Ryne Williams

CS 330 7-1 Final Project

Southern New Hampshire University

For this project, I decided to use my kitchen stove and the cabinets on either side of it to render through OpenGL. I chose these objects as they would be able to be made using multiple different primitive shapes, as laid out in the guidelines for this project. I originally chose to render each object individually using separate methods to render each shape. However, as I worked on the project, I decided that it would look better and create cleaner and easily understood code to create all shapes in one method. This also allowed me to create methods for some of the vertex attribute and vertex buffer objects creations that I could call once, instead of calling them for each shape that I rendered. In the render method, I decided to create two “if” statements to handle the change between the orthographic and perspective view changes and allowed for the “O” and “P” keys to change between the two. I found that it was the easiest way to ensure that the program successfully changed between the two view types without it affecting the program’s ability to render the objects.

When using this program, the user is able to navigate the 3D world by using the “WASD” keys to move the camera around. The “WASD” keys can be used to move the camera in the forward, left, backward, and right directions, respectively. The user is also able to move the camera up and down by using the “Q” and “E” keys. This will allow the user to move the camera around freely and view the rendered objects much more easily. The mouse is also used to turn the camera in any direction to change the angle or direction of the camera view. This function will help the viewer to look around the scene and change direction easily when moving the camera using the “WASD” keys. Which will allow the user to see the scene from every angle at any point in the scene.

I used an object-oriented approach when coding this project, separating specific functions into their own methods to help create clear and concise code that is easy to understand. I made sure to create functions for the code that creates the textures and also for the code that creates the shapes. This allows for the program to be changed in other areas in ways that will not affect the code that does not need to be changed. I also created methods for the code that loads the textures, as well as the code that handles the keyboard and mouse input, allowing this code to be easily reused whenever it is needed elsewhere. Finally, by separating the code that draws the shapes and for the render method that is looped through in the main method, I am able to easily make changes to these methods and do not have to worry about other parts of the code acting in ways that are not intended, as they will not be affected by changing code in other parts of the program.