**Module 7 Reflection**

Jermaine Wiggins

Southern New Hampshire

CS-330: Computer Graphics and Visualization

Prof. Phillip Enkema

June 22, 2025

**Module 7 Reflection**

**Justify development choices for your 3D scene. Think about why you chose your selected objects. Also consider how you were able to program for the required functionality.**

Before building my scene, I examined each object and listed how I would construct it. For example, I planned to use four stretched box legs for the table. This step was very valuable because it allowed me to follow a somewhat structured plan. If an object didn’t look quite right, I would rethink the structure. For instance, I originally used boxes combined with flattened cylinders for the couch cushions. The object looked okay, but I wanted the cushions to appear softer and more distinguishable from the couch frame. In the end, I decided to use modified cylinders for the cushions. While not perfect, this approach provided a more natural, curved look that better resembled cushions.

Choosing good textures was more challenging than I expected. Some textures looked very realistic, while others didn’t quite fit. I wanted the three main objects to stand out and for the scene to feel unified, as if everything belonged together. For example, I found a popcorn texture that made the wall look very realistic, more so than the textures on the end table and couch. However, this texture was very bright, and with the lighting I set up, it appeared too bright and out of place. Ultimately, I chose textures that complemented the overall scene better, even if they were less detailed.

To adjust the lighting, I took a similar approach to planning the objects. I initially used two lights, one for general lighting and another for the lamp. To prevent the point light above the lamp from overpowering the scene, I added another light on the opposite side and adjusted both lights to their lowest possible values to achieve uniform but subtle lighting. However, this caused the lamp light to appear nonexistent. To fix this, I found a new texture for the lamp shade that gave the appearance of being lit, enhancing the overall effect.

**Explain how a user can navigate your 3D scene. Explain how you set up to control the virtual camera for your 3D scene using different input devices.**

The user navigates the scene using a keyboard and mouse. The Q and E keys allow the camera to move up and down along the y-axis, while the WASD keys move the camera left, right, forward, and backward along the x and z axes. This functionality is achieved by checking the status of the keyboard each time the display window refreshes and mapping the camera movements to the corresponding keys.

Moving the mouse controls the camera’s orientation, allowing the user to look around the scene. To improve precision, the scroll wheel adjusts the speed of the camera’s movement along all axes. Mouse movements were handled differently, instead they are captured through callbacks that react only when the mouse moves, rather than continuously checking its status.

**Explain the custom functions in your program that you are using to make your code more modular and organized. Ask yourself, what does the function you developed do and how is it reusable?**

After reviewing all the header and source files, I found the code to be well-structured and modular. Many of the functions needed to build objects in the scene, such as drawing partial shapes, were already provided. The parts I added followed this clean, organized structure to maintain consistency. One change I made was having RenderScene calls separate functions to render each object individually. Inside each of these object specific functions, the code for creating the object is wrapped in curly braces with comments describing the object. This structure helped keep the code organized and made it easier to locate and update specific objects later on.