7-1 Assignment

Lauren-Ann Javier

CS 330

12/11/2024

Southern New Hampshire University

* **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.

In developing the 3D scene, I carefully selected the box, a cylinder, a sphere, and a prism based on their visual appeal, geometric diversity, and their ability to demonstrate different lighting interactions. Boxes are clear to model and in texture, making them a good variety for structure in the scene. The cylinder is because of its diversity in shape and simple geometry. A sphere has a smooth surface and can catch and reflect light and the contrast is much different from a box since there are no sharp angles. Prisms are very versatile, so I wanted to use them for this example. It adds a visual interest to its angles. These objects provide a broad range of reflections and shadow effects due to their varied surfaces and enable a more dynamic and attractive scene when illuminated with lights. The decision to use these specific objects allows for a stronger interaction with the lighting model, which is important for showcasing the effects of ambient, diffuse, and specular lighting. Additionally, the variety in shape helps in demonstrating how different textures and lighting affect the perception of 3D space and material properties. From a development perspective, programming for this required making sure that each object was correctly loaded and textured. This involved setting up texture tags appropriately and making sure all paths to texture files were correct.

* **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.

For user navigation within the 3D scene, I chose to implement controls that mimic common gaming and virtual reality navigation patterns. This makes the experience natural and accessible. The setup includes keyboard and mouse controls for movement. This is where the arrow keys and mouse movements are used to move forward, backward, left, and right, and to look around the scene. The camera control is programmatically linked to user input through these devices, allowing the user to explore the scene dynamically. I also incorporated features like zooming in and out using the mouse scroll wheel to provide a more immersive experience. The camera’s position and orientation are updated in real time based on user input, which allows for a seamless transition between different viewpoints. This setup not only enhances user interaction with the scene but also makes it more engaging by allowing free movement within the 3D space.

* **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?

The development of custom functions significantly contributed to the organization of the code. These functions encapsulate specific tasks, such as setting up lighting, handling texture loading, updating the camera position, or managing object rendering. For instance, the function SetupSceneLights() consolidates all the lighting setup tasks, ensuring that all lights are initialized correctly and efficiently, and avoiding redundant code. This makes the main rendering loop cleaner and easier to follow, as each function serves a specific purpose, like setting up different light types or handling user inputs. These custom functions are reusable across different scenes or projects with minor adjustments. They allow for easier debugging and maintenance since each function's logic is isolated and well-defined. For example, by isolating texture loading in a separate function, it becomes straightforward to switch or update textures without affecting other parts of the code. The modular approach not only simplifies code management but also enhances the scalability and maintainability of the application, making it easier to expand the 3D scene with new features or content in the future.

References

ambientCG. (n.d.). Categories. Retrieved December 11, 2024, from <https://ambientcg.com/categories>