**Zakk Car**

**Reflection**

For my 3D scene, I chose to recreate a simple desktop workspace. The objects in the scene include a flat floor plane, a rectangular monitor, a book, a keyboard, and a basic lamp composed of simple geometric primitives like boxes and cylinders. I selected these objects because they are familiar, easy to represent with basic shapes, and align well with the project’s requirement to replicate a scene from a 2D image. Additionally, using low-polygon models allowed me to focus on implementing core OpenGL functionalities such as transformations, texture mapping, and lighting without getting bogged down in complexity.

To meet the required functionality, I programmed the scene using modular methods such as PrepareScene() and RenderScene(). These methods ensure that all geometry, textures, and transformations are initialized cleanly and rendered in the correct order. I used helper functions like SetTransformations(), SetShaderTexture(), and SetShaderMaterial() to simplify repetitive logic and allow for better code reusability.

User navigation in the scene is controlled by a virtual camera system. I implemented camera movement using keyboard input (WASD for forward, backward, and strafing; QE for up and down movement) and mouse movement to control the view orientation. The scroll wheel adjusts the zoom level, making it intuitive to explore the scene. These controls were integrated through the ViewManager class, which captures and processes input to update the view and projection matrices dynamically.

To keep the code organized and maintainable, I created custom functions such as CreateGLTexture() to handle texture loading, and SetTextureUVScale() to easily adjust how textures tile across different surfaces. These reusable functions abstract away OpenGL calls and make it easier to apply consistent behavior across multiple shapes in the scene. The modular design also makes it easy to expand the project in the future by simply adding new shapes or materials and calling the existing helper functions.

Overall, this project helped reinforce key graphics programming concepts and taught me how to structure a 3D application using best practices for modular design, input handling, and scene composition.