Optimization: GPU Budget Intro

For pre-rendered 3d-graphics, rendering a single frame can take anywhere from a few minutes, to a few hours, to a few days. You can imagine, then, that rendering a scene at 60 frames per second comes with some significantly limitations. The key to making impressive realtime graphics are learning the limitations of your system and being able to work economically within them. Distributing these limitations across your scene is referred to as budgeting.

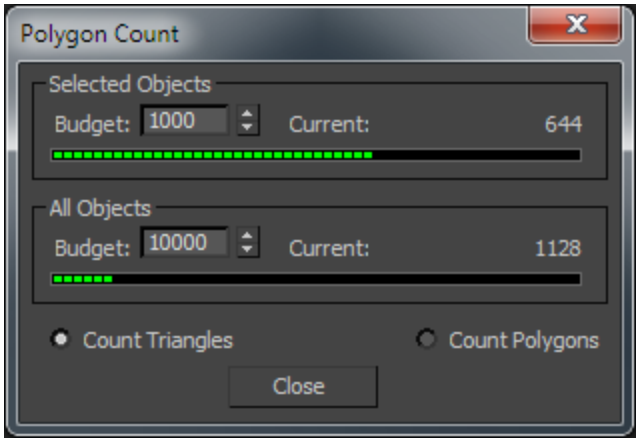
Common limiting factors:

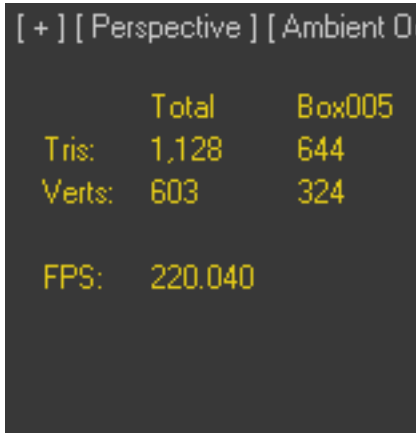
1. Texture memory footprint of all loaded texture images
2. The number of triangles on screen
3. The number of separate objects (drawables)

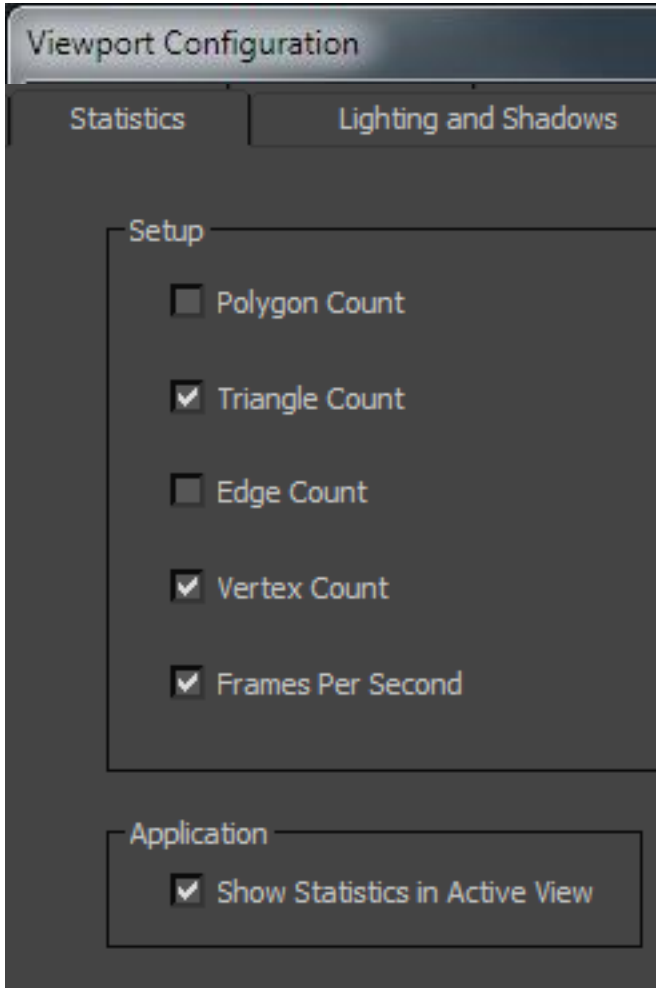
Less common limiting factors:

1. Texture call overhead
2. Character animation overhead
3. GLSL Shader overhead

See the second page for brief descriptions of each of these factors.







WORLD VIZ®

Notes

- To check your usage in Vizard, load a model and press F4 twice. This will bring up a box showing your framerate, primitives (triangles), drawables, and the amount of system and GPU memory being used.

- Texture Footprint: The uncompressed size of texture maps in GPU memory. Check you graphics card specs to find out how much memory it has, or press F4 twice within Vizard to see your capacity and your usage.

- Triangles: AKA polycount, AKA primitives. All “mesh” objects are made up of triangles. The power of your graphics card limits how many you can quickly process at any given moment. A modern graphics card can easily handle 1 million tris.

- Drawables: The number of individual objects, further split up by the number of separate materials on each object. These have much more overhead. Try to keep the number under two thousand.

- Texture call overhead: Each new texture that has to be processed also has it’s own overhead. Depending on distribution, it can also increase the number of drawables. This will usually be a smaller issue than the other factors.

- Character Animation: [need to test this out more - # of bones, # of verts driven by bones, # of avatars. Using a lot of avatars can slow down a scene to the point where occluders and other tweaks become necessary. Using lots of animations also can take a lot of ram and greatly increase load times]

- Shader Overhead: shaders are programs that control the surface properties of objects. You won’t need to worry about this one unless you are writing your own GLSL code.