

☒ Preferences ☐

Init ZBrush

- Config
- Interface
- Icolors
- Picker
- Mem
- Marker
- Zscript
- Importexport

iFlipX iFlipY iFlipZ
eFlipX eFlipY eFlipZ
iSwitchYZ eSwitchYZ
Unify Scale @
NormalMapFlipX
NormalMapFlipY
NormalMapFlipZ
NormalMapFlipXY
NormalMapFlipVert

• Deformation

Unify XYZ
Mirror
Smart ReSym
ReSym
Offset
Rotate
Size XYZ
Bend
SBend
Skew
SSkew
RFlatten
Flatten
SFlatten
Twist
Taper
Squeeze
Noise
Smooth
Inflat
Spherize
Gravity
Perspective

Group and Material

Group by: Object

☐ Use material

☐ Create material

Geometry

☐ Rotate

Faces: Quads

☒ Texture

☐ Normals

☐ Smooth

Vertex scale

File

of Digits

☐ Complex

☐ Relative

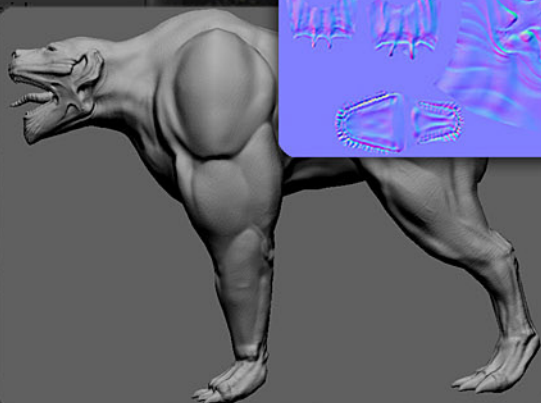
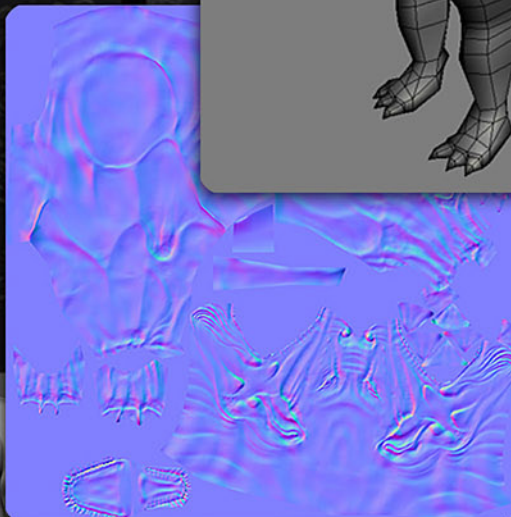
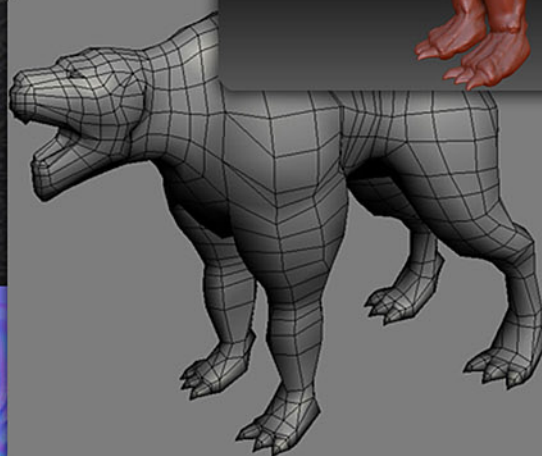
OK

Cancel

NORMAL MAPS

baking, creating & correcting

by Waniek Rafał alias EDI



EVERMOTION
WWW.EVERMOTION.ORG

1. Introduction

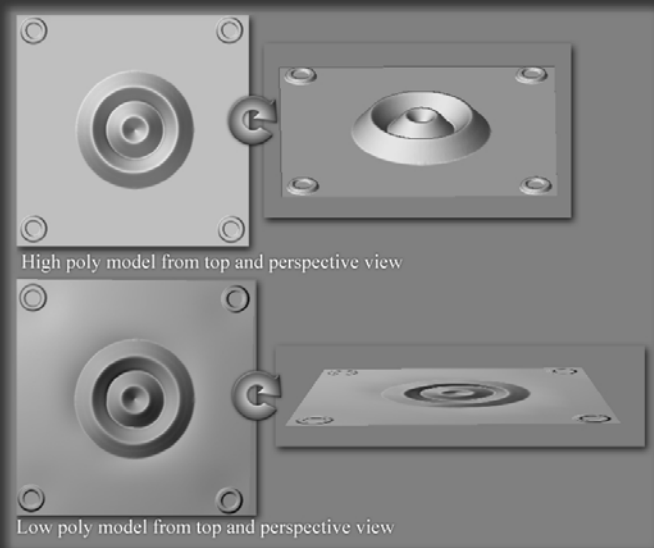
In this tutorial I will try to explain some techniques and show you how to bake and create normal maps, which become very popular in this days, specially in game industry.

Software that I will use :

- Zbrush 2 – for baking normal maps from high poly models. Currently ver 3.1 of Zbrush is available, but the baking techniques are the same as in the ver 2.0.
- For baking normal maps I will use a ZMAPPER plugin, which you can download for free from pixologic website
- 3D studio max – this program I will use for modeling low poly models and UVW mapping. To do this you can use program whatever you want, because for baking purpose we will export the model to obj format.
- Photoshop CS2 – I will use it for creating normal maps and editing baked maps.

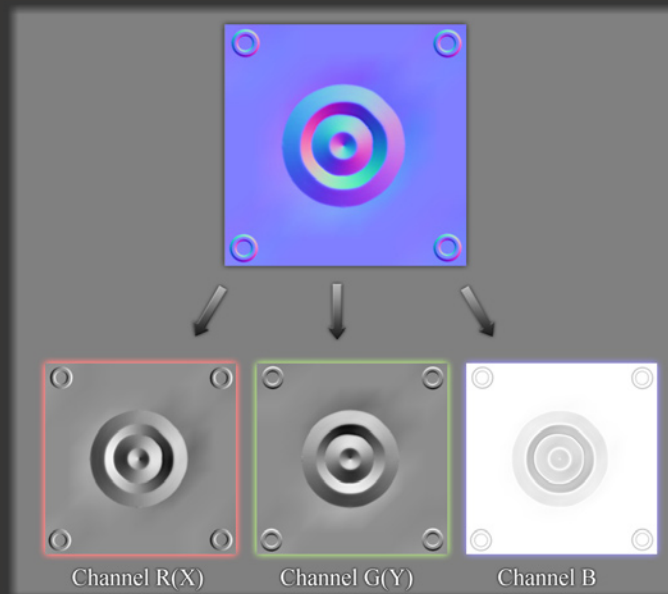
2. What are normal maps, and what are their for?

The purpose of normal maps is to create an illusion of more complex geometry than it really is. Normal mapping doesn't affect the geometry of the low poly model, like the displacement maps do. So if our low poly model is very simple and edgy, the normal map will not change it. Below you see a comparison of high poly model and a simple plane with normal map.



This simple example shows how normal maps works.

Now lets take a look on the normal map itself.



The whole effect of normal maps lies in their RGB channels, specially in R and G channel, often define as X and Y in baking parameters. If the normal map doesn't look like it should be, when displayed in 3dsmax or other 3d rendering (or real time) engines, the problem is in wrong interpretation of R and G channels by the engine. Sometimes you need to swap X and Y channel before baking the map (you can also do this after baking in photoshop) or invert the X or Y channel.

For baking normal maps we will use ZMapper, a free Zbrush plugin which have a lot of presets that will choose the right parameters for us.

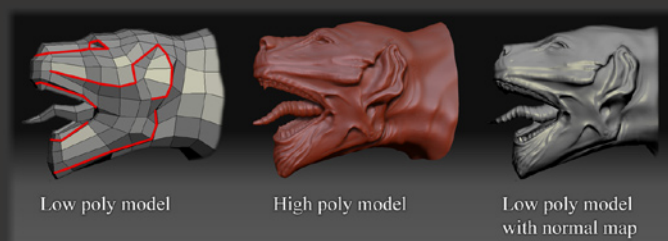
3. Tips for a good start

Before we started to create a high poly models with a lot of details, it is a good idea to do some planning ahead for our model. For example: if there will be some larger elements, like big pockets, large muscles or larger material wrinkles, we will get better results by adding some polygons in our low poly model, because like I said before, the normal map will not change our low poly geometry, and in some camera, angles things, like large muscles or big pockets, will look very flat.

That's why it's better to plan ahead before we started to sculpt our high poly model.

The best solution in this case seams to be a detailed concept art, or at least a simple sketch.

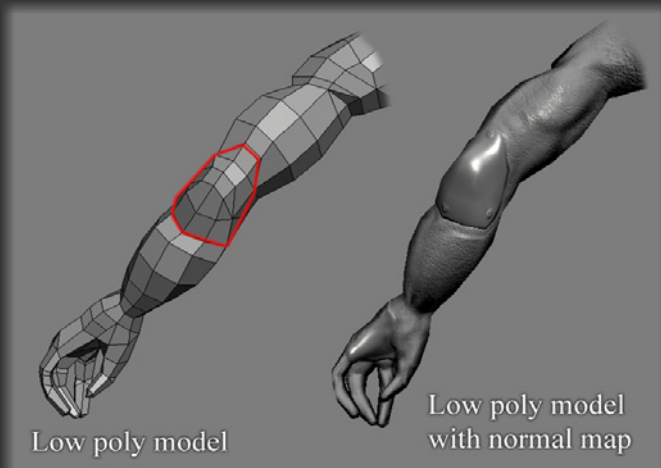
Below you can see an example of low poly model, which is a very simple geometry, but have some high poly geometry characteristics.





EVERMOTION

WWW.EVERMOTION.ORG



Low poly model

Low poly model
with normal map

We need also to remember that the best normal map is nothing without a good color map. That's why normal maps should be a nice addition to a color map, but not otherwise. The best effect we will achieve by perfect matching color, normal and specular map.



