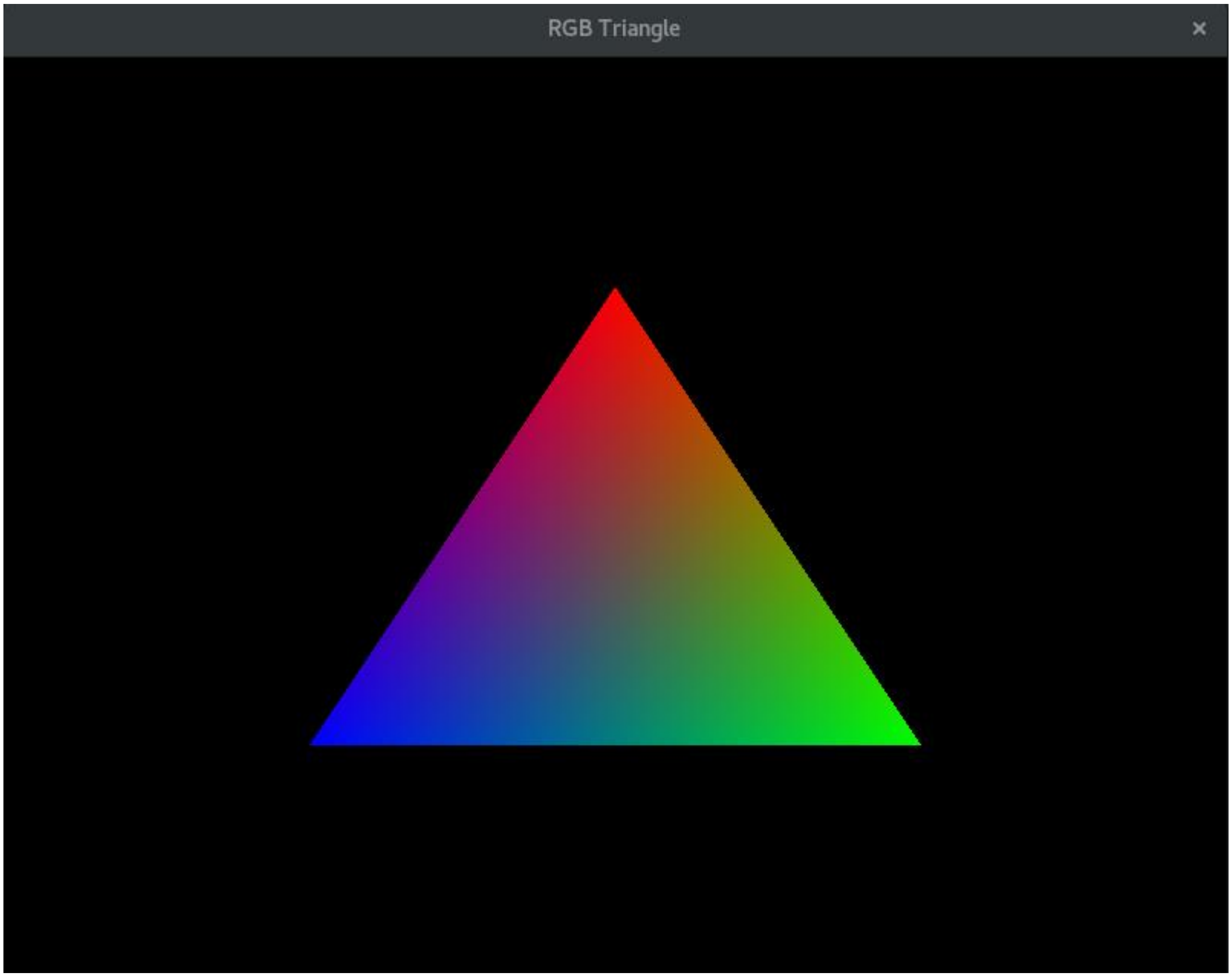
---- Week 3 ----

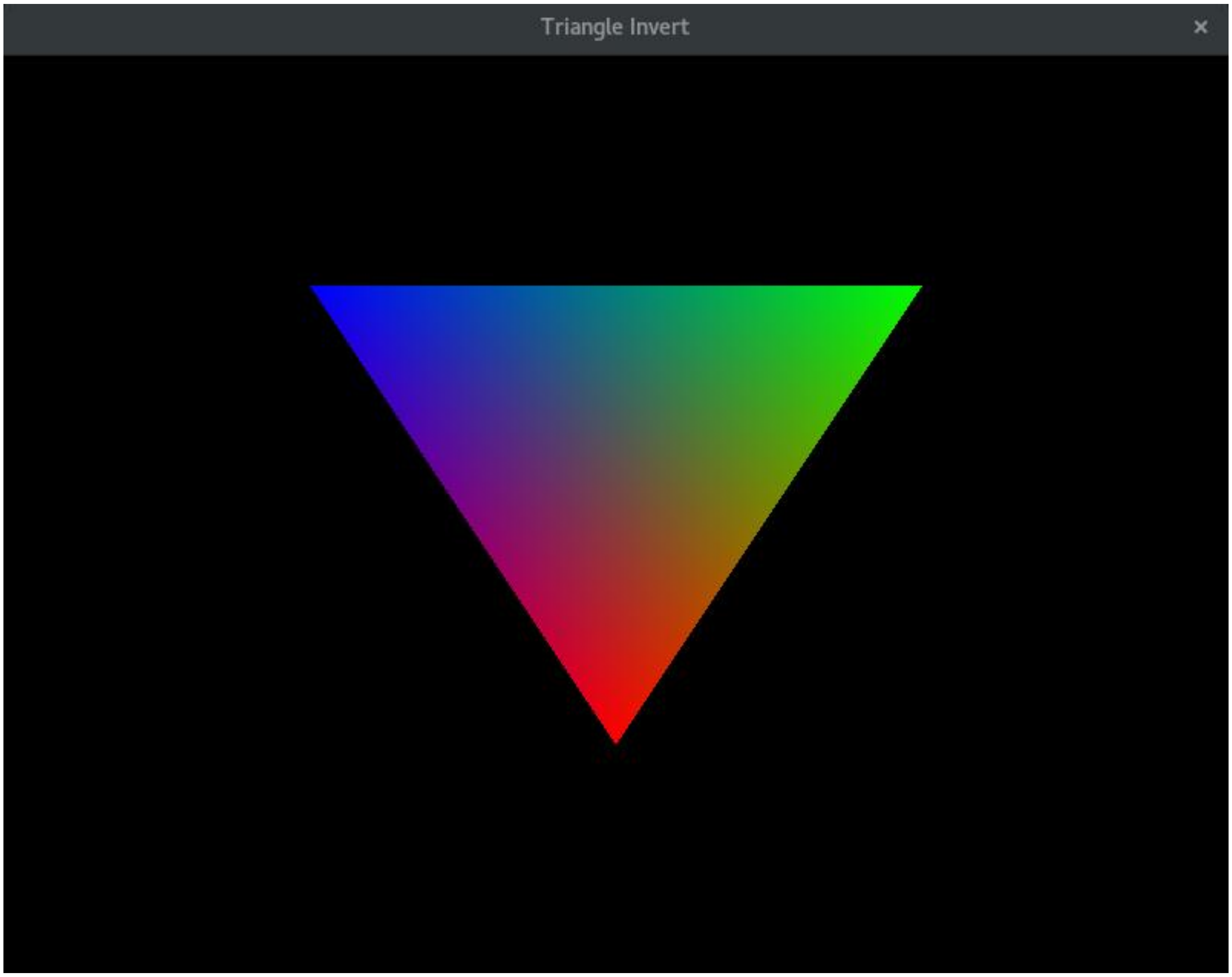
Changes to `triangle_color.cpp`:



The vertex shader now takes in color as an input and outputs color to the fragment shader (lines 33,35,40). The fragment shader then takes in that color and assigns it as another output variable to be referenced later (lines 64,65,69).

