa. Los fotones de esta luz tienen la capacidad de ser reflejados en superficies difusoras. De este modo los cálculos de "fuga de color" de radiosidad serán añadidos en el renderizado, para dar un efecto realista de Iluminación Global. Esto se puede hacer también mediante una luz Path, pero una luz Photon cuidadosamente ajustada es mucho más rápida.

Luz Caustic (cáustica): Las cáusticas son concentraciones de luz causadas por la refracción de los objetos transparentes. La luz Photon del tipo Caustic permiten modelar este fenómeno correctamente. Antes de proseguir, es necesario decir que si están realizadas sobre objetos con material inadecuado, esta luz no genera ningún efecto. De hecho, la escena, para tener cáusticas, debe contener una fuente de luz A, un objeto opaco C *recibiendo* las cáusticas, y un objeto transparente B situado entre estos dos *generando* las cáusticas ([Figure 29-16](#)).

Figure 29-16. Configuración de cáusticas.



- A debe ser una luz Photon de tipo Caustic;
- B debe tener un material capaz de producir Cáusticas (asignándole *como mínimo* Caustic IOR y Caustic Tcolor);
- C debe tener un material capaz de recibir los efectos de radiosidad (para ello el material por defecto podría ser bueno, porque este material exporta un sombreador simple que, sin especificar los valores de radiosidad recibidos y emitidos deja la opción a YafRay, que normalmente deja estos valores activados. Si en la práctica se usa un material genérico, no olvidemos activar el botón conmutador *Receive Radio*).

La luz Photon de tipo Caustic y de tipo Diffuse tienen los mismos parámetros. el botón de flecha se usa para imponer el mismo valor a más de una variable (por ejemplo el mismo color).

El botón *Photons* (fotones) define el número de fotones que serán disparados desde la lámpara; en escenas comunes unos pocos miles de fotones son suficientes, pero para obtener mejores resultados en términos de calidad, se recomienda seleccionar 50000 fotones o más. El botón *Depth* (profundidad) define el número de rebotes/transmisiones que los fotones pueden sufrir antes de que el trazado de rayos de YafRay los abandone, y es un dato

particularmente importante en el cálculo de la Iluminación Global. 3 es un buen valor para resultados aceptables. *Search* (búsqueda) indica el número de fotones que pueden usarse para iluminar un solo punto de la superficie del objeto; se usan valores altos para tomar en consideración también zonas que reciben pocos fotones, con un efecto sombreado de iluminación, mientras los valores bajos se usan para dar luz únicamente a los puntos realmente alcanzados por los fotones completamente, con bordes más definidos y "duros". *Angle* es el ángulo del cono de proyección con el que los fotones son disparados: se usan valores altos para cubrir un área importante, pero con desvanecimiento cuando nos alejamos del centro. *Fixed* es la abreviatura de *fixedradius* (radio fijo), y representa el radio en el que el número de fotones definido por *Search* debe caer para considerar el punto de la superficie como iluminado. *Cluster* es la porción de superficie iluminada más pequeña capaz de contener un fotón. Cuanto más alto sea este número más grande es la anchura del cluster y, consecuentemente, menos definido es el efecto de iluminación. *QMC* es, nuevamente, el método quasi-MonteCarlo.

[Prev](#)[Instalando el script](#)[Home](#)[Up](#)[Next](#)[Ejemplo de utilización de Yable](#)

29.4. Ejemplo de utilización de Yable

Pertinente a Blender v2.31

Figure 29-17. Escena de cáusticas.





Lo que se ve en ([Figure 29-17](#)) es una imagen realizada por Xavier 'richie' Ligey, modelada con Blender y renderizada con YafRay. La exportación ha sido hecha con Yable. No hay ninguna luz en la escena, solamente una luz Hemi con un fondo HDRI. Xavier ha sido tan amable de proporcionarnos los pantallazos de los ajustes que ha usado en Yable.

[Figure 29-18](#) muestra la pintura del coche, [Figure 29-19](#) muestra el material de cromo y [Figure 29-20](#) muestra el material de cristal.

Figure 29-18. Pintura del coche.

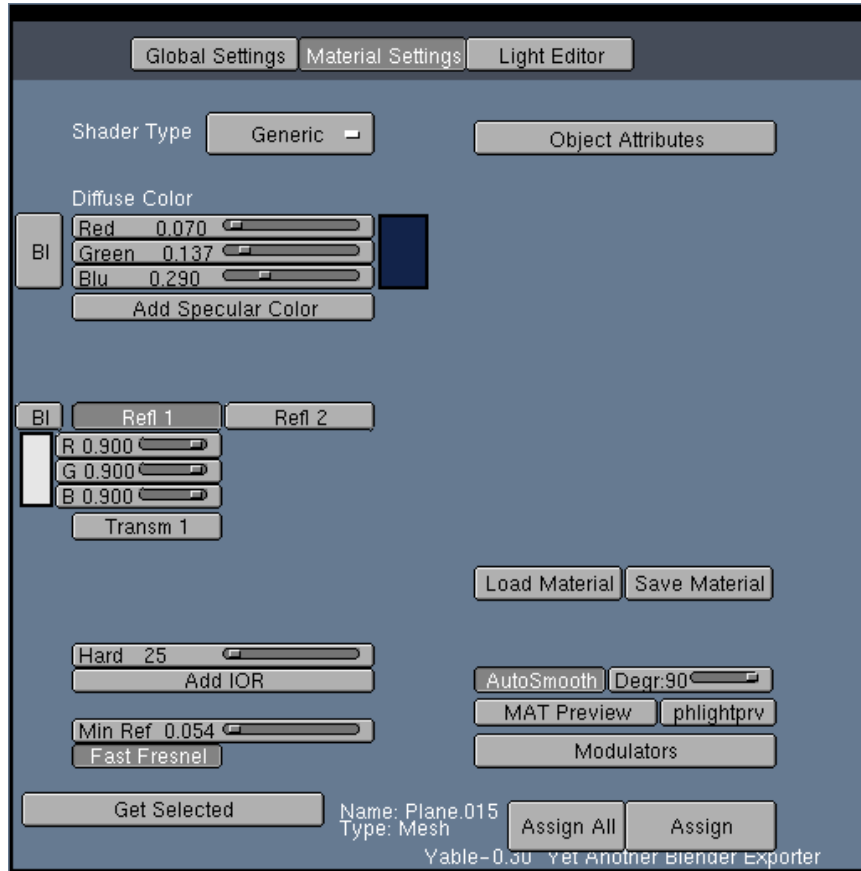


Figure 29-19. Elementos de cromo.

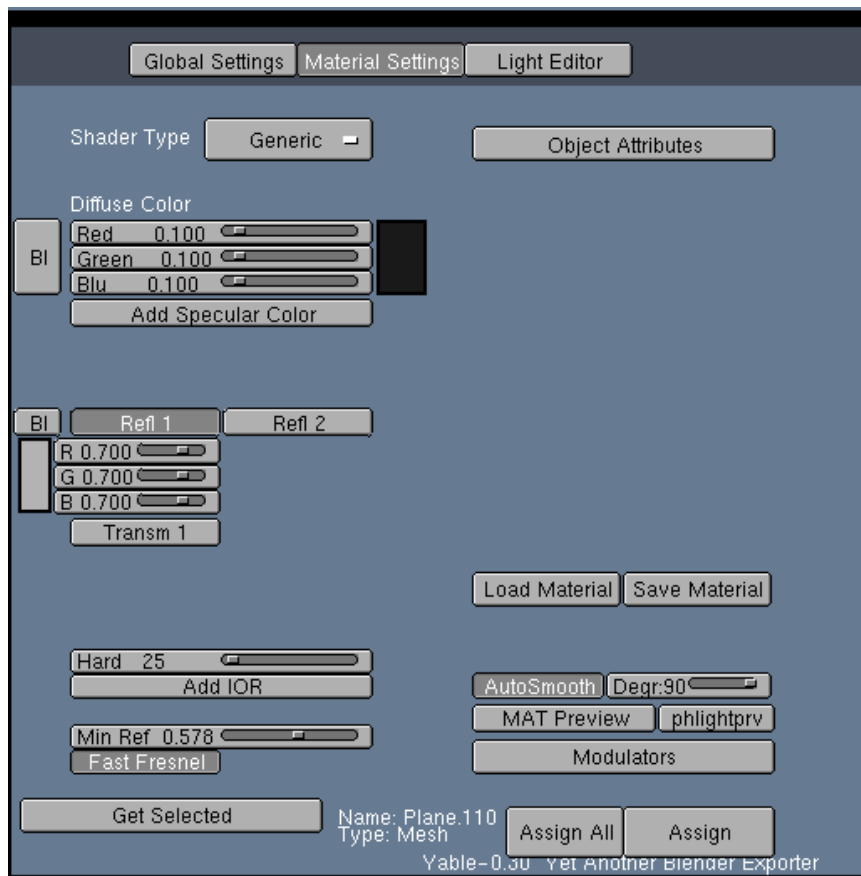
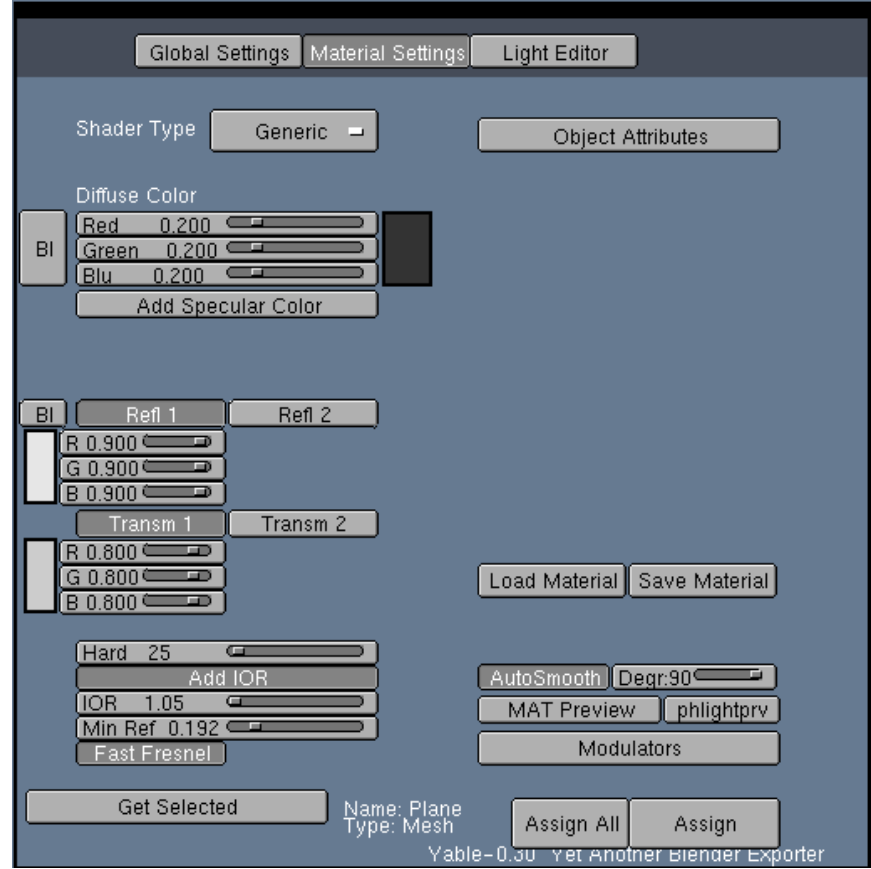


Figure 29-20. Parabrisas y otros cristales.



Un especial agradecimiento a Alessandro Braccili, quien me ayudó a comprender Yable/YafRay, y a escribir este capítulo.

Chapter 30. YafRay

por Alejandro Conty Estevez, Chris Williamson, Johnny Matthews.

30.1. Introducción

Relativo a Blender v2.31

por Alejandro Conty Estevez,

Cuando comencé a trabajar en YafRay, estaba probando algunos exportadores de blender para renderizadores externos como BMRT y Lightflow. Mientras escribía el código de sombreado y exportación, empecé a interesarme en cómo debería escribirse un trazador de rayos. Así que cuando la época de exámenes estaba a pleno, me aburrí (tan raro como pueda sonar) y empecé a escribir la estructura principal del programa. Después de hacer algunos renders de prueba, lo postergué por un año, hasta el verano siguiente. Luego escribí el cargador XML y Yafray, llamado "noname" para esa época, comenzó a ser un programa utilizable.

Alfredo se unió al desarrollo al mismo tiempo. Fue una gran ayuda. Un mes después, un montón de las cosas necesarias, como aceleración, se habían terminado y Alfredo portó mucho de su código a Yafray, como la famosa hemilight.

