

PROJECT 4: In addition to original project 2 & 3

New Features:

Features include the new walls, door, and window into the cabin. The roof was omitted simply so the inside is easier to see. The window is rendered with translucency in the Phong model. Textures are applied to the cabin walls, water, and the door itself is just a door texture on a rectangle.

Interactive Controls:

'L' will enable the indoor lighting. While 'D' will disable it.

'F' enables the bonfire lighting & cylinder representing the flame. 'K' will remove it.

'A' will animate flowing water, which is really just a scrolling texture. 'Z' will revert it to a static scene.

New Things that Caused You Difficulty

The primary issue I had was getting appropriate images for textures that looked good, but were not too large. Initially the water texture was a way oversized png, and the screen would take a while to load when the program was run. It was compressed into a much smaller jpg file. There is still some delay, but minor.

Anything Else New that You Want Me to Notice

The animation portion was mostly just to see if I could do it, and is very basic and fairly hacked in, as it was not something I had ever intended to need support for. Several GLFWcontroller methods had to be overwritten to add support without being able to modify the originals. Render() for the basic animation timings, and handleAsciiChar() for enabling and disabling it. I used the chrono library to work with framerate and sleeping the cpu to keep it from looping at 100% when we are waiting to render the next frame.

The glass has some slightly special handling in the fragment shader, that allows it to be illuminated from either side, basically we just disregard the fact that the normal is on the opposite side, and act as if it isn't. This creates a more realistic look. For example the light from the fireplace can be seen on the window from the outside.

PROJECT 3: In addition to original project 2 below.

New Features:

Primarily the new features include the entire outdoor section. This includes the small pond and stream, tree, and campfire. For the campfire, as well as the old fireplace, there are now light sources defined in Model Coordinates within each one, to give a realistic feel.

New Interactive Controls:

Shift-I, or 'L' will enable the indoor lighting. While 'D' will disable it. This lighting has specific fragment shader support to clamp its effects to within the interior of the room, as if there were actually 4 walls, and they were blocking the light from escaping.

