# CS 330 Final Project Reflection

For my final project in CS 330, I recreated a realistic 3D scene inspired by a reference photo of a simple, warmly lit tabletop arrangement featuring a mug, a bowl of apples, and a hanging lamp. The purpose of this project was to demonstrate my ability to model, texture, light, and render a cohesive scene using OpenGL, while also applying modular code design and interactive camera controls. This reflection explains the development choices made for my scene, how user navigation is implemented, and how custom functions were designed to maintain an efficient and organized structure.

## Development Choices

I chose this reference image because it presented a realistic challenge that combines soft lighting, textured materials, and accurate object placement. The scene includes a wooden table, a ceramic mug, a bowl filled with apples, and an overhead lamp. I constructed each object using simple primitive shapes: the table was made from a box and cylinders for the legs, the mug used a cylinder with a torus handle, the bowl was a tapered cylinder, and the apples were created from multiple spheres. The hanging lamp consists of a tapered cylinder for the shade and a thin cylinder for the stem, demonstrating shape composition.

Three textures were applied across the scene: a rustic wood texture for the table and floor, a ceramic texture for the mug and bowl, and an apple texture for the fruit. Each texture was carefully scaled and mapped to match realistic proportions. Lighting was a critical focus of my project. I implemented a warm overhead light to emulate the glow of the hanging lamp, combined with ambient lighting to soften shadows. This lighting setup used the Phong shading model, incorporating ambient, diffuse, and specular components for realism. The specular intensity and shininess values were adjusted per object to produce the reflective qualities of ceramic and the glossy surface of the apples.

## User Navigation

The user can navigate the 3D scene using intuitive controls that provide freedom of movement and viewing. The WASD keys control forward, backward, and side-to-side motion, while the Q and E keys move the camera downward and upward, respectively. The mouse is used to control the camera’s orientation, allowing smooth pitch and yaw rotation to view the objects from any angle. Additionally, the mouse scroll wheel adjusts the camera movement speed dynamically, allowing finer control when inspecting details. These controls provide a natural, fluid navigation system that enhances interactivity and mirrors typical 3D camera behavior in modern engines.

## Custom Functions and Code Organization

To ensure my program was organized, modular, and maintainable, I utilized several custom helper functions to manage transformations and rendering. For example, the SetTransformations function accepts scale, rotation, and position values to handle matrix transformations consistently across all objects. The SetShaderColor function simplifies the process of updating color uniforms in the shader program, improving readability and reducing redundant code. These functions promote clean structure and make it easier to modify or expand the scene in future projects. Each object in the scene follows a clear pattern: transformation setup, shader configuration, and mesh rendering. This modular approach ensures consistent lighting behavior and keeps the RenderScene function well-organized.

Overall, this project successfully demonstrates the integration of 3D modeling, lighting, texturing, and camera control within OpenGL. Through modular programming practices, realistic materials, and careful attention to lighting, the final render closely replicates the chosen reference photo. The process has reinforced my understanding of computer graphics fundamentals and the importance of iterative refinement when building a complex 3D scene.