# Design Decisions and Reflection

For my final 3D OpenGL scene, I selected a practical workspace setup as depicted in a reference image featuring a laptop, a coffee mug, and a wireless mouse arranged neatly on a wooden table. I chose this arrangement because it allowed me to work with common real-world items that could be constructed using a variety of basic 3D shapes while still presenting challenges with texturing, scaling, and spatial alignment. The scene also lends itself well to realistic lighting and shading, creating a grounded and relatable virtual environment.

The four required objects—table, laptop, mouse, and mug—are all constructed using combinations of basic primitive shapes such as boxes, planes, cylinders, spheres, and a torus. The table was built first using a scaled box for the top and cylinders for the legs and supports. The laptop was constructed using two boxes: one for the keyboard base and one for the screen, with the screen rotated back slightly to mimic a realistic open angle. The mouse was represented using a scaled sphere to form a smooth ergonomic body. The mug body was made using a vertical cylinder and the handle was modeled using a torus, rotated and scaled to attach cleanly to the side.

Textures played a significant role in adding realism to the scene. A wood grain texture was applied to the table to make the surface visually appealing and consistent with the reference image. The laptop keyboard used a realistic key layout texture mapped onto a flat plane above the base. The mug uses a custom ceramic green-and-brown porcelain texture to give it a rich, tactile appearance. All textures were carefully UV-scaled to ensure they did not appear stretched or pixelated.

To illuminate the scene, I implemented two light sources: one white point light and one colored directional light. Both lights use the Phong shading model, incorporating ambient, diffuse, and specular components to realistically simulate light behavior on different surfaces and materials. The lights are positioned so that they capture all objects in the scene and highlight surface textures without casting hard shadows or dark regions that obscure the geometry.

User navigation in the 3D scene is handled via keyboard and mouse inputs. The WASD keys move the camera forward, backward, left, and right, while the Q and E keys allow for vertical movement along the Y-axis. The mouse cursor is used to change the pitch and yaw of the camera, enabling users to look around the scene from a fixed position. Scrolling the mouse wheel adjusts the speed of the camera movement, giving users greater control when exploring the scene.

From a coding perspective, I modularized my project using helper functions such as SetTransformations(), SetShaderTexture(), and SetShaderColor(). These functions abstract away repetitive OpenGL commands, making the codebase easier to read and maintain. For instance, SetTransformations() takes parameters for scale, rotation, and position and handles the creation of the final transformation matrix. This makes it easy to transform multiple objects without duplicating transformation logic. Similarly, SetShaderTexture() allows for dynamic assignment of texture IDs by tag, keeping the scene render loop cleaner and more focused on object definitions.

Overall, this project demonstrates a solid understanding of 3D object modeling, lighting, texturing, and camera control using OpenGL. Each object was purposefully selected to fulfill specific project requirements while also encouraging experimentation with primitive composition, texturing, and interactive viewing. Through this process, I gained deeper experience in building reusable graphics functions and composing realistic, interactive virtual environments.