7-1 Final Project Submission

Reflection

*Christopher Davidson | CS-330 Comp Graphic and Visualization | 4/20/25*

For this final project, I created a 3D scene that approximated a reference image of a PCBA sitting on an ESD mat. The image was based on a real setup I worked with, and the textures were drawn from actual design elements. The goal was to recreate the visual structure and material feel of the board using simplified geometric shapes and shader-based rendering techniques within OpenGL. I selected components L901, L903, C234, and U902 from the reference photo as focal elements. While the surrounding context was kept minimal to direct attention toward those highlighted objects.

The base plane of the scene represented the ESD mat on a workbench. I applied a large scale plane mesh to act as the surface and overlaid it with a textured material simulating the padded, matte surface of a typical anti-static mat. Above this, the PCBA was placed and modeled using a scaled box mesh. The texture used here was not a generic one; instead, I generated it myself from a real board's silkscreen layer. I took a PNG export of the layer, laid it over a solid green background to mimic the standard solder mask color, and applied this as the top-facing texture on the board mesh.

My object choices were focused on the components I found most visually distinct and structurally informative for this scene. These included two inductors, a capacitor, and a surface-mount IC. L903 and C234 were particularly detailed in terms of structure, combining multiple shapes such as boxes, spheres, and toruses to model their bodies, leads, and fillets. This allowed me to capture the volumetric complexity of how these components sit on a real board. The solder pads, leads, and wire connections were differentiated using specific shaders and materials to reinforce the contrast between metal surfaces and plastic or composite bodies. I designed these materials by tuning ambient, diffuse, and specular components based on whether the part was meant to be matte, metallic, or semi-reflective. The brushed metals and plastics came from royalty-free sources, including AmbientCG and image searches filtered for usage rights.

Lighting in the scene was handled using a combination of a directional light and a colored point light. The directional light simulated general illumination, while the point light, subtly tinted to reflect warm workspace lighting, added depth and contour to the components. This helped distinguish overlapping shapes, especially the curved edges of the toroidal inductors and the rounded solder joints. I also used materials with varying shininess to reflect how different substances respond to light: matte FR4 for the PCB, shiny metal for solder, and more diffused reflections for black plastic.

User navigation was a critical component in how I structured the scene. The camera was programmed to traverse in all directions using the WASD keys for planar movement and QE for vertical motion. The mouse controlled the pitch and yaw of the camera’s orientation, allowing the user to freely look around the board as if examining it on a desk.

Additionally, mouse scroll allowed the user to dynamically adjust camera speed, improving both fine inspection and broader movement. These inputs were implemented through GLFW callbacks and mapped to smooth transitions in the view matrix. To support different inspection contexts, I implemented the option to toggle between orthographic and perspective projection modes, giving users the ability to switch from a technical overview to a more realistic depth view of the scene.

The SetTransformations function was used extensively to apply scaling, rotation, and translation vectors to each object before rendering. This allowed me to treat object placement in a repeatable and flexible manner. I abstracted shader management and texture binding into their own functions, which made it easy to reuse materials and surface properties across different objects. The SceneManager class handled the entire sequence of loading textures, defining materials, setting up lights, and rendering meshes, which kept my main() function clean and focused on initializing the application and maintaining the render loop.

Looking back, I recognize where I could have made better choices during the planning phase. The reference photo I used was not ideal. It was busy and more complex than it needed to be. A more effective approach would have been to set up a mock board with only a few strategically placed components. This would have allowed me to better meet the scope of the assignment while focusing on higher visual fidelity for fewer elements. I also should have spent more time understanding the final project objectives early on. Time constraints in my real-world schedule unfortunately limited how much iteration I could do before settling on the final scene.

Nonetheless, the process of bringing together modeled shapes, custom textures, real lighting interactions, and interactive camera control provided a comprehensive experience that deepened my understanding of 3D graphics and scene composition. This project gave me the chance to build a complete, interactive, and visually representative model using fundamental techniques, and it helped me better appreciate the importance of early planning, clean code structure, and real-world constraints in graphics development.