**Reflection**

Carl LaLonde

Southern New Hampshire University

CS-330: Computer Graphics and Visualization

Instructor John Graham

June 22, 2025

Throughout the development of my 3D scene, a lot of thought went into the choices about what to include as well as how to put everything together. My 3D scene was simple, a workbench with a couple of basic tools and a toolbox laying on the tabletop. The original picture also included tools hanging from a pegboard behind the workbench. These items were excluded from my final project so I could spend more time on details for the few items that I did include. These items where a hammer, wrench, and a toolbox… and of course the workbench. The hammer was constructed from a few different meshes. For example, I made the body of the head using a box mesh, a cylinder for the pounding part of the head, and two tapered cylinders for the claw. The handle was a long cylinder with a wooden texture.

I found one of the more complicated pieces being the wrench. Using basic mesh shapes for the open end of the tool proved to be a challenge. I ended up combining a flattened box mesh and two flattened prism meshes to create a somewhat similar look. The handle of the wrench was crafted from a cylinder mesh and the closed end of the cylinder was created out of a torus. For the toolbox, I used a box shape and added a red texture to make it look like red painted metal. My entire scene was crafted out of a total of six basic meshes. Planes for the floor and back wall, cylinder meshes, tapered cylinder meshes, box meshes, torus meshes, and prism meshes.

Controls were added to allow users to navigate the scene. The camera was able to respond to both keyboard and mouse. Any PC gamer is familiar with the W, A, S, and D to move forward, backward, left, and right. The Q and E buttons allow you to move the camera up and down. The mouse was set up to control where the camera is viewing, similar to moving a player’s head/view in a video game. I used a cross-platform library called GLFW to handle the keyboard and mouse inputs and update the camera, so it moves smoothly around the 3D scene. This allows anyone to look at the workbench and tools from all angles.

I also wrote some custom functions to keep my code neat and organized. Each complex object had its own function, like RenderHammer(), RenderWrench(), RenderToolbox(), and RenderWorkBench(). Each of these functions puts the parts of the objects in the correct positions, applies texture, and takes care of things such as scale and rotation of shapes. Following the format made my programming easier to read as well as fix if something didn’t work correctly.

Other functions were used like SetTransformations() and SetShaderMaterial() so I didn’t have to repeat the same code over and over again. SetTransformations() let me set the size, angle, and position of any shape quickly. This also helped save time when adjustments were needed. SetShaderMaterial() allowed me to change the look and shininess of different objects depending on what material I set them to.

Overall, I am proud of the 3D scene that I have built. Time was spent thinking about which shapes were going to be used to build the different objects I have. The camera controls make it easy for anyone to have a look around. And writing reusable functions helped stay organized and made my overall project cleaner and easier to read.