**Final Project Reflection**

Nathalie Morales

CS-330

Southern New Hampshire University

10/19/2024

The 3D scene was designed to represent a modern desk setup, featuring objects like a lamp, a monitor, and a pencil holder. These objects were chosen for several reasons:

1. Relevance: They represent common items found in a typical workspace, making the scene relatable and realistic.

2. Geometric Variety: Each object offers different geometric challenges (e.g., the cylindrical lamp base, the flat monitor screen), allowing for a demonstration of various 3D modeling techniques.

3. Lighting Opportunities: Objects like the lamp provide natural focal points for lighting, enhancing the scene's visual depth.

Programming for the required functionality involved:

1. Modular Design: Creating separate functions for each object (CreateLamp(), CreateMonitor(), etc.) to improve code organization and reusability.

2. Shader Utilization: Implementing shaders to handle lighting and textures, allowing for realistic object rendering.

3. Texture Mapping: Applying different textures to objects to enhance visual detail and realism.

4. Transformation Matrices: Using transformation matrices for precise object positioning and scaling within the 3D space.

This approach allowed for a flexible, maintainable codebase that can be easily extended or modified for future enhancements.

Navigation in the 3D scene is controlled through a virtual camera, which is manipulated using various input devices:

1. Keyboard Controls:

- WASD keys: Used for forward, backward, left, and right movement respectively.

- QE keys: Used for upward and downward movement.

- This setup allows for intuitive 3D movement similar to many first-person video games.

2. Mouse Controls:

- Mouse movement: Changes the camera orientation, allowing the user to look around the scene.

- Mouse scroll wheel: Adjusts the speed of camera movement, providing finer control over navigation.

3. Implementation Details:

- The camera is implemented using a Camera class that stores position, orientation, and movement parameters.

- Input is captured in the ProcessKeyboardEvents() and Mouse\_Position\_Callback() functions.

- The camera's view matrix is updated based on these inputs and passed to the shader for rendering.

This combination of keyboard and mouse input provides six degrees of freedom, allowing users to fully explore the 3D environment from any angle or position.

Several custom functions were developed to enhance code modularity and organization:

1. CreateGLTexture(const char\* filename, std::string tag):

- Purpose: Loads texture images and converts them to OpenGL texture data.

- Reusability: Can be used for any texture in the project, identified by a unique tag.

2. SetTransformations(glm::vec3 scaleXYZ, float XrotationDegrees, float YrotationDegrees, float ZrotationDegrees, glm::vec3 positionXYZ):

- Purpose: Sets up transformation matrices for objects in the 3D scene.

- Reusability: Can be applied to any object, allowing for consistent positioning and scaling.

3. CreateLamp(), CreateMonitor(), CreatePencilHolder():

- Purpose: Each function creates a specific object in the scene.

- Modularity: Encapsulates all the steps needed to create each object, making the main rendering loop cleaner and more organized.

4. SetupSceneLights():

- Purpose: Configures all lighting parameters for the scene.

- Modularity: Centralizes lighting setup, making it easier to adjust and maintain.

5. DefineObjectMaterials():

- Purpose: Sets up material properties for different objects.

- Reusability: Can be easily expanded to include new materials as needed.

These functions contribute to a more organized and maintainable codebase by:

- Encapsulating specific functionality, adhering to the Single Responsibility Principle.

- Providing clear interfaces for complex operations, improving code readability.

- Allowing for easy reuse of common operations across different parts of the project.

- Facilitating easier updates and modifications to specific aspects of the 3D scene.