The array of floating point vertices was also changed to include color values for the triangle. The color was then retrieved from the shader attribute using `glGetAttribLocation()`. Since the color was placed in an array alongside the triangle vertices, then the stride had to be changed as well, to show that the attributes occurred after every 5 elements. For color, the base pointer also had to be increased by the size of 2 floats so that the vertex data would not be read as color data.

Changes to `triangle_invert.cpp`:



This part really only took one change on line 39, where

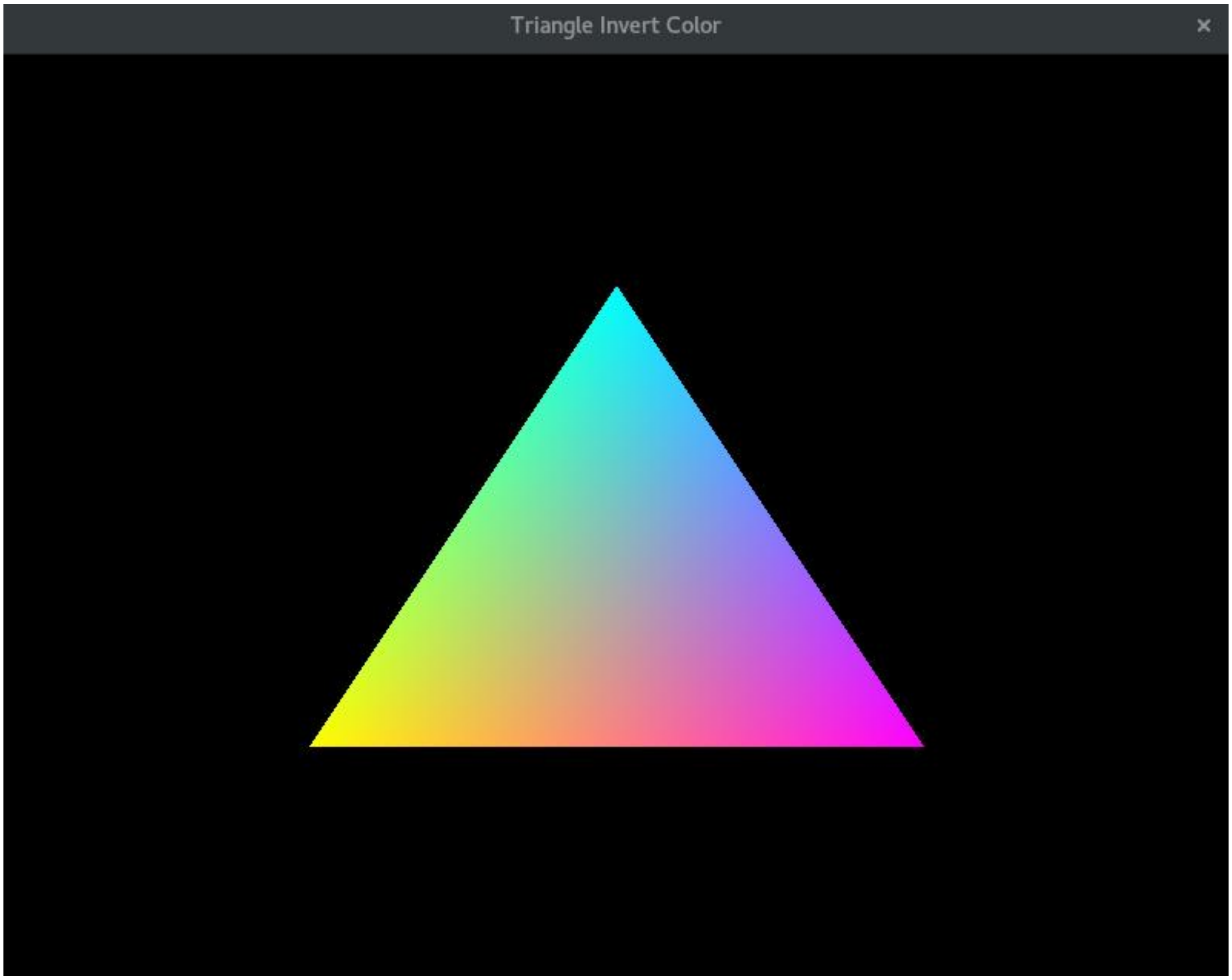
```
gl_Position = vec4(position, 0.0, 1.0);
```