Luego Luis Fernando Ruiz, un amigo y compañero de clases, se unió para darnos un buen sitio web. Así que dijimos adiós al aburrido sitio web de puro texto. Tuvimos también la oportunidad de ver a Yafray renderizado en varias computadoras cuando Luciano Campal escribió su parche para que YafRay pudiera trabajar en forma distribuida gracias a Mosix. Fue muy excitante cuando tuvimos accesos a una sala con 20 computadoras para hacer pruebas. Cuando Andrea apareció con Yable, las cosas se empezaron a ver muy promisorias. Un script de exportación experimental para un renderizador experimental que terminó en un larguísimo hilo de imágenes impresionantes en Elysiun. Vimos las primeras imágenes buenas hechas con Blender y renderizadas en Yafray gracias a él.

No esperábamos ese éxito. Ni Alfredo ni yo. Por supuesto, era el alucinante script de exportación lo que estaba atrapando a la gente, exportar fácilmente desde Blender a un trazador de rayos. Nos excitamos mucho con todo el apoyo de la comunidad. Todavía me impresiona lo que la gente puede hacer con una herramienta tan simple como esta.

Ahora más gente se está involucrando y ayudando. Empezamos a tener una buena sección de documentación y recursos, la mayor parte escrita por Chris Williamson. Básicamente es todo lo que ve en este capítulo. Pero el no es el único. YafRay se está volviendo cada vez más fácil de usar desde Blender gracias a Johnny Matthews. Creo que él consume cada minuto de su tiempo escribiendo Extractor: un nuevo script de exportación para Blender. Facilita muchísimo la exportación, obteniendo los datos directamente desde Blender con prácticamente ninguna interacción con el usuario.

La potencia actual de Extractor y su rápido desarrollo indican que este podría ser el futuro esquema oficial de exportación desde Blender. De cualquier manera, se están realizando esfuerzos para escribir un exportador incluido en el mismo Blender. Alfredo contribuyó con un montón de código de compatibilidad de sombreadores y realizó algunos experimentos. Así que parece que en algún momento podremos comparar las alternativas en Python con la exportación incorporada.

YafRay comenzó como un experimento y aún lo es. No está terminado y carece de muchas características si lo comparamos con otros motores de render. Siempre pienso que no es lo suficientemente bueno y es difícil imaginar qué es lo que la gente ve en él. Como a la gente le gusta por alguna razón, queremos convertirlo ahora en un motor de renderizado completo que merezca ser llamado "renderizador". Llevará tiempo para divertirse programando. Queremos agregar todo aquello de lo que YafRay carece (partículas, efectos, etc) y mejorar la iluminación global. Pero actualmente solo Alfredo De Greef y yo estamos programando YafRay, por lo que para mantener el desarrollo en un ritmo aceptable, debemos conseguir más programadores, más desarrolladores. Espero que esto ocurra tarde o temprano.

Finalmente quisiera agradecer a la comunidad de Blender, que apoyó este proyecto. Todas esas bellas imágenes son lo que atrae a la gente a YafRay. Del mismo modo, gracias a toda la gente que aportó ideas para mejorar YafRay en los foros, y a Juan David G. Cobas por su valiosísima asistencia en matemática.

[Prev](#)[Ejemplo de utilización de YafRay](#)[Home](#)[Up](#)[Next](#)[Instalación](#)

30.2. Instalación

Relativo a Blender v2.31

YafRay está disponible para Linux, Windows y Mac OSX. Descargue el paquete apropiado para su SO de www.yafRay.org.

30.2.1. YafRay para Windows

Ejecute el programa instalador. Creará una carpeta llamada "yafRay" en su disco c:\ . Este diorectorio contiene el ejecutable yafRay.exe y el fichero de gramática que es utilizado por el cargador . El instalador copia también tres ficheros dll en el directorio system de Windows . Estas dll so necesarias para el soporte cygwin. Finalmente se copia un fichero de procesameinto por lotes (yafRay.bat) en la carpeta Windows (necesitamos este fichero en la variable PATH).

Ejecutar YafRay es sencillo. Simplemente abra una ventana MS-DOS, vaya al directorio de trabajo y escriba "yafRay fichero.xml" o "yafRay fichero.xml.gz". Por ejemplo, si deseara trabajar en *e:\raytracing\work* sobre un fichero XML ubicado en *C:\Docs\xmls* llamado *test.xml*, abra una ventana MS-DOS y:

```
c:\windows\>
c:\windows\> e:
e:> cd e:\raytracing\work
e:\raytracing\work> yafRay c:\Docs\xmls\test.xml
```

Se crearán uno o más ficheros targa, la salida del renderizado, en el directorio *e:\raytracing\work*.

30.2.2. YafRay para Mac OSX

Expanda el fichero tarball. (El expansor stuffIt también puede expandir ficheros tarball). Haga doble click sobre el paquete expandido para ejecytar el instalador. YafRay debe instalarse en el dispositivo Root (donde fue instalado MacOSX), *no puede elegir ningún otro disco*.

La ubicación de los ficheros instalados es: */usr/sbin/yafRay* y */usr/etc/gram.yafRay*

La utilización de YafRay no difiere apreciablemente entre los tres SO, por lo que puede consultar en la sección anterior.

YafRay solo consta de dos ficheros ' /usr/sbin/yafray ' y ' /usr/etc/gram.yafray' . Pero normalmente un usuario común no puede acceder a estos directorios a través de Mac OSX GUI, por lo que el OS X *Package Manager* (OSXPM) puede ayudarle a desinstalar los paquetes de su disco.

30.2.3. YafRay en Linux

Expanda el tarball.

```
tar xvzf yafray-#.#.#.tar.gz
```

Vaya al directorio recién creado y configurelo para su sistema.

```
./configure
```

Asegúrese de tener habilitado el soporte para zlib y jpeg. Si no fuera así, necesitará instalar los paquetes de desarrollo para libjpeg y libgz (verifique en su distribución).

¡ Compilelo !

```
make
```

Si la compilación falla, puede intentar

```
cd src
make yafray
```

El ejecutable es *yafray* y es un programa de línea de comando , cuyo uso es análogo a lo descrito en la sección "Windows".

30.3. Repaso del Lenguaje de Descripción de Escenas

Relativo a Blender v2.31

Una fichero de descripción de escena YafRay es un fichero XML que se atiene a las definiciones de esta sección. El renderizador analiza el XML de arriba hacia abajo. Por eso, si el Bloque1 está referenciada antes que Bloque2, debe ser definido antes que Bloque2 (debe estar líneas arriba en el XML).

```
<scene>
```

```
<shader type = "generic" name = "Default">
  <attributes>
    <color r="0.750000" g="0.750000" b="0.800000" />
    <specular r="0.000000" g="0.000000" b="0.000000" />
    <reflected r="0.000000" g="0.000000" b="0.000000" />
    <transmitted r="0.000000" g="0.000000" b="0.000000" />
  </attributes>
</shader>
```

```
<transform
m00 = "8.532125" m01 = "0.000000" m02 = "0.000000" m03 = "0.000000"
m10 = "0.000000" m11 = "8.532125" m12 = "0.000000" m13 = "0.000000"
m20 = "0.000000" m21 = "0.000000" m22 = "8.532125" m23 = "0.000000"
m30 = "0.000000" m31 = "0.000000" m32 = "0.000000" m33 = "1.000000"
>
```

```
<object name = "Plane" shader_name = "Default" >
  <attributes>
</attributes>
  <mesh>
    <include file = ".\Meshes\Plane.xml" />
  </mesh>
```

```
</object>
</transform>
```

```
<light type="pathlight" name="path" power= "1.000000" depth "2"
```

```

samples = "16" use_QMC = "on" cache="on" cache_size="0.008000"
angle_threshold="0.200000" shadow_threshold="0.200000" >
</light>

<camera name="Camera" resx="1024" resy="576" focal="1.015937" >
  <from x="0.323759" y="-7.701275" z="2.818493" />
  <to x="0.318982" y="-6.717273" z="2.640400" />
  <up x="0.323330" y="-7.523182" z="3.802506" />
</camera>

<filter type="dof" name="dof" focus = "7.97854234329" near_blur
"10.000000" far_blur "10.000000" scale "2.000000">
</filter>

<filter type="antinoise" name="Anti Noise" radius = "1.000000"
max_delta = "0.100000">
</filter>

<background type = "HDRI" name = "envhdri" exposure_adjust = "1">
  <filename value = "Filename.HDR" />
</background>

<render camera_name = "Camera" AA_passes = "2" AA_minsamples = "2"
AA_pixelwidth = "1.500000" AA_threshold = "0.040000"
  raydepth = "5" bias = "0.300000" indirect_samples = "1" gamma
= "1.000000" exposure = "0.000000" background_name"envhdri" >
  <outfile value="butterfly2.tga"/>
  <save_alpha value="on"/>
</render>
</scene>

```

No se asuste ! No es tan complejo como parece. Concéntrese en las etiquetas resaltadas en negrita.

Las etiquetas funcionan muy parecido a las etiquetas HTML (también como corchetes) cada etiqueta debe tener su correspondiente etiqueta de cierre. Dos etiquetas juntas, con parámetros encerrados, es un bloque. Un bloque puede decirle al renderizador cómo sombrear algo, el tamaño en que se renderizará una imagen, como debe verse la forma de un objeto, dónde se encuentra, etc etc.

En el ejemplo de arriba, primero se define un sombreador, luego un objeto (que está envuelto en su Matriz de Transformación), luego es añadida una luz, una cámara, un filtro, un fondo y finalmente los parámetros de renderizado (observe la etiqueta de cierre `</scene>`).

[Prev](#)

Instalación

[Home](#)

[Up](#)

[Next](#)

Sombreadores

30.4. Sombreadores

Pertinente a Blender v2.31

30.4.1. Sombreadores Base

Estos bloques de sombreadores determinan la Función de Reflectividad BiDireccional (BDRF) o el Modelo de Iluminación con el que el objeto está sombreado. Cada tipo diferente de sombreador base tiene varias entradas que pueden recibir las salidas de otros bloques de sombreadores, alterando las características de superficie.

30.4.1.1. Constant

Un sombreador uniformemente constante.

```
<shader type = "constant" name = "Sphere.mat">
  <attributes>
    <color r="15.000000" g="15.000000" b="15.000000" />
  </attributes>
</shader>
```

30.4.1.2. Generic

El sombreador más versátil.

```
<shader type = "generic" name = "Sphere.mat">
  <attributes>
    <color r="0.800000" g="0.800000" b="0.800000" />
    <specular r="1.000000" g="1.000000" b="1.000000" />
    <reflected r="0.000000" g="0.000000" b="0.000000" />
    <reflected2 r="1.000000" g="1.000000" b="1.000000" />
    <transmitted r="0.197183" g="0.197183" b="0.225352" />
    <transmitted2 r="1.000000" g="1.000000" b="1.000000" />
    <hard value = "25.000000"/>
  </attributes>
</shader>
```

```

        <IOR value = "1.592105"/>
        <min_refle value = "0.200000"/>
        <fast_fresnel value = "off"/>
    </attributes>
</shader>

```

30.4.1.3. Phong

Sombreador Phong clásico.

```

<shader type="phong" name="phongshader">
    <attributes>
        <environment value="fresnel"/>
        <color value="rgb"/>
    </attributes>
</shader>

```

30.4.2. Procedural

Estos bloques de sombreado crean varios patrones de procedimiento con valores paramétricos. No precisan entradas.

