**Victor Ngetich**

**CS 330**

**8/16/2025**

**Final Project Reflection**

I chose to include a cup, a pencil, a book, and a detailed instrument to create a realistic desktop scene. These items are typical of what one might find on a desk, and they allowed me to showcase different modeling techniques by making use of primitive shapes.

**Approach to Modeling**

**Cup**: I modeled the cup using cylinders for the body and a half torus for the handle, accurately reflecting the typical form of a mug.

**Pencil: T**he pencil was constructed from multiple primitives a cylinder for the body, cones for the tips, and smaller cylinders for the ferrule and eraser adding both realism and detail.

**Book:** The book was created with box primitives for the cover and pages, which provided a simple yet effective structure for applying textures.

**Lamp (Instrument):** The lamp was built using a combination of cylinders and spheres, demonstrating how simple geometric shapes can be combined to represent a more complex object.

**Textures and Materials**

Texture Choices: A wood texture was applied to the table surface for realism, the book cover features a green leather-like texture, and the lamp incorporates a glass texture to simulate transparency.

Texture Application: I ensured textures were accurately mapped by adjusting UV scales and using high-resolution, royalty-free images to avoid distortion and maintain detail.

**Lighting Setup**

**Directional Light**: A directional light simulates natural sunlight streaming through a window, providing ambient illumination for the scene.

**Point Light:** A point light positioned inside the lamp represents a light bulb, with warm tones to create a cozy and realistic atmosphere.

**Lighting Effects:** The Phong shading model was applied, incorporating ambient, diffuse, and specular components to enhance object realism and highlight surface details.

**User Navigation in the 3D Scene**

**Keyboard Controls:**

W / S: Move forward and backward along the camera’s facing direction.

A / D: Move left and right.

Q / E: Move upward and downward along the Y-axis.

These controls provide full freedom of movement across all three dimensions.

**Mouse Controls:**

**Orientation:** Mouse movement adjusts the camera’s pitch (up/down) and yaw (left/right), enabling smooth scene exploration.

**Speed Adjustment:** The scroll wheel modifies camera movement speed, allowing for precise navigation when examining detailed parts of the scene.

**View Mode**s:

P Key: Switches to perspective projection, providing depth perception in the 3D environment.

O Key: Switches to orthographic projection, offering a flattened, 2D-like view useful for checking alignments and proportions.

Custom Functions for Modularity and Organization

**Classes and Structure**:

SceneManager: Handles overall scene setup, including object creation and texture application.

**ViewManager**: Manages the camera, user input, and projection toggling.

**ShaderManager**: Responsible for loading, compiling, and applying shaders during rendering.

This modular design improves code readability, maintainability, and scalability.

**Custom Functions:**

**RenderShapeWithEdges:**

Purpose: Renders a shape in both fill and line modes, showing the solid object along with its wireframe edges.

Reusability: Ensures consistent rendering of different objects without duplicating logic.

**RenderTransparentShape:**

Purpose: Manages rendering transparent objects, such as the glass lamp and bulb, by configuring appropriate blending modes and depth masks.

Reusability: Streamlines the rendering of transparent materials while maintaining correct visual effects.

**Benefits:**

**Code Function Benefits**

These functions are great because they cut down on code duplication, make the rendering process more efficient, and let you update or change how things are rendered without affecting the rest of your code.

Reflecting on the Development Process

**Challenges I Faced**

-Complex Modeling: Building the instrument was tricky. I had to be precise with placing and scaling multiple shapes to get the look just right.

- Texture Mapping: Getting textures to correctly project onto unusual shapes took a lot of trial and error with UV scaling.

- Lighting Effects: It was a balancing act to get the lighting right I had to constantly tweak the intensity and position of the lights to avoid spots that were too bright or too dark.

**What I Learned**

- I now have a much better grasp on how to build complex 3D models using basic shapes.

- My skills in applying and manipulating textures to create realistic effects have improved significantly.

- I enhanced my knowledge of OpenGL lighting, especially using the Phong shading model.

**Future Improvements**

**Optimization**: I'd like to try reducing the polygon count to improve performance without losing visual quality.

**Advanced Texturing**: I plan to explore normal or bump mapping to add more depth to the textures.

**Interactivity**: I want to add user interaction, like being able to pick up or move objects within the scene.