Finley Blakeman

2.16.2021

Assignment 2

# The Graphics Pipeline

The graphics pipeline is the common process in graphics visualization of rendering images using program data that is turned into vector data. This vector data is then turned into a raster image which can be used to project the pixels seen on a screen (1.3 – The 3D Graphics Pipeline). There are two pieces of hardware used in this process the CPU (central processing unit) and the GPU (graphics processing unit). The CPU takes in the programming data and turns it into vector data. The way I like to differentiate between the CPU and the GPU is as follows: *We mess with what goes in the CPU. The CPU then takes our mess and tries to change it to a form that the GPU can read. Then the GPU is where the graphics pipeline takes place and creates an image using our programs and data.* These vectors are then sent the GPU for vertex shading, clipping and assembly, rasterizing, fragment shading, compositing, and finally drawing (Interactive Computer Graphics and 1.3 – The 3D Graphics Pipeline). A vertex shader is a type of program that is set up in the CPU that is then run in the GPU. It creates and stores the coordinates for each vertex. Clipping is the process of getting rid of data that would be outside of the graphics canvas. Rasterization is the process of determining the pixel boundaries for each shape (1.3 – The 3D Graphics Pipeline and Tim Urness' Class Notes). A fragment shader outputs a color for each pixel and creates the primitive shape from the rasterization process. Compositing is the process that blends colors of pixels from the fragment shading process (1.3 – The 3D Graphics Pipeline). Honestly, I'm not sure how stretch this to 350 words, I think I understand the process and how it works to the point where my explanation makes sense. However, there is a very real possibility that this explanation is so off course that I might as well be writing about a different class. I hope it is the former.