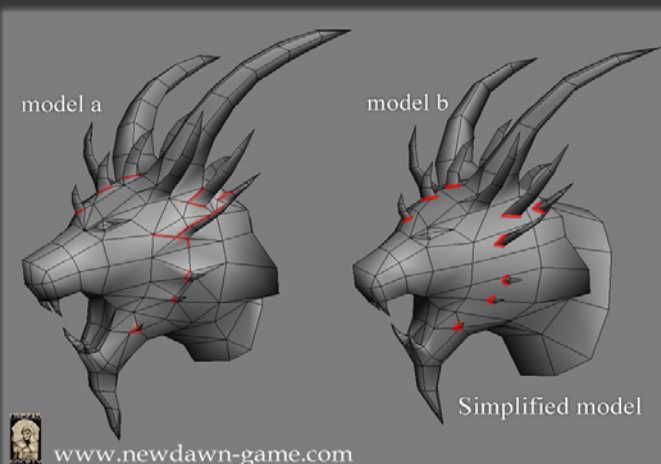
Color

Normal map

normal + color +
specular map

Low poly geometry – things we must keep in mind

During creating low poly models, we often try to use as less triangles as possible, and we are simplifying some elements such as ears, horns, teeth etc. by using separate elements. Such lack of „cohesion” of the models leads to numerous errors when burning the normal maps. So it's better to use more triangles, here and there, than later try to correct the normal maps in photoshop. Below you can see how the simplified geometry influence the normal map. Both normal maps are baked from the same high poly model using ZMapper (Zbrush plugin) and projection method.



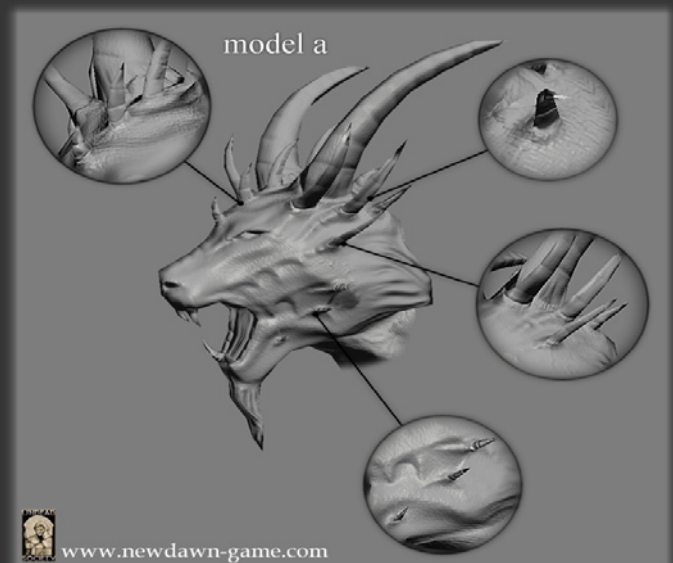
model a

model b

Simplified model



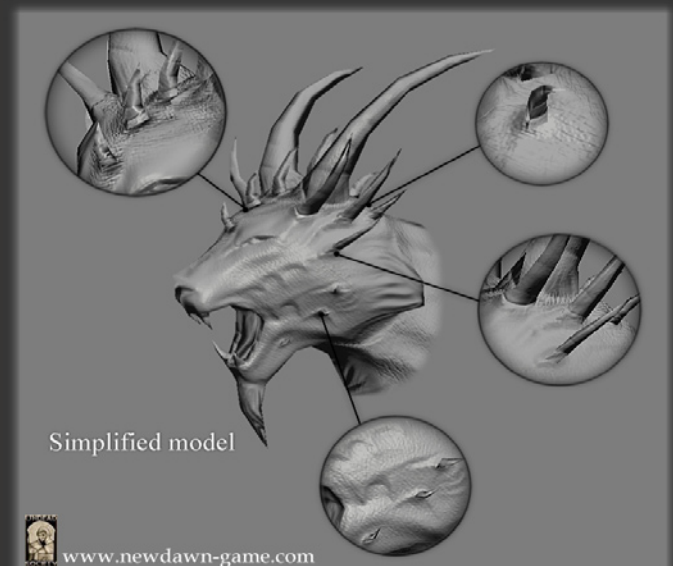
www.newdawn-game.com



model a



www.newdawn-game.com



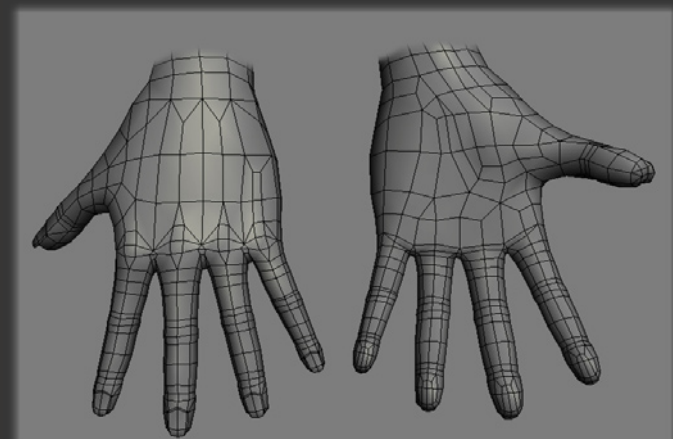
Simplified model



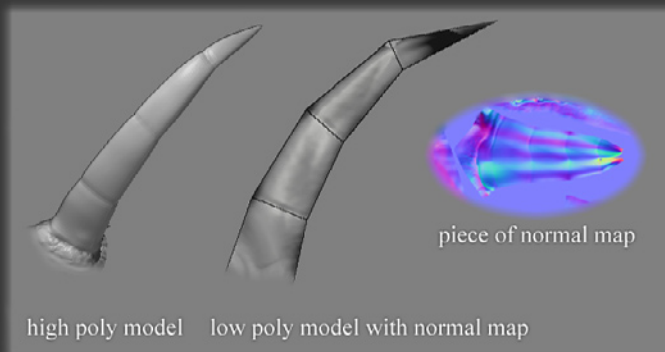
www.newdawn-game.com

A small baking errors we can eventually hide under the color map. However with larger elements it could be difficult to hide.

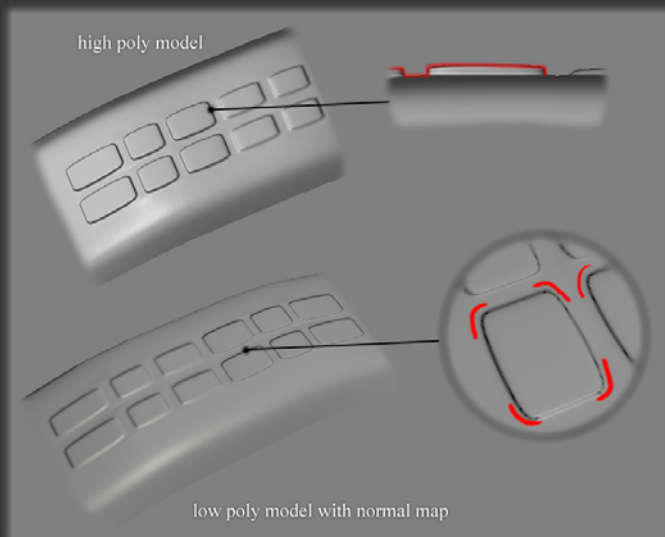
There could be more of problematic places in our low poly models. For example - hands - tight places between the fingers. We can avoid it by spreading out the fingers a little bit, before exporting into Zbrush.



Other problematic places : all kind of sharp ended cones or polygon spikes.



During sculpting high poly models we have to avoid creating deformation and details that are perpendicular to the models surface, like it is shown on the screen below. During displaying normal map we see dark spots at places, where the high geometry details are perpendicular to the surface. On normal map those places are red or dark blue.



Preparing models for export

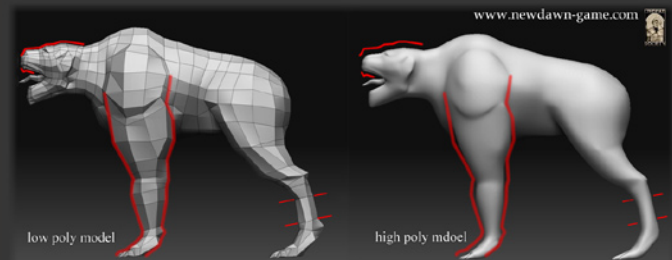
Here are some tips how to prepare our mesh, before we start to export it into Zbrush.

In case of low poly models, where every triangle counts, it is often that in some spots the mesh is dense and regular, and in other the mesh contains large polygons in a form of rectangles or triangles. The general principle is that the mesh should be as regular as possible, and should contain (as far as possible) only quads.

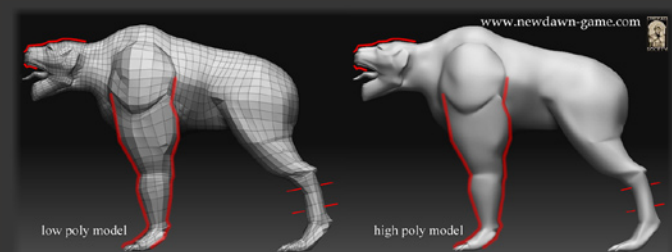
Here, as well, we need to do some planning ahead. Sometimes we need to add more details in some spots, so there is no need to dense the mesh everywhere.

Nowadays, in the time of next generation hardware, we often can spare some extra triangles for our low poly models, so the mesh is regular for itself.

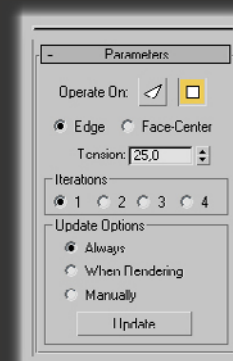
Even when our model is regular and contain mostly quads, after exporting to Zbrush and dividing (DIVIDE) the geometry several time, model can loose it's broad outline.



