**Final Project Reflection**

**Justin Hancock**

**CS – 330 Computational Graphics and Visualization**

**02.25.2024**

My original scene was inspired by normal everyday objects. The objects were a computer mouse, pen, remote control, and two batteries. The purpose of choosing the objects was to try and select something that most other people wouldn’t pick, but I wanted to select items that weren’t just made from simple boxes. I wanted to have items that required multiple shapes to make up the items to add a bit of complexity. If I could go back, there might be a couple of objects that I might have exchanged for simpler objects, but in week two of this course, I didn’t know any better. Using the mesh.cpp to create these shapes helped tremendously by saving time with the object creation. If each object had been built from different vertices, I probably never would have finished. With the material throughout the course being misleading, I really had to focus on what Professor Battersby said in his emails and in our meetings to piece all the information together to come up with this final project. I was able to program for the required functionality primarily by trial and error. Creating the shapes took many hours of time to get them exactly how I wanted them to represent each object in my scene. The requirements stated that the objects should be resting on top of the plane, no object should protrude through the plane, and the objects should also look like the original scene in scale. The textures were not as difficult to apply, but I did struggle with a few. Textures were not necessary to match exactly to the original scene which I think helped me add a bit of creativity to the scene. I do think that the objects in my scene look very realistic because of the textures applied to them. Lighting had a big influence on the textures, and this was the source for most of my texture struggles. Now that the lighting is coded how it should be, the textures look much more realistic in comparison to what the object should look like from the original scene. I began by applying too many different lights to the scene which oversaturated the scene with light. After meeting with Professor Battersby, I further understood that each light source for the individual objects needed to be overwritten to apply the light properly. The combination of specular and diffuse lights on an object can reflect the way that the textures appear from the object like something that should be shiny, dull, or glow. Specular lighting is typically used for an object that would glow or be shiny, and diffuse lighting would help to add light for an object that is dull or flat. The texture chosen for a particular object can be a deciding factor in which type of lighting should be applied. The most difficult of these choices for me was the ends of the batteries. The original textures that I selected looked great on the sides of the object, but not the ends. Once adding the appropriate lighting to the end, I feel like the lighting presented the realism of the object.

A user can navigate through my scene by using the WASD key pattern. WASD key patterns are common in computer video games for movement. W is forward, S is backwards, A is left, and D is right. I have also added Q and E to allow the user to move the camera up and down, respectively. The mouse is also another tool that the user can use to maneuver through the scene. Moving the cursor rotates the camera orientation up, down, left, and right. The scroll wheel on the mouse is used to speed up (scroll forward) or slow down (scroll backward) the movement of the camera when the WASDQE keys are being pressed. The last navigation keys that the user has an option of using are O and P. The O key changes the scene to orthographic mode which allows the user to see the image from a 2D vantage point and disables the use of camera movement. The P key changes the scene back to perspective mode which puts it back into 3D and adds camera movement back to the user.

Throughout the code for this project, there are many different functions that are set up to be reused that help make the code modular and organized. The most helpful functions to me were the functions in the mesh.cpp file that Professor Battersby created for us to use for each of the different shapes. Inside the URender() function, we were able to call the shape from the mesh.cpp file, recreate the shape, and edit the scale, rotation, translation, texture, and lighting for each shape within the object. I used those functions repeatedly to create each of the shapes for all my objects by copying and pasting the code and changing the shape and line of code that drew the triangles. After changing those lines of code, all that was left to do was reposition the shape in its correct place within the object and apply the correct texture and lighting.

While graphic design will more than likely not be what I do professionally after completing this degree, I have enjoyed the challenges that have come along with this course. I feel like I have further developed my programming skills, and I hope to continue doing that through the remainder of this program.