Module 7-1: Final Project Reflection

Stanley Niles

Southern New Hampshire University

CS-330: Computer Graphics & Visualizations

Professor Jeff Phillips

June 18th, 2023

The main goal and choice of the 3D scene that I chose was it was something familiar to me as it was my own property, and I could provide a creative aspect of taking my own pictures of objects inside of my house to apply for textures. My family really likes the Fallout series (“Universe”), and I had an array of objects that would pose different challenges while also being interesting. The other aspect about choosing a scene from within my house is knowing beforehand where to place objects such as the lighting, which made it easier to program for the lighting requirements of the project. Part of the requirements were to have an object that requires two or more primitive objects to recreate, hence the inclusion of the PS5 controller and headphones. However, I struggled greatly with the scaling, rotation, and translation of the cylinder objects that I withheld the headphones from the image. It was more the rotations were not performing the way that I expected, and it was difficult to get the handles of the controller to line up appropriately. One thing I did to make the rectangular objects easier was define a basic cube object with vertices and elements and transform them into 6 other objects within the scene.

For the navigation of the scene, I implemented the required functionality that allowed the camera to pan in all directions using the W, A, S, D, Q and E keys respectively while also implementing the functionality of moving the camera with the mouse and changing the speed of the camera with the mouse scroll wheel. These implementations allow the user viewing the scene to traverse the entirety of the scene, which would allow different views and angles of various objects. An aspect of this that I am proud of is that if you go towards the back corner of the scene and look at the “initial” left wall when the rendering starts, a prominent specular highlight can be seen which represents the ceiling flood light and the blue light coming from my television in the room. When panning around the “front” of the table and the objects on top of it, you can see a faint blue glow coming from the TV. There are also other lighting objects to provide general ambient/diffuse lighting.

The modularity and organization of my code is something that I believe is well implemented. Towards the end of the project, I skipped further abstraction of certain functions and aspects of the overall system due to time constraints. I abstracted out numerous classes such as the Mesh, Shader, and Texture classes so that when it was time to initialize and render the final objects, it would be much easier to manage the growing list of models.

Within the mesh class, there is a simple Draw() function that accepts the vertices and elements of an object to create a Vertex Array Object, Vertex Buffer Object and Element Buffer Object to represent a drawable 3D object. Within the Shader class, I abstracted the necessary functions to allow values to be passed in an initialized for any number of necessary objects.

The Texture class is much the same, in which multiple lines of code have been reduced to simple function calls to allow a texture to be loaded from a file and applied to a specific object.

Finally, to increase modularity, I placed all the vertices and elements of every object I planned to render inside of a types.h file. Certain functions would modify the existing vertices and elements of simple objects such as a cube and apply them to a new/copied set of vertices. This was, when it came time to perform the function calls and set up my scene, the amount of code for that portion of the program was significantly shorter and more precise.

All this modularity and organization also allowed for easier debugging, maintenance, and future modifications of the code.