Matthew R. Leclerc

4/19/24

CS-330 Final Project Reflection

Southern New Hampshire University

When setting up the 3D environment for a workspace in a computer, I paid attention to making it immersive, like a gaming setup. The use of 3D planes, textured in the paint and wood for the walls and floor, created the first hint of a background, reflecting common choices in interior design that serve only to augment the relatability and visual appeal of the scene. Objects are exhaustively listed, such as the keyboard keys and associated computer peripherals, and decorative, for example a desk microphone and a desk drink, to further add reinforcement that this scene is as enriched, lifelike, and inclusive as it could be.

The selection of the objects—the desk, legs of the desk, and the computer monitor—is a part of the idea to bring out a scene that does not only look appealing but also brings forth a typical modern computer workspace. Each object has been modeled in such a way that all the aesthetics and functionality of the scene are met. An example might be a desk modeled from a combination of LoadBoxMesh(), stretched into a 'desk' form, and LoadCylinderMesh() for desk legs were pre-made and taken advantage of as it allowed for “not-too-complex" geometry but simple and perfectly practical results in a robust and realistic-looking desk.

It features an interface where the user can browse through the 3D model with intuitive keyboard and mouse controls. It utilizes WASD for movement direction, 'Q', and 'E' for up and down direction and utilizes the mouse for the control of the orientation of the camera, which makes the whole model come with a total video game kind of navigation system, making the user feel at home. It is easy to toggle between orthographic and perspective views with the 'P' and 'O' keys, respectively, to facilitate the user in exploring and interacting with the environment based on how they would prefer it to be seen.

The virtual camera has been rigged to be dynamic following user inputs through the keyboard and mouse. This setting was achieved by capturing the real-time input data for the adjustment of the camera in 3D space, both in position and in orientation, in a way that responds naturally, just like a physical camera. Meanwhile, the sensitivity of movements across the space is made possible by the scroll wheel of the mouse, giving smooth and responsive camera dynamics for a smooth experience for a user.

There is a need to create custom functions for certain renderings, such as RenderKeyboardKeys(), RenderDesktop(), and various others, to maintain a neat and organized code base. These functions enclose every detail concerning rendering for a certain part of the scene, making the code modular and manageable. This makes the code readable, debuggable when needed, and helps to reuse those parts of the components across various places of the application or other projects. It could be compared to using a table of contents in a book, where every function acts like a precise entry point: this makes the base of the code navigable, hence making development easy and efficient.

The 3D scene development choices of aesthetics versus realism, usability, and code maintainability are therefore balanced. The application has further been designed to take advantage of modular programming with an environment designed from carefully chosen objects that add up to coherent. As such, the development remains both adaptive and easy to manage.