**Introduction**

For my final project, I created a 3D desk scene in OpenGL featuring a coffee cup, a book, and a lamp on a wooden desk. The goal was to replicate a realistic workspace from a 2D reference image while applying textures, lighting, and camera navigation. This project explored object modeling, texture mapping, lighting, and interactive camera controls in a cohesive scene.

**Object Selection and Construction**

Shapes were chosen to simplify modeling while accurately representing objects. The coffee mug uses a cylinder for the body and a torus for the handle. The book is modeled with two boxes for the cover and pages. The lamp combines a box base, two cylinders for the stand, a tapered cylinder for the head, and a sphere for the bulb. The desk is represented by a plane.

Objects are positioned to match the reference image: the coffee mug is on the right, the book to its left, and the lamp further left and slightly back. This placement creates a balanced scene and allows textures and lighting to interact realistically.

**Texture Application**

Textures enhance realism. The coffee mug uses a ceramic texture to avoid artifacts seen with porcelain. The book has a leather cover and white pages. The desk uses a wood texture, and the lamp has a metal texture. Textures were tiled and aligned to each object to maintain consistency and minimize distortion.

**Lighting**

The scene uses two main lighting types. Ambient light softly illuminates the room, while a directional light simulates the lamp, casting focused light on the desk, mug, and book. This combination creates depth and emphasizes textures, making the scene visually dynamic.

**Camera Navigation**

Camera controls allow free exploration. WASD and QE keys move the camera along X, Y, and Z axes, the mouse adjusts pitch and yaw, the scroll wheel changes speed, and a keyboard toggle switches between orthographic and perspective views. This ensures all objects can be interactively examined.

**Modular Functions and Code Design**

The scene was built with modular functions for reusability and maintainability. The SceneManager class organizes preparation and rendering. PrepareScene() loads meshes, textures, and materials, centralizing operations so objects can be modified easily.

Objects are transformed using SetTransformations(), which separates scaling, rotation, and position, standardizing application and reducing code duplication. Materials and textures are handled with SetShaderMaterial() and SetShaderTexture(), providing a clear interface.

Lighting setup is also modularized. InitializeShadowMap() prepares the lamp’s shadow map, and PrepareScene() configures ambient and directional lights. RenderScene() applies transformations, materials, and textures consistently, ensuring proper shading. Emissive properties, like the lamp bulb, are applied through shader flags, supporting flexibility. This structure allows new objects or lighting effects to be added with minimal changes.

**Conclusion**

The desk scene demonstrates practical 3D modeling, texture mapping, lighting, and camera control. Using low-polygon shapes, carefully selected textures, and interactive navigation, I created a visually cohesive, realistic environment. The modular code design ensures maintainability and scalability, reflecting thoughtful aesthetic and functional choices.

2D Image

A black and white drawing of a lamp and a book

AI-generated content may be incorrect.

3D Image

A desk lamp and a book

AI-generated content may be incorrect.