Hong Luu

Professor Philip Enkema

CS 330

October 21, 2023

Design Decisions

**Justify development choices for your 3D scene:**

In designing my 3D scene, I selected specific objects such as a cube, cylinder, sphere, and plane to demonstrate a range of geometric shapes, each serving a distinct purpose. The cube and cylinder were chosen for their versatility in representing stylized objects, while the sphere allowed me to explore the challenges of curved surfaces. The plane was used to create a stable ground for the scene. My choice of objects was driven by a desire to diversify the scene and enhance the learning experience.

To program the required functionality, I opted to use OpenGL due to its flexibility and widespread usage in graphics programming. This choice allowed me to work with vertices, textures, and shaders effectively. For user interaction, I implemented keyboard and mouse controls, enabling users to manipulate the virtual camera. I also incorporated lighting effects to enhance the visual appeal of the objects, as it’s a fundamental aspect of 3D graphics.

**Explain how a user can navigate your 3D scene.**

Users can navigate my 3D scene using keyboard and mouse inputs. The arrow keys or W, A, S, D keys control the camera’s movement in the scene, allowing users to move forward, backward, and strafe left or right. Meanwhile, mouse movements control the camera’s orientation, enabling users to look around and change the perspective. This combination of keyboard and mouse controls creates an immersive and interactive experience, making it intuitive for users to explore the 3D environment.

To ensure smooth camera movement, I implemented camera control functions that calculate the camera’s position and orientation based on user inputs. These functions take user input as parameters and update the camera’s position and orientation accordingly, creating a responsive and engaging navigation experience.

**Explain the custom functions in your program that you are using to make your code more modular and organized.**

In my program, I developed custom functions to enhance code modularity and organization. One example is the "updateCameraPosition" function, which takes user input as parameters and calculates the new camera position based on the user’s movement commands. This function makes the code more organized and readable by encapsulating the camera movement logic.

Another custom function, “applyTexture” allows me to apply textures to 3D objects. This function takes the object and texture as parameters, making it reusable for different objects in the scene. By encapsulating this texture application logic in a function, I can easily apply textures to various objects without duplicating code, enhancing code maintainability.