**Victor Udeh  
CS330 Final Project Reflection  
Module 7-1  
03 OCT 2024**

**Reflection Document**

**1. Justify Development Choices for Your 3D Scene**

**Selection of Objects:**

I selected a cup, a pencil, a book, and an intricate instrument to replicate a realistic desktop scene. These objects are common items found on a desk, and they provided an opportunity for me to demonstrate a variety of modeling techniques using primitive shapes.

**Approach to Modeling:**

* **Cup:** I created the cup using cylinders for the body and a half torus for the handle to capture the typical shape of a mug.
* **Pencil:** Composed of multiple primitives, I used a cylinder for the body, cones for the tips, and small cylinders for the ferrule and eraser to add detail.
* **Book:** I modeled the book using boxes for both the cover and pages, which allowed me to apply textures effectively.
* **(Lamp) Instrument:** Constructed using cylinders and spheres, I represented a complex object to showcase the ability to combine simple shapes into a more intricate model.

**Textures and Materials:**

* **Texture Choices:** I applied a wood texture to the table surface to give it a realistic look. The book cover uses a green book texture, and the lamp has a glass texture to simulate transparency.
* **Texture Application:** I ensured textures were accurately projected by adjusting the UV scales and using high-resolution, royalty-free images.

**Lighting Setup:**

* **Directional Light:** I implemented a directional light to simulate natural light coming from a window, providing ambient illumination to the scene.
* **Point Light:** Placed within the lamp to represent a light bulb, I used warm colors to create a cozy atmosphere.
* **Lighting Effects:** I used the Phong shading model with ambient, diffuse, and specular components to enhance the realism of the objects.

**2. Explain How a User Can Navigate Your 3D Scene**

**Keyboard Controls:**

* **Movement:**
  + **W:** Move forward along the camera's facing direction.
  + **S:** Move backward.
  + **A:** Move left.
  + **D:** Move right.
  + **Q:** Move upward along the Y-axis.
  + **E:** Move downward.

These controls allow the user to navigate freely around the scene in all three dimensions.

**Mouse Controls:**

* **Orientation:**
  + **Mouse Movement:** Controls the camera's pitch (up and down) and yaw (left and right), allowing the user to look around the scene smoothly.
* **Speed Adjustment:**
  + **Scroll Wheel:** Adjusts the camera's movement speed, enabling precise control when exploring intricate parts of the scene.

**View Modes:**

* **Projection Toggle:**
  + **P Key:** Switches to perspective projection, providing a 3D view with depth perception.
  + **O Key:** Switches to orthographic projection, offering a flat, 2D-like view of the scene.

This functionality helps users to view the scene from different perspectives, which can be useful for examining object alignments and proportions.

**3. Explain the Custom Functions in Your Program to Make Your Code More Modular and Organized**

**Code Modularity:**

* **Classes and Organization:**
  + **SceneManager:** I designed this class to handle the preparation and rendering of the scene, including object creation and texture application.
  + **ViewManager:** Manages the camera, user input, and projection modes.
  + **ShaderManager:** Deals with loading and using shaders for rendering.

This separation of concerns improves code readability and maintainability.

**Custom Functions:**

* **RenderShapeWithEdges:**
  + **Purpose:** Renders a shape twice—first in fill mode and then in line mode—to display both the solid object and its wireframe edges.
  + **Reusability:** Allows for consistent rendering of various objects with edges without duplicating code.
* **RenderTransparentShape:**
  + **Purpose:** Handles rendering of transparent objects, such as the glass lamp and light bulb, by setting appropriate blending modes and depth masks.
  + **Reusability:** Simplifies the process of rendering any transparent shape with correct settings.

**Benefits:**

These functions reduce code duplication, make the rendering process more efficient, and allow for easy updates or changes to rendering logic without affecting other parts of the code.

**4. Reflection on the Development Process**

**Challenges Faced:**

* **Complex Modeling:** Creating the instrument required careful placement and scaling of multiple primitives to achieve the desired look.
* **Texture Mapping:** Ensuring textures were accurately projected on irregular shapes took some trial and error with UV scaling.
* **Lighting Effects:** Balancing the lighting to prevent overexposure or dark areas involved tweaking light intensities and positions.

**Learning Outcomes:**

* I gained a deeper understanding of how to construct complex 3D models from basic shapes.
* Improved my skills in texture application and manipulation for realistic effects.
* Enhanced my knowledge of lighting in OpenGL, particularly using the Phong shading model.

**Improvements for the Future:**

* **Optimization:** I would consider reducing the polygon count further for better performance without sacrificing visual quality.
* **Advanced Texturing:** Plan to explore normal mapping or bump mapping to add more depth to textures.
* **Interactivity:** Aim to implement user interactions with objects, such as picking up or moving items within the scene.