Michael Scott

Graphics Pipeline

The purpose of graphics is to display 3D polygons onto a 2D surface/screen. This is done efficiently by using the graphics pipeline. The pipeline is a step-by-step process which takes shader attributes and shader uniforms as inputs and outputs an image. It used to be that no part of this process was programmable by the user, but these days there are some parts which the user can program which allows more freedom and customization. The order that this is done in is setting up the data, executing the vertex shader on each vertex, clipping to account for field of view, mapping the model onto pixels of a raster, determining which pixels are used to represent each polygon, executing a fragment shader which colors the pixels that are part of the shapes, combining pixel colors appropriately, and then outputting the final raster as an image. Every step of this process except for the first one is done by the GPU, which has specialized parts and its own memory. The first step is done by the CPU. The benefit of the pipeline is that all of the steps are run parallel, which means that it is generally faster. Depending on what kind of transformation is being done, there are different matrices that can be used to alter the data/points in the pipeline. Additionally, there are some tests that are done after the shader is done. For example, there is the scissor test, which basically creates a rectangle that defines the field of view for that image and crops out portions that will not be in the field of view, and therefore those sections are not rendered. Then there is the alpha test, which determines transparency of an object. It compares two alpha values and, depending on which is larger, determines which object is more transparent. Next is the stencil test, which uses an extra buffer to can be used to make shadows and reflections. After that is the depth test which compares the fragment depth to the buffer depth to determine if the fragment will be displayed. Finally, there is blending. Which combines the color of the fragment with the color in the buffer.

Sources:

https://shot511.github.io/2014-06-08-tutorial-04-what-is-programmable-rendering-pipeline/

https://en.wikipedia.org/wiki/Graphics_pipeline