**Design Decisions Reflection**

**Scene Objects and Development Choices**

For my 3D scene, I chose a realistic desktop environment with a table surface, wall backdrop, stack of books, glass mug, laptop, and desk lamp. These objects were selected because they are composed of basic geometric primitives (planes, boxes, cylinders, torus meshes) that can be transformed and textured to create familiar real-world items. This made the project an opportunity to demonstrate how simple shapes can be combined into more complex models, while also giving the scene a cohesive “study desk” theme.

The books provided an opportunity to showcase multiple textures, including leather-like covers and paper pages, which required distinct material properties. The mug and coffee allowed me to experiment with transparency and reflective surfaces by applying glass and liquid materials. The laptop was included because it combines metallic and screen surfaces, requiring multiple materials with different shininess and diffuse properties. Finally, the lamp demonstrates articulated components with various scaling and rotation transformations, showing how modular functions such as SetTransformations can be reused across multiple objects.

**Lighting Design**

I set up four light sources in the scene to ensure all objects are fully visible and realistic shadows could be emphasized. The first two lights were placed diagonally opposite each other to provide general ambient and diffuse illumination. A third light was placed above the scene to simulate an overhead room light, and the fourth was positioned at desk level to represent the lamp. The use of multiple sources allowed balanced lighting without washing out textures, while still giving enough highlights to show the shininess of glass, metal, and polished surfaces. The shader values were adjusted to balance ambient, diffuse, and specular effects across different materials.

**Texture and Material Decisions**

Textures were chosen carefully to reinforce realism. The table used a wood texture with tiling, while the wall had a subtle, light-colored pattern to avoid drawing focus away from the desk items. Books used multiple cover textures and separate paper textures to highlight contrast between glossy and matte surfaces. The mug and coffee required two different textures, one emphasizing transparency and reflectivity for glass, and the other giving the liquid a dark, specular finish. The laptop used two textures as well, metal for the body and a glowing screen image.

Each texture was paired with a corresponding material definition that set ambient, diffuse, specular, and shininess values. For example, the “leather” material used muted specular values with warm diffuse tones, while “laptopMetal” used higher shininess and specular intensity to achieve a soft metallic effect. This modular material setup, stored in DefineObjectMaterials, allowed reusability across multiple objects while keeping the rendering consistent.

**Camera and Navigation Controls**

User navigation was enabled through both keyboard and mouse inputs. Movement along the horizontal plane is controlled with the W, A, S, and D keys, while vertical movement is handled with the Q (up) and E (down) keys. This provides full six-degree freedom of movement within the scene. Mouse movement controls camera yaw and pitch, allowing the user to look around naturally. The scroll wheel adjusts keyboard control speeds, giving smooth control over perspective.

These controls were implemented in the ViewManager class using GLFW input callbacks. Functions such as ProcessKeyboardEvents, ProcessMouseMovement, and ProcessMouseScroll modularize the input handling, allowing the camera to update its position and orientation consistently each frame. This ensures that navigation feels intuitive and responsive, much like in a 3D modeling program or game environment.

**Modular Code Functions**

Several custom functions helped keep the code modular and reusable. The SetTransformations function encapsulates scaling, rotation, and translation for each object, avoiding repeated matrix code in the rendering section. SetShaderTexture and SetShaderMaterial abstract away texture binding and material parameter setting, allowing quick application of visual properties to any object. Similarly, LoadSceneTextures, DefineObjectMaterials, and SetupSceneLights centralize initialization logic, keeping the rendering code clean and focused only on drawing.

By breaking functionality into these reusable functions, I ensured that the program remained organized and easy to extend. For example, adding a new object to the scene requires only defining its transformations, material, and texture references without duplicating lighting or shader setup code. This modularity supports scalability and makes debugging simpler.

**Conclusion**

In summary, my design choices were guided by the goal of building a realistic, desk-themed 3D scene while demonstrating key OpenGL techniques. Objects were chosen for their ability to be constructed from basic meshes, textures were applied to highlight surface differences, and materials were tuned to reflect real-world lighting interactions. Navigation was designed to provide users with full control of the camera using both keyboard and mouse, while modular functions kept the program organized and reusable. This combination of scene design, lighting, materials, and navigation ensured a functional and visually engaging 3D environment.