CS 330 Module 7 Project

Christina Jimenez

[christina.jimenez3@snhu.edu](mailto:christina.jimenez3@snhu.edu)

Southern New Hampshire University

1. 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.

I chose to render a variety of objects in my scene to create a realistic and engaging environment. The objects include books, a desk, a bookshelf, a window, walls, a floor, a pot, and a cat. These selections allowed me to practice rendering different basic shapes and combining them into more complex forms. The coding process was tedious, but many parts of the code were reusable. For example, I reused and adapted code for positioning, scaling, transforming, and selecting appropriate mesh types for each object. The most challenging part was setting up the lighting to ensure the scene was properly illuminated, which required careful attention to the direction and intensity of light sources. Texturing was another detailed task, involving finding suitable textures and properly loading them into the textures folder for use in the scene.

2. 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.

To navigate the 3D scene, the keyboard and mouse are utilized. The keyboard controls include:

W/S: Move forward and backward

A/D: Move left and right

Q/E: Move upward and downward

The mouse allows for camera rotation within the scene, and the scroll wheel adjusts the camera’s speed for smoother or faster navigation. To set this up, I implemented input handlers that capture keyboard, mouse inputs, and translate them into corresponding camera movements. This system ensures seamless control of the virtual camera and enhances user interactivity with the scene.

3. 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?

To improve modularity and organization, I created custom functions for rendering individual meshes. Each object in the scene was assigned its own mesh function, which simplifies tracking and troubleshooting. These functions enable the reusability of code by allowing adjustments to scale, position, and rotation without duplicating effort. For example, specific parts of complex objects, such as the shelves of a bookshelf, were rendered by reusing and modifying these mesh functions. By keeping the code modular, I reduced redundancy, streamlined error correction, and enhanced readability.