**Project Two**

**Design Decisions**

Jessica Ayer

CS 330 Comp Graphic and Visualization

Instructor: Kurt Diesch

08/14/2022

The 3D scene is a recreation of four glass objects resting on a flat surface. A sphere on a stand, a cube, a pyramid, and a candle holder. They are arranged so that the user can weave through the objects and view their three-dimensional forms from any angle. I selected these objects to demonstrate my ability to render various basic three-dimensional shapes using indices and vertices both from pre-defined vectors and from functions which generated them. There were many challenges associated with this project from drawing the shapes, to texturing them, to adding keyboard and mouse input, and to creating lighting. These challenges as well as gained insight as to how OpenGL works over the last few weeks led me to change some of my design plan from my initial proposal.

Originally, I had planned to use cylinders for the candle holder and a torus and cylinder for the sphere’s stand. In the end I made the decision to represent these shapes with cubes. I struggled construct a cylinder for the candle holder and import it into the main window so I recreated it with cubes. I became quite fond of the end results and though it’s not an exact replica of the picture I made the artistic choice to leave it as is. When building the sphere, I quickly discovered how many triangles a more complex object requires and the challenges of getting the object to render correctly so rather than go with my initial proposal to utilize so many round objects, I simplified it down to only the necessary sphere. Perhaps with more time I will gain better abilities to create round objects that can be created more efficiently but I also felt keeping it simple was a practical choice with the project criteria putting a cap on the number of triangles allotted for the scene.

Navigation tools were added to the three-dimensional scene to allow for a more interactive experience. I opted to build my own camera class to help with learning/understanding the functions and used a combination of techniques from *Learn OpenGL* by Joey de Vries and *Computer Graphics with Modern OpenGL and C++* by Ben Cook. One can use the mouse and keyboard to move the virtual camera, or one’s point of view all around the scene or travel through it. The user can use the W and S keys to move forward and backwards on the z axis replicating a zoom function, A and D keys to move left and right on the x axis panning around the scene, or Q and E to move up and down on the y axis. The O and P keys can be used to change the view angle between an orthogonal view and a perspective view replicated and 2D vs 3D view. The mouse can be used to keep the scene within the viewing window while moving the camera around with the keyboard.

Multiple textures and lights were added to the scene to give the objects their own unique visual appearances. I found it challenging to find textures that matched my vision and rendered properly so I used GIMP to create several of my own. Due to all of the objects being glass, I gave them a shiny material finish whereas the floor and candle were given a dull, less reflective surface. Directional, point, and spot lights were used to give the scene a finished look and make the objects stand out. There is one directional light to give the whole seen a base lighting, three-point lights to make the candle holder glow and the cube pop, and three spot lights to add a flame to the candle, and some color accents to the pyramid and sphere.

After the first few weeks of this course the code started to get lengthy and scrolling through it to work became cumbersome so I followed the *Computer Graphics with Modern OpenGL and C++* video tutorials which showed me how to “clean up” the code and separate the functions into different classes. In some ways using Cook’s techniques made things more challenging because I had to re-configure everything we learned in this course, however in other ways Cook’s format made everything easily reusable. All of the shapes I created can be easily multiplied by using the push\_back function to add another object into a mesh vector. I can add and remove as many shapes as I like without specifying how many VAO’s or VBO’s are needed. Overall, I really enjoyed the opportunity to learn graphics. There is still a lot of knowledge to be gained but I feel the base code I have created is a solid foundation that easily be re-used and expanded on. With each piece being sorted into separate classes it is quick to navigate the code and make minor adjustments as needed.

**References:**

Cook, Ben (March 2020). *Computer Graphics with Modern OpenGL and C++*. Udemy. <https://www.udemy.com/course/graphics-with-modern-opengl/>.

de Vies, Joey (June, 06 2020). *Learn OpenGL*. https://learnopengl.com/.