30.4.2.1. Marble

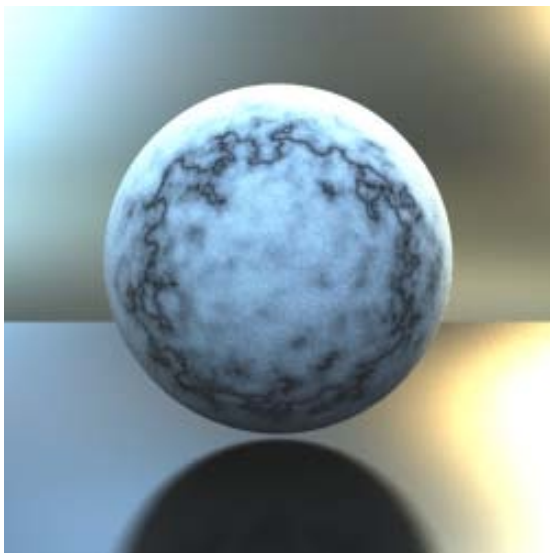
```

<shader type="marble" name="Marble" size="4.00" depth="4" hard="off" turbulence="5" sharpness="5.00">
    <attributes>
    </attributes>
</shader>

```

- *size*: Tamaño del efecto mármol, números más bajos = menos vetas, números más altos = más vetas.
- *depth*: Controla el número de iteraciones (número de frecuencias de ruido añadidas al remolino).
- *hard*: Controla el tipo de ruido; cuando se le asigna 'off', el ruido varía suavemente, mientras que asignándole 'on', mostrará cambios de color más bruscos.
- *turbulence*: Controla la cantidad de turbulencia del ruido.
- *sharpness*: Controla la finura del color 1 comparada con la del color 2; a mayor valor, más fina será la veta de color 1. Este efecto es similar a los

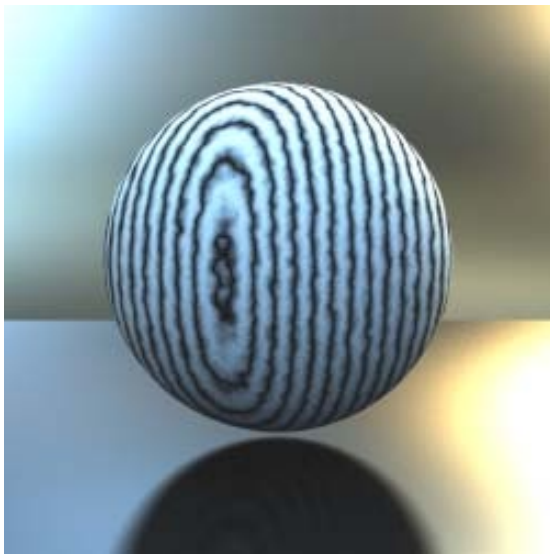
parámetros soft/sharp/sharper de la textura mármol de Blender; la diferencia es que aquí resulta más controlable. El valor debe ser 1 como mínimo.



30.4.2.2. Wood

```
<shader type="wood" name="Wood" size="5.00" depth="5" hard="off" turbulence="40">
  <attributes>
    <ringscale_x value="5" />
    <ringscale_y value="5" />
  </attributes>
</shader>
```

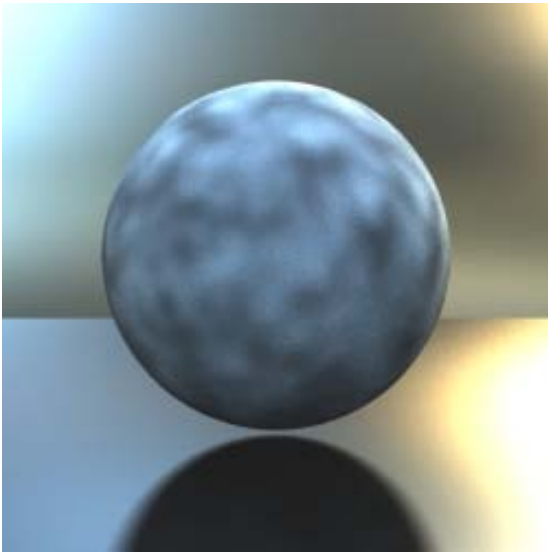
- *size*: Tamaño del efecto madera, números más bajos = menos vetas de madera, números más altos = más vetas.
- *depth*: Controla el número de iteraciones (número de frecuencias de ruido añadidas al remolino).
- *hard*: Controla el tipo de ruido; cuando se le asigna 'off', el ruido varía suavemente, mientras que asignándole 'on', mostrará cambios de color más bruscos.
- *turbulence*: Controla la cantidad de turbulencia del ruido.
- *ringscale_x*: Controla la anchura de los anillos de madera en el eje x.
- *ringscale_y*: Controla la anchura de los anillos de madera en el eje y.



30.4.2.3. Clouds

```
<shader type="clouds" name="Clouds" size="5.000" depth="3">  
  <attributes>  
  </attributes>  
</shader>
```

- *size*: Tamaño del efecto nubes.
- *depth*: Controla el número de iteraciones (número de frecuencias de ruido añadidas al remolino).



30.4.3. Meta Sombreadores

Permiten la modificación de otros sombreadores y la construcción de "cadenas" de sombreadores simples para elaborar un sombreador complejo.

30.4.3.1. Color2float

Recoge un color como entrada y devuelve un número en punto flotante (float) a la salida.

```
<shader type="color2float" name="c2f" input="input" >
  <attributes>
  </attributes>
</shader>
```

- *input*: Color de entrada a convertir a float.

30.4.3.2. Colorband

Genera un color desde un valor de entrada y un degradado. Un número ilimitado de moduladores añaden nodos al degradado. El sombreador interpola los valores del color de los nodos en el punto definido por el valor de entrada. En el ejemplo más abajo, un valor de entrada de 0.12 generaría un color entre el primer y segundo nodo, que son negro y naranja. Por lo tanto el resultado sería un naranja oscuro.

```
<shader type="colorband" name="Colorband" >
```

```

    <attributes>
      <input value="Wood" />
    </attributes>
    <modulator value="0.00"><color r="0.00" g="0.00" b="0.00" /></modulator>
    <modulator value="0.26"><color r="1.00" g="0.36" b="0.00" /></modulator>
    <modulator value="0.66"><color r="1.00" g="1.00" b="0.00" /></modulator>
    <modulator value="1.00"><color r="1.00" g="1.00" b="1.00" /></modulator>
  </shader>

```

30.4.3.3. Conetrace

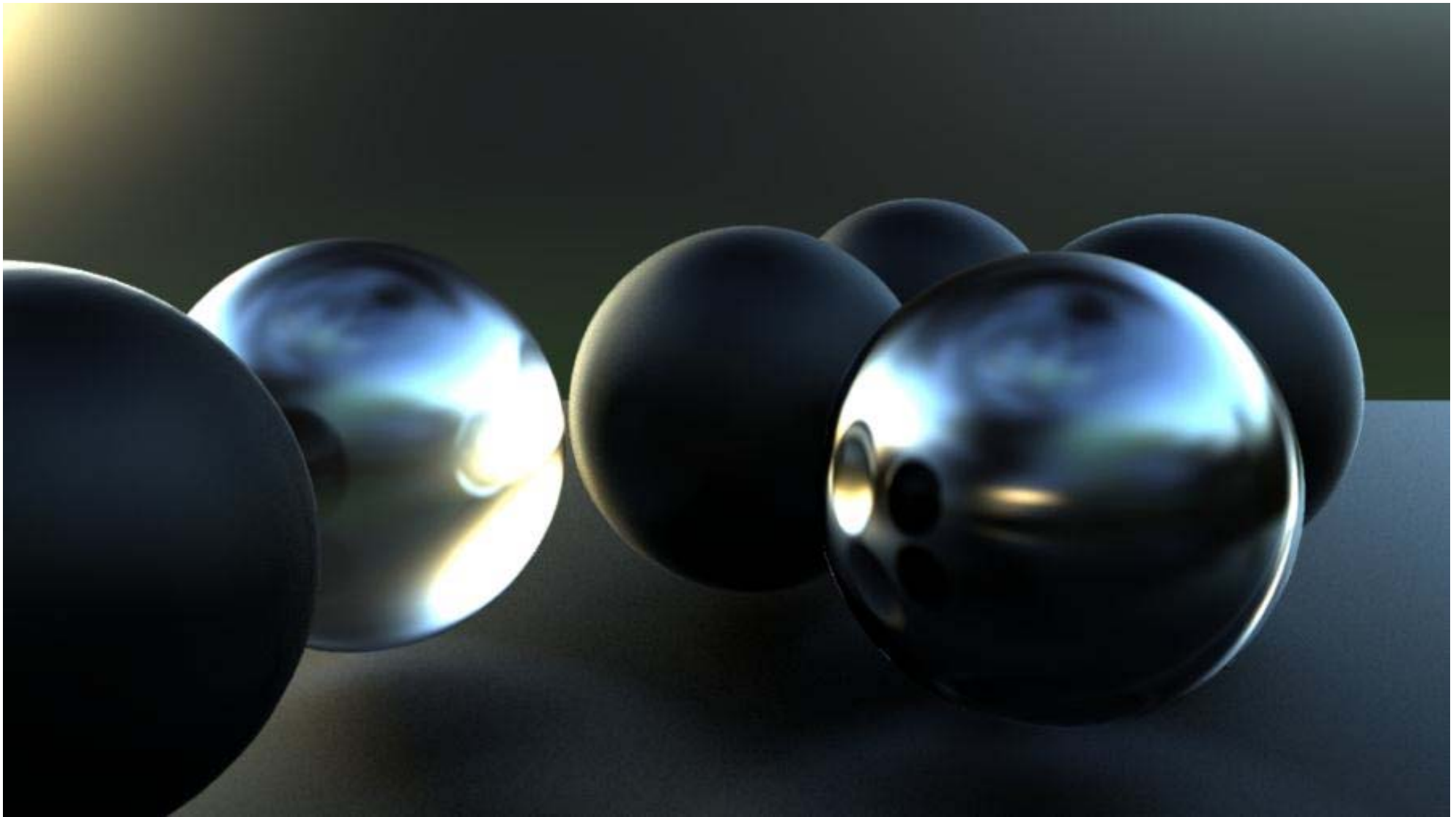
Puede usarse para conseguir reflejos o color transmitido por el entorno. Pero también se puede usar para conseguir efectos borrosos.

```

<shader type="conetrace" name="env1" reflect="on/off" angle="number"
  samples="number" IOR="number">
  <attributes>
    <color ... />
  </attributes>
</shader>

```

- *reflect*: 'on' reflejará los rayos, 'off' los refractará.
- *angle*: Ángulo del cono (alrededor del rayo) a ser muestreado; 0 para una reflexión/refracción nítida simple.
- *samples*: Número de muestras a tomar dentro del cono.
- *IOR*: Índice de refracción.
- *color*: Color para filtrar la luz incidente.



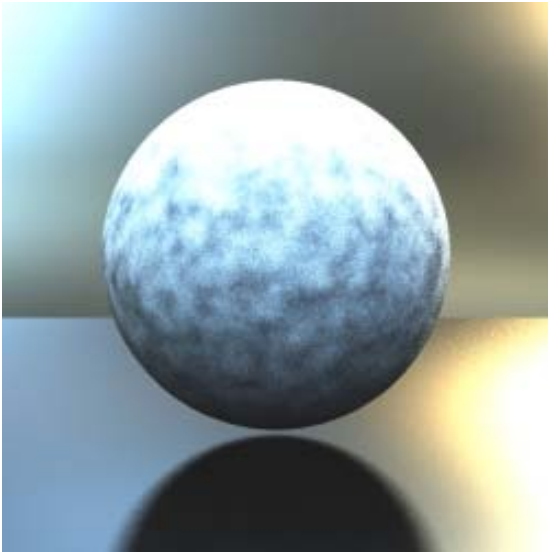
Esferas con varios niveles de reflexiones y refracciones borrosas y un fondo HDRI.

30.4.3.4. Coords

Genera un número en punto flotante (float) basado en las coordenadas del objeto.

```
<shader type="coords" name="PosY" coord="Y" >  
  <attributes>  
  </attributes>  
</shader>
```

- *coord*: Coordenada a utilizar: o bien X, o Y, o Z.



Sombreador Clouds y sombreador Coords (z) en un sombreador Multiply.

30.4.3.5. Float2color

Recoge un número en punto flotante (float) como entrada, y devuelve un color a la salida.

```
<shader type="float2color" name="f2c" input="input" >  
  <attributes>  
  </attributes>  
</shader>
```

- *input*: Float de entrada a convertir a color.

30.4.3.6. Fresnel

Puede usarse para conseguir reflexiones/refracciones realistas basadas en el ángulo de incidencia.


```
<shader type="fresnel" name="fresnel1" reflected="..." transmitted="..."
  IOR="number" min_refle="number">
  <attributes>
  </attributes>
</shader>
```

- *reflected*: Entrada a utilizar como color reflejado (normalmente una salida de Conetrace).
- *transmitted*: Entrada a utilizar como color transmitido (normalmente otra salida de Conetrace).
- *IOR*: Índice de refracción.
- *min_refle*: Cantidad mínima de reflexión.



Mezcla de bloques Fresnel y Conetrace.

30.4.3.7. HSV

Genera un color desde, o bien una entrada, o desde valores paramétricos para componentes HSV.

```
<shader type="HSV" inpuhue="..." inputsaturation="..." inputvalue="..."
  hue="number" saturation="number" value="number" >
</shader>
```

Como en el sombreador RGB, si se omiten las entradas, se usan los valores paramétricos de tono/saturación/valor (hue/saturation/value).

