Final project

**Design Decisions and Implementation of the 3D Scene**

**Introduction**  
The purpose of this project was to develop an interactive 3D scene using OpenGL, incorporating fundamental computer graphics concepts such as modeling, lighting, texturing, and camera navigation. The scene includes multiple 3D objects, dynamic lighting, and user-controlled navigation. This document outlines the design choices, implementation strategies, and modular programming principles applied to achieve the project’s objectives.

**3D Scene Design Choices**  
The objects in the scene were chosen based on their composition of fundamental 3D primitives. The scene consists of a **plane (table surface), a box, a cylinder (mug), a torus (mug handle), and a sphere**. Each object was created using **basic shapes such as planes, boxes, cylinders, and toruses**, ensuring an optimal polygon count under 1,000 triangles per object.

Textures were applied to enhance realism, with a **wood texture for the table surface and a metal texture for the mug**. These textures were loaded using the stb\_image.h library and mapped correctly using UV coordinates to ensure proper alignment. Texturing was applied using OpenGL’s texture binding functions.

**Lighting Choices**  
To improve the visual realism of the scene, **two light sources were implemented**:

1. A **white point light** positioned above the scene to simulate natural lighting.
2. A **red-colored light** to add contrast and dynamic visual appeal.

Both lights were implemented using the **Phong reflection model**, which consists of **ambient, diffuse, and specular** components. These light properties were passed to the shaders to simulate realistic shading and reflections on the objects.

**Camera and Navigation Implementation**  
User navigation was implemented to allow **free movement around the scene**. The controls are as follows:

* **W, A, S, D**: Move forward, left, backward, and right.
* **Q, E**: Move up and down.
* **Mouse Movement**: Controls the camera’s pitch and yaw, simulating a first-person perspective.
* **Mouse Scroll**: Adjusts movement speed for better control.
* **P and O Keys**: Switch between **perspective (3D)** and **orthographic (2D)** views, keeping the camera orientation unchanged.

**Code Modularity and Functions**  
The project was designed with a modular structure, ensuring code reusability and clarity. The following classes manage different aspects of the 3D scene:

* **SceneManager**: Handles object creation, transformations, rendering, and texture mapping.
* **ViewManager**: Manages camera navigation, keyboard/mouse inputs, and projection mode switching.
* **ShaderManager**: Handles shader programs, passing uniform variables such as lighting properties and transformations.

Functions such as SetTransformations(), SetShaderTexture(), and PrepareSceneView() were used to **keep rendering efficient and maintainable**. The modular approach ensures scalability, making it easy to add new objects or modify scene properties.

**Conclusion**  
This project successfully demonstrated **core concepts in 3D computer graphics**, including **object modeling, texturing, lighting, and camera navigation**. The modular design allowed for efficient development and easy extensibility. Implementing interactive navigation and dynamic lighting added a layer of realism to the scene. Future improvements could include adding **shadows, reflections, and animation** to enhance interactivity and realism further.