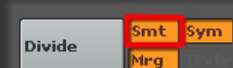
We can correct this by reshaping the model in Zbrush, or use the tessellate modifier in 3dsmax before exporting the model. This modifier will not break our uvw mapping, and will divide the polygons in such a way, that after dividing it again in Zbrush, the model will keep it's low poly shape.



Here are some tessellate modifier parameters that I've used.



Another way to achieve this, is to turn off subdivide smoother option when dividing the mesh to the second level of subdivision, and then turning it back when dividing it again to the third level.



After this operation we must to smooth a little bit the entire model, by using smooth in the deformation options.

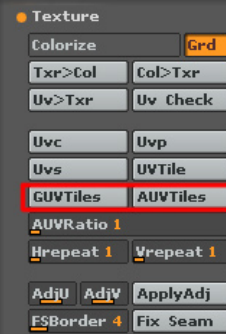
4. Importing and exporting models.

At this point we will concentrate with what parameter, and in what kind of format we will export our low poly models. Before we export our model we have to remember some things:

- **convert the model to editable poly** or just collapse the modifier stack(some of the modifiers can damage our mapping when exporting with them).
- **Reset the xform**, especially if we had scaled or mirrored model. Resetting the xform can flip the model's normal, but you can use the normal modifier to flip it back and collapse the stack again.
- **Mapping** – in case of Zbrush we do not need to unwrap the model before export, because we can do this anytime, by exporting the low poly model back to 3dsmax or any other 3D software, even after finishing the high poly sculpting.

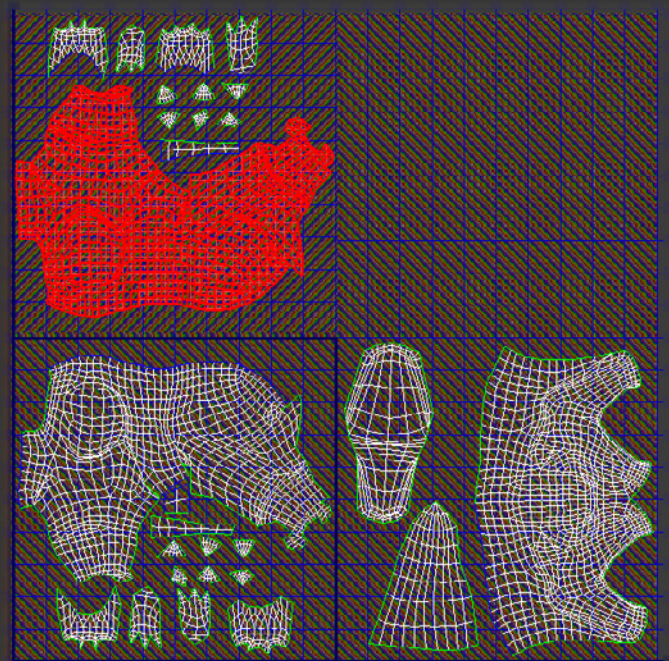
If we do a mapping before export, it's good to remember a few things: keep all the unwrap elements and vertices in the main mapping area.

Sometimes we do a first part of high poly model in other software than Zbrush, and then we add some details in zbrush and bake our normal map using projection, so the high poly model do not need to be unwrapped. If we try to run ZMapper it is possible that we receive the same message as above. The easiest way to deal with this problem is to use automated mapping in Zbrush (AUV tiles or GUV tiles), which you can find in **TOOL>>Texture**.

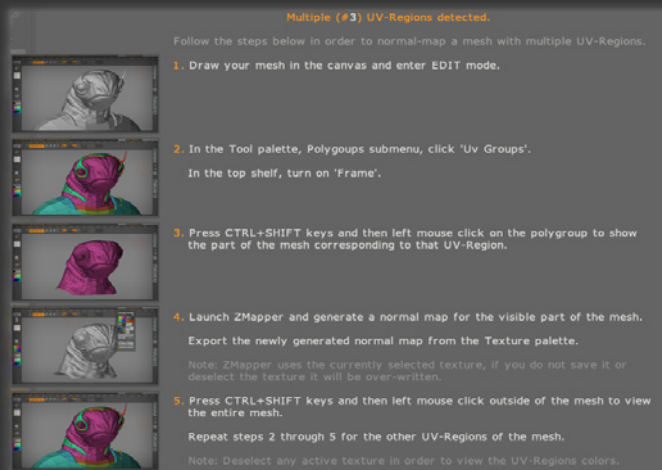


Putting some unwrap elements outside the main mapping area have it purpose as well. During creating complex models for high resolution renders, the resolution of our normal map could be to low, even if we are using 4096x4096 map size.

Below you can see mapping that allows us to bake 3 different maps for the same model, each with maximum resolution of 4096x4096 (or 8000x8000 in Zbrush 3).



If we do not keep all the elements in main mapping area, when baking the normal map using ZMapper, we will see the following message:

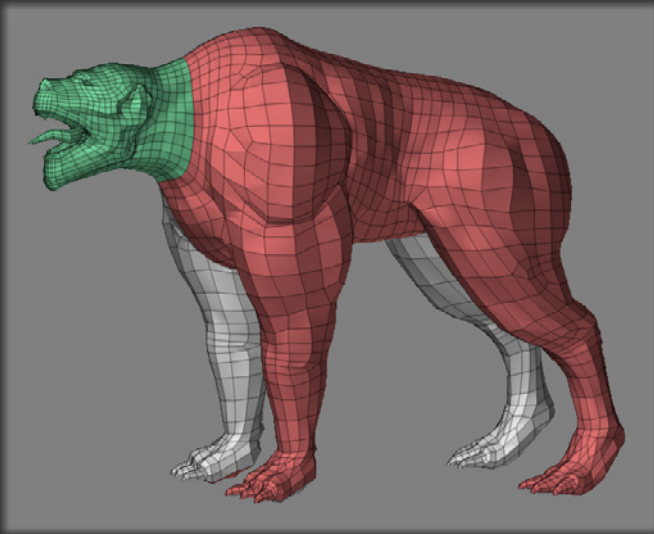


The reason of this messages are small mapping errors.

After importing our model to Zbrush go to **TOOL>>Polygroups** and turn on **UV GROUPS**

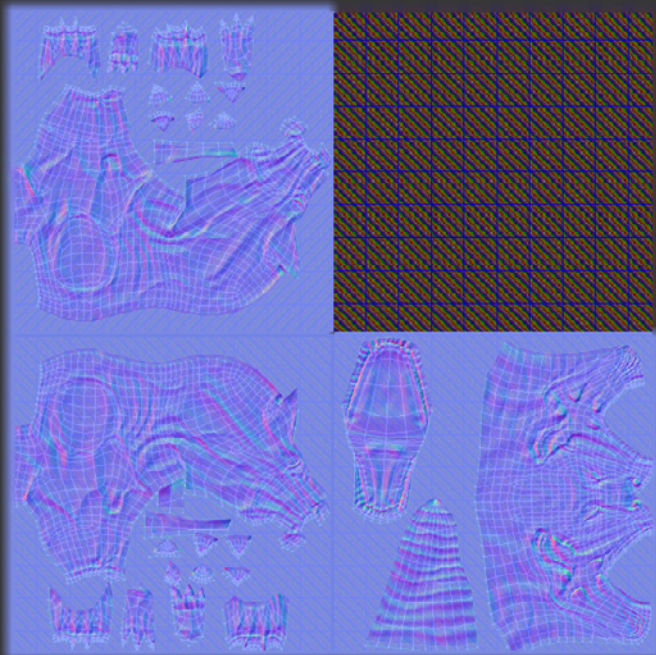


The result of his operation you can see below – the model has been divided into groups, considerate our untypical mapping.



Before we start to bake the normal map we need to choose one of the groups (Ctrl+Shift+left click on desired group), even if we will not do this, ZBrush will remind us, by displaying the message that was shown above (multiple UV regions detected).

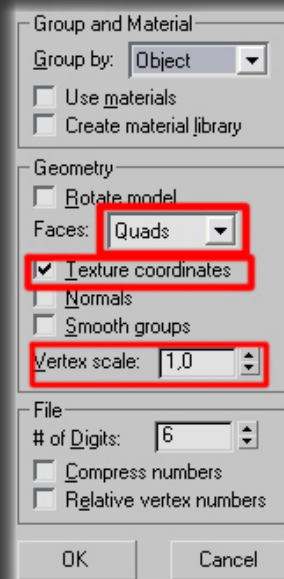
After baking the map for one group, we do the same for others. At the end we will get three different normal maps (up to 8000x8000 for each one) for one model.



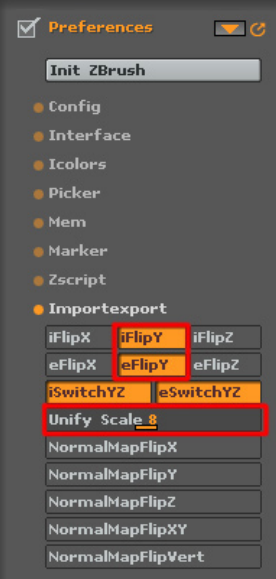
We can bake more than three maps for each model if we need to.

Now let's get to the proper export and import format.

In case of Zbrush, there are two formats for importing the model that we can use: .obj and .dxf format, but in practice we will only use .obj format, because the dxf format does not have mapping coordinates. Obj format is a very popular and universal format, so most of current 3D software have the option to export models into this format. Take a look at the export parameters from Autodesk 3ds max :



Export parameters for 3ds max



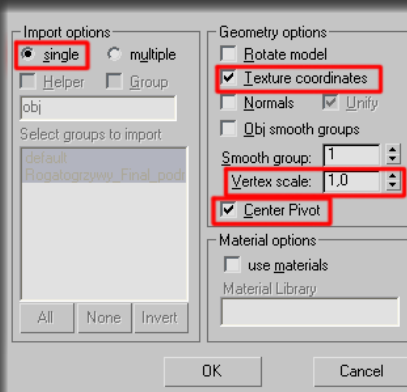
Import parameters for Zbrush

The other very important thing is the proper model import into Zbrush (screen that is shown above). This time we need to setup the parameters only once and save the configuration (In case of Zbrush 3 there is no UNIFY SCALE parameter).

