Jonathan Biedermann

Barbara Hecker

CS-330

17 April 2022

Module Seven: Project Reflection

When I first constructed the setting for my 2D scene, I was unsure what would be required in order to transform it into a 3D scene, complete with lighting and camera movement as well as applied texture. However, looking at the scene, one can see how simple 2D shapes could be used to construct the more complex 3D shapes. Implementing movement controls along with camera help ensure the user can view the entirety of the 3-Dimensional shapes. By utilizing the proper keys to move the camera on certain axes, the program ensures the user can use their other hand to operate the mouse to move the camera on the other axes. The functions contained within the program are designed in a way so that they can be reused in a different program and produce similar results.

The shapes that are contained in my 2-Dimensional image are simple enough to be reconstructed as 3D geometric shapes in OpenGL using vertices and triangles. As I have had little to no experience using OpenGL, it took a lot of trial and error to figure out how the vertices and coordinate system worked in cohesion to form triangles and then use these triangles to form, and then progress again and use these simple squares to make cubes. After getting the correct shapes that I wanted, I needed to figure out how to get the color of those shapes to closely identify with the shapes in my image. I was unable to texture the images correctly so I needed to use the color functions get as close as I could to match the objects in my image. The objects that I chose to represent from image are a calculator and a notebook. I used a rectangular cube that is shorter on the z-axis to represent the notebook and a rectangular cube that is shorter on both the y-axis and the z-axis to represent the calculator.

When creating functions to handle camera movements, the first choice might be to code them to the up, down, left, and right arrow keys accordingly.