Blake Kemp

Southern New Hampshire University

CS-330

Final Project

In this paper, I will justify the development choices made for my 3D scene and explain the functionality implemented in the scene. Even more than that, I will discuss the navigation options provided to the user and the use of custom functions to enhance code modularity and organization.

Development Choices for Object Selection: The objects chosen for the 3D scene were carefully selected based on their shapes and relevance to my immediate environment, specifically my office. The decision to incorporate objects found in my office allows for a more personal and relatable experience. By using familiar objects, such as cubes, cylinders, and planes, the user can easily connect with the virtual environment. This choice enhances the immersive nature of the scene, creating a sense of familiarity and engagement.

Functionality and User Navigation: To provide a seamless user experience, I implemented a camera control system that allows users to navigate the 3D scene. Users can manipulate the virtual camera using different input devices. By using the mouse, users can adjust the camera's orientation, effectively changing their viewpoint in the scene. Scrolling the mouse wheel enables users to control the camera's movement speed, allowing for smooth exploration. In addition to mouse control, keyboard inputs provide more navigation options. Users can use the WASD keys to move the camera position horizontally and vertically, enabling them to explore different areas of the scene. To enhance the user's interaction with the scene, I also incorporated the ability to switch between different perspectives by pressing the P and O keys. This feature allows users to see the scene from different angles, enhancing their understanding of the objects' spatial relationships, and the impact of lighting effects.

Custom Functions for Code Modularity and Organization: To promote code modularity and organization, custom functions were developed within the program. These functions encapsulate specific tasks and improve code readability, making it easier to troubleshoot and maintain. One notable example is the creation of shape meshes, such as cubes, cylinders, planes, and pyramids. Although these functions may not be the most efficient implementation, they serve as building blocks for constructing objects in the scene. By encapsulating shape generation in individual functions, it becomes easier to modify and customize object properties, such as position, rotation, and texture mapping. This modularity enhances code reusability, allowing for the creation of multiple instances of the same shape with varying attributes. It also facilitates the addition of new objects in the future by providing a consistent and organized structure.

Conclusion: The development choices for the 3D scene, including object selection, aim to create a relatable and visually engaging experience for the user. Implementing user navigation controls through mouse and keyboard input enhances the interactive nature of the scene, allowing users to explore and see different perspectives. Custom functions contribute to code modularity and organization, improving readability and facilitating future expansion of the scene. Through these choices, the 3D scene achieves its intended functionality and provides an immersive and educative environment for users to explore.