Sometimes imported models have flipped faces normals, in this case go to TOOL>>Display properties and click flip.



We will also need to import the models into 3ds max from Zbrush. See the parameters below :



If we want to import a high polygon model, it is recommended to choose from import option multiple and select groups from the list. Those groups are the same as polygroups in ZBrush. The reason for doing this is that 3dsmax can better handle multiple models with less polygons, than one model with many polygons.

5. Baking maps from high polygon models.

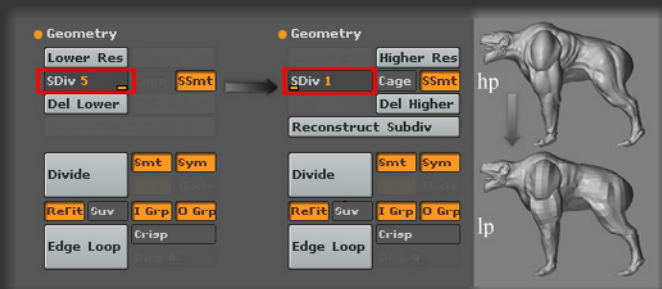
a) standard baking

By standard, I mean baking from a high poly model which was entirely done in ZBrush (except the low poly model) and have non symmetrical mapping.

STEP 1.

The low poly model was imported to zbrush without mapping coordinates, and then was divided and sculpted to a high poly model.

We go down with subdivision levels TOOL>>geometry:



STEP 2.

We export the low poly model (without mapping coordinates) to a obj file, and then we import it to 3dsmax (or any other 3d program) to do the unwrapping. We can map the model or even change it's geometry, **but we can not change the amount of vertices, or change the vertex IDs by adding the symmetry modifier for example.** When the mapping is done, we export our model to an obj file format again.

STEP 3.

Having our model in ZBrush still in EDIT MODE, at the lowest subdivision level, we import the mapped low poly model. Then we can check the UV mapping by using the UV Check option in TOOL>>Texture.

Zbrush will crate a test texture, so we can see all the UV elements and possible mapping errors(overlapping) shown in red.



STEP 4.

We choose the resolution of normal map that we are going to bake (2048x2048 by default). To choose the resolution we go to TEXTURE>> width and height and then click NEW.



STEP 5.

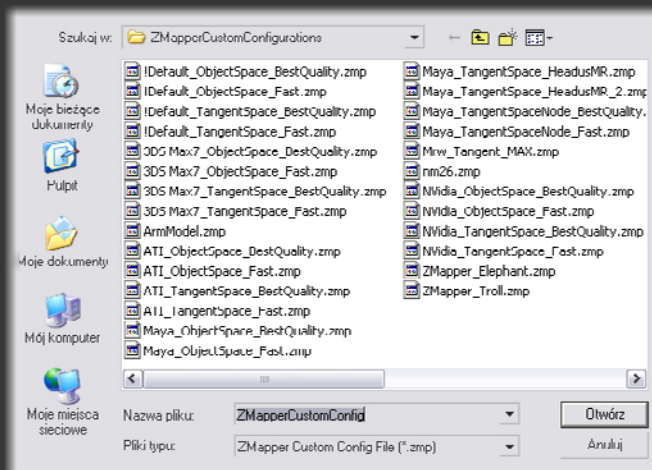
Baking normal map

Start the ZMapper plugin.

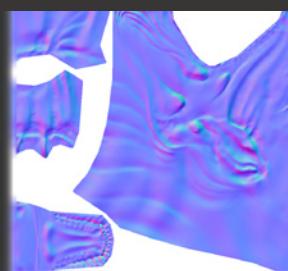
Before we start to bake our map, I will shortly discuss the ZMapper parameters.



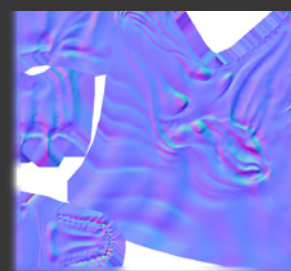
Open/Save configuration – Like I said before, ZMapper offers us a lot of presets for different kind of 3d programs. In case of settings for 3ds max ver 7, the baked maps will not display properly in 3dsmax 8 and 9, therefore I have attached a download link to my own 3ds max settings.



Seam overpaint – this parameter decided how much the borders of the uv elements are interpolated.



Small amount of seam overpaint



High amount of seam overpaint

Inflat Hires Mesh details – this parameter decide how much the details will be inflated – if the value is set to high it could lead to bake errors.

Sharpen Hires Mesh details – this parameter decide how much the details will be sharpen – very useful in a case of low resolution normal maps.

Inflat/Sharpen Bumpmaps details – Zbrush offers the possibility to connecting our normal map with a bump map while baking it. With this parameter you can either sharpen or inflate the bump map detail. How to create those king of connected normal map I will describe latter.

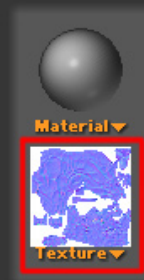
Interpolate/Raytrace – it is the way how the normal map is generated, interpolate is faster but the quality is much lower than in the case of raytrace, but raytrace is much slower. Those options are useful when we are testing something and we do not want to loose too much time for it. We can also use Render RGN option (render region) - it allows us to generate normal map, only in a small regions that we choose.

Create normal map – it starts the normal maps baking process.

Other settings in expert pass 1 and 2 we will pass over, because the presets that we use will setup all those parameters for us. But we will take a look on one more parameters in MISC section – **Raycasting Max Scan Distance**. Sometimes after baking normal map we will get the result as shown below:



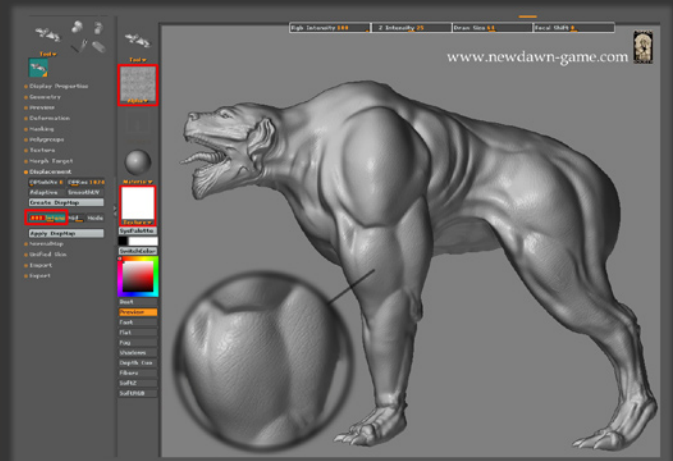
It's because the high poly deformation is so different from the low poly geometry, that the automatic projection can not handle it – the Raycasting Max Scan Distance parameter is set to low. So we need to change the value of it a little bit, till we get the desired result.



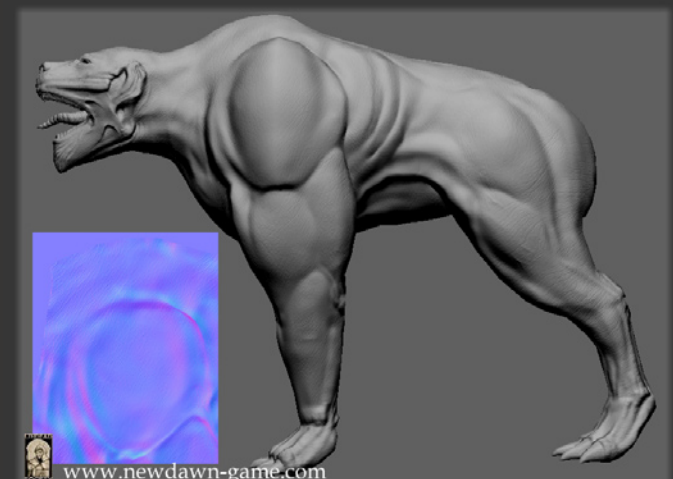
After we have baked the map, hit escape and leave the ZMapper. Our normal map is shown in the TEXTURE slot.

Connecting the normal map with bump map, using the ZMapper plugin.

Before we turn on ZMapper, we must import (or load) our bump map as an alpha map (bump map must be an RGB map). To see our alpha map as a bump map we must go to TOOL>>Displacement and enter the desired value in the intensity slot. We must also create a blank texture to see the bump map in viewport.

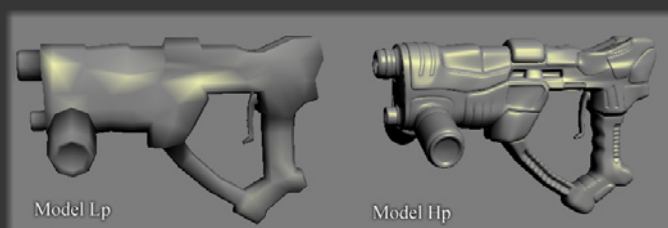


The intensity value should be something around 0.005, or a little bit higher, if we want. In screen above, the model is shown in his highest subdivision level to better see the bump map effect. Now we can set the subdivision level to minimum and turn on ZMapper. Below you can see the result of connecting the normal map with bump map.



6. Baking maps from high polygon models using projection.

Baking normal maps using projection is useful when the part of high poly model was made in other program than the ZBrush, like it is shown in the example bellow. In this case, the low poly and the high poly model was done in 3ds max

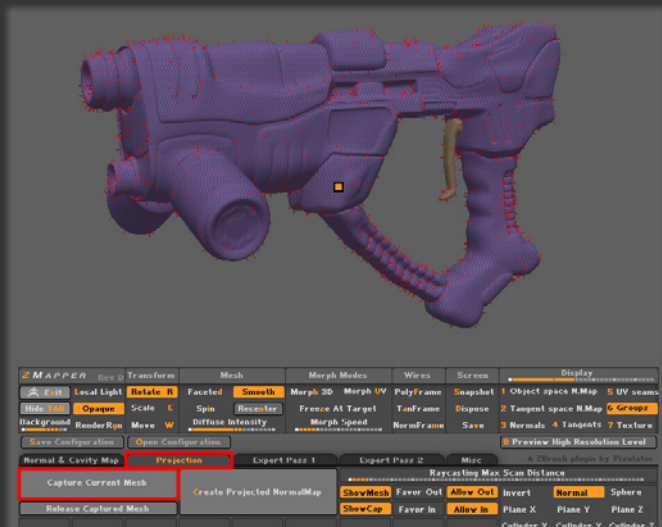


STEP 1.