was changed to

```
gl_Position = vec4(position.x, -position.y, 0.0, 1.0);
```

so that only the y axis of the triangle would be flipped.

Changes to `triangle_invert_color.cpp`:



This only took one change as well, where on line 69

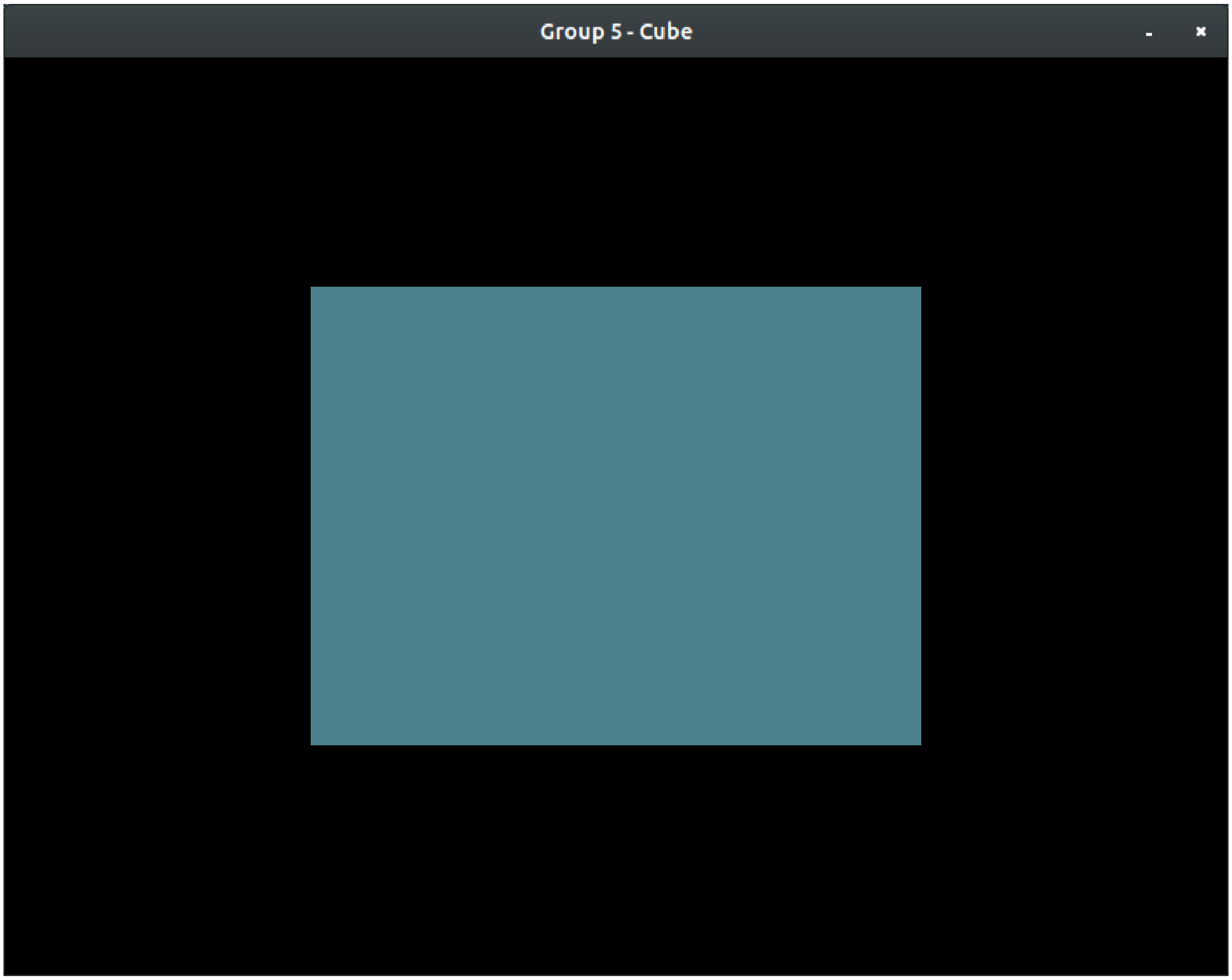
```
outColor = vec4(color_out, 1.0);
```

