Drew Pepin

Southern New Hampshire University

CS330

December 18, 2022

Final Project Reflection

**Justify development choices for your 3D scene**.

For my final scene, I used several primitive shapes to construct the objects necessary to build the lamp scene that I chose. To start, I used different sized cubes for the base, to give the layered look to the bottom of the lamp base. This gave the look of the layers being stacked on top of each other, which is what the lamp looks like. The side pieces of the lamp box were also using the cube mesh, with some translation and scaling to get the long, thin look for the four separate side pieces. To render the top of the lamp base, I used a pyramid texture to give the “roof” look of the lamp base. I went back to using cubes for the “stacked” look of the two different sized cubes that are on top of the pyramid. For the “tube” that connects the base to the light bulb, I rendered a cylinder that needed to be rotated and scaled to give the look of the “tube”. For the light bulb, I rendered the sphere mesh to get the look of the lightbulb as best I could. Finally, the lamp shape is made using a tapered cylinder to get the look of the lamp shade as accurate as possible. In terms of the textures used, I used a different texture for the table, the lamp base, the light bulb, and the lamp shade.

**Explain how a user can navigate your 3D scene**.

The user can navigate my scene using the mouse pad, which can allow the user to look around the scene from any angle. The user can also navigate the scene using different inputs. As the user inputs a specific key, WAS or D, they will move the camera forward / backward, and left / right. If you user inputs the Q or E key, the camera will move up or down. This allows the user to view the scene from any angle.

**Explain the custom functions in your program that you are using to make your code more modular and organized**.

To make my code more organized, I stored all my meshes in a separate file, to not overload a single file with indices. In meshes.h, I created all the primitive shapes I’d be using for my scene. In main.cpp, I imported the meshes.h file, and was able to render whichever shape I wanted to use from meshes.h by making a call to that specific mesh. First, I had to render specific mesh, by calling glBindVertexArray and making a call to vao within that function. Then, I draw the mesh, by calling a specific glDrawArrays function for that specific mesh. This method allows my code to be reusable, and can render any shape I want multiple times without excess code.