**Design Decisions Document**

**Project Name:** OpenGL 3D Scene with Camera Controls  
**Author:** Umar Asif  
**Date:** 3/2/2025

**1. Introduction**

This document outlines the key design decisions made in the development of an OpenGL 3D scene with camera controls. The project involves implementing interactive camera movements, including panning, zooming, and orbiting, using both keyboard and mouse inputs.

**2. Project Goals & Requirements**

The main objectives of this project are:

* To create a navigable 3D scene using modern OpenGL.
* To implement interactive camera controls (pan, zoom, orbit).
* To ensure efficient rendering and smooth performance.
* To structure the code for readability, modularity, and reusability.

**3. Design Decisions**

**3.1 Choice of Graphics API**

* **Decision:** Used OpenGL (version X.X) for rendering.
* **Reasoning:** OpenGL is a widely used, cross-platform API that provides flexibility for implementing custom rendering techniques.

**3.2 Camera Implementation**

* **Decision:** Implemented a perspective camera with first-person and orbital controls.
* **Reasoning:** A perspective camera provides a more natural depth perception for 3D scenes. The combination of first-person movement and orbiting allows for flexible navigation.

**3.3 Camera Control Mechanisms**

* **Keyboard Input:** Used for basic movement (WASD for translation).
* **Mouse Input:** Used for rotation and zoom.
* **Reasoning:** This control scheme is intuitive and follows industry standards in 3D applications.

**3.4 Scene Representation & Object Loading**

* **Decision:** Objects are loaded using vertex buffers and shaders.
* **Reasoning:** Using vertex buffers and shaders ensures efficient rendering and enables advanced graphical effects.

**3.5 Shader Program Design**

* **Decision:** Implemented separate vertex and fragment shaders.
* **Reasoning:** This separation enhances flexibility, allowing modifications to lighting and material properties without affecting geometry calculations.

**3.6 Lighting Model**

* **Decision:** Used Phong shading for realistic lighting.
* **Reasoning:** Phong shading provides smooth shading and specular highlights, enhancing the visual appeal of the scene.

**3.7 Performance Optimization**

* **Decision:** Implemented frustum culling and reduced draw calls.
* **Reasoning:** These optimizations improve rendering performance by only processing visible objects and minimizing state changes.

**3.8 Code Structure & Modularity**

* **Decision:** Separated concerns into different classes (e.g., Camera, Shader, Renderer).
* **Reasoning:** This modular approach makes the code easier to maintain and extend.

**4. Challenges & Trade-offs**

* **Challenge:** Balancing performance and visual quality.
* **Trade-off:** Chose real-time rendering optimizations over highly detailed models.
* **Challenge:** Managing input handling smoothly across multiple control schemes.
* **Solution:** Implemented a state-based input manager to handle conflicts between keyboard and mouse inputs.

**5. Future Improvements**

* Implementing a skybox for a more immersive environment.
* Adding support for additional camera modes (e.g., free-look).
* Optimizing shadow mapping for improved realism.

**6. Conclusion**

This document outlines the major design decisions made in the OpenGL project. The chosen approaches balance performance, usability, and visual quality, resulting in an interactive and well-structured 3D scene.