**Design Decisions and Reflection**

# Justification of Development Choices

For the 3D scene, I selected a variety of simple objects that could be easily represented using primitive shapes such as cubes, spheres, and cylinders. These objects were chosen because they are representative of common real-world items, and using low-polygon versions of these shapes allowed me to create an optimized 3D environment with minimal rendering cost. One of the key objects in the scene was created by combining multiple shapes, such as a cylinder and a sphere, to mimic a more complex real-world item. This modular approach to building objects helped reduce the complexity of the scene while still achieving a recognizable result.

The use of basic shapes allowed for a straightforward application of textures. I selected royalty-free textures for two of the objects to enhance their realism. Textures were applied using the texture loading function built into the code. This function handles image loading and applies textures efficiently to 3D objects in the scene, ensuring that the objects appear polished and visually appealing. The lighting in the scene was designed to create a balanced and polished presentation, with both ambient and colored light sources illuminating the objects. I included point light and directional light, ensuring that objects could be viewed clearly from different angles.

# User Navigation

The user can navigate the 3D scene using a combination of input devices. The WASD keys control the camera movement in the forward, backward, left, and right directions, while the QE keys enable vertical movement. Mouse movement allows the user to change the camera’s orientation, offering flexibility in viewing objects from different angles. Additionally, the mouse scroll function was implemented to adjust the speed at which the camera moves. This allows the user to have more precise control over their navigation through the scene, making the experience smooth and intuitive.

The camera controls were designed with modularity in mind, allowing for easy adjustments to the speed or range of motion. The inclusion of both perspective and orthographic views provides the user with different ways to visualize the 3D scene, which can be toggled with a keyboard input.

# Modularity and Reusability

The program was designed with modularity as a key focus. Custom functions were developed to handle specific tasks such as texture loading, lighting setup, and camera control. For instance, the texture loading function accepts a file path and handles the entire process of loading and binding the texture to a 3D object. This function can be reused for any additional objects that require textures, without the need for re-writing code.

Similarly, camera controls were abstracted into reusable functions that handle input and movement. The modular nature of these functions allows the camera system to be easily modified or extended, whether for adding new input controls or adjusting the behavior of the camera in different scenes. This design choice ensures that the codebase is maintainable and scalable, allowing new features to be integrated with minimal changes to existing functions.