Chandni Singh

CS330

Justification of Development Choices for 3D Scene

**Object Selection and Scene Composition**

In developing my 3D scene, I carefully selected objects that would provide a balanced and realistic environment. The inclusion of a green lighter, an orange, a water bottle, and a green ceramic glazed cup on a wooden floor creates a still-life composition that mirrors everyday objects. These items were chosen for their contrasting textures and shapes, which allowed for a rich exploration of material properties and lighting effects within the scene.

For example, the orange's matte texture contrasts sharply with the reflective surface of the green lighter, which was intentionally made shiny to capture the light differently. The water bottle, with its clear plastic and detailed cap, adds a level of realism, while the ceramic cup introduces a different kind of surface reflection. The wooden floor ties the scene together, providing a warm, natural base that complements the more artificial materials.

**Programming for Required Functionality**

To achieve the desired functionality, I relied on a combination of shader programming and careful material definition. By using shaders, I was able to control how light interacts with each object, adjusting parameters like ambient, diffuse, and specular components to create realistic effects. For instance, the lighter's reflective material was defined with high specular values to ensure it would appear shiny under the light source.

The lighting model was implemented using the Phong reflection model, which allows for realistic rendering of surfaces by simulating how light reflects off them. I used two light sources in the scene to add depth and dimension, one representing natural sunlight and the other providing additional illumination to highlight specific objects.

The materials were defined using a custom structure that includes ambient, diffuse, and specular colors, as well as shininess and tags for easy identification. This modular approach allows for easy modification and reuse of materials across different objects.

**User Navigation and Virtual Camera Control**

Navigating the 3D scene is made intuitive through the implementation of a virtual camera system that responds to user input from various devices, such as a keyboard, mouse, or game controller. The camera is set up to allow for both first-person and third-person perspectives, providing flexibility in how the user views the scene.

The camera is controlled using a set of custom functions that handle movement and rotation based on user input. For instance, keyboard inputs are mapped to functions that move the camera forward, backward, or side to side, while mouse input controls the camera's orientation. This setup ensures smooth and responsive navigation, allowing users to explore the scene from different angles and distances.

The virtual camera was programmed to include functions for setting the field of view, adjusting the aspect ratio, and handling perspective projection. These features ensure that the scene is rendered correctly on different screen sizes and resolutions, providing a consistent experience regardless of the display used.

**Custom Functions for Modular and Organized Code**

To keep the codebase organized and maintainable, I developed several custom functions that encapsulate specific tasks, making the code more modular and reusable. One such function is SetTransformations, which handles the scaling, rotation, and positioning of objects in the scene. By centralizing these operations in a single function, I can easily adjust the transformations for different objects without duplicating code.

Another key function is SetShaderMaterial, which applies the desired material properties to an object before rendering. This function takes a material tag as an argument, looks up the corresponding material properties, and sets the appropriate shader parameters. This approach not only keeps the code clean but also allows for easy experimentation with different materials.

The SetShaderTexture function follows a similar pattern, applying the correct texture to an object based on its tag. By separating the texture and material application into distinct functions, I can mix and match materials and textures with ease, providing greater flexibility in the scene's appearance.

In summary, the development of this 3D scene involved careful consideration of object selection, material definition, lighting, and user interaction. By organizing the code into modular functions, I ensured that the scene is both visually compelling and easy to maintain, with a user-friendly navigation system that enhances the overall experience.