30.4.3.8. Image

Asigna una imagen de mapa de bits a un objeto según sus coordenadas UV (genera Color a la salida).

```
<shader type = "image" name = "bitmap">
<attributes>
  <filename value = "c:\filename.tga" />
</attributes>
</shader>
```

- *filename*: Ruta y nombre del mapa de bits a aplicar.

30.4.3.9. Mix

Mezcla 2 entradas de diferentes maneras, dependiendo del modo utilizado.

```
<shader type="mix" name="mixMode" input1="Colorband0" input2="Colorband" mode="add">
  <attributes>
  </attributes>
</shader>
```

- *input1*: Primera entrada a mezclar.
- *input2*: Segunda entrada a mezclar.
- *mode*: Posibles modos de mezcla (nota: algunos modos generan resultados diferentes en la salida dependiendo del orden de las entradas).

Son: Add, Average, Color Burn, Color Dodge, Darken, Difference, Exclusion, Freeze, Hard Light, Lighten, Multiply, Negation, Overlay, Reflect, Screen, Soft light, Stamp, Subtractive.

30.4.3.10. Multiply

Multiplica (números en punto flotante, float) valores de entrada, o bien valor de entrada y valor constante, genera un float.

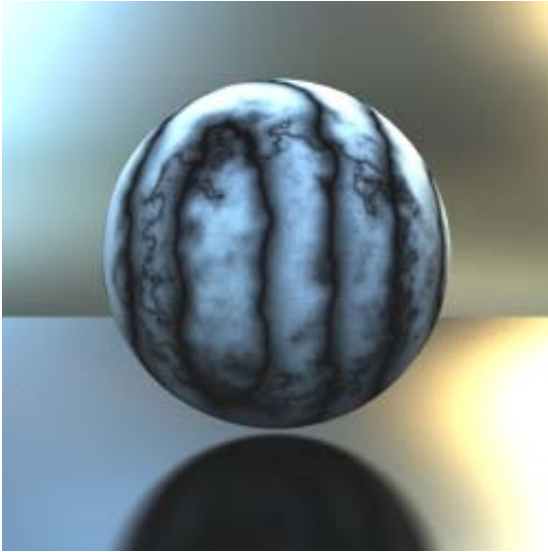
```
<shader type="mul" name="Multiply" input1="input" input2="null" value="5.30">
```

```

    <attributes>
    </attributes>
</shader>

```

- *input1*: Primera entrada a multiplicar.
- *input2*: Segunda entrada a multiplicar (si es nulo, input1 se multiplica con el valor del parámetro 'value').
- *value*: Valor a multiplicar si input2 es nulo.



Sombreador Multiply con Wood y Marble como entradas.

30.4.3.11. RGB

Genera un color desde, o bien una entrada, o desde valores paramétricos para componentes RGB.

```

<shader type="RGB" inputred="..." inputgreen="..." inputblue="..." >
  <color ...>
</shader>

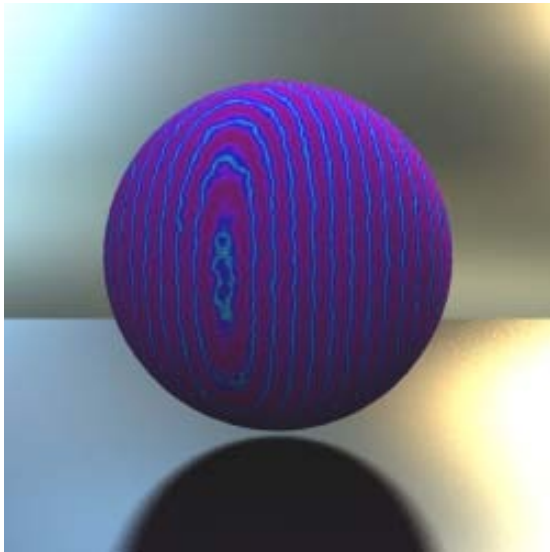
```

Si una de las entradas se omite, entonces para esa entrada se usa el color por defecto definido en la etiqueta 'color'.

30.4.3.12. Sin

Genera valores en punto flotante (float) basados en onda senoidal y entrada.

```
<shader type="sin" name="Sin" input="input" >  
  <attributes>  
  </attributes>  
</shader>
```



El sombreador Wood es entrada para Sin, el cual a su vez es entrada para el canal 'hue' de un sombreador HSV - el valor del color de un sombreador Phong.

[Prev](#)[Repaso del Lenguaje de Descripción de Escenas](#)[Home](#)[Up](#)[Next](#)[Objetos renderizables](#)

30.5. Objetos renderizables

Relativo a Blender v2.31

Actualmente, YafRay soporta únicamente objetos de malla. Un ejemplo de un plano simple triangular es:

```
<transform
m00 = "0.997525" m01 = "0.070303" m02 = "0.001329" m03 = "0.115816"
m10 = "-0.018745" m11 = "0.284097" m12 = "-0.958612" m13 = "1.522439"
m20 = "-0.067771" m21 = "0.956215" m22 = "0.284711" m23 = "3.272361"
m30 = "0.000000" m31 = "0.000000" m32 = "0.000000" m33 = "1.000000"
>

<object name = "Plane.002" shader_name = "Plane.002.mat" caus_IOR =
"1.500000" recv_rad = "on" emit_rad "on"
    shadow = "on">
    <attributes>
        <caus_tcolor r = "1.000000" g = "1.000000" b =
"1.000000"/>
        <caus_rcolor r = "1.000000" g = "1.000000" b =
"1.000000" />
    </attributes>
    <mesh      autosmooth = "30.0" >
<points>
    <p x="4.403727" y="-4.403728" z="0.000000" />
    <p x="-4.403727" y="-4.403727" z="0.000000" />
    <p x="-4.403725" y="4.403728" z="0.000000" />
</points>

<faces>
    <f a="0" b="2" c="1" />
    </faces>
</mesh>
</object>
</transform>
```

30.5.1. Etiqueta Transform

La etiqueta Transform define la matriz de transformación para el objeto encerrado (posición, escala y rotación desde el punto de origen del mundo).

30.5.2. Etiqueta Object

La etiqueta Object define la geometría del objeto, y tiene una serie de parámetros:

- *caus_IOR*: Índice de refracción para fotones caústicos.
- *recv_rad*: Recibir o no radiación (luz rebotada generada desde lámparas fotónicas) ('on' u 'off').
- *emit_rad*: Emitir o no radiación ('on' u 'off').
- *shadow*: Proyectar o no una sombra ('on' u 'off').
- *caus_tcolor*: Color para teñir los fotones transmitidos (colores de los caústicos transmitidos).
- *caus_rcolor*: Color para teñir los fotones reflejados (colores de los caústicos reflejados).

30.5.3. La etiqueta Mesh:

La clave en la definición de un objeto, es la etiqueta Mesh, la cuál define su geometría.

- *autosmooth*: Ángulo umbral para el algoritmo de suavizado (smoothing) (omitir para los objetos encarados).

Dentro de la etiqueta Mesh un bloque <points> define los vértices de una malla, mientras que el bloque <faces> define caras triangulares mediante los índices de sus tres vértices.

[Prev](#)[Sombreadores](#)[Home](#)[Up](#)[Next](#)[Luces](#)

30.6. Luces

Pertinente a Blender v2.31

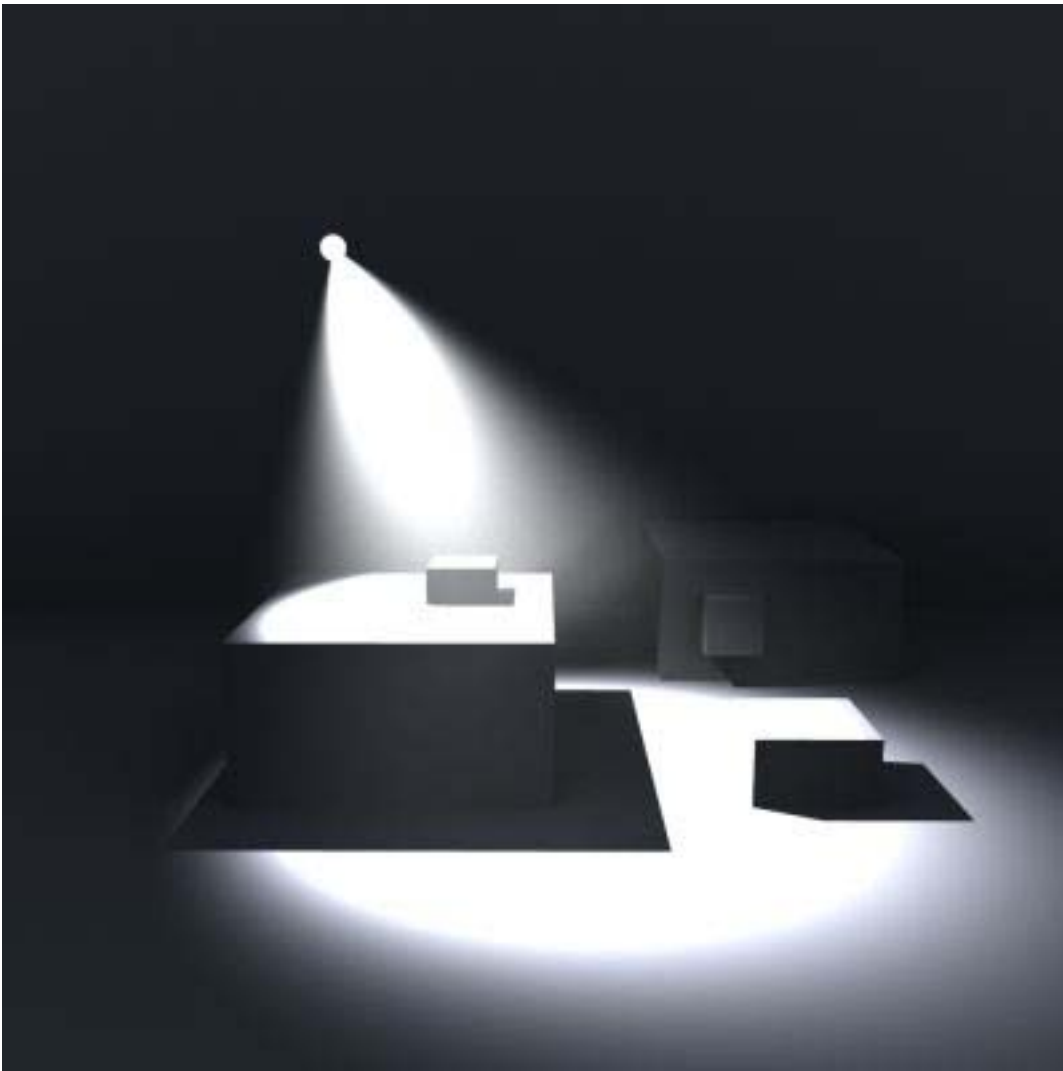
YafRay proporciona varios tipos de luces:

30.6.1. Luz Spot

Es bastante similar a la luz Spot de Blender.

```
<light type="spotlight" name="spot" power="30.0" size="80" blend="10"
beam_falloff="2" halo="on" res="512" blur="0.0" fog_density="0.20">
  <from x="-0.140436" y="4.175604" z="8.336139" />
  <to x="-0.140436" y="4.175604" z="0" />
  <color r="1" g="1" b="1" />
  <fog   r="1" g="1" b="1" />
</light>
```

- *size*: Ángulo de cono (¡la mitad que el de Blender!).
- *blend* y *beam_falloff*: Igual que en Blender.
- *halo*: Controla si hay que proyectar luz volumétrica y sombras, o no.
- *res*: Resolución del mapa de sombras (sólo para sombras volumétricas, por ahora).
- *blur*: Borrosidad aplicada al rango de sombras volumétricas desde 0.0 a 1.0.
- *fog_density*: Cantidad de niebla en el halo.
- *samples*: Número de muestras a utilizar para renderizar el halo. Por defecto, igual que 'res'. A más muestras, menos ruido.
- *from*: Posición de la luz.
- *to*: Objetivo de la luz.
- *color*: Color de la luz.
- *fog*: Color de la luz volumétrica.

