7-1 Final:

Reflection

William Marshburn

CS-330

Gene Rodriguez

Southern New Hampshire University

6/14/22

7-1 Project: Scene Reflection

Per the rubric and requirements, a scene was created looking down on a table with 4 objects. The objects consist of a laptop, pen, card box and a candle. To complete the scene there had to be four primitive shapes making up the objects while one was considered a complex object, being made up of two separate primitive shapes. I selected this scene after reviewing my options of shapes to pick from because I knew I could utilize multiple primitive shapes to help create this scene.

The background of the scene consisted of a table which was reconstructed by creating a large plane. On top of the plane, the laptop lies in the center of the image which coordinates were altered to give off a slim appearance to mock that of the actual laptop, essentially a rectangular cube. The initial basis of the image was relatively easy to put together but including the additional objects increased in difficulty. Rather than creating my own cylinder, pyramid, and second cube, I was able to import them thanks to OpenGL resources I found on Github (Behrens, n.d., DeVries, n.d.).

To finish creating the scene, I placed an enlarged cylinder to the left of the laptop and a cube to the right, lastly, I included a pyramid and a cylinder below the laptop. I then manipulated the model matrix, translate, and rotate functions of all objects to help properly place them to mock that of the picture. The first cylinder to the left acts as the candle and the cube to the right, a card box. The second cylinder and pyramid were combined to create the complex object, the pen as the cylinder can act as the base for the pen, while the pyramid represents the tip of the pen.

As previously stated, I choose the scene and setup because I knew the primitive shapes, we were given to work with could be used to recreate all the objects in the picture. I did run into some difficulties placing (pen and pen tip) and texturing certain objects (could not find a green candle texture, so opted for red) and while the image is not exactly reproduced how I had initially imagined, I feel I made the best choices for creating the 3D objects. Unfortunately, I was unable to properly implement lighting, I had ideas of using different lights in this scene, especially since in real life, a light was on directly overhead, but this is something I will have to revisit in the future.

After working through the modules, I wanted the user to be able to navigate and traverse the scene so they may see from all angles. The mouse is set up to allow the user to pan around the scene from multiple angles while the images’ location is not moving. The user can do this by simply moving the mouse cursor around. The camera is set up so the user can traverse all axes (X, Y, Z) and get a view of all the objects created in the scene and view them from multiple angles. The user can perform these movements by utilizing the keys “W” - Forward, “A” - Left, “S” - Backwards,” D” – Right, “Q” – Down, “E” – Up. The camera setup in combination with mouse movement creates an immersive scene. Lastly, the user can also switch between orthogonal (2D) and perspective (3D) displays by pressing the “P” key.

Most of the code in this project consists of either imported classes or re-used functions of shaders, VBOs or VAOs. This makes the code modular and portable. I have used different set ups and different coding styles to try to create this image. The best way I found was to develop multiple VAOs and VBOs to store the objects. I felt like this was an easier way to track all the objects and manipulate them accordingly, especially when texturing more than one object. Certain functions or aspects in this setup were not abstracted away into another class or file but have been coded so that if this is necessary to do, it can be executed easily.

# Bibliography

Behrens, F. (n.d.). *https://github.com/florianbehrens/opengl-demos/tree/master/shader-fundamentals*. Retrieved from github: https://github.com/florianbehrens/opengl-demos/tree/master/shader-fundamentals

DeVries, J. (n.d.). *https://github.com/JoeyDeVries/LearnOpenGL*. Retrieved from github: https://github.com/JoeyDeVries/LearnOpenGL