We export our high poly model to a obj file, and then import it to ZBrush, where we can add the amount of polygons and sculpt it a little bit more. Our exported high poly model must not be mapped to bake a normal map from it.

STEP 2.

If we have divided the model (DIVIDE) after importing the model, then we must go down with subdivision levels to minimum and turn on the ZMapper.



We choose projection and then click CAPTURE CURRENT MESH. We leave the ZMapper by hitting the esc key.

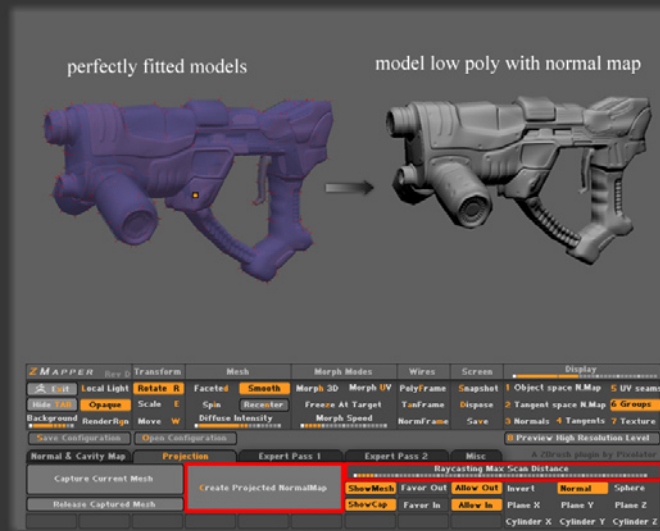
STEP 3.

Turn off EDIT MODE and then import our low polygon model. It is necessary to change the tool to anything that is not polymesh, and clean the canvas by hitting ctrl+n. This time low polygon model must have mapping coordinates before we export it to obj format from 3ds max or any other 3d software.

STEP 4.

Normal map baking

After importing low poly model we must go to EDIT MODE. We chose the resolution of our normal map (TEXTURE >> width and height then click NEW), then turn on ZMapper.



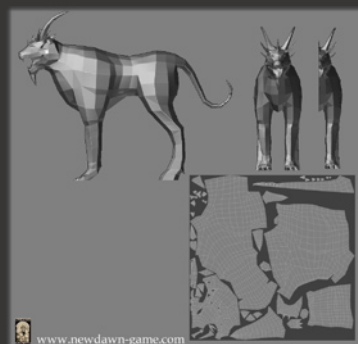
If the export parameters are set like in point 4, and the model have the same orientation of pivot points, both models should perfectly fit one on the other. If not, we must correct the pivot point of one of the model in 3dsmax, and then repeat step 3.

Next we open desired settings (open configuration) and click **Create Projected NormalMap**. After the map is baked we leave the ZMapper by hitting escape and the normal map will be displayed in the texture slot.

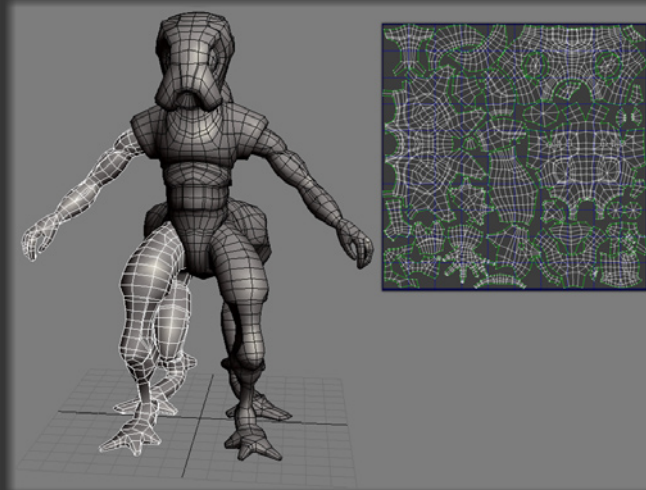
7. Baking normal maps with symmetrical mapping coordinates.

To save a texture space, it is good to use symmetry mapping, but some graphic engines do not display normal maps properly with those kind of mapping. In this section I will show you how to bake normal maps with symmetry mapping on our low poly model. First, a few words about what we understand with complete and partially symmetry mapping.

Symmetry mapping is complete when we have 100% symmetrical model, which can have fully symmetrical mapping like the model below:



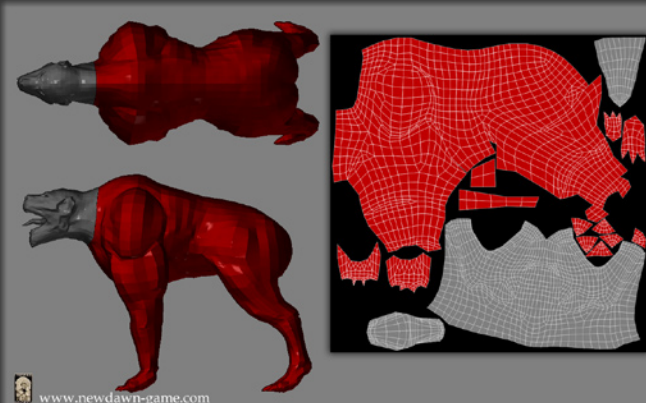
Symmetrical mapping is partially when the model is not fully symmetrical, or it is, but we don't want to have symmetrical texture on the entire model. Like on the screen below – symmetrical mapping on the torso doesn't look so good, so the symmetrical mapping is only on hands and partially on the legs.



In case of model with symmetrical mapping, we can use either standard normal map baking or use projection - **standard normal map baking**.

STEP 1.

The process of preparing a model for normal map baking is the same as in point 5a, till step three. But when checking mapping coordinates by using UV check, symmetrical uv elements will be shown in red. It's ok this time, because in case of symmetry mapping the UV elements are overlapped.



This model is partially symmetrical (only the torso has symmetry mapping).



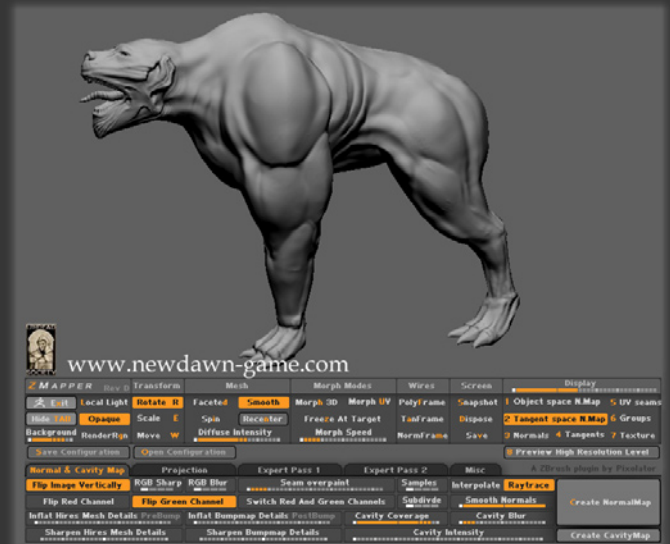
STEP 2.

Next we hide the part of model which has symmetry mapping – in this case half of the torso, and then we turn on ZMapper.

STEP 3.

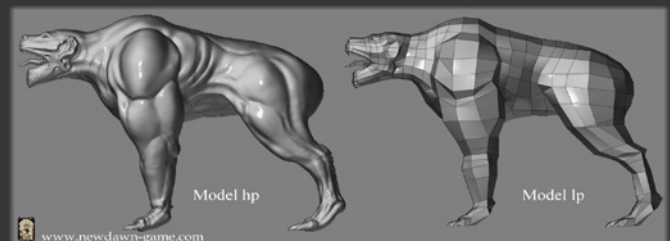
Baking Normal Map

Here we do like in point 5a – we choose the proper settings and then we generate the normal map.



Baking normal map by using projection.

In this case I will use the model from previous point, but the projection mapping we will do on the model with less polygons, than the previous one.



STEP 1.

First we open or import the high polygon model into ZBrush. In case of imported high poly models which don't have subdivision levels, we can use **RECONSTRUCT SUBDIVISION** function. If the imported model contains more than 250 thousand polygons, the ZMapper will not run and we must use **RECONSTRUCT SUBDIVISION**.



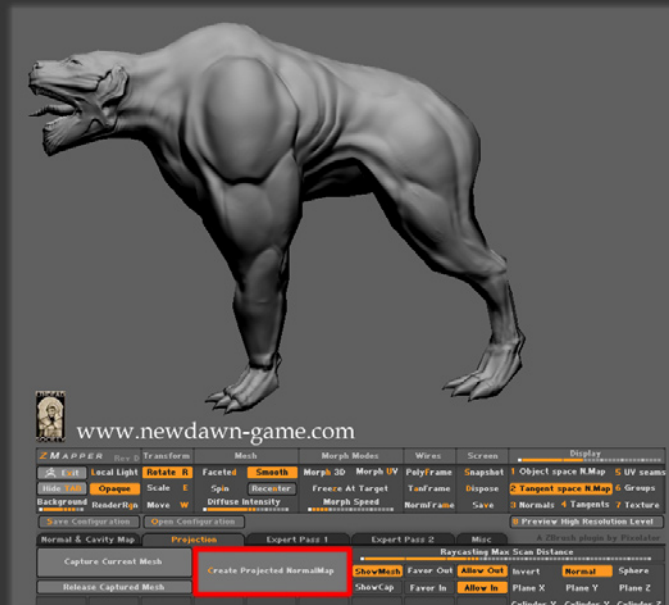
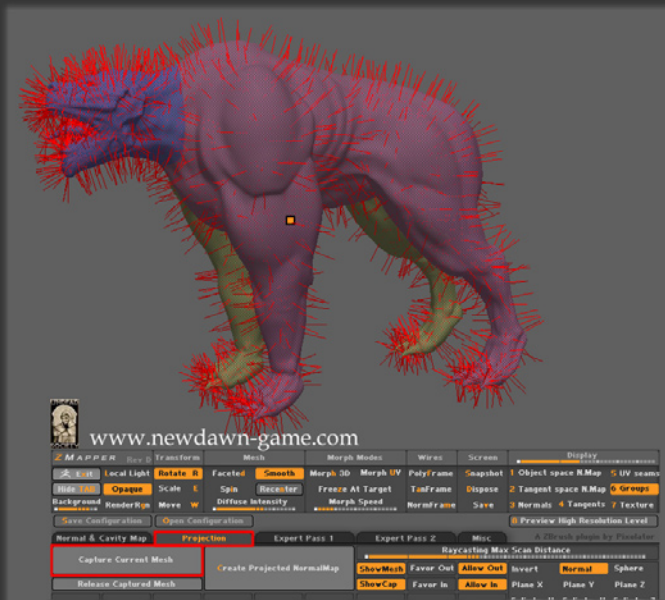
STEP 2.

