Back to the Root Page of The No-Joke Topology Guide For Serious 3D Modelers

Revised / Modern (Base) Game Modeling Topology

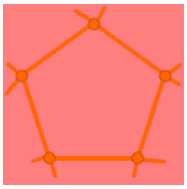
The new convention about topology in game modeling has been modified greatly. Though it has lost some of the restraints it had in the original, allowing the artist more visual freedom, with the growing popular demand for better graphics and better general visuals in a game, as well as excelling video-card technologies, firmware, and other aspects of hardware pipelines, a lot of animation-related restrictions have now come into play.

Now, game graphics are slowly catching up to animation graphics, as well as the solely motion aspects in games (in-game animation.) Entertainment-wise, games (console games as well as computer games) have one undisputable advantage before an animated feature / film which is most unlikely to ever change. That is **interactivity**. Interactivity tends to attract people, get them more interested, sometimes "suck them in," but it still does not come without cost.

Since game engines are interactive, they use a physics engine, which should work hand-in-hand with the rendering engine when the game is being played. This constitutes that physics algorithms (namely, collision algorithms) have to work on objects made with some rules.

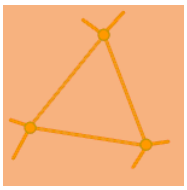
The constraint on face convexity is long since removed, since the physics engine will choose the optimal algorithm for a given dynamic object upon import of this object. If the object is convex, the optimal convex algorithms will be used, if not, well - the concave algorithms, which can still be handled well by modern computers. There are, however, still problems with holes. Most engines will make an object with holes (especially dynamic) look funky. Other ones simply will not accept them, returning an error when trying to import. Covering holes up may work sometimes, but, due to polycount restraints, it is in your interest to maintain a single-layer surface on all of your geometry. High-quality collision algorithms require surface of an object or an element within that object to be continuous, with no breaks, because they do not "know" how to make things collide with holes in the geometry.

There still are problems with high polycounts. This time, however, the poly-cap requirements for a half-decent new engine at least doubled since Quake III, for better engines tripled and quadrupled. This means that 2-4 times more polygons can be put into a model. You should still keep count. Good figures to go off are: 1,000-2,000 quads (x2 triangles) per simplistic character, 2,000-3,000 per detailed humanoid character, 3,000-10,000 per an advanced, very detailed, or very big character that only appears in quantities of one or two at a time in the game. This makes triangle counts 2,000-4,000; 4,000-6,000; and 6,000-20,000 respectively. Level/map/static-object polycount should be monitored, but in a different way - make sure that your target game platform can perform at a reasonable rate (45-60 fps) at any point in playing the level.



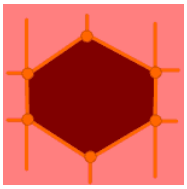
N-gons

As in animation topology, N-gons generally tend to screw up your character's look when moving, no matter how far away they are located from the joints.



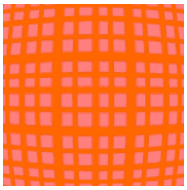
Triangles

Triangles may somewhat help to keep the polycount down to the minimum. They are also good for times when you need some sort of sharp "lip" on a rigid shape. However, keep in mind that triangles can only be used in such static areas where you are sure that the vertexes defining the triangle will not move with respect to each other during the animation. These areas are never on the joints / grace-lines.*



Holes

As already described, no holes, no covering up of holes, no holes PERIOD**.** Covering them up will blow your poly cap and screw up your texture / detail baking. They don't work with collision either.

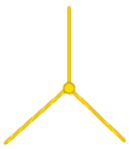


Dense / High Polycount Areas

Medium polycount is usually optimal for animation. Only a big company with a large-scale production can afford a network render with more than 16-24 mental ray render nodes (2-3 seats) that can handle real high-res geometry with good lighting at decent-enough rendering times.

5-or-more-edged Vertexes

Vertexes that are surrounded by (define one end of) five edges are okay to use in the modern days game topology convention, as long as they don't mess up your grace-lines.*



3-Edged Vertexes

These are also ok to use with this convention. Again, watch your grace-lines.

* Grace-lines are described in greater detail in the **Animation Topology** description, in the **Triangles** section.

IMPORTANT: this new game modeling topology convention is widely accepted today, though many modelers, especially those who model for animation, stress to minimize or even abolish the use of triangles. Nonetheless, Kevin Lanning, one of the greatest Gears of War game modelers from Epic, continues to use triangles and comes up totally mind-blowing high-quality models. His models look great both static and dynamic. Keep in mind that, though he has triangles, they only appear on the hardest, most unyielding surfaces which do not shape-shift during animation, like certain parts of the boots, helmets, and, sometimes, non-moving skin regions (where skin meets a bone or skull.) Though his models are deemed so great, Kevin's methods also have their drawbacks: check the (Strict) Quad Topology convention description for details.

Back to the Root Page of The No-Joke Topology Guide For Serious 3D Modelers