Danica Hesemann

CS 330

June 20, 2021

Final Project Reflection

The 3D scene I created was based on a photo from my backyard. I chose the scene and the items I did because I thought it would be easy to create the objects from a variety of basic shapes, but they also left room for more detail and complex shapes. For example, the shed could have been a simple cube and a pyramid, but I chose to build a more realistic and complex shape. including a doorknob, which I initially did not intend to do. Other shapes, like the chairs and firepit, I simplified a bit. I made the chairs with just vertical cylinders for the legs and planes for the seat, and I made the firepit a cylinder on the ground with a short tube I made by modifying the code for a cylinder. I also simplified and took some liberties with the positioning. I did not angle the shed or chairs like in the photo, and I moved the trees from the back of the scene to the front because I thought it worked better with the lighting. The light objects fit very naturally into my scene, a sphere for the moon, and colored pyramids for a fire. One thing I did to try and code more efficiently was to use the same shapes wherever I could. This way, I could re-use VAOs and VBOs for multiple objects. Originally, I was going to use cones for the trees, but I was using pyramids for the fire and decided to use them for the trees as well, which I ended up liking much better.

To navigate my scene, a user can move the camera forward, left, backward, and right using the W, A, S, and D keys (respectively). The camera can be moved up with the Q key, or down with the E key. The camera can also be rotated by moving the mouse cursor to the area of the screen that aligns with the direction they wish to look. For example, move the cursor to the left side of screen to rotate the camera to look to the left. A user can also press the P key to switch between a perspective display and an orthographic one, or they can press the escape key to exit the scene.

Some of the functions I included in my program include functions to create shaders or textures, process various inputs, and render the scene. The functions to create a shader program and to create a texture can be used as many times as necessary in the code. The function to create a shader program compiles specified shader files and links them together into a shader program. I used the function to create a shader once for the shader program I used for my textured objects and once for the shader program I used for my light objects. The function to create a texture loads an image, flips it (using another function in my program), sets the wrapping and filtering parameters, and generates the mipmap. I used this function every time I loaded a new texture into my program. The other functions I used make the code more organized and easier to modify or reuse for other programs. For example, the render function in my program defines the vertex data for the various shapes, creates the VAOs and VBOs, and draws the shapes. The function is long and I only use it once, but if I wanted to create an entirely new project, I could use the same main function, make small modifications, and just write a new render function to define and draw the objects for a new scene. Similarly, have three functions to handle various input: processing keyboard input to move the camera, positional mouse input to rotate the camera, and scrolling mouse input to adjust the camera speed. If I wanted to omit or change any of these input methods in a new project, I could keep the same main function (or function to initialize the window) and rewrite the input functions, making only small modifications to the main code.