Joseph Valle

3D Scene

CS 330

06/18/2023

The original picture I took when we first began the journey of creating a 3D scene in this course was a picture of my work desk. On my desk I had a coffee cup sitting on my desk from Dunkin Donuts. When I first took the picture there were many other objects on my desk that would assist in learning to create 3D object. I specifically chose to recreate the Dunkin donuts coffee cup and the desk because it had multiple elements that would challenge me more than a simple rectangular keyboard would. The coffee cup entailed creating two separate primitive shapes that together would create the coffee cup. Instead of a cube or triangle the shapes that I needed to build were a sphere and cylinder that I felt were more challenging. The additional shape I aimed to create was a plane that my objects would sit on to represent the desk. I thought of this because placing an object on an additional shape would be an opportunity to learn on how to build around a main object. In this case the plane would be the main object representing the desk that I would build upon by adding the coffee cup on top of it and so forth. To program for the required functionality, I made sure that I would set my objects up in a way that would make sense. For example, it would take specific placement of each object to resemble a coffee cup. First, I would need to make sure I adjust the cylinder to be able to properly sit on the plane. Next, I needed to appropriately place the sphere within the cylinder to make it resemble the cap to the cup.

In my 3D scene you can navigate around the entire plane the all the objects will sit on. You can use user inputs like “W, A, S, D” to move forwards, backwards, left, and right. I also added a functionality of where we can move the camera up or down using the “Q” and “E” keycap. In this scene I also made sure that the camera was able to take input from the mouse to move the view of the camera around through it. With giving simple directional inputs through a keyboard and mouse this can easily be translated to other input devise. Mainly because the amount of input doesn’t overwhelm a controller, or any other input device.

Certain functions that are in my program that help organize it include, processInput, and renderScene. These functions helped me organized my code that was needed to navigate my 3D scene using user input. In this specific function I was able to effectively set my code that needed to be in one place so it didn’t not get mixed in with all the other functions. Additionally, renderScene is another function that handles the rendering of my scene. I also added the initializeOpenGL function that sets up the openGL context and initialized the GLFW and GLEW as well as creating the window. Lastly another function that was added was the createShaderProgram that compiles all the shaders and links them into a shader program. By simplifying the code into smaller self-contained modules, it makes the code more modular and organized. The code being more modular and organized allows for the reusability and others ability to reference when looking to implement similar functions as to these in their own code. It makes it that much easier to read the code and truly understand it.