REFLECTION

For my 3D scene, I chose to replicate a modern workspace based on a 2D reference image. I selected real world objects such as a desk, dual monitor setup, keyboard, digital clock, lamp, and a vase, aiming to create a believable and functional desktop scene. I chose these objects because they each presented different modeling challenges and lead me to incorporate various basic shapes like boxes, cylinders, and spheres. This variety helped me meet the project’s requirement of using at least four distinct primitive shapes. I also combined multiple shapes into a single object, such as using a cylinder and a box to form the lamp. As well as a box and cylinder for the monitors.

My development choices were guided by simplicity, I used low-polygon models to stay under the triangle limit and carefully worked on proportions, placement, and rotation so that all objects were spaced realistically. Textures were applied to four objects: the desk, using a wood texture, and the lamp and keyboard, using a aluminum texture and the monitors using a plastic texture. For materials, I added glossy effects to the clock and vase to and to reflect lighting more naturally. I adjusted lighting a lot to create contrast and depth: a soft cool white light simulates daylight from the right, while a warm lamp light from the left adds mood and highlights. This addressed earlier feedback I had got suggesting that my scene lacked lighting contrast and realism.

To allow users to navigate the scene, I implemented a free roaming camera controlled by keyboard and mouse input. The WASD keys are used for forward, backward, left, and right motion, while the Q and E keys move the camera vertically. Mouse movement controls the angle of the camera, and the scroll wheel adjusts the camera speed for moving control. This interaction setup ensures the user can explore the scene fully in all directions and view it from multiple angles.

My code is structured around modular and reusable functions. For example, the SetTransformations() function handles all position, rotation, and scale settings for any mesh object, which simplifies rendering multiple shapes. Similarly, the SetShaderMaterial() and SetShaderTexture() functions apply material and texture settings without reusing code. This modular approach made it easier to test and revise different visual elements throughout development. It also follows coding best practices.

Overall, this project helped me understand how to simulate a real-world environment using basic geometry and lighting principles. I learned how to structure OpenGL code for clarity and performance, and how to add interactivity and visual design to create a complete and polished 3D experience.