The Graphics Pipeline **CS 147** **Dominic Sherman** **Sept. 11th**

The graphics pipeline consists of 6 main steps. It begins as a number of arrays of vertices for the points of the primitive object in 3d space, and other attributes, such as color or texture. These arrays of vertices, called vertex buffers, are then run through the vertex shader in the second step of the process. The vertex shader is a program that takes in these vertex buffers as input and does something to the to produce an output of values that will get passed to the rasterizer. It could simply find the position that is projected for the vertex on the screen, or it generate a number of other varying outputs, such as color or texture. After the vertex shader, the output goes through primitive assembly. It connects the vertices that were projected by the vertex shader, according to the type of primite object it is assembling. If it is assembling a triangle, it simply connects groups of threes and a face in between them. If it is assembling a triangle strip, it knows to use the last two vertices of the previous triangle as the first two of the next one. For a triangle fan, it connects the first element to every other vertex that comes after it. The result of this is then passed to the rasterizer. The rasterizer is where the pixelation occurs, and where graphics outside of the scope of the screen are discarded. The vertex shaders outputs are spread across the rasterized surface so that the so that each primitive object is assigned a smooth gradient of colors. The pixels, or fragments, generated by the rasterizer are then passed on to the fragment shader, along with the varying outputs that the vertex shader produced. The fragment shader, (or fragment processor or pixel processor) calculates depth and color values for the pixels. It has a number of operations it can do, such as texture mapping and adding lighting. The fragment shader does not look at the graphic as a whole, but rather looks individually at each pixel, which allows it to generate a multitude of special effects. Because of this though, it is the slowest part of the process and the most sensitive to affect the performance of the graphics. After the fragment shader is completed, the results are then finally sent to the framebuffer. The framebuffer is the final step of the graphics pipeline process. A framebuffer contains more than just a 2d image, and most modern openGL implementations allow the programmer to make framebuffer objects. A framebuffer has at minimum one or more color buffers, but it can also have in addition to those a depth buffer and a stencil buffer, which filter the pixels before they are drawn to the framebuffer. The frame buffer also performs alpha blending, which looks at each pixel and calculates the final depth, stencil, and color values that will be drawn into each buffer.

Sources:

http://duriansoftware.com/joe/An-intro-to-modern-OpenGL.-Chapter-1:-The-Graphics-Pipeline.html

https://www.geeks3d.com/20110704/3d-graphics-pipeline-explained/