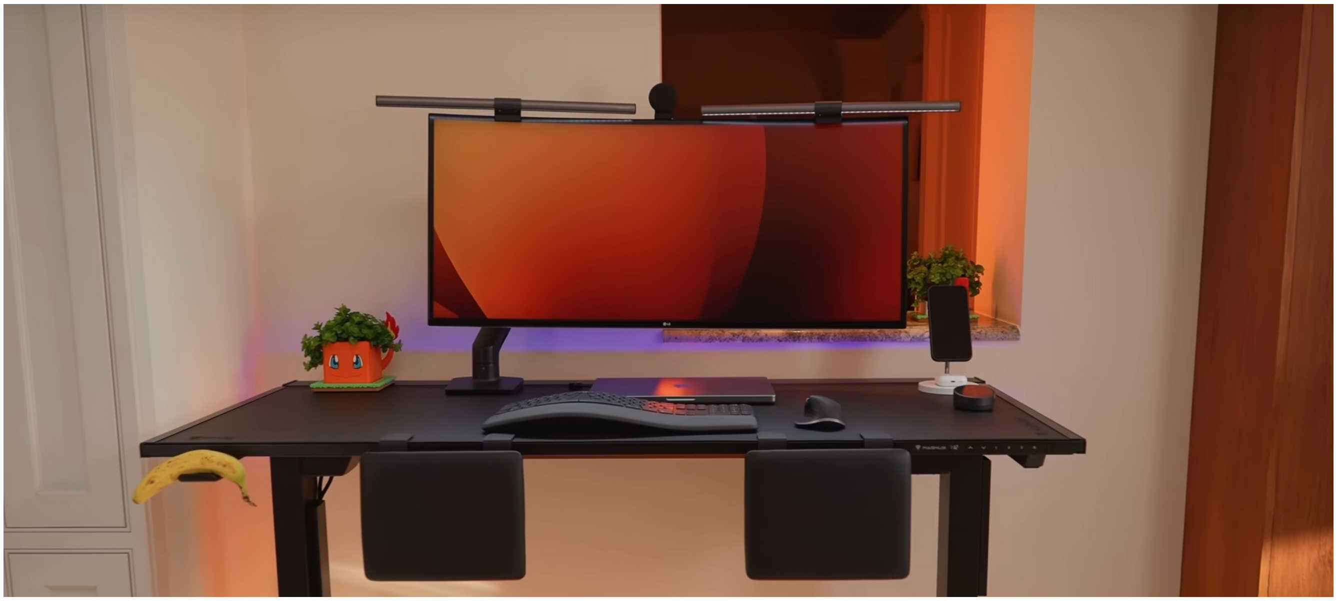
Daniel Aguiar
03/02/2025

Introduction

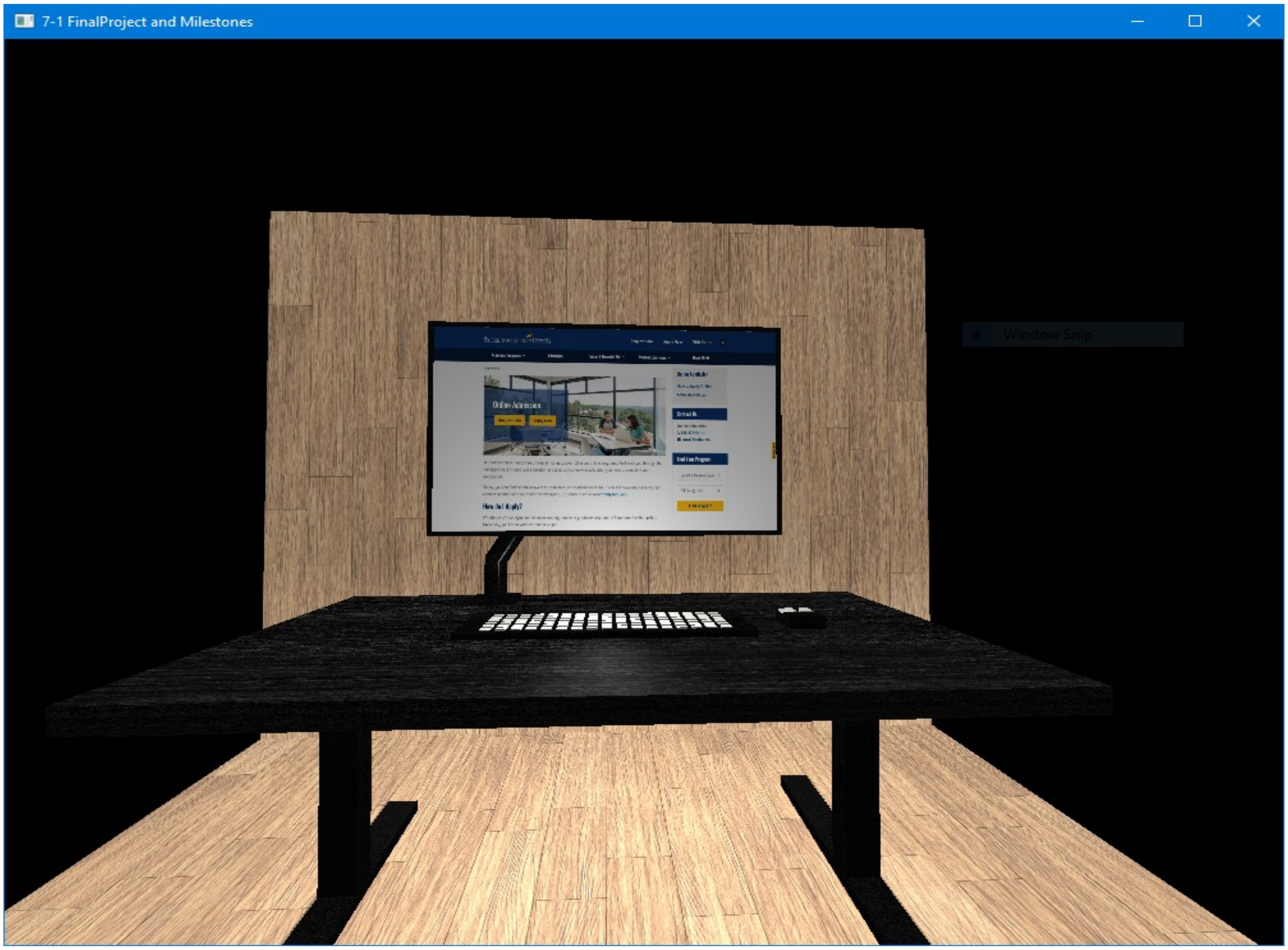
This project tried to recreate a basic 3D scene from a 2D reference image, showcasing basic modeling, texturing, lighting, and implementing camera movement features using OpenGL and C++. The completed scene features a desk, monitor, keyboard, and mouse. I did use some creative liberties when creating my version of the by for example displaying Southern New Hampshire University's (SNHU) webpage on the monitors screen.

Reference Image:



Daniel Aguiar
03/02/2025

My Version of the Scene:



3D Objects and Materials

All of the objects were made with low-poly primitive forms (box and cylinder meshes) to simplify the modeling process and build on the idea of combining primitives to build more complex shapes/geometry. The desk was made from several box meshes rescaled to create the tabletop and legs, with a different material applied for wood and metal finishes. The stand and bezel of the monitor also use box and cylinder meshes, and the screen itself uses an image-based texture on the SNHU homepage screenshot. The mouse and keyboard were also constructed from box meshes, smaller boxes forming individual buttons and keys. Every object has its own individual material, e.g., a black brushed metal with increased specular values for the metallic components or a black wood texture for the desk surface, to attain differentiated surface qualities and textures.

Lighting

