Jacob Winters

CS-330

April 17, 2022

**Final Project Journal**

As a C++ and OpenGL 3D graphics developer for Triangle & Cube Studios, I have created a 3D version of a 2D image provided by my client. The 2D scene was a representation of a computer workspace, which includes several objects that I have recreated in my 3D environment. To begin my work, I needed something to represent a plane, or something that my objects would be placed around. So for this project, I chose to use a wooden desk as the plane, as it would serve this purpose while contributing to the overall scene. Furthermore, I added table legs to the desk to ground the object within the scene.

In addition, I decided to create a 3D representation of a computer monitor display, keyboard, and mouse. Each of these objects has high-quality custom textures designed in Adobe Photoshop, with accurate projections. As a small easter egg of sorts, I had the monitor screen texture based on an early revision of the actual source code of my OpenGL application. During the earlier design stages, I had considered additional objects, such as a laptop. I decided against these as I wanted to focus on achieving all of the project goals that I had set forth, without getting lost in the process due to an overabundance of object code.

In conclusion, I believe that my 3D scene does an excellent job representing a likeness to the selected 2D image. The chosen objects make sense within the scene, are placed in accurate positions, and are nicely-rendered representations of these real-world objects. In order to add realism to the scene, I included lighting sources that implement all components of the phong shading model, including ambient, diffuse, and specular lighting effects. The only portion of my lighting technique that I would have liked to implement differently was the monitor screen itself. I think that it would have been a very cool idea to have an actual reflective light source coming from the screen to add even more realism to my scene.

Moving on to user navigation, I made some slight modifications to the camera.h file in order to achieve several goals in terms of camera input. First, I wanted the user to be able to use the W, A, S, D keys for general movement, while also being able to use Q and E for upward/downward movement. The camera view can also change between a perspective and orthographic projection upon the input ‘P’. The mouse is capable of allowing the user to look around the scene. The scroll wheel input has been altered so that scrolling forward or backward will change the speed of the camera movement, which I found especially helpful in looking at smaller details within the 3D environment. The combination of keyboard and mouse input provides a comfortable way to navigate within the OpenGL application.

Lastly, there are some custom functions within my program that allow my code to be both modular, and organized. For example, the keyboard and mouse input implementation allows for me to make interesting changes as to how the user interacts within my environment. For example, I could add additional movement keys or remap the scroll wheel to a zoom function. While developing my OpenGL scene, I tried to maintain consistent and clean code for the mesh generation functions in order to make my code reusable. This helped by allowing me to easily create new objects without having to make substantial changes to my program. Overall, OpenGL provides some interesting ways to utilize C++ in order to create fun, visually pleasing results, and I look forward to learning more about it in the future.