was changed to

```
outColor = vec4(1 - color_out, 1.0);
```

to invert the colors

---- Week 4 ----



Sorry for the ambiguous view. I could not figure out how to change the viewing angle.

Changes from the 2-D file (originally `triangle.cpp`):

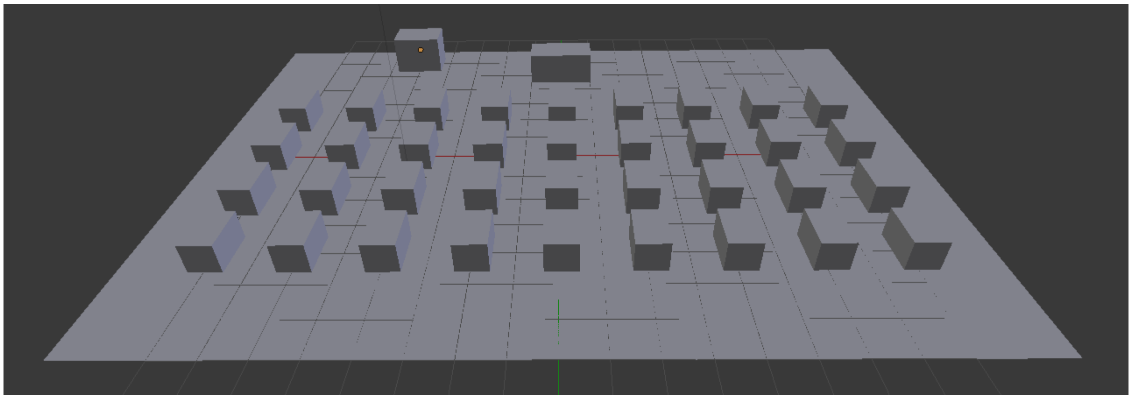
- Line 32: `in vec2 position;` ---> `in vec3 position;` to account for 3-D coordinates (x, y, and z)
- Line 35: `gl_Position = vec4(position, 0.0, 1.0);` ---> `gl_Position = vec4(position, 1.0);` to account for the change on line 32
- Lines 133-179: added all the new vertices of the cube to the vertices array, 36 points in total, each point having 3 values, one for x, one for y, and one for z
- Line 283: `glVertexAttribPointer(postAttrib, 2, GL_FLOAT, GL_FALSE, 0, 0);` ---> `glVertexAttribPointer(postAttrib, 3, GL_FLOAT, GL_FALSE, 0, 0);` to account for `position` now being a `vec3`
- Line 310: `glDrawArrays(GL_TRIANGLES, 0, 3);` ---> `glDrawArrays(GL_TRIANGLES, 0, 36);` to account for all the new vertices added