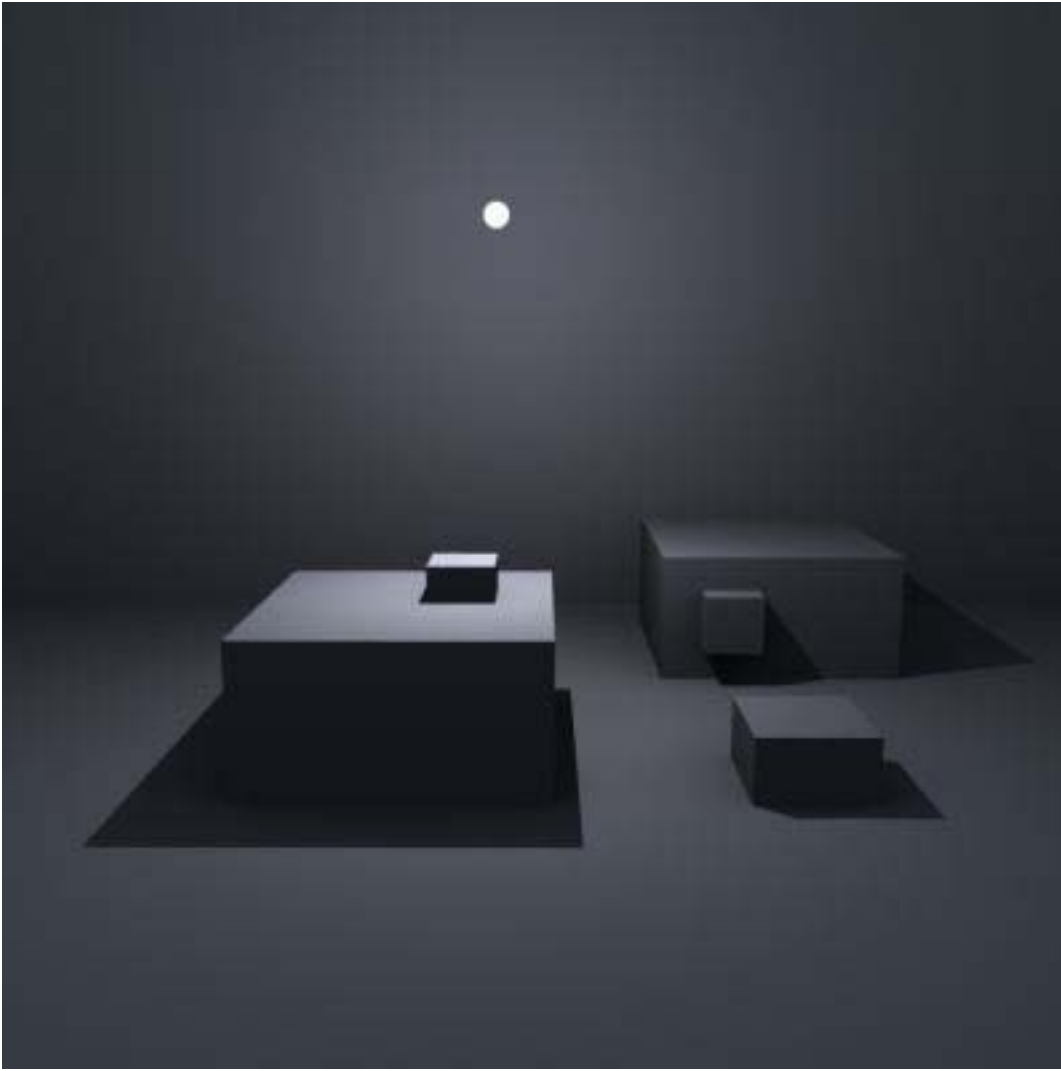


30.6.2. Luz Point

Es bastante similar al Lamp de Blender... ¡Pero proyecta sombras!

```
<light type = "pointlight" name= "omniLight" power= "1.000000"
cast_shadows= "on" >
    <from x="6.372691" y="3.340035" z="2.815973" />
    <color r="1.000000" g="1.000000" b="0.000000" />
</light>
```

- *power*: Intensidad de luz.
- *cast_shadows*: Si proyecta (raytrace) o no sombras ('on' o 'off').
- *from*: Posición de la luz.
- *color*: Color RGB de la luz.



Luz Point con luz Hemi (el punto blanco representa la posición de la luz).

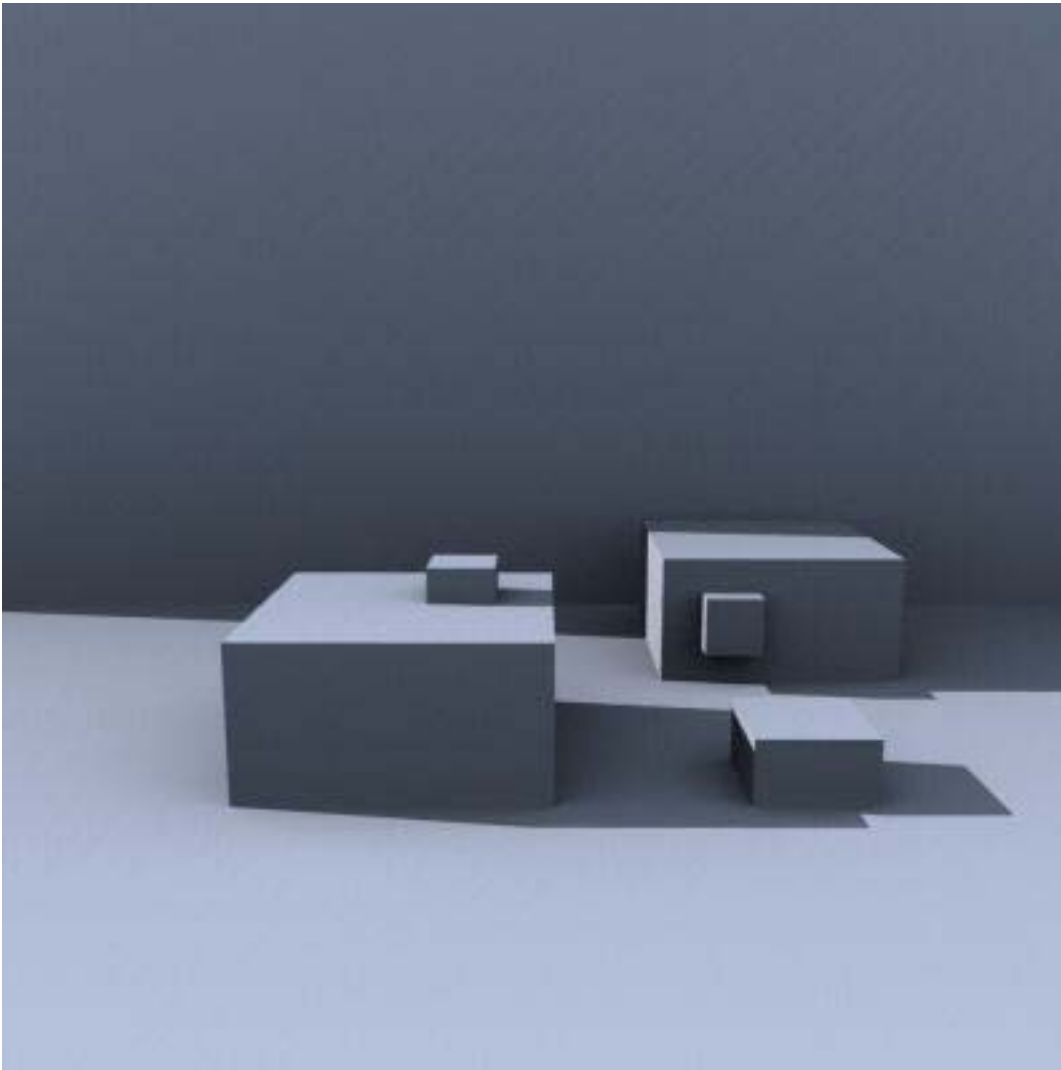
30.6.3. Luz Sun

Es, una vez más, similar al Sun de Blender... ¡pero proyecta sombras!

```
<light type ="sunlight" name="Lamp.001" power="1.000000"
cast_shadows="on">
    <from x="0.026929" y="-0.071142" z="3.552329" />
    <color r="1.000000" g="1.000000" b="1.000000" />
</light>
```

- *power*: Intensidad de la luz solar.
- *cast_shadows*: Si proyecta o no sombras ('on' o 'off').
- *from*: Posición de la luz (¡la dirección apunta automáticamente hacia el origen!).

- *color*: Color de la luz.



Luz Sun, con luz Hemi para proporcionar algo de sombreado difuso.

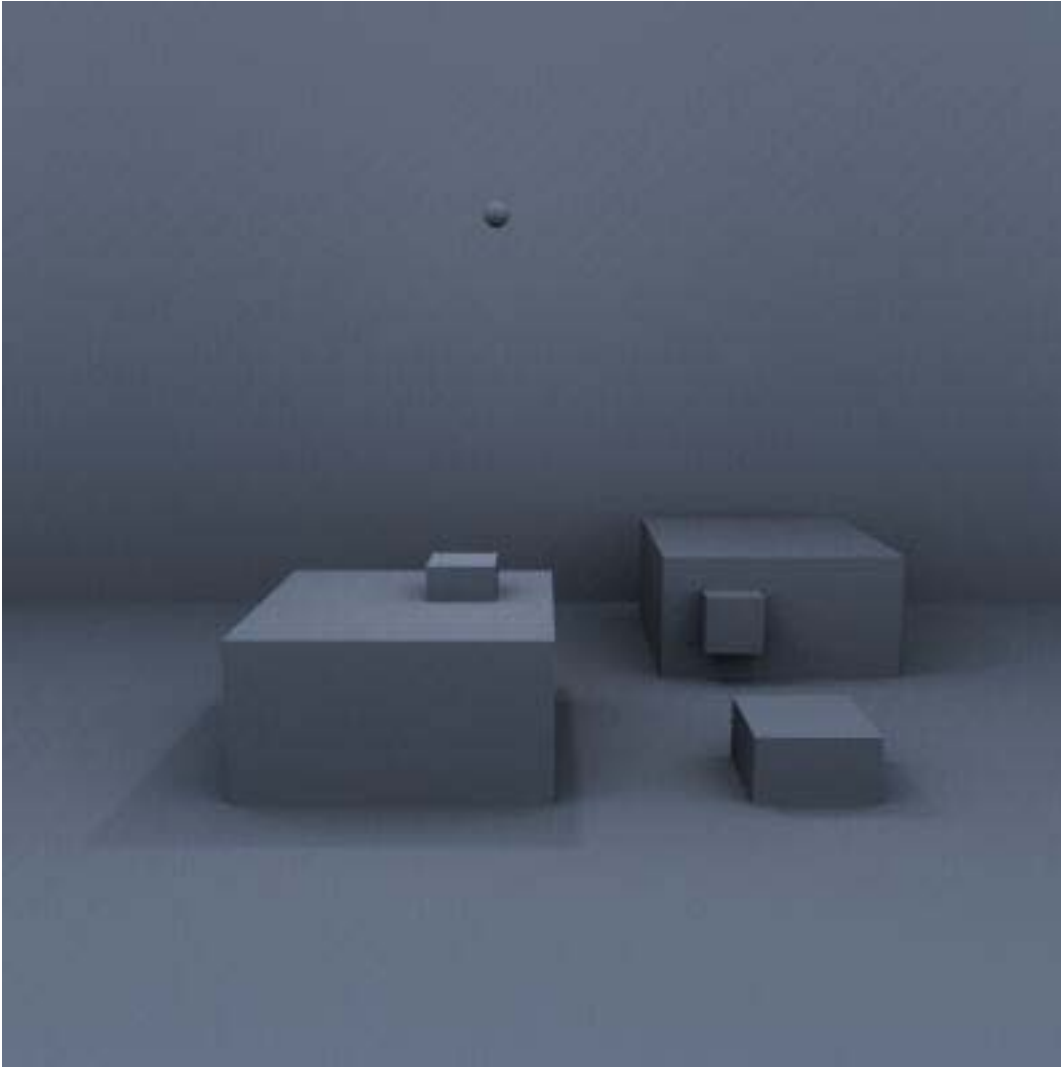
30.6.4. Luz Soft

```
<light type= "softlight" name="softomni" power="1.000000" res="512"
radius="5.000000" bias="0.300000" >
    <from x="0.026929" y="-0.071142" z="3.552329" />
    <color r="1.000000" g="1.000000" b="1.000000" />
</light>
```

- *power*: Intensidad de la luz.
- *res*: Resolución del mapa de sombras.
- *radius*: Radio de la borrosidad (entre las áreas sombreadas y no sombreadas, crea la apariencia de

suavidad).

- *bias*: Tendencia del mapa de sombras. Cercanía de la sombra al objeto; si tenemos una sombra que inunda áreas que no debiera, intentaremos reducir el bias de sombras.

