Reflection

When I began working on this project, I initially chose objects that I believed would be relatively easy to code and manipulate. This proved to not be the case when the first object we needed to created was the complex object in our scene. The complex item for my scene consisted of a sphere resting on top of a torus or cylinder. We were given a code snippet that allowed us to create both spheres and cylinders for the project, however it took me several days of trial and error before I was competent enough to complete the project. Once objects were created the texturing wasn’t terribly hard. The most complicated part of texturing was making sure I made my vertexes correctly so the image didn’t come out weird. Lighting proved to be the hardest challenge due to how easy it was to accidently break your program when messing with the shader program. It took me three days of trials before I was able to make a shader program that didn’t mess up all of my textures.

When creating the camera, I maintained the normal standard of “movement” using WASD. Panning the camera up and down would have been easy to understand on the arrow keys, however for the convenience of finger placement I used Q and E to handle vertical movement. The mouse callback functions are made to allow users to move the camera’s orientation with the mouse and scroll wheel, and using P will cause the scene to shift from a first-person perspective to more of a fixed camera view from a distance.

The only modular functions I managed to use in my programing are the processInput function and UCreateTexture. The processInput function is good for easy sorting and changing of various inputs that you want the program to be able to accept. UCreateTexture is a great function for quick creation of textures without a massive amount of repeat code. I would have moved more shape building and light programing into functions, however I had quite a bit of trouble moving those functions without inadvertently breaking my program.