When first choosing objects for the photo I was going to recreate in OpenGL, I had to ask myself what would be realistically feasible for the parameters of the project. After misunderstanding what was expected once, making sure to create something that would be accepted was my highest priority. At first it was a struggle to choose items that weren’t simply different variations of boxes but eventually I stumbled across an old martini glass that would be perfect for the multi-shaped object requirement. This led to another obvious choice of a bottle to go along with it, another multi-shaped object that could add a bit more complexity and interest to the project. Finally, with two complex objects already in the scene, a simple wooden box and spherical candle were added to round everything out.

While I felt like the objects chosen were a good fit, their design was mostly flat. The glass and bottle were simply clear glass, the candle was white wax with no real detail, and the box was just a box. So instead of keeping my render completely true to life, I took some artistic license and added some more details. The first attempt saw a stained-glass texture on the martini glass, but after more consideration it felt like it looked like an error in the render when viewed from too far away. After going back to the drawing board, the primitives were changed into a Halloween theme and looked much better. This allowed for greater expression and a much more interesting scene for the viewer while still meeting all the project’s requirements. The higher contrast in color also showcases the lighting in a much more striking, easy to see way. Overall, the design choices were successful; they met every requirement set forth in the project and came together in a cohesive, visually appealing manner.

The scene is set up to be traversable using both the mouse and keyboard to move the camera. Different vector directions were assigned to the W, A, S, D, Q, and E keys: W for up, S for down, etc. There are also two different camera options accessible via the O and P keys, with O enabling an orthographic view and P returning to a perspective view respectively. While in the orthographic view mode, the scene is locked in a 2D perspective, allowing the user to scan up, down, left, and right but keeping the z-axis locked. Perspective mode is the default mode the program launches in, allowing for full camera movement. The mouse acts similarly to that in a first-person video game with the camera following wherever the cursor moves in any direction while the perspective view mode is enabled.

There are multiple functions that exist in the code of the program that make it much more modular and easier to read. Things such as vertex matrices for primitives were coded so that instead of creating a brand-new matrix for each shape, adding hundreds of lines to the code, a basic shape could simply be called and transformed to fit whatever was needed. Similarly, when shader materials were created to define how light would interact with an object based on what it is made of, they were made so that they could be referred to as many times as required by a tag instead of redefining them for every object.