Kendal Guizado O’riley

Final Project

For this project, I selected several objects to replicate the chosen scene. These objects consisted of cylinders, boxes, tapered cylinders, pyramids, and planes. The plane was chosen to represent the ground level of my scene. With an added texture and lighting the plane replicated a concrete patio. The next object created was a BBQ smoker. For this object, five cylinders of different sizes were used to create a main cooking chamber, a smokestack, as well as a firebox for the fuel. For the patio umbrella, a box mesh was utilized for the base, followed by two cylinders that represented the main pole as well as the arm that connected to the umbrella portion which was in the same as a pyramid. A trash can object was created utilizing the tapered cylinder shape with a 180 Z rotation to turn it upside down and appear as a trash can. Lastly, the last object on the scene is a table which consists of five different box shapes of different sizes to form the top and four legs. I chose all these objects after I took a deep look at the scene and compared which shapes would best represent the objects I was looking at. To program them for the required functionality I first worked on creating the object to a size that I found appropriate. Once the object had all the right components, I thought about locating them in the appropriate portion of the scenes. Once that was complete, I applied textures to the objects with textures that were similar or appropriate for the scene.

Users can navigate the scene using keyboard movements using the keys WASD and QE which allow users to move up and down, forward, and back, as well as left and right. The mouse can also be utilized to move the angle at which we view an object, and the mouse scroll well controls how quickly these actions are performed. These tools were programmed on the viewManager header file, as well as the viewManager cpp file. There were callbacks created for the mouse position which first verifies the position of the mouse to appropriately replicate mouse movements in the future. Calculations were added to measure the amount of movement required. A mouse wheel call-back method was added as well to process the mouse scroll distance. All keyboard functions were created in the process of keyboard events methods. This method processes what happens when a certain key is pressed. For example, if W is selected it would cause the camera to move forward.

The first function generates 3D objects based on specified parameters such as shape, size, and position. By parameterizing the creation process, this function can be reused to generate various objects throughout the scene. Different combinations of parameters allow for the creation of diverse objects without duplicating code. Next is the Texture Mapping Function, which applies textures to 3D objects, mapping images onto their surfaces accurately. As textures need to be applied to multiple objects, this function can be reused to efficiently texture each object. By passing different texture images and object tags, the same function can be used to texture various objects in the scene. The Light Source Function configures the properties of light sources, such as position, color, and intensity. Since the project requires light sources with different characteristics, this function allows for easy configuration of each light source. By specifying parameters for each light source, such as position and color, the function can be reused to set up various lighting scenarios. The Camera Movement Function controls the movement of the virtual camera, including translation and rotation. By encapsulating camera movement logic into a separate function, it can be reused to implement different types of camera controls throughout the project.