---- Week 11 ----

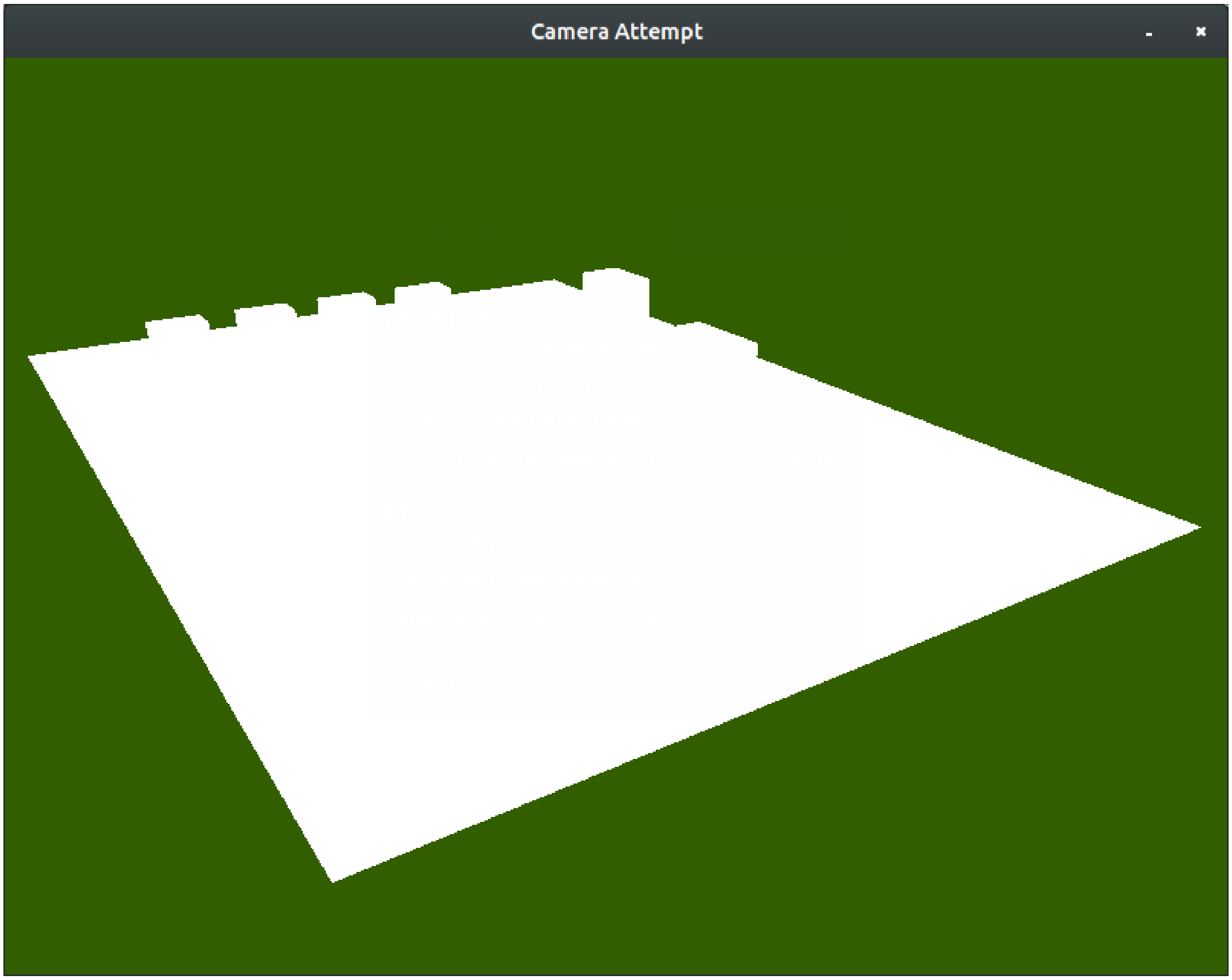
- Created floor plan object in Blender

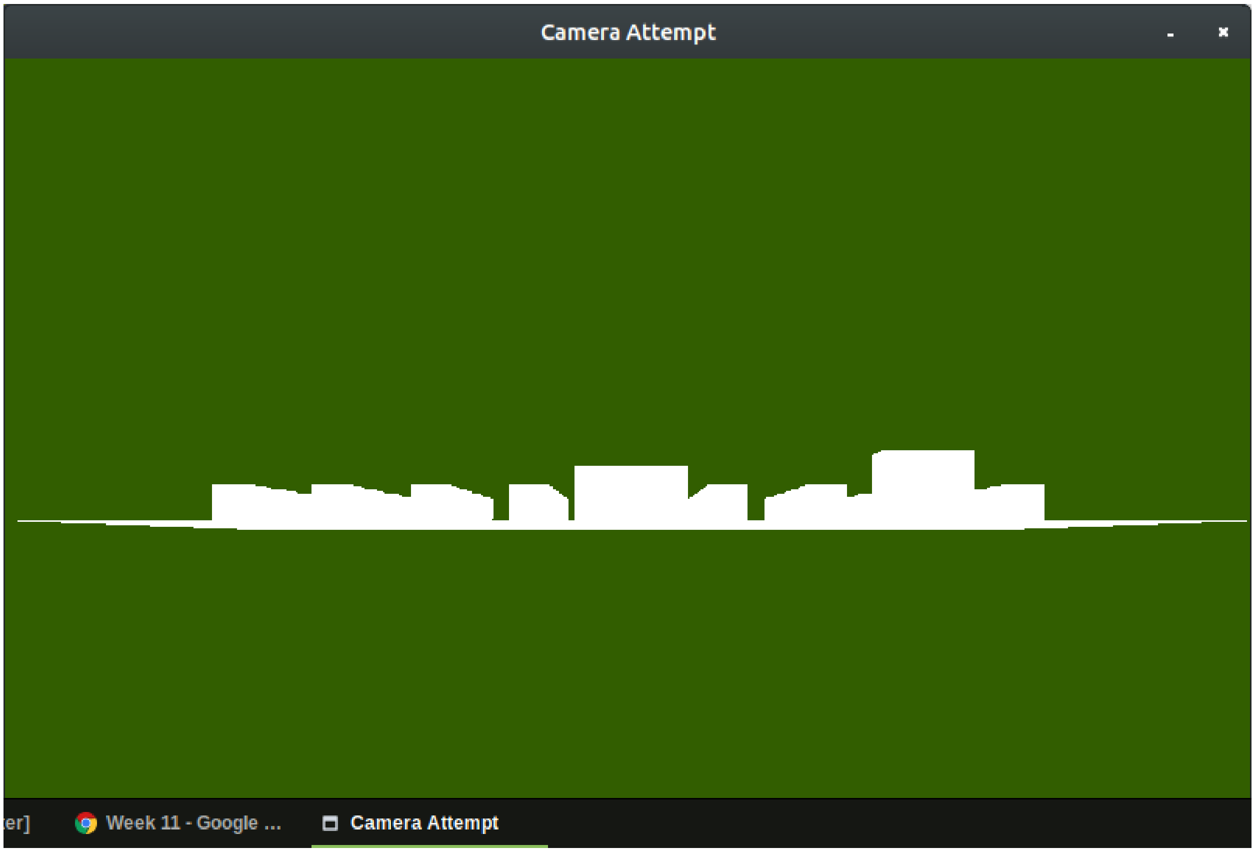
- Imported object into working OpenGL project

Screenshot of object in Blender:



Screenshots of object in OpenGL:





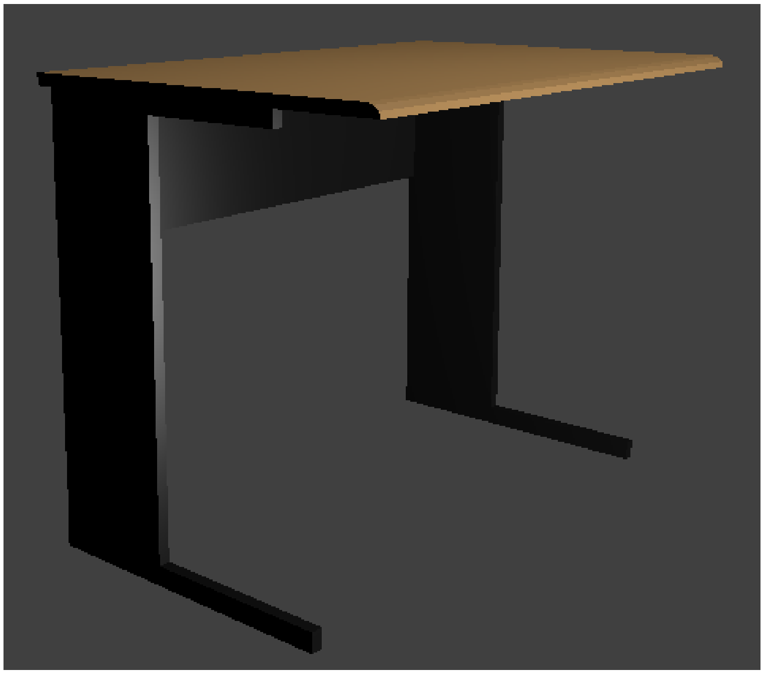
---- Week 12 – Week 15 ----
Working on final project, learning blender, and refining OpenGL code.

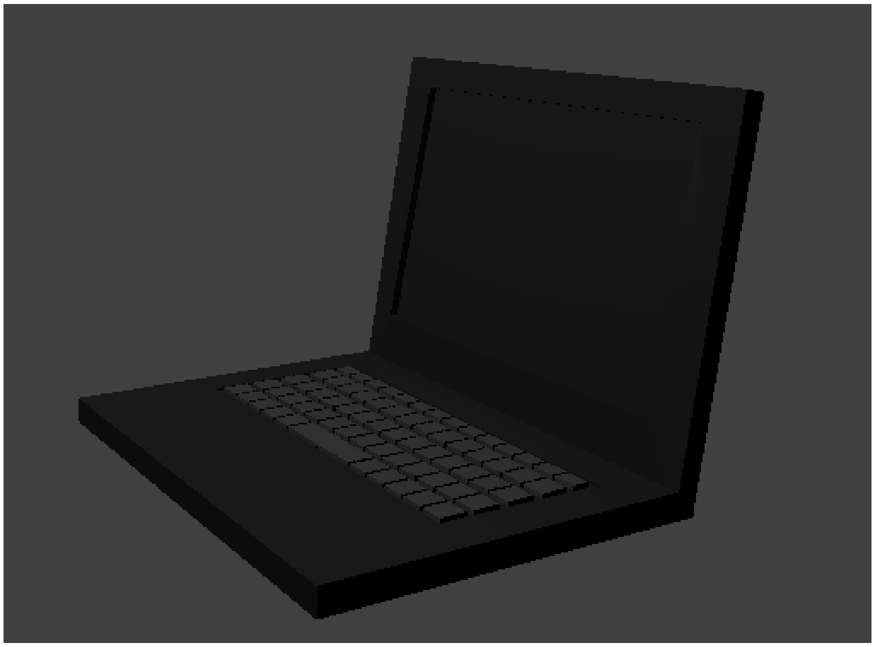
----- Final Project -----
Our goal was to recreate Cramer 239. We got most of the way there but neglected to include many objects and texture simply because there was a time constraint and Cramer 239 has many objects to recreate. The scene we created also has no lighting applied, as we simply could not figure it out.

Most of our models and paintings are included in the museum_models folder in the git repo. There are 4 models in the folder that are rendered in blender. Paintings.png shows our 2 paintings in the image (the board and the wall textures). Cramer239.png contains the floor plan for the recreation of the classroom model.

The git repo also contains the walkthrough video for the project, titled 'video_for_gl_final.mp4'

The models/paintings are also included below:





Paintings shown are: The board writing, the texture on the wall, and the texture on the ceiling.