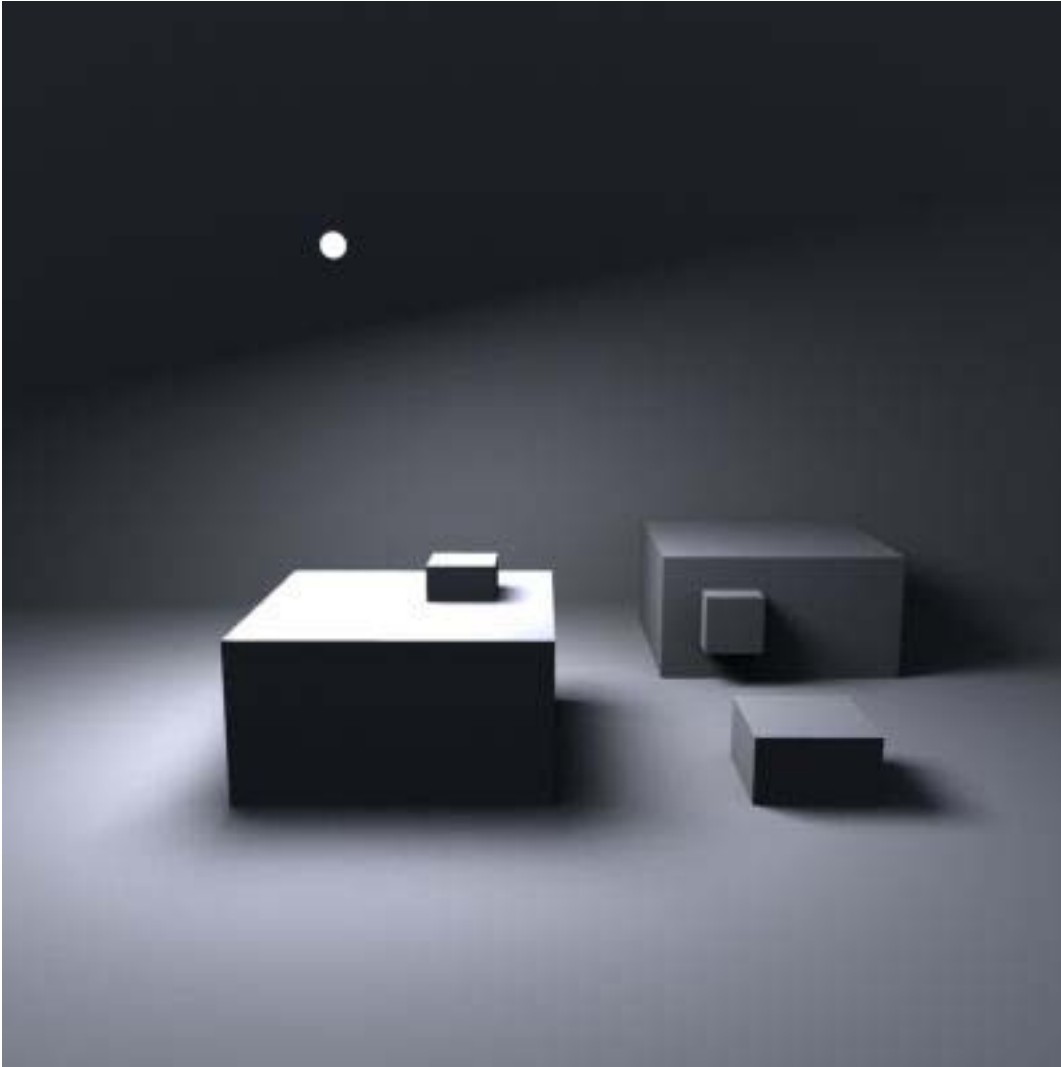


30.6.5. Luz Area

Es una luz que emite uniformemente por un rectángulo.

```
<light type "arealight" name"Lamp.001" power="25.000000"
samples="64.000000" psamples="64.000000">
  <d x = "-2.800729" y = "-0.566380" z = "3.235599" />
  <c x = "-0.686398" y = "-1.759042" z = "3.833231" />
  <b x = "0.541875" y = "0.418420" z = "3.833231" />
  <a x = "-1.572455" y = "1.611082" z = "3.235599" />
  <color r="1.000000" g="1.000000" b="1.000000" />
</light>
```

- *power*: Intensidad de la luz.
- *samples*: Número de muestras a través de la superficie de la luz.
- *psamples*: Predicción de penumbra (calidad de los bordes borrosos de la sombra).
- *a, b, c & d*: Posiciones de las 4 esquinas del rectángulo que define la luz. (¡Cuidado con la orientación!)



Luz Area, con luz Hemi (el punto blanco representa la posición de la luz).

30.6.6. Luz Path

Este sistema indirecto de iluminación produce 'iluminación global' tomando luz de objetos difusos y de fondo. Usa un algoritmo Monte Carlo (MC) de trazado de rayos, o bien un algoritmo Quasi Monte Carlo (QMC) de trazado de rayos. Los resultados de cualquiera de los dos sistemas pueden ser renderizados usando un Cache de Irradiación.

Puesto que MC usa un muestreo aleatorio, los resultados pueden contener algo de ruido. Cuantas más muestras se tomen, menos ruido aparece. Naturalmente ello redundará en un mayor tiempo de renderizado. El muestreo QMC, por otro lado, produce menos ruido, pero en ocasiones puede generar patrones visibles en el sombreado de los objetos. Ambos ruido y patrones pueden reducirse con el filtro incorporado de yafray Anti Noise.

La luz Path producirá bellos efectos de radiosidad. Puede asimismo producir cáusticas; sin embargo, puesto que los fotones que producen las cáusticas no están enfocados en una dirección concreta como en la luz Photon, los patrones cáusticos serán más suaves, a no ser que se tome un elevado número de muestras.

```
<light type = "pathlight" name = "path" power = "1.000000" depth =
"2" caus_depth = "4" samples = "16" use_QMC = "on" cache = "on"
cache_size = "0.008000" angle_threshold = "0.200000" shadow_threshold
= "0.200000">
</light>
```

- *power*: Intensidad de la luz.
- *samples*: Número de muestras a tomar por píxel. Para conseguir una rápida previsualización borrador del render, puede usarse un número bajo, y aumentar su valor para conseguir el render final.
- *depth*: Número de rebotes del rayo para cada muestra, como mínimo 2 para conseguir luz indirecta.
- *caus_depth*: Número de rebotes del rayo pasando a través de objetos cáusticos.
- *use_QMC*: Cuando está 'on', usará un muestreo Quasi Monte Carlo.
- *cache*: Cuando está 'on', Yafray efectuará una pre-pasada del render para generar un cache de irradiación.
- *cache_size*: El tamaño de la rejilla en el cache de irradiación. Valores pequeños significarán un cache de irradiación de mayor resolución (y una pre-pasada de mayor duración).
- *angle_threshold*: El ángulo entre normales de superficie que determina si el algoritmo de cache considera la superficie como 'lisa'. Si la variación entre normales de superficie es superior a este valor, el algoritmo de cache toma más muestras.
- *shadow_threshold*: La mínima distancia desde el punto de muestreo a la que un objeto puede estar antes de que el algoritmo de cache tome más muestras.

30.6.6.1. Uso de la luz Path con cache

El uso del Cache de Irradiación puede ser delicado, aunque los resultados bien valen la pena, puesto que se puede conseguir con frecuencia la misma calidad de imagen en un lapso de tiempo de render inferior.

El tamaño del cache es el tamaño de una rejilla en la que la escena es dividida. A medida que los rayos son disparados dentro de la escena e intersectan con objetos, el punto en el cual el rayo intersecta con el objeto caerá por lo tanto en uno de los cuadros de la rejilla de cache. En el momento en que el rayo alcanza este punto, el renderizador pregunta primero:

1. "¿Hay algún otro punto de muestreo dentro de este cuadro?" Si la respuesta es no, se toma una muestra; si hay otras muestras en el cuadro, pasa a la siguiente pregunta:
2. "¿Las normales de superficie de las otras muestras son diferentes a mi punto actual?" (el ángulo de la diferencia está definido por 'angle_threshold'). Si la respuesta es sí, se toma una muestra; si las normales de superficie son todas iguales, pasa a la siguiente pregunta:
3. "¿El Punto de Intersección está próximo a algún otro objeto?" (la distancia umbral está definida por 'shadow_threshold' y la distancia entre el punto de intersección y el punto de la muestra actual juega una parte). Si la respuesta es sí, se toman más muestras, si no, se salta el punto de muestra y el renderizador pasa al siguiente punto de intersección.

Haciendo esto, el renderizador encuentra áreas de la imagen que necesitan más muestras (áreas de detalle alto), y áreas que necesitan menos muestras (áreas con poco detalle, tal como muros lisos). La toma de muestras es la parte de Iluminación Global que consume más tiempo; tomando muestras únicamente donde son necesarias, la luz Path con cache puede producir imágenes fantásticas en un lapso de tiempo comparativamente corto.

Se necesita algo de ajuste manual para encontrar el 'punto dulce' de la configuración de una escena determinada. Si el tamaño del cache es demasiado bajo, prácticamente cada punto de muestra será usado de todas formas, puesto que la respuesta a la pregunta 1 será casi siempre 'no'. Por otro lado, si el tamaño del cache es demasiado alto, la distancia entre puntos de muestreo (la cual juega una parte en determinar si un objeto está considerado 'cercano') será tal que la respuesta a la pregunta 3 será casi siempre 'sí'. Ambas situaciones resultarán en una toma de más muestras y un renderizado más largo.

30.6.7. Luz Hemi

Este sistema de iluminación indirecta realiza lo que se denomina comúnmente 'pase de oclusión'. Esto produce una luz difusa rápida en la escena ignorando las propiedades de superficie (color) de los objetos, y únicamente determinando si el punto en cuestión está en la sombra o no. Por esta causa, la luz Hemi no producirá sangrado de color entre objetos (al contrario que la luz Path). Utiliza un algoritmo Monte Cristo (MC) de trazado de rayos, o bien un algoritmo Quasi Monte Cristo (QMC) de trazado de rayos. Puesto que MC usa un muestreo aleatorio, los resultados pueden contener algo de ruido. Cuantas más muestras se tomen, menos ruido aparece. Naturalmente ello redundará en un mayor tiempo de renderizado. El muestreo QMC, por otro lado, produce menos ruido, pero en ocasiones puede generar patrones visibles en el sombreado de los objetos. Ambos ruido y patrones pueden reducirse con el filtro incorporado de yafray Anti Noise.

La luz Hemi asume que la escena está iluminada uniformemente, como si una gran esfera rodeara la escena, iluminándola con el color especificado en la etiqueta 'color'. Si se omite la etiqueta color, la luz muestreará el fondo (background) del render en caso de existir (se puede usar esto con fondos HDRI para conseguir simulaciones de iluminación realistas rápidas para composición con una escena real).

```
<light type="hemilight" name="sky" power= "0.500000" samples = "20"  
use_QMC = "on">  
    <color r ="0.800000" g ="0.900000" b ="1.000000" />  
</light>
```

- *power*: Intensidad de la luz.
- *samples*: Número de muestras a tomar. Más muestras significará un resultado más suave, pero tiempos de render más largos.
- *use_QMC*: Cuando está en 'on' usará un muestreado Quasi Monte Carlo.
- *color*: Color de la luz difusa.



Luz Hemi con valor de color azul claro (modelo de www.amazing3d.com)



30.6.8. Luz Photon

Es una luz con foco que produce radiosidad y efectos cáusticos.

```
<light type= "photonlight" name="Lamp.002caus" power = "100.000000"
photons = "50000" depth = "3" search = "100" angle = "15.000000"
    mode = "caustic" fixed_radius = "0.100000" cluster =
"0.010000" use_QMC = "off" >
    <from x="6.372691" y="3.340035" z="2.815973" />
    <to x="0.285646" y="0.149627" z="1.397566" />
    <color r="1.000000" g="1.000000" b="1.000000" />
</light>
```


- *name*: nombre de la luz Photon.
- *power*: Escala el efecto de la luz Photon, ya sea cáustica o difusa.
- *mode*: Establece la luz Photon en difusa o cáustica, tal como se detalla más abajo.
- *photons*: Número de fotones a trazar; a más fotones, más información de la cual generar el mapa de fotones. Generalizando, se necesitan menos fotones para luces Photon difusas.
- *depth*: Cantidad de reflexiones (rebotes) o refracciones que realizarán los fotones.
- *search*: Número de fotones a recoger al sombrear. Valores más altos suavizarán el efecto (cuando se incrementa la búsqueda (*search*), también debería incrementarse '*fixed_radius*').
- *fixed_radius*: Radio de búsqueda al buscar fotones (el número de fotones buscados es definido por '*search*').
- *cluster*: Define la unidad más pequeña en el mapa de fotones creado. Cuanto menor sea el número, más definición tendrá el mapa de fotones.
- *use_QMC*: Cuando está en 'on' usará el [trazado de rayos Quasi Monte Carlo](#).
- *from*: Posición de la luz.
- *to*: Objetivo de la luz.
- *angle*: Similar al valor '*size*' de la luz Spot. Ángulo del haz de fotones.
- *color*: Color de la luz.