While in EDIT MODE turn on ZMapper, and like in point 5a step 2, we go to projection and click capture current mesh. We leave ZMapper (ESC).
Even that the low poly model will have symmetrical mapping, we do not hide any part of high poly model.

STEP 4.

Baking Normal Map

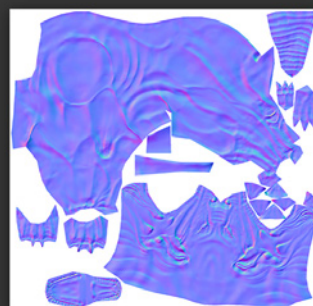
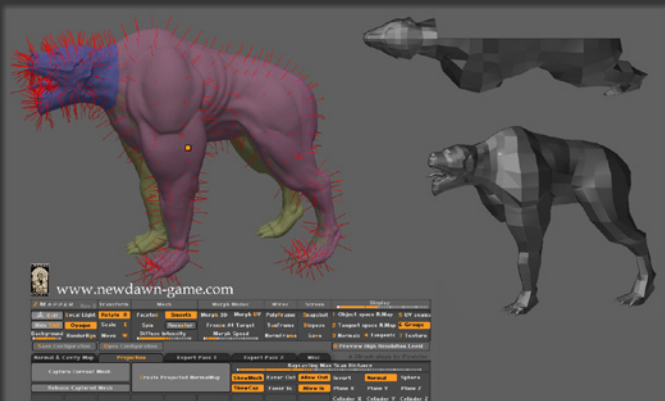
Now we hit Create Projected NormalMap and wait till its done.



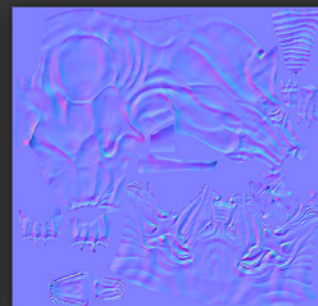
STEP 3.

After leaving ZMapper, turn off EDIT MODE, switch to non polymesh tool, and import the low poly model with symmetrical mapping on it.
Next we hide one part of the model , like in previous point (step 2), and then turn on the ZMapper again.

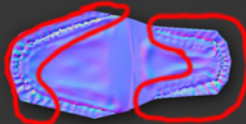
8. Correcting and detailing normal maps.



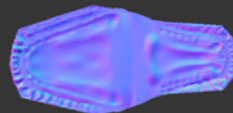
normal map generated from high poly mesh



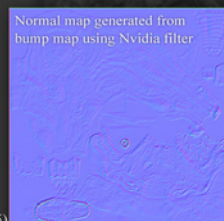
background filled with neutral color (RGB 128,128,255)
bump map created from color map



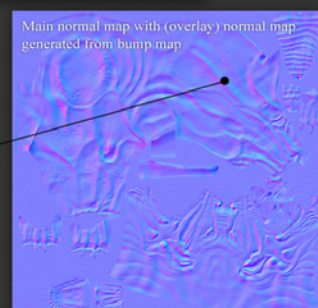
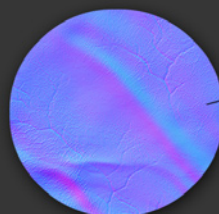
baking errors



Corrected baking errors.
(overpainted with neutral color RGB128,128,255)



Normal map generated from bump map using Nvidia filter



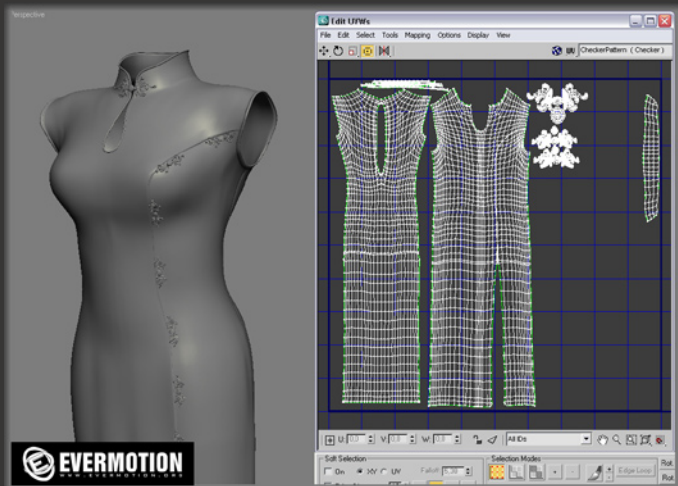
Main normal map with (overlay) normal map generated from bump map



Even our normal maps are well done in many cases, and low polygon models are good prepared, sometimes on baked normal maps we can see small baking errors in a form of red spots. In those cases the best way to deal with those errors is photoshop correcting. Below I have shown step by step how to repair our normal maps. Beside the error correction we can add some nice details to the normal map. To achieve this I will use a free Nvidia plugin which can make a normal map from any RGB map.

9. Hand made normal maps in photoshop.

Sometimes we want to add so small details on the normal map, that sculpting it in Zbrush could take to long. In this case the photoshop and Nvidia normal map plugin will do the trick. How we can use this Nvidia plugin I will show by making fine ornaments on the cloth. First and very important thing is the proper UV mapping. In case of small ornaments every deformation of UV map can be seen.

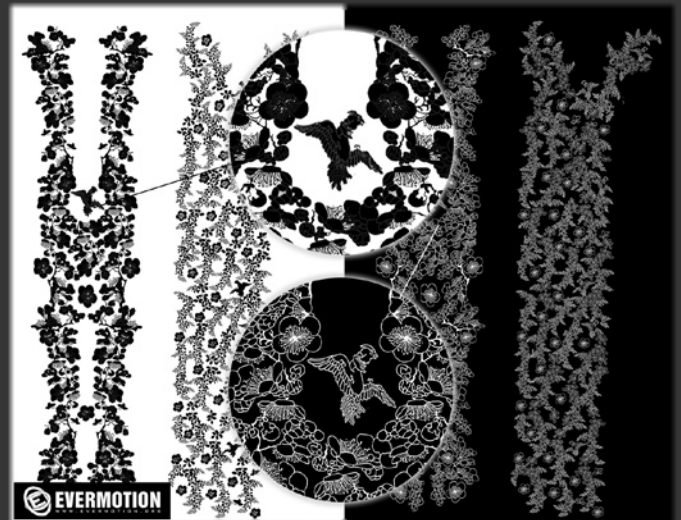


Next step is creating the pattern, by hand or by composing it from partially finished patterns. We can also use Zbrush and his texturing tools for making the patterns. Using Zbrush will also help us if the UV mapping is not too good. But in this case I will use only photoshop, and in most cases I do or use Zbrush only for seams correction. Below you can see finished mask, which will help us to create normal map, specular and even color map.

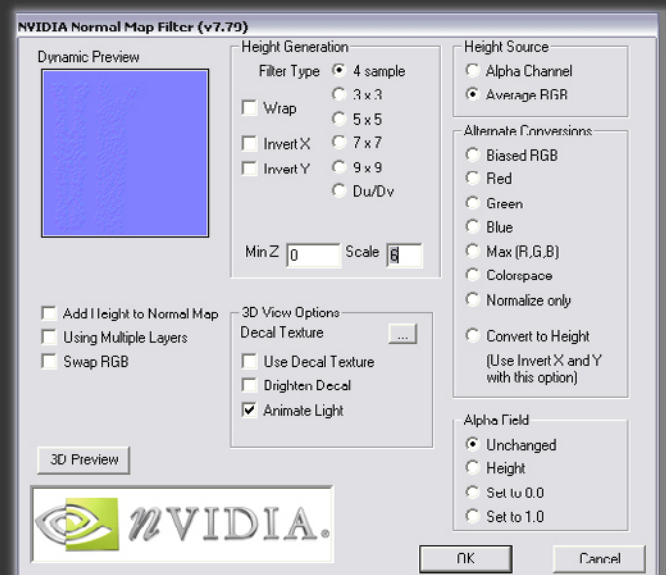


The look of normal map hardly depends on the look of our mask. I recommend to do some experimentation with the color of pattern elements to achieve the result we want – just make a piece of pattern, convert it to normal map and check how it looks in a shader.

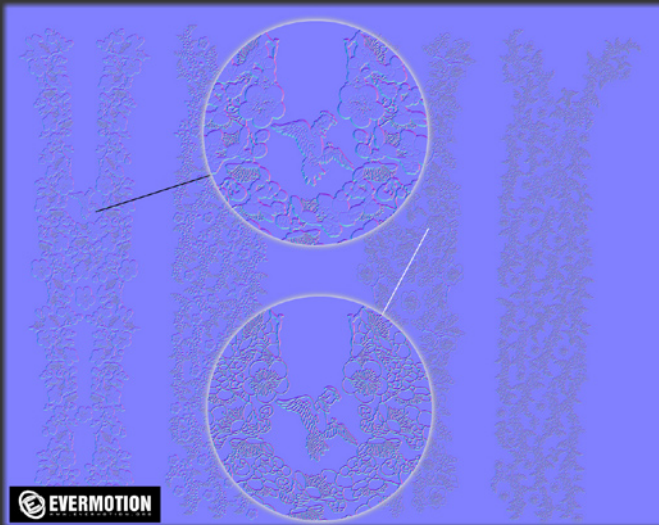
Below you see two similar masks that I will use to create one normal map. The mask on the left will bring out the main pattern, and the mask on the right will sharpen a little bit the edges of the main pattern and put more details within it.



