Kaley Lavery

CS-330: Computational Graphics and Visualization

Dr. Aly

October 16, 2025

Final Project

For my final 3D project, I created a small desk workspace scene consisting of a laptop, soda can, pencil cup, and pencils placed on a desk surface. This scene was inspired by my original 2D image reference from Milestone One. Each model was constructed using low-polygon primitive shapes to maintain efficiency and clean geometry. The desk uses a plane mesh, the laptop combines box and tapered cylinder meshes for the base and hinges, the soda can combines a box and sphere, and the pencil cup and pencils use cylinders and cones.

All models were scaled and positioned carefully using transformation matrices in SetTransformations() to match the proportions of the reference image. The entire scene uses under 1,000 triangles per object to ensure real-time performance and rendering clarity.

Two primary textures were used. Metal texture for the laptop surfaces to simulate a brushed aluminum appearance. Rubber/Wood texture for the desk surface to emulate a practical workspace feel. These were loaded using the CreateGLTexture() and BindGLTextures() functions, then applied within the RenderScene() method through SetShaderTexture() and SetTextureUVScale(). This established realistic surface variation and contrast between objects.

To enhance visual realism, I implemented two point light sources in SetupSceneLights(). A white key light positioned above and to the right provided the primary illumination, using balanced ambient, diffuse, and specular values. A pink fill light placed to the left softened shadows and added color tone variation for a polished appearance. Both lights use the Phong lighting model, with parameters tuned to avoid overexposure. The plane (desk surface) is fully lit and reflective, while smaller props (laptop, pencils, and can) are rendered without active lighting for performance and clarity.

Material properties were defined in DefineObjectMaterials(), setting diffuse and specular reflectivity and a shininess factor of 16. The result is a cohesive, well-lit environment that meets the milestone criteria for ambient, diffuse, and specular components.

Camera navigation was implemented through the ViewManager and Camera classes. W/A/S/D was implemented to move forward, left, backward, right, while Q/E controlled movement up and down. Mouse movement was used look around for yaw/pitch rotation. Scroll wheel adjusts movement speed dynamically. O/P keys were utilized toggle between Orthographic and Perspective projection views. This control scheme allows the user to explore the entire 3D environment interactively, meeting all rubric requirements for horizontal, vertical, and depth navigation.

The project code is modular and adheres to best practices through the usage of:

- PrepareScene() loads all meshes, textures, and initializes lights.

- RenderScene() contains all transformation, texturing, and lighting toggles for object rendering.

- SetupSceneLights() and DefineObjectMaterials() modularize illumination setup and material definitions.

- PrepareSceneView() handles real-time camera updates, including projection matrix application and user inputs.

- Mouse\_Scroll\_Callback() dynamically modifies camera speed to enhance control precision.

Code readability was improved with consistent indentation, descriptive comments, and logically separated methods.

This final project demonstrates my ability to apply C++ and OpenGL programming techniques to create an interactive, textured, and illuminated 3D environment. The workflow—from object modeling to texturing, lighting, and navigation—reflects both computational graphics principles and industry best practices. The finished scene achieves realistic depth and lighting while maintaining simplicity, efficiency, and intuitive user control.