Las luces Photon tienen dos modos, 'cáustico' y 'difuso'. En el primer modo la luz dibujará fotones reflejados y transmitidos, causando que la luz forme patrones cáusticos de luz que viajan a través de objetos transparentes, es decir, objetos que tienen los valores apropiados en su etiqueta *Object*. En el modo 'difuso' los fotones son reflejados por superficies difusas en direcciones aleatorias para producir 'radiosidad' o 'Iluminación Global'. En ambos modos sólo la luz indirecta es almacenada (fotones que han rebotado al menos una vez), por lo que la luz directa todavía debe realizarse con una luz normal.

¿Por qué no trabajar con ambos modos? Normalmente se usan valores de fotones diferentes para cáusticas que para radiosidad. Las necesidades cambian, por lo que es mejor tener una luz diferente para cada tarea.

30.6.8.1. Afinar las luces Photon

30.6.8.1.1. photons

Seleccione un valor adecuado a la tarea a realizar; para radiosidad se necesitarán pocos fotones. Para cáusticas dependerá de qué resolución se desee en las formas.

30.6.8.1.2. search, fixed_radius y cluster

Estos parámetros están firmemente relacionados; es necesaria una combinación adecuada de estos tres parámetros para conseguir buenos resultados. El parámetro '*search*' define cuántos fotones hay que buscar desde un punto concreto, '*fixed_radius*' define a cuanta distancia de este punto buscar los fotones. Una vez los fotones han sido recogidos, el área dentro de '*fixed_radius*' es dividida en pequeñas regiones

(clusters), el tamaño de las cuales está definido por el parámetro 'cluster'. Los fotones que quedan dentro del mismo 'cluster' son reducidos a un solo resultado: el valor medio de todos ellos (equivalente a 1 píxel en el mapa de fotones).

Si los efectos difusos o cáusticos aparecen fracturados en formas geométricas, el algoritmo no está encontrando en número necesario de fotones (search) dentro del radio definido (fixed_radius). Para solucionarlo, debemos asegurarnos de que hay suficientes fotones dentro del radio de búsqueda para alcanzar la cantidad buscada. Para conseguir esto, puede incrementarse la cantidad total de fotones (photons), incrementar el radio de búsqueda (fixed_radius), o reducir la cantidad de fotones buscada (search). Incrementar el número de fotones ralentizará la primera pasada, y seguramente habrá que añadir *muchos* más fotones para poder apreciar mucho cambio. Incrementar el radio de búsqueda mostrará los resultados rápidamente pero depende de que haya fotones suficientes en la escena inicialmente. Grandes diferencias entre los parámetros fixed_radius y cluster (por ejemplo fixed_radius grande y cluster pequeño) incrementarán el tiempo de renderizado (lo cual tiene sentido, pues cada muestra, que será grande debido al gran fixed_radius, está dividida en muchas pequeñas regiones debido al pequeño tamaño de cada cluster). Una buena regla general es:

$$\text{fixed\_radius}/\text{cluster} = \text{sqrt}(\text{search})$$

Esto significa que si se está ensayando con 100 fotones (search = 100), entonces fixed_radius dividido por cluster debería ser igual a 10 ($\text{sqrt}(100) = 10$), por lo que si definimos un tamaño de cluster de .01, entonces fixed_radius debería ser alrededor de .1 ($(10 * .01 = .1) = (.1 / .01 = 10)$).

[Prev](#)

Objetos renderizables

[Home](#)[Up](#)[Next](#)

Fondo

30.7. Fondo

Relativo a Blender v2.31

Añade una imagen de fondo (mapa ambiental), color o cielo a tu render.

Pathlight and Hemilight pueden muestrear el color de fondo y la intensidad para simular luz del mundo real.

30.7.1. Fondo de imagen normal

```
<background type="image" name="envnorm" power="1.000000">  
  <filename value="C:\directory\image.jpg"/>  
</background>
```

- *type - image*: Permite utilizar una imagen bitmap de cualquiera de los formatos de textura soportados como fondo. La imagen es mapeada alrededor de la escena como una esfera, por lo que la imagen deberá estar en formato latitud/longitud (radio 2:1).
- *name*: Nombre del fondo.
- *Power*: Nivel de brillo del bitmap. Por defecto es 1.0, números mayores aumentan el brillo, números menores disminuyen el brillo.
- *filename*: Ruta completa y nombre del archivo de la imagen, incluyendo la extensión del archivo.

30.7.2. Fondo HDRI

```
<background type = "HDRI" name = "envhdri" exposure_adjust = "0">  
  <filename value = "C:\directory\image.hdr" />  
</background>
```

- *type - HDRI*: Permite utilizar una imagen HDR como fondo. La HDRI es mapeada alrededor de la escena como un mapa angular, no en latitud longitud como en el fondo de imagen normal.
- *name*: Nombre del fondo.

- *exposure_adjust*: Similar al 'power' de un fondo de imagen normal. Por defecto es 0, incrementándolo aclarará el HDR, decrementándolo lo oscurecerá (equivalente a ajustar el f-stop de una cámara física).
- *filename*: Ruta completa y nombre de la HDRI, incluyendo la extensión del archivo.

30.7.3. Fondo constante

```
<background type="constant" name = "constbackg">
  <color r="1.000000" g="1.000000" b="1.000000"/>
</background>
```

- *type - constant*: Permite asignar un color único al fondo.
- *name*: Nombre del fondo.
- *color*: r - valor de color rojo (0.000000 - 1.000000) g - valor de color verde(0.000000 - 1.000000) b - valor de color azul (0.000000 - 1.000000). 1/1/1 es blanco, 0/0/0 es negro.

30.7.4. Fondo de sol/cielo

```
<background type="sunsky" name ="Sun1" turbidity ="4.000000"
add_sun="on" sun_power="1.000000"
  a_var="1.000000" b_var="1.000000" c_var="1.000000" d_var="1.000000"
e_var="1.000000" >
  <from x="-0.007401" y="8.589217" z="3.737965"/>
</background>
```

- *type - sunsky*: Permite asignar un cielo realista con sol opcional.
- *turbidity*: Densidad atmosférica(pe neblina/niebla) cuanto menor es el número, menos visible es el cielo. El valor 4 corresponde a cielo despejado.

30.8. Cámara

Relativo a Blender v2.31

```
<camera name="Camera" resx="1024" resy="576" focal="1.015937">  
  <from x="0.323759" y="-7.701275" z="2.818493"/>  
  <to x="0.318982" y="-6.717273" z="2.640400"/>  
  <up x="0.323330" y="-7.523182" z="3.802506"/>  
</camera>
```

- *Name*: Nombre de la cámara.
- *resx*: Resolución horizontal (anchura de la imagen renderizada en pixeles).
- *resy*: Resolución Vertical (altura de la imagen renderizada en pixeles).
- *res*: Resolución de la sombra (solamente para las sombras volumétricas por ahora).
- *focal*: Campo Visual. Equivalente a la longitud de la lente en cámara fotográfica en el mundo real - 1.093 equivale 35mm, 6.25 equivale 200mm.
- *from*: Posición de la cámara.
- *to*: Dirección de la cámara.
- *up*: La cámara 'Up Vector' - define que está considerada la dirección 'hacia arriba' de la cámara.

30.9. Renderizado

Relativo a Blender v2.31

La salida del archivo de imagen, se basa en la entrada de la vista de 'Cámara'.

Tip Multiples vistas

Puedes tener varias cámaras y/o bloques de render dentro del mismo XML, para guardar la misma visión, con diferentes valores de salida, o distintas vistas (cámaras) de varias resoluciones, con diferentes fondos.

```
<render camera_name = "Camera" AA_passes = "2" AA_minsamples = "2"
AA_pixelwidth = "1.500000" AA_threshold = "0.040000"
      raydepth = "5" bias = "0.300000" gamma = "1.000000" exposure
= "0.000000" background_name="background"
      <outfile value="C:\yablex\texture sample\texture sample.tga"/
> <save_alpha value="on" />
</render>
```

- *AA_passes*: Selecciona el valor de los pasos anti-alias para realizarse. Un valor 0 significa que no hace ningun anti-aliasing.
- *AA_minsamples*: Selecciona el número de muestras por paso.

Hay varias maneras de realizar estos parámetros. Puedes elegir '*AA_minsamples*' a cierto valor, y seleccionar '*AA_passes*' a 1, entonces después de un primer paso de render todos los pixels que lo necesiten serán anti-aliaseados usando el numero completo de las muestras seleccionadas con '*AA_minsamples*'. El viejo método es equivalente a fijar '*AA_minsamples*' a 1 y '*AA_passes*' al numero de muestras, entonces todos los pixels serán continuamente chequeados si todavía necesitan pruebas adicionales, esto hara que funcione más lento la generación de imagenes de rayos (raytracing), pero puede ser más rapido cuando renderizamos con el hemi- o el path-light. Sin embargo, debido a las limitaciones internas no se trabaja bien con los altos valores de la muestra. Puedes combinar ambos métodos, comprueba los pixeles a cada paso y tomar más muestras por paso en el mismo tiempo, por ejemplo para 16 muestras, tendrás que poner los valores *AA_pases* y *AA_minsamples* a 4 (4 x 4=16).

- *AA_pixelwidth*: Fija la cantidad de coincidencia de los pixeles usados para AA.

El rango es 1 a (las opciones comunes son 1.5 or 2.0), el mas alto, el mejor y el más liso el AA, pero dependiendo de tus preferencias puede ser que encuentres en la iamgen un pequeño velado. Un valor de 1.0 es equivalente a el viejo metodo.

- *AA_threshold*: Fija el valor de umbral en qué punto se considera un pixel para el anti-aliasing.

La gama es a partir de la 0.0(anti-alias cada pixel) a 1.0 (no anti-aliasing).

Puesto que se utiliza QMC, los ajustes con los valores bajos de la muestra como en el ejemplo arriba sin embargo pueden producir resultados absolutamente buenos.

Anti-aliasing también se hace en el canal alfa.

- *save_alpha*: AL guardar un rederizado de la imagen con el canal alfa selecciona el valor 'on'.

[Prev](#)

Cámara

[Home](#)

[Up](#)

[Next](#)

Filtros

30.10. Filtros

Relativo a Blender v2.31

30.10.1. Filtro anti ruido

```
<filter type="antinoise" name="Anti Noise" radius = "1.000000"  
max_delta = "0.100000">  
</filter>
```

- *type-antinoise*: Fija los procesos la imagen renderizada, reduciendo el ruido resultando de demasiado poco en la trayectoria de la luz, semi-iluminación, o muestras concentradas.
- *name*: Nombre del filtro
- *radius*: Cantidad de desenfoque a aplicar en las áreas consideradas de tener ruido.
- *max_delta*: Tolerancia que se fija para el ruido. Con valores más altos, más la imagen considerará 'ruido' y tendrá la falta de definición aplicada a ellos.

30.10.2. Profundidad de Filtro de Campo

```
<filter type="dof" name="dof" focus = "12.5" near_blur = "10.000000"  
far_blur = "10.000000" scale = "2.000000">  
</filter>
```

- *type-dof*: Fija los procesos la imagen renderizada, usando la información de la profundidad para aplicar el efecto del foco.
- *name*: Nombre del filtro.
- *focus*: Distancia de la cámara que está en foco (los objeto que esten más lejos y más cerca estarán desenfocados).
- *near_blur*: Cantidad de desenfoque de los objetos delante del punto de foco.
- *far_blur*: Cantidad de desenfoque objects detrás del punto de foco.
- *scale*: Escala el área que está en foco. Los valores más altos disminuirá el efecto de la profundidad de campo como las áreas fuera de foco seran llevadas fuera del area de foco.

El filtro de la profundidad de campo es un filtro 2D, por ej. fija la tecnica del proceso, y como tal, tiene sus ventajas e inconvenientes. Si usas el renderizado de imagen, sumas la memoria intermedia de Z (le dice al filtro como está de lejos cada pixel desde la cámara) para calcular si se velan o no se velan de pixeles.

Porque es un efecto 2D tiene la ventaja de ser extremadamente rapido. Al menos hay algunas desventajas:

Las reflexiones no las muestra correctamente. Si te fijas en una reflexión, veras que le falta definición de la reflexión está basada en la distancia de la cámara al *plano de reflexión*, no al *objecto en la reflexión*.