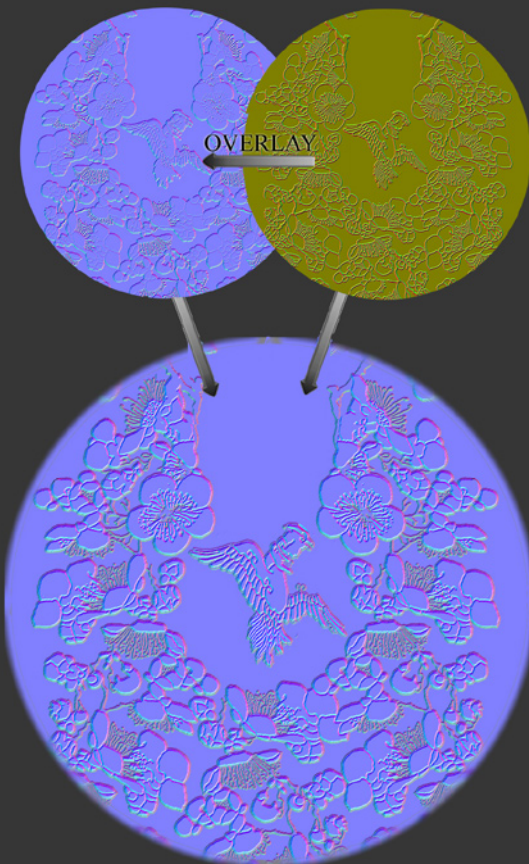
Next I've used a Nvidia normal map filter on the two masks with a default settings, only one thing that I've changed is the scale of the normal map deformation. In most cases the default setting works fine.



Below you see both generated normal maps.



The last thing to do is to connect two normal maps – I just put the right map on the top of the left map using overlay option in photoshop layers. The map on the right has been inverted (invert CTRL+i) to achieve the right effect. Below you see small piece of final normal map.



Using the same technique as above I have added more details on the map, and finally connected the pattern normal map with a normal map generated in zbrush from a high poly model (model with some cloth wrinkles). I have also used my masks to create the color and specular map which you can see below.



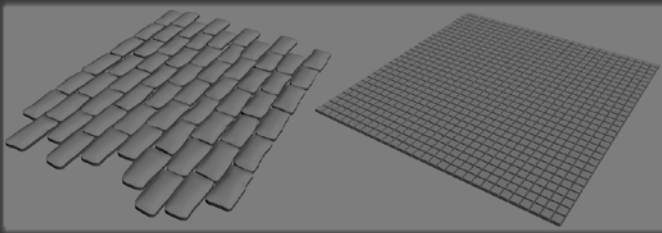
Final effect with color + normal + specular map.



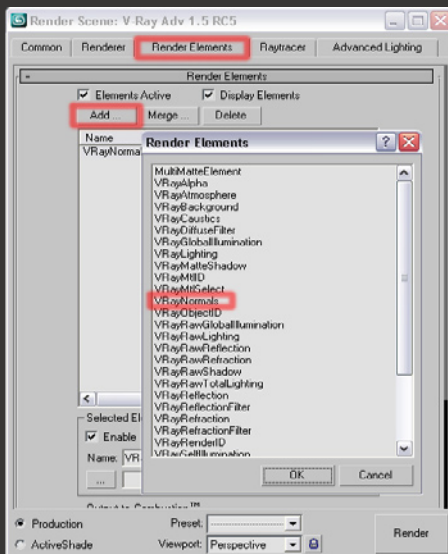
10. Baking simple maps in Zbrush and 3ds max using Vray renderer.

Mostly, normal maps are used in organic modeling and for detailing other kinds of models. We also can use normal maps for environment texturing, such as walls, ground etc., not only in game environments but also in architectural visualization, by creating tiled normal maps of bricks or stones for example.

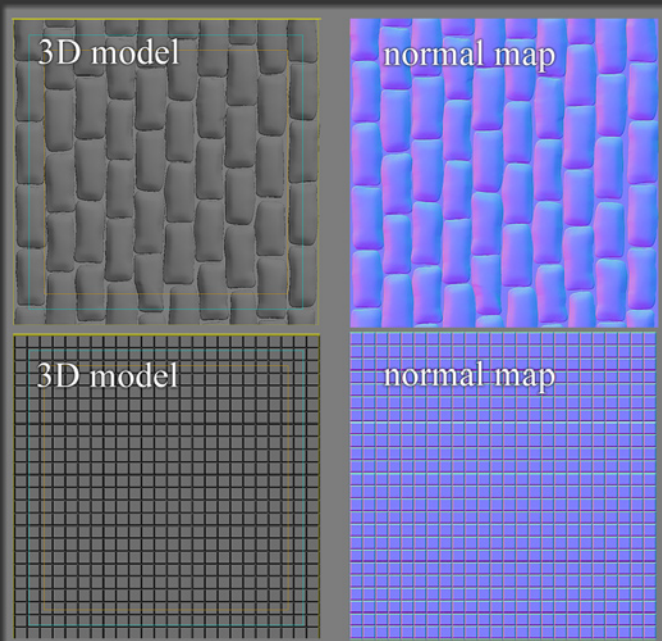
At beginning we will bake a simple normal map from two models which you can see below. I will use V-ray renderer to bake those maps.



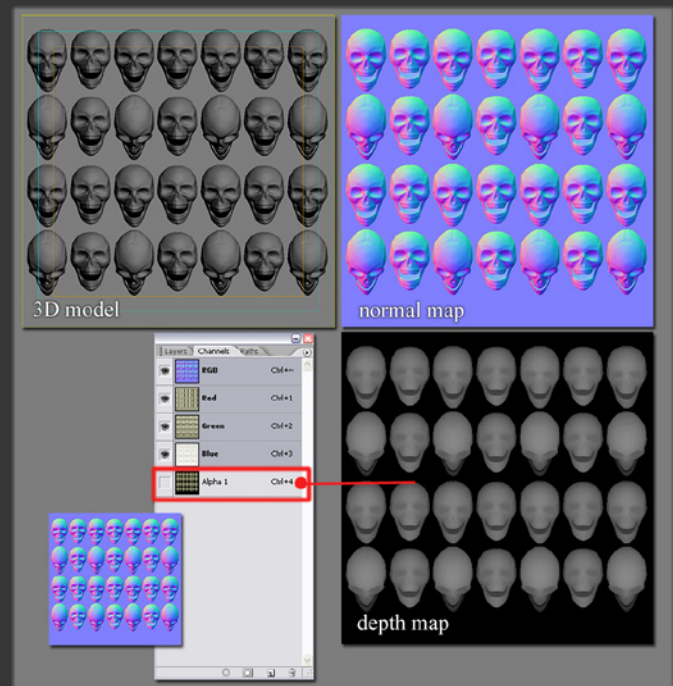
First, we must choose the Vray renderer in render scene window (F10). Then, in render elements we click add and choose V-Ray Normals from the list.



Now go to the top view, choose the resolution of the render (512x512 for example), and hit render. Below you see the final normal maps.



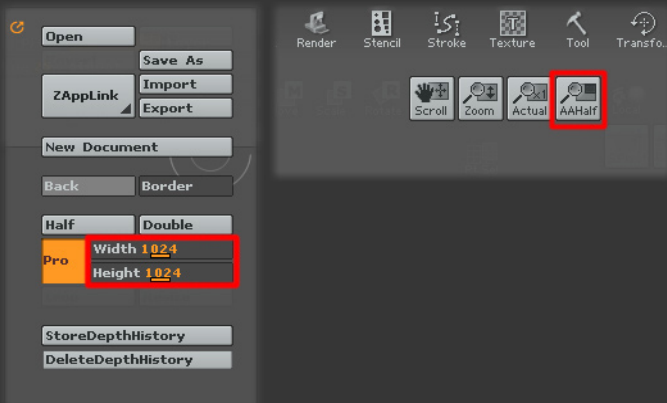
This technique we can also use on more complex models, baking besides the normal map also the depth map (VrayZdepth), which we can put in alpha channel of a normal map to achieve more realistic look of normal map's depth..



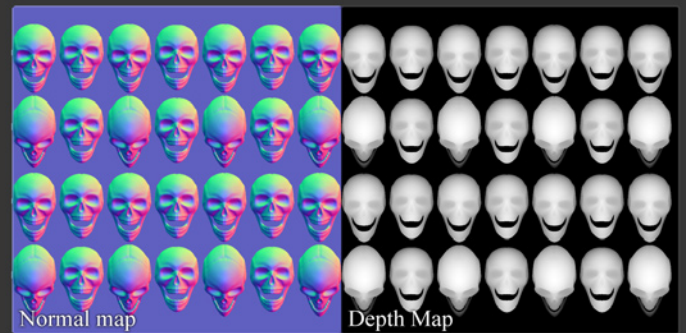
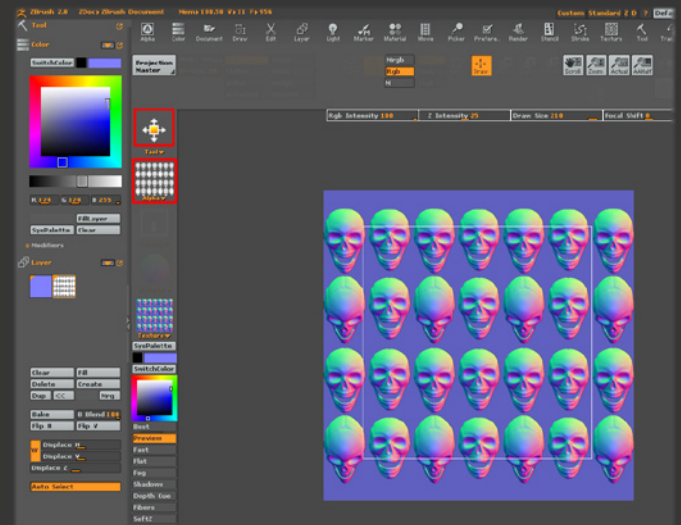
Below you can see the comparison of normal map without depth map (on the left) and with it (offset normal mapping). For displaying the normal map I've used "DX display of standard material" option in material editor. For displaying the offset normal map I've used a free shader from Ben Cloward site (http://www.bencloward.com/resources_shaders.shtml). Very important thing : the offset normal mapping needs a color map to display properly.



