The objects chosen for this projects, and their placement, were a deliberate effort on my part to challenge myself in areas of software development where I have fallen short in the past, in addition to the specifications for the project. The cup and books were chosen for the complexity requirement, and to create a more aesthetically interesting scene. The thermometer was a choice to also fulfill the complexity requirement, and also challenge myself to experiment with lighting and blending tools. The 20-sided die was meant to challenge myself to become more accustomed to longer, more repetitive and complex tasks within software development.

For the implementation, I did simplify the image in the following ways: The small baubles within the thermometer were not made visible, the 20-sided die does not have a number printed on each side, a different texture was used for the cup instead of ceramic, and the book covers were made blank. I was unable to find a royalty-free ceramic texture that would mesh seamlessly on the surface of the cup, and so used a combination of the provided stainless steel and brushed gold textures to create the cup. I chose to use blank textures for the books and dice model for both simplicity, as rendering each face with a texture would require significant alteration to the Icosahedron method that I had not prepared for, and I was unsure of the potential copyright constraints of using an image of the actual book cover and spine.

While I was able to make the body of the thermometer transparent, and created models for the small baubles, due to the way OpenGL models transparency these objects were not visible inside of the thermometer normally. I did discover that it is possible to rectify this issue using depth-scanning to adjust the order in which objects are rendered, I was unable to work this solution into my code-base.

I built the thermometer using a cylinder mesh, cone mesh, and toroid mesh for the base. The cup was constructed using a sphere mesh, three toroid meshes for the base, lip and handle, and a cylinder to raise the lip above the top of the sphere. The books, I created from four stretched cubes each, and the Icosahedron, or 20-sided dice, was a custom 3D mesh that I created in the ShapeMeshes class file.

For the camera, I have utilized the following controls:

* Directional, 2D navigational control:
  + W: Forward.
  + A: Strafe Left.
  + S: Backward.
  + D: Strafe Right.
* Vertical Navigation:
  + Q: Down.
  + E: Up.
* Viewpoint Shifting:
  + O: Orthographic view toggle.
  + P: Perspective view toggle.
* Free-Look Camera:
  + Controlled by mouse or touchpad controller, based on configuration.
* Look and Navigation Speed:
  + Mouse Scroll-up: Increase speed of all input effects.
  + Mouse Scroll-down: Decrease speed of all input effects.

These controls will work as along as a keyboard and mouse are connected, or a keyboard and trackpad are connected. In the case of the latter, all input is inverted.

Over the course of the project, I encountered the issue of potentially too many instances of a particular 3D object mesh being called and increasing the load on the system. While objects such as my cup item are composed of several meshes, with three Toroids, a Sphere and a Cylinder being present, this is an acceptable load for the system. The construction of the 20-sided die presented the potential usage of 20 draw calls, and up to 80 potential sustained lighting models if done correctly. To resolve this, I had to create a custom 3D mesh in the ShapeMeshes class that could be drawn in, textured and shaded as a cohesive structure. I plotted and recorded the planar relations between the twelve vertices of an Icosahedron, and calculated the planar normal vectors for each set of vertices composing a face. This allows me to place the dice anywhere in the frame as a cohesive object, without overtaxing the program.