This project runs using visual studio cmake. When running notable keys include

1 = cube

2 = bumpy cube

3 = bunny

8 = wireframe

9 = flat shading

E = clears screen

Not Working

O = orthographic view

P = perspective view

Keys that were going to be implemented, but were not

0=phong shading

W=pan up

A=pan left

S=pan down

D=pan right

H=rotate left

J=rotate right

K=scale up

L=scale down

1.1 adding and deleting objects

There are 3 different types of objects you can add. The cube, the bumpy cube, and the bunny. The bunny and bumpy cube have the vertexes stored in a .off file. The information is read in from the file and used to create the objects on the screen at origin. To keep it colorful I also implemented a color matrix that has random colored triangles to give it a fun geometric look. The individual delete function isn't in place because I was confused how to keep track of the different objects when they have different vertices sizes. However, to make it easier to view the objects I did implement a clear screen function done by clicked E.

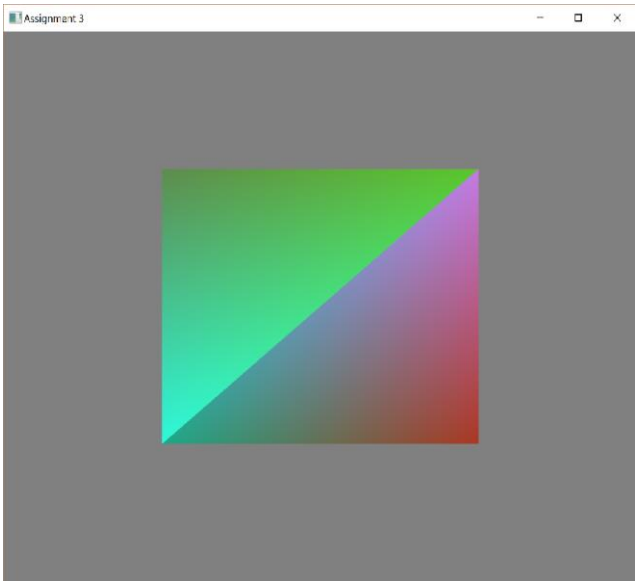
1.2 editing objects and rendering

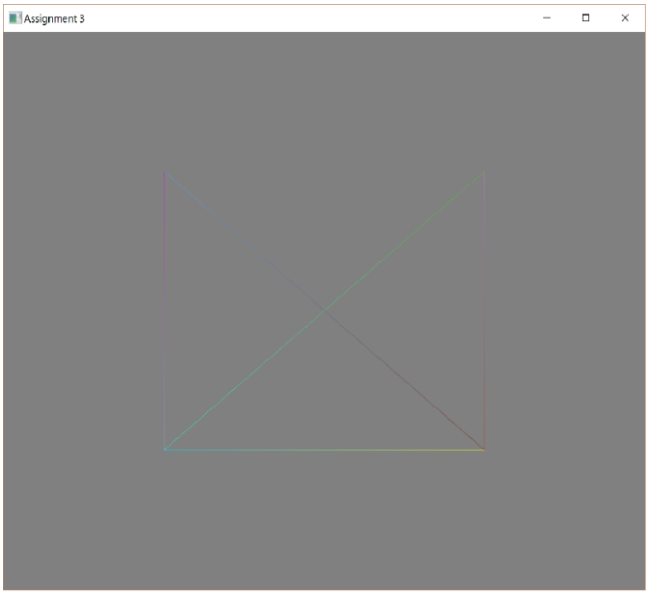
For editing the objects not much really had the chance to be worked with. I brought over my code for rotate, scale, and translate from assignment 2, but was struggling to apply the equations to the whole object. It also was frustrating having my objects have different sizes of vertices. I debated the idea of re editing the square to have more triangles but didn't get the chance to try it out. As for rending it changes everything in the scene not just one object. It you press 8 you will get wire framing and if you press 9 you will get flat shading. I did not get a chance to do phong shading however I do know that to implement it you have to edit the projection matrix. By calculating the ray direction from a light source, you can figure out the shading for the triangles.

1.3 camera views

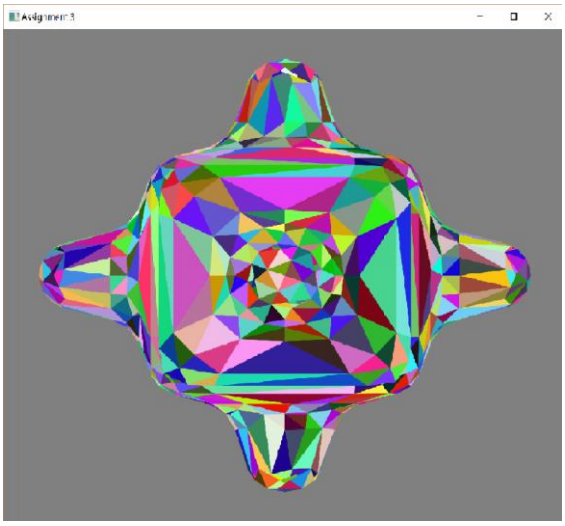
For Camera I didn't get the chance to work on it's ability to pan the scene. I know this is edited in the view matrix and that by calculating the world view you can then transform the matrix to shit the camera. I also attempted to do orthographic and perspective views however neither of these changes the