The same kind of maps (normal and depth) we can create using Zbrush. First step is to setup the resolution of our final map. In document settings we must change the width and height value – this time I've change it to 1024x1024 but the final resolution I want to achieve is 512x512. The reason I've created 1024x1024 document is that I will use untypical aa option – antialiased halfsize (AAHalf) which shrink's the canvas for times.



Next step is to make a background for our normal map – go to COLOR and fill in (R128,G128,B255), then in LAYER fill the active layer (FILL). Now create a new layer (CREATE) and put (or import) on this layer our 3D model. In this case the model are skulls imported from 3dsmax. It often happens that the model partially goes through the background (layer behind), then we must correct the depth of the layer by changing the DISPLACE Z value by 100 for example. WARNING, by changing the depth of the layer, EDIT MODE automatically goes off, and will not be able to edit the tool (model) anymore. Now all we need to do is load the proper material, which is not in material editor, but you can load it from ZBrush material folder (ZBrush2Zmaterials). The name of the material is NormalRGBMat. If for some reason the EDIT MODE is off, we must add the material to our 3D model by hand, using brush tool with Zadd turned off, and M turned on, and remember to paint on the right layer.



Next we do the same as previously, with maps generated on 3Dsmax – we put the depth map in normal map's alpha channel.

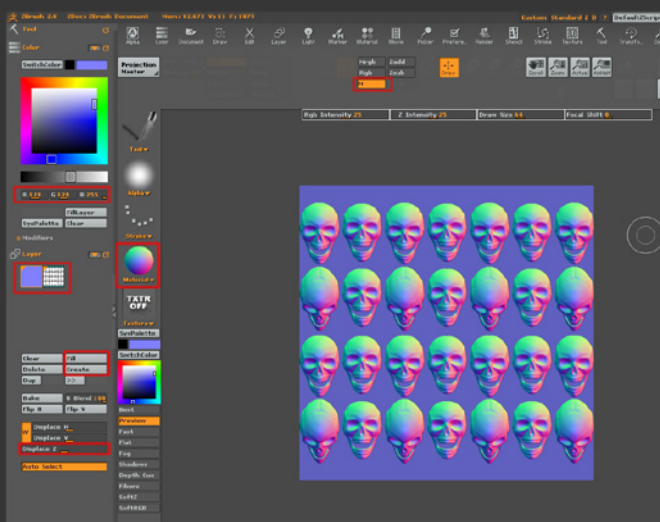
11. A few tips at the end.

a) Symmetry mapping

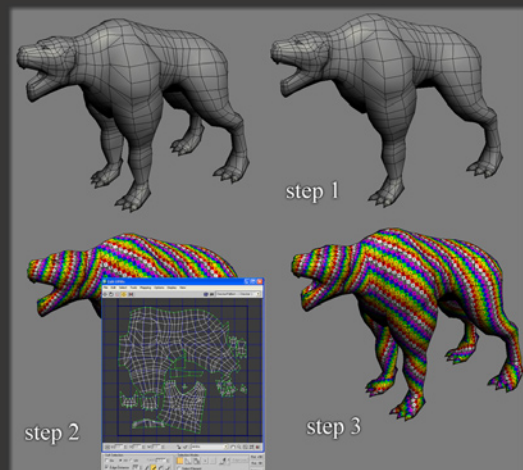
In some points above I had mentioned a symmetry mapping completely and partially. In next few points I will show you how to do those kind of mapping in 3dsmax.

– completely symmetrical mapping

The technique of creating symmetrical mapping is really easy and similar on both cases.



Our normal map is ready to use, all you need to do is just to export it – go to DOCUMENT and EXPORT it. Now we will create a depth map. To do this go to TOOL and chose MRGBZGrabber from the list, and simply drag over the canvas. We will get an alpha map with our depth map.



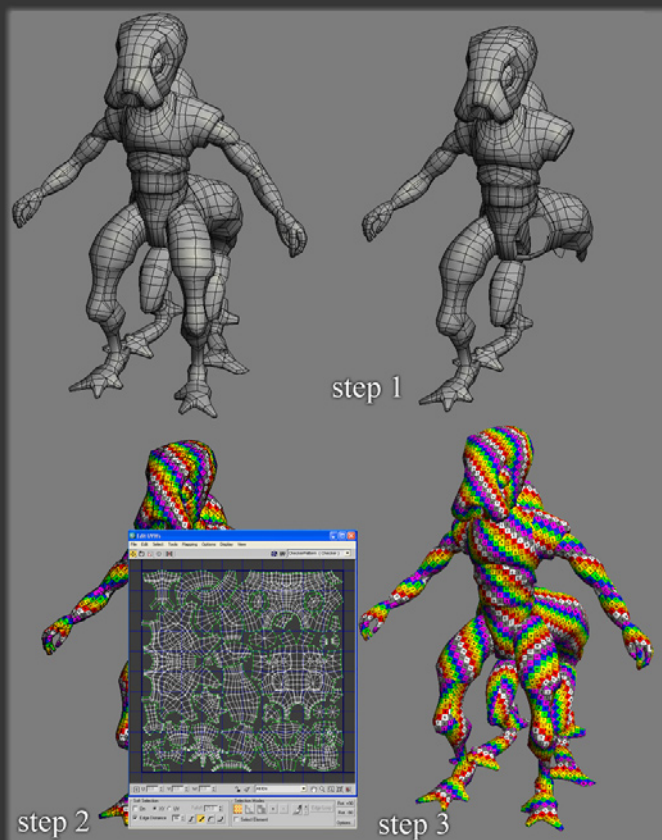
After the modeling process has been done, we remove half of our model (if the model is entirely symmetrical of course) **(STEP 1)**

Next, we unwrap our half of the model, using UNWRAP UVW modifier **(STEP2)**.

After the mapping is done we choose the SYMMETRY modifier from the list with proper parameters (x, y or z), and it's done **(STEP 3)**. Sometimes we must correct the symmetry point (or pivot point) and delete some vertices which were created here and there.

- partially symmetrical mapping

In this case we only remove those parts of models where we want to use symmetrical mapping – in model below arms and legs **(STEP 1)**.



Next step is to unwrap the model using UNWRAP UVW modifier **(STEP 2)**. Now we detach the parts, which will have symmetrical mapping (hand and leg), and use SYMMETRY modifier on it. The last thing we need to do is to attach the hands and legs to the rest of the model **(STEP3)**. Remember to set the pivot point in write place before using Symmetry modifier.

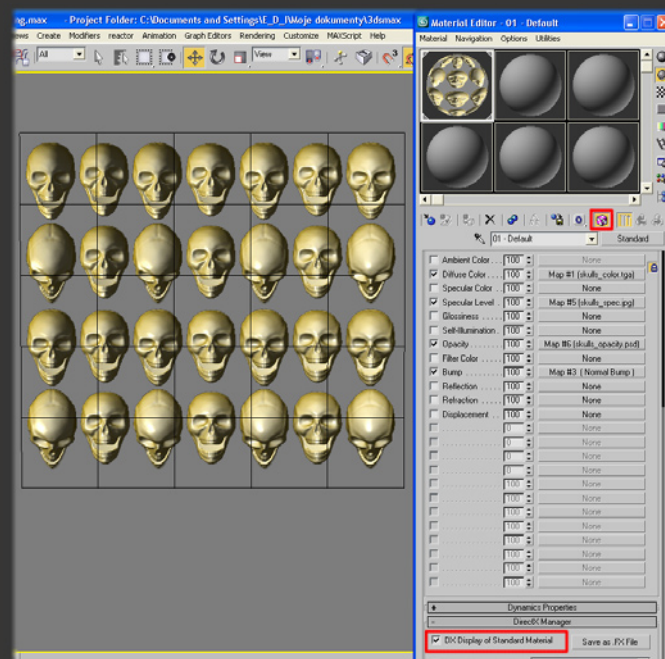
b) Displaying real time shaders in 3dsmax viewport.

In most cases we use normal maps for low poly game models which will be used in real time 3D game engines. In 3dsmax we also have the possibility to display shaders with normal maps in real time in viewport.

There are a few possibilities to display shaders with normal maps in 3dsmax viewport. The first one is the easiest - we use standard material.

For bump map slot we choose normal bump from the list, then we load our normal map in the right slot. Next we choose the kind of our normal map: tangent of object space. Value of bump map is set by default at 30, and when rendering this value do not affect the normal bump effect, but when using real time shaders in viewport, this value will affect it – so it's best to set it to 100. We can also add some color and specular map.

To display our standard material as an DX shader we go to DIRECT X MANAGER in material editor, and turn on the DX DISPLAY OF STANDARD MATERIAL. To see all the maps you also need to turn on SHOW MAP IN VIEWPORT.



Other way to see real time shaders in viewport is to use Direct-x shader material. Just click on standard icon in material editor and choose Direct-x shader from list. Within the Direct-x material we can choose one of the 3dsmax's standard shaders (FX files) or load one of our own shaders. In this case I will use a free shader from Ben Cloward's site (http://www.bencloward.com/resources_shaders.shtml). In different kind of shaders we have different kind of maps slots. In this case we have a color map slot and normal map in uncompressed and compressed (DX5 – files in .dds format) file format. All maps are displayed right after load.