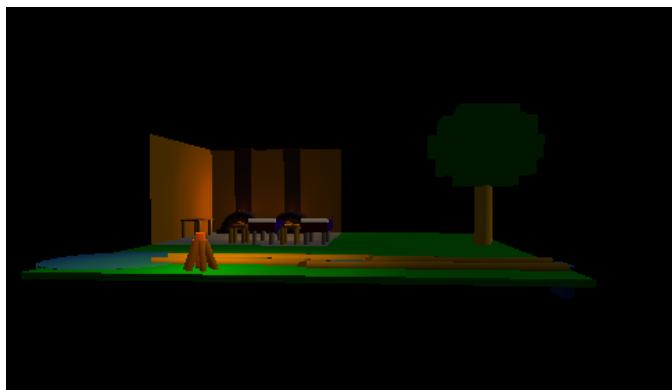
New Things that Caused You Difficulty

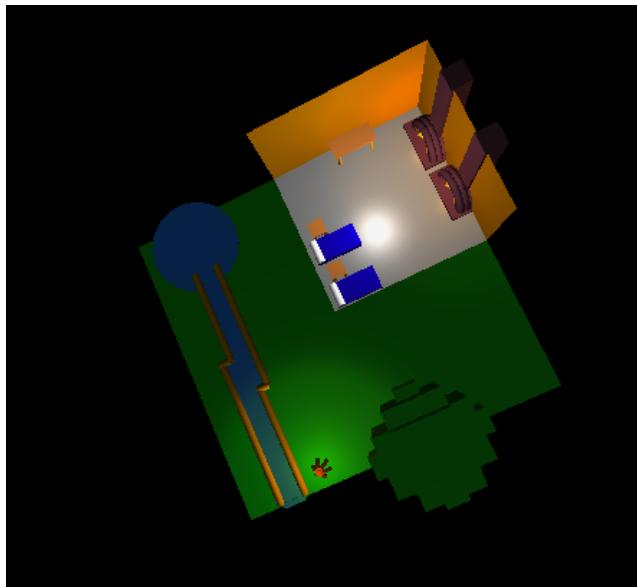
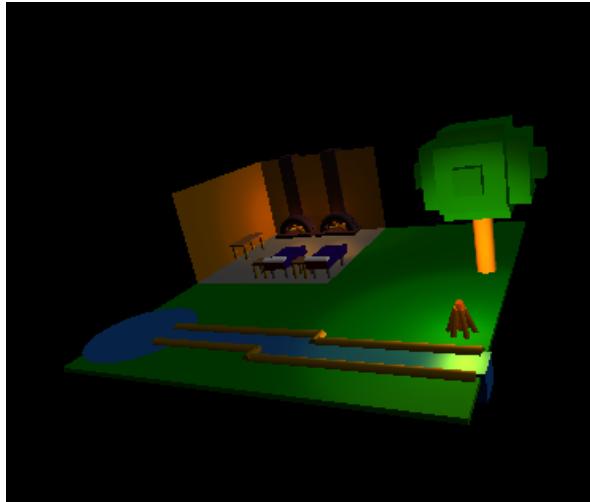
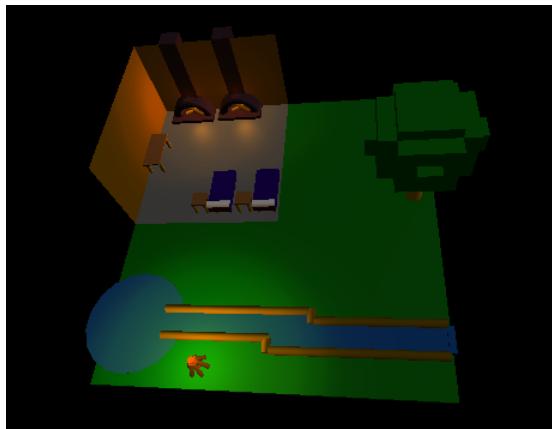
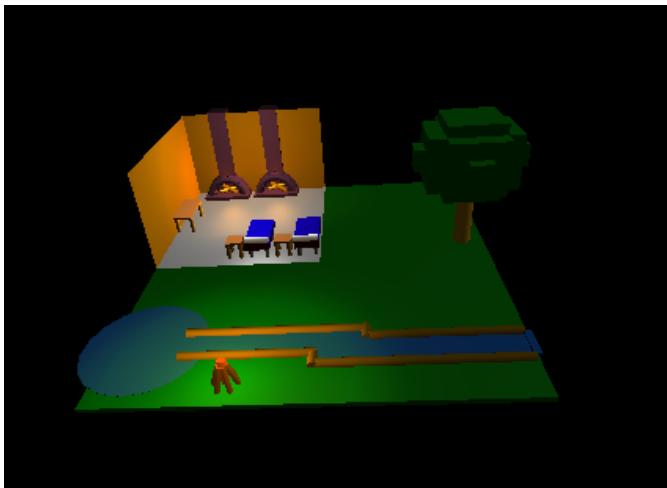
The primary issue I had was that my ModelView classes were implementing the handleChar methods, which was overriding/hiding the SceneElement's implementation, and projection changes weren't working.

Anything Else New that You Want Me to Notice

Nothing particular. The fragment shader will handle fire source attenuation differently, while physically unrealistic it creates a better looking scene. Reflecting the fire off of the water looks rather pleasant. Table legs are also reflective brass, but only a few angles will show it.

The vector-based light is coming from over the tree and into the cabin.





PROJECT 2:

Project Origin:

This simply started as me needing something to model, and so I began with my bedroom starting with a bed and bedside table. It then extended to trying to think of other features to add, so for some reason what came to mind was a fireplace/brick oven. So it is no longer a bedroom model, but just a room with various household features.

Model Generation:

For starters I created a model container for the bed, which would contain 4 vertical cylinders, 1 rectangular prism and 1 horizontal cylinder. The table just included 4 cylinders and a prism. To properly make the pillow, and aspects of the fireplace I needed to improve the cylinder class, allowing it to have the 2 end caps, and be arbitrarily placed and rotated, as well as scaled in radius and height. The fireplace contains a bottom and back panel and the chimney stack, which are prisms. The fireplace itself is made of 3 layers of cylinders, plotted over (50,y,z) where y and z are calculated to create a semicircle, the angle of these cylinders is a function of where they are on the semicircle. The logs are again cylinders, with an (x,y,z) rotation.

Components are broken into containers like the fireplace, and individual models like the cylinder. Inheriting from one of these classes extends a lot of the functionality over them. MyContainer extends features such as rendering, memory management, bounding box management, and dynamic rotation(which was unused). Container objects are passed to the controller as a ModelView instance, but upon calling render will pass the call onto all of the models in that container by iterating over the vector of components.

Meeting Project Specifications:

My 3 main features are found in Bed.c++ Table.c++ and Fireplace.++, which are all combinations of various geometric figures. The two walls are included in Fireplace.c++, as in order to avoid the wall clipping into the fireplace, the wall had to be rendered without including the semicircle that the fireplace takes, like the crude ascii art below.



Found in the scene are 3 tables, 2 beds, and 2 fireplaces. As well there are the walls and a floor panel.

Difficulties:

Primarily, the hardest part was using the basic geometry to generate more advanced models, having to deal with orientation, as well as location with respect to other components of the model.

Unique:

A minimal amount of this scene is hard coded coordinate data, the floor and walls are all I believe. Instead it is generated at runtime from a combination of class constants and provided parameters. Certain components have complete control of rotation, size, and position, and some can just be sized and moved. I attempted to abstract away a lot of common functionality, and as such, models like the fireplace only need a constructor to generate data as their sole function, as inherited methods will handle everything else. The simple components like the cylinder simply had the ability to change color properties pulled out into inheritable methods.

A few different lighting locations and perspectives:



