**7-1: Final Project: Design Decisions**

Jerris English

Southern New Hampshire University

CS-330 Comp Graphic and Visualization

Prof Malcolm Wabara

August 24, 2025

**Project Concept and Object Selection**

For my final project, I decided to recreate a scene from the hallway of my home. It includes a lamp placed on top of a cabinet with a small mirror mounted above. I walk past this setup every day, so it felt like a personal and familiar reference point to work from. I chose this scene because I knew the layered lamp structure and its texture details would push me creatively. I wanted to challenge myself this term by focusing on transparency, texture realism, and overall object placement.

**Modeling Approach and Scene Layout**

To construct the 3D scene, I started by breaking the lamp down into core primitive shapes, primarily cylinders and spheres, and carefully scaled and stacked them to match the structure from my reference image. I created the cabinet using box meshes and added small leg pegs to give it a grounded and realistic presence. The mirror frame was assembled from simple textured box tiles and positioned directly on the wall above the rest of the scene to anchor the layout visually.

Throughout the modeling process, I focused heavily on small details, even down to the tiny details in the gold clasps on the cabinet doors that appear in my actual hallway. I also replicated the epoxy resin shine of the floor and the subtle sheetrock texture on the wall to bring more realism into the final scene. To ensure texture accuracy, I took high-resolution close-up photos of these real-life surfaces and used them to create custom shader textures that mapped cleanly onto the appropriate objects. This included specialty textures such as the chevron fur for the decorative box and zebra fur for the mirror panels.

From a technical perspective, I built the entire scene using OpenGL in Visual Studio. I used the mesh setup functions from the course’s starter files to load base shapes like cylinders, boxes, and spheres. All objects were positioned precisely in the 3D space using X, Y, and Z coordinates based on the layout in my reference photo. I used the STB image loader to bring in texture files and mapped them onto surfaces using UV scaling through the shader manager. This setup ensured that transparency, reflections, and lighting behaved as intended. I also used a variety of shader materials such as frosted glass for the dome, copper metal for the base, and dark-stained wood for the cabinet. These visual details helped match the tone of the original reference while keeping each object under the one thousand triangle limit.

**Lighting Setup and Effects**

To bring the entire scene to life and control how materials respond, I implemented a six light system using the full Phong shading model. This included a mix of directional and point lights for layered realism and glow simulation.

First, I placed a soft directional light overhead to evenly illuminate the room and provide a neutral base tone across all surfaces. To simulate the lamp bulb’s glow, I added two point lights. One was placed directly at the center of the bulb to act as the main emission source, and a second glow boost was positioned slightly offset to reinforce the light diffusion through the frosted glass dome.

To balance shadows and enhance visibility across the entire setup, I added two warm point lights on the left and right sides of the scene. These helped highlight the mirror and cabinet edges, soften harsh shadows, and bring out the texture in the wall and floor materials. Finally, I introduced a central neutral point light positioned slightly forward to brighten the immediate area and anchor the scene’s core visual space.

Each of these six light sources incorporates ambient, diffuse, and specular components, allowing polished surfaces like the copper base, resin coated top, and glossy floor to reflect natural highlights based on camera angle. This lighting strategy helped push the realism of the scene while keeping the object details clear and grounded.

**Navigation and Camera Controls**

The original source code already included basic camera movement mapped to the W, A, S, and D keys for navigating forward, left, backward, and right. I expanded on those controls to support full 3D movement and improve scene interactivity. Specifically, I added Q to move the camera upward and E to move it downward, allowing users to explore the scene vertically as well as horizontally.

The yaw and pitch controls for mouse input were already integrated into the project, allowing users to look around by dragging the mouse. However, I adjusted the sensitivity settings to create a smoother and more manageable experience. I brought the yaw speed down to 0.2, which reduced the jumpiness of the camera and gave the user more precise control when orbiting around the scene. I also connected the scroll wheel to adjust the camera’s movement speed dynamically, making it easy to slow down for detailed views or speed up when covering longer distances.

To round out the camera system, I added quick toggles for switching projection views. Pressing O activates an orthographic view, which flattens the 3D depth and gives everything a straight-on, photograph-like appearance as if the camera is facing objects directly. Pressing P switches back to perspective view for a more natural 3D layout with depth. These extra touches made it much easier to test and view the scene from different angles without having to manually reposition the camera every time.

**Modular Functions and Reusability**

To keep the code organized and flexible, I made consistent use of several built-in helper functions throughout the project. These included SetTransformations, SetShaderColor, SetShaderMaterial, SetShaderTexture, and SetTextureUVScale. Each function played a key role in simplifying my workflow. They allowed me to position, scale, color, and texture objects with just a few clear lines of code. This helped keep the overall scene logic clean and modular, making adjustments much easier as the project evolved.

For example, SetTransformations allowed me to apply scale, rotation, and position all at once, which kept object placement efficient and readable. I reused this method across every shape in the scene, from the lamp’s intricate layers to the textured frame of the mirror. Similarly, SetShaderMaterial and SetShaderTexture let me swap out shader visuals quickly, whether I was applying frosted glass, copper metal, or textured zebra fur, without duplicating code. By working with these reusable functions, I was able to focus more on the creative side of layout, lighting, and material behavior while still keeping the codebase clean and professional.

**Conclusion**

This project pushed me to take my OpenGL skills further than I expected. I learned how to work with lighting, transparency, and texture blending in ways that made the final scene feel polished and immersive. I put a lot of care into high-detail elements like the transparent frosted glass lamp shade, the soft glow of the bulb with layered lighting, the custom fur textures on the mirror and box, and the small trim pieces on the cabinet that helped ground the space. Building the hallway scene from a real reference gave me a chance to combine technical execution with personal design choices, which kept the process both creative and challenging. Overall, I’m proud of how closely the final layout and materials match the original image and how smooth the user navigation feels within the 3D space.