A directional light approximates a sunlight-like source with a moderate ambient term and equal diffuse and specular values. A point light above the desk is used to fill in shadows. Both lights use the Phong shading model, which blends ambient, diffuse, and specular contributions to provide smooth lighting from numerous viewing angles.

Camera Navigation

Movement within the scene is controlled by the W, A, S, and D keys to move horizontally, Q and E to position vertically, and movement of the mouse to adjust pitch and yaw. The scroll wheel on the mouse controls the speed of movement. This allows users to see each angle of the scene so that each object and texture can be viewed in detail.

Code Structure

PrepareScene deals with material, texture, and light setup, whereas *RenderScene* regulates the rendering of objects by invoking my custom added methods (*RenderDesk*, *RenderMonitor*, *RenderKeyboard*, and *RenderMouse*). Transformation and texture assignment routine tasks are encapsulated in helper functions (*SetTransformations*, *SetShaderMaterial*, and *SetShaderTexture*) for convenient maintenance and future upgrade.

Daniel Aguiar
03/02/2025

Conclusion

By employing a mix of basic geometry, fitting materials that match appropriately, well-adjusted lighting, and natural 3D camera movement, this project effectively replicates a 3D work/computer environment from my selected 2D reference image. While my virtual 3D scene is far from perfect or photo-realistic, I feel there is enough to showcase proficiency in 3D graphics fundamentals and hope you enjoy my project!