Noah Sherry

SNHU CS 330

Final Project Design Decisions

February 21, 2021

A picture containing text, electronics, display

Description automatically generated For my project, a lot of consideration went into exactly what scene I wanted to recreate and how I wanted to use my knowledge of OpenGL to do so. The scene I chose included plenty of objects to comply with the project requirements, and some of the objects would require multiple primitive shapes to create. Namely, the lamp and the potted plant would require multiple primitive shapes to accurately create. Here is a copy of the image I attempted to recreate:

I initially had a lot of struggles with my scene. Trying to figure out exactly how to use/create multiple Vertex Array Objects and Vertex Buffer Objects to create multiple shapes was very difficult for me. Unfortunately, these difficulties occurred at the same time as the birth of my first child, so I did not have the amount of time I would have liked to have to finish this project. For these reasons, I have decided to omit both the computer monitor and the computer mouse from my scene, so that I could focus more on getting the texturing/lighting/objects right for my other objects. I also hade some difficulty getting the normal and texture coordinates to properly apply to the cylinders and spheres, thus they look not quite correct in my scene.

To navigate my program, the user can use the WASD keys to navigate forward, left, backwards, or right depending on the direction the camera is facing. Using the Q and E keys will allow the user to move the camera up and down. The user can also move the mouse to change the camera angle. To switch between a perspective and orthographic view of my scene, the user may press the P key. The user may also adjust the camera movement speed by using the mouse’s scroll wheel.

To make my program as clean and easy to understand as possible, I made use of separate functions to Create/Destroy the shaders, object meshes, and textures. The main render loop also exists within its own function in the program. For organization, I made liberal use of the #region tags within Visual Studio to organize each section of code for each shape. There are also separate functional callbacks for each input method.