7-1 Final Project Submission

Ashlee Wood

When I was choosing what I wanted for my 3D scene I considered a few scenes such as a night sky or house. I felt a simple living room scene was sufficient and not too exhaustive to implement shapes for. It also had common objects that would not be too simple or too hard to play with. I wanted a challenge but not the “dark souls' level” of the game type of challenge. The couch had me struggling to make it functional from all angles in the 3D scene. I finally changed it up a bit in this last assigment so it was easier to see from all angles. The lamp I added was not hard and simple. Cylinders can be scaled pretty easily. The bookshelf was fun to make and add the texture of books to it. It was a common box shape.

I was able to program the required functionality from the announcements as a guide. It was very helpful to watch how it could be done step by step. I am a hands-on learner and sometimes online feels a little detached. I was able to program the functions and requirements by continuously playing with the shapes, scale, positions and colors.

The input devices that were used to control the virtual camera in my 3D scene were the keyboard and mouse. This was implemented into the code by adding the mouse scroll and zoom for the mouse options. The keyboard allows the movements of forward, backward, left, right, up, and down directions. The WASD keys are the forward, backward, left and right movements. Q and E will be the up and down movements in the program, while the P and O keys are the perspective and orthographic modes. These were programmed into the code with a callback function. The movement to manage the camera and view configurations in the ViewManager class in the C++ class.

The OpenGL and tutorials that were provided helped me to implement these functions and stay modularized and organized in the programming. Using customized functions in programming can be reused to improve code’s readability. A function that I developed was the object material weird material function. It was set and then called into my scene manager: Prepare scene method. I was able to reuse it in my program a few times to render the material in my 3D scene. It can be called a few times without having to rewrite the code. This allowed my code to be maintainable and organized.