Design Reflection

The 3D scene presents a cozy and realistic breakfast table setup, featuring a wooden table with a plate of food, a napkin, a coffee mug, a small plate with grapefruit, and a set of cutlery. The goal was to capture the familiar details of a simple breakfast environment while adding enough realism to create an immersive experience. Textures such as "woodfortable.jpg" for the table, "baconstrips.jpg" for the bacon, and "stainlesssteel.jpg" for the cutlery were selected to replicate real-world materials. Lighting and material properties were carefully configured to enhance the scene's visual appeal, including the subtle reflections on the plate and mug and the metallic sheen of the cutlery.

To maintain functionality and organization, modular programming techniques were implemented. Functions like CreateGLTexture manage the loading and binding of textures efficiently, reducing redundancy in the code. The SetTransformations function centralizes scaling, rotation, and positioning, enabling consistent object placement with a single reusable method. For instance, RenderTable handles the table’s dimensions and position, while RenderCutlery creates the fork and knife with detailed proportions and realistic textures. This modular approach ensures cleaner code and allows for quick adjustments or expansions of the scene.

User navigation in the 3D scene is intuitive, achieved through simple keyboard and mouse controls. The WASD keys enable horizontal and vertical movement of the camera, while mouse input adjusts the view angle by modifying yaw and pitch. Additionally, the P and O keys allow toggling between 2D and 3D views, providing flexible exploration. The navigation system leverages the glm::lookAt function to dynamically update the view matrix, ensuring smooth and responsive controls for an engaging user experience.

Several custom functions were created to ensure organization and reusability in the code. The SetTransformations function simplifies scaling, rotation, and positioning, allowing for consistent application across all objects. Efficient texture management is achieved through CreateGLTexture, which prevents duplication and optimizes memory usage. Supporting functions like SetupSceneLights and DefineObjectMaterials manage lighting configurations and material properties, such as directional lighting to simulate sunlight or reflective surfaces for objects like the cutlery. For example, RenderEgg combines a flattened cylinder for the egg white and a sphere for the yolk, each with unique textures, to create a realistic representation of a sunny-side-up egg.

The final scene strikes a good balance between simplicity and realism. Modular programming and reusable functions kept the development process organized and easy to manage, while the intuitive controls make navigating the scene smooth and visually appealing. Overall, the project was really fun, and I believe I’ve created a solid foundation for any future improvements or additions.