Porque el DOF se hace en una imagen 2D, en vez de una escena 3D, el desenfoque no puede saber que está detras de cada objeto, por lo tanto los bordes de un objeto extremadamente desenfocados en el primer plano quedara a menudo manchados o sucios.

Si tienes presentes estas limitaciones, el filtro de Profundidad de campo puede producir grandes efectos de campo en vistas rapidamente.

[Prev](#)[Renderizado](#)[Home](#)[Up](#)[Next](#)[Glossary](#)

[Prev](#)

Glossary

A-Z

Activo

Blender distingue entre *seleccionado* and *activo*. Sólo un objeto o elemento puede estar *activo* en un momento dado, por ejemplo para permitir la visualización de sus datos.

Un objeto activo es el que está en Modo Edición, o se puede conmutar inmediatamente a Modo Edición (normalmente con **TAB**). No hay más de un objeto activo al mismo tiempo. Típicamente, el último objeto seleccionado es el activo.

See Also: [Selected](#).

Actuador (Actuator)

Un bloque lógico (LogicBrick) que actúa como un músculo de un ente vivo. Puede mover un objeto, o también producir un sonido.

See Also: [LogicBrick](#), [Sensor](#), [Controller](#).

Alpha

El valor "alpha" en una imagen denota su opacidad, usada para mezcla (blending) y antialiasing.

Luz ambiental (Ambient light)

La luz que existe en todas partes sin una fuente particular. La luz ambiental no genera sombras, pero rellena las áreas sombreadas de una escena.

Anti-aliasing

Un algoritmo diseñado para reducir el efecto escalera que se produce cuando se dibujan primitivas gráficas en una cuadrícula.

AVI

"Audio Video Interleaved". Un formato contenedor para vídeo con audio sincronizado. Un fichero AVI puede contener varios flujos de vídeo y audio comprimidos.

Back-buffer

Blender usa dos buffers en los que dibuja la interfaz. Este sistema de doble buffer permite que se dibuje un buffer mientras se dibuja en el otro. Para algunas aplicaciones en Blender el back-buffer se usa para almacenar información de selección basada en el color.

Bevel

El FIXME:"beveling":chaflán? elimina los bordes afilados de un objeto extruido añadiendo material adicional alrededor de sus caras. Los FIXME"Bevels" son especialmente útiles para logotipos, y animación en general, dado que reflejan luz adicional desde los bordes de un objeto además del anverso y los lados.

Caja de contorno (Bounding box)

Un cubo que aparece en la pantalla y que representa la extensión máxima de un objeto. La caja más pequeña en la que cabe el objeto.

Bump map

A grayscale image used to give a surface the illusion of ridges or bumps. In Blender bumpmaps are called Nor-maps.

Channel

Some DataBlocks can be linked to a series of other DataBlocks. For example, a Material has eight *channels* to link Textures to. Each IpoBlock has a fixed number of available *channels*. These have a name (LocX, SizeZ, enz.) which indicates how they can be applied. When you add an IpoCurve to a channel, animation starts up immediately.

Child

Objects can be linked to each other in hierarchical groups. The Parent Object in such groups passes its transformations through to the Child Objects.

See Also: [Parent](#).

Clipping

The removal, before drawing occurs, of vertices and faces which are outside the field of view.

Controller

A LogicBrick that acts like the brain of a lifeform. It makes decisions to activate muscles (Actuators), either using simple logic or complex Python scripts.

See Also: [LogicBrick](#), [Sensor](#), [Python](#), [Actuador \(Actuator\)](#).

DataBlock (or "block")

The general name for an element in Blender's Object Oriented System.

Doppler effect

The Doppler effect is the change in pitch that occurs when a sound has a velocity relative to the listener. When a sound moves towards the listener the pitch will rise. When going away from the listener the pitch will drop. A well known example is the sound of an ambulance passing by.

Double-buffer

Blender uses two buffers (images) to draw the interface in. The content of one buffer is displayed, while drawing occurs on the other buffer. When drawing is complete, the buffers are switched.

EditMode

The mode for making intra-object graphical changes. Blender has two modes for making changes graphically. EditMode allows intra-object changes (moving, scaling rotating, deleting, and other operations on selected vertices of the active object). By contrast, ObjectMode allows inter-object changes (operations on selected objects).

Switch between EditMode and ObjectMode with Hotkey: **TAB**.

See Also: [ObjectMode](#), [Vertex \(pl. vertices\)](#).

Extend select

Adds new selected items to the current selection (**SHIFT-RMB**).

Extrusion

The creation of a three-dimensional object by pushing out a two-dimensional outline and giving it height, like a cookie-cutter. It is often used to create 3D text.

Face

The triangle and square polygons that form the basis for Meshes or for rendering.

Field

Frames from videos in NTSC or PAL format are composed of two interlaced *fields*.

FaceSelectMode

Mode to select faces on an object. Most important for texturing objects. Hotkey: **FKEY**.

Flag

A programming term for a variable that indicates a certain status.

Flat shading

A fast rendering algorithm that simply gives each facet of an object a single color. It yields a solid representation of objects without taking a long time to render. Pressing **ZKEY** switches to flat shading in Blender.

Fps

Frames per second. All animations, video, and movies are played at a certain rate. Above ca. 15fps the human eye cannot see the single frames and is tricked into seeing a fluid motion. In games this is used as an indicator of how fast a game runs.

Frame

A single picture taken from an animation or video.

Gouraud shading

A rendering algorithm that provides more detail. It averages color information from adjacent faces to create colors. It is more realistic than flat shading, but less realistic than Phong shading or ray-tracing. The hotkey in Blender is **CTRL-Z**.

Graphical User Interface (GUI)

The whole part of an interactive application which requests input from the user (keyboard, mouse etc.) and displays this information to the user. Blender's GUI is designed for an efficient modelling process in an animation company where time equals money. Blender's whole GUI is done in OpenGL.

See Also: [OpenGL](#).

Hierarchy

Objects can be linked to each other in hierarchical groups. The Parent Object in such groups passes its transformations through to the Child Objects.

Ipo

The main animation curve system. Ipo blocks can be used by Objects for movement, and also by Materials for animated colors.

IpoCurve

The Ipo animation curve.

Item

The general name for a selectable element, e.g. Objects, vertices or curves.

Lathe

A lathe object is created by rotating a two-dimensional shape around a central axis. It is convenient for creating 3D objects like glasses, vases, and bowls. In Blender this is called "spinning".

Keyframe

A frame in a sequence that specifies all of the attributes of an object. The object can then be changed in any way and a second keyframe defined. Blender automatically creates a series of transition frames between the two keyframes, a process called "tweening."

Layer

A visibility flag for Objects, Scenes and 3DWindows. This is a very efficient method for testing Object visibility.

Link

The reference from one DataBlock to another. It is a "pointer" in programming terminology.

Local

Each Object in Blender defines a *local* 3D space, bound by its location, rotation and size. Objects themselves reside in the *global* 3D space.

A DataBlock is *local*, when it is read from the current Blender file. Non-local blocks (library blocks) are linked parts from other Blender files.

LogicBrick

A graphical representation of a functional unit in Blender's game logic. LogicBricks can be Sensors, Controllers or Actuators.

See Also: [Sensor](#), [Controller](#), [Actuador \(Actuator\)](#).

Mapping

The relationship between a Material and a Texture is called the 'mapping'. This relationship is two-sided. First, the information that is passed on to the Texture must be specified. Then the effect of the Texture on the Material is specified.

Mipmap

Process to filter and speed up the display of textures.

MPEG-I

Video compression standard by the "Motion Pictures Expert Group". Due to its small size and platform independence, it is ideal for distributing video files over the internet.

ObData block

The first and most important DataBlock linked by an Object. This block defines the Object *type*, e.g. Mesh or Curve or Lamp.

Object

The basic 3D information block. It contains a position, rotation, size and transformation matrices. It can be linked to other Objects for hierarchies or deformation. Objects can be "empty" (just an axis) or have a link to ObData, the actual 3D information: Mesh, Curve, Lattice, Lamp, etc.

ObjectMode

The mode for making inter-object graphical changes. Blender has two modes for making changes graphically. ObjectMode allows inter-object changes (moving, scaling rotating, deleting and other operations on selected objects). By contrast, EditMode allows intra-object changes (operations on selected vertices in the active object).

Switch between ObjectMode and EditMode with Hotkey: **TAB**.

See Also: [EditMode](#).

OpenGL (OGL)

OpenGL is a programming interface mainly for 3D applications. It renders 3D objects to the screen, providing the same set of instructions on different computers and graphics adapters. Blender's whole interface and 3D output in the real-time and interactive 3D graphic is done by OpenGL.

Orthographic view

An orthographic view of an object makes it appear flat and two-dimensional, like a plan or an elevation. All the points of the object are perpendicular to the viewing plane, and are projected in parallel.

See Also: [Perspective view](#).

Oversampling (OSA)

See: [Anti-aliasing](#)

Overscan

Video images generally exceed the size of the physical screen. The edge of the picture may or may not be displayed, to allow variations in television sets. The extra area is called the overscan area. Video productions are planned so critical action only occurs in the center safe title area. Professional monitors are capable of displaying the entire video image including the overscan

area.

Parent

An object that is linked to another object, the parent is linked to a child in a parent-child relationship. A parent object's coordinates become the center of the world for any of its child objects.

See Also: [Child](#).

Perspective view

In a perspective view, the further an object is from the viewer, the smaller it appears.

See Also: [Orthographic view](#).

Pivot

A point that normally lies at an object's geometric center. An object's position and rotation are calculated in relation to its pivot-point. However, an object can be moved off its center point, allowing it to rotate around a point that lies outside the object.

Pixel

A single dot of light on the computer screen; the smallest unit of a computer graphic. Short for "picture element."

Plug-In

A piece of (C-)code loadable during runtime. This way it is possible to extend the functionality of Blender without a need for recompiling. The Blender plugin for showing 3D content in other applications is such a piece of code.

Python

The scripting language integrated into Blender. [Python](#) is an interpreted, interactive, object-oriented programming language.

Quaternions

Instead of using a three-component Euler angle, quaternions use a four-component vector. It is generally difficult to describe the relationships of these quaternion channels to the resulting orientation, but it is often not necessary. It is best to generate quaternion keyframes by manipulating the bones directly, only editing the specific curves to adjust lead-in and lead-out transitions.

Render

To create a two-dimensional representation of an object based on its shape and surface properties

(i.e. a picture for print or to display on the monitor).

Rigid Body

Option for dynamic objects in Blender which causes the game engine to take the shape of the body into account. This can be used to create rolling spheres for example.

Selected

Blender makes a distinction between *selected* and *active* objects. Any number of objects can be *selected* at once. Almost all key commands have an effect on *selected* objects. Selecting is done with the right mouse button.

See Also: [Activo](#), [Extend select](#).

Sensor

A LogicBrick that acts like a sense of a lifeform. It reacts to touch, vision, collision etc.

See Also: [LogicBrick](#), [Controller](#), [Actuador \(Actuator\)](#).

Single User

DataBlocks with only one user.

Smoothing

A rendering procedure that performs vertex-normal interpolation across a face before lighting calculations begin. The individual facets are then no longer visible.

Transform

Change a location, rotation, or size. Usually applied to Objects or vertices.

Transparency

A surface property that determines how much light passes through an object without being altered.

See Also: [Alpha](#).

User

When one DataBlock references another DataBlock, it has a user.

Vertex (pl. vertices)

The general name for a 3D or 2D point. Besides an X,Y,Z coordinate, a vertex can have color, a normal vector and a selection flag. Also used as controlling points or handles on curves.

Vertex array

A special and fast way to display 3D on the screen using the hardware graphic acceleration. However, some OpenGL drivers or hardware doesn't support this, so it can be switched off in the InfoWindow.

Wireframe

A representation of a three-dimensional object that only shows the lines of its contours, hence the name "wireframe."

X, Y, Z axes

The three axes of the world's three-dimensional coordinate system. In the FrontView, the X axis is an imaginary horizontal line running from left to right; the Z axis is a vertical line; and Y axis is a line that comes out of the screen toward you. In general, any movement parallel to one of these axes is said to be movement along that axis.

X, Y, and Z coordinates

The X coordinate of an object is measured by drawing a line that is perpendicular to the X axis, through its centerpoint. The distance from where that line intersects the X axis to the zero point of the X axis is the object's X coordinate. The Y and Z coordinates are measured in a similar manner.

Z-buffer

For a Z-buffer image, each pixel is associated with a Z-value, derived from the distance in 'eye space' from the Camera. Before each pixel of a polygon is drawn, the existing Z-buffer value is compared to the Z-value of the polygon at that point. It is a common and fast visible-surface algorithm.

[Prev](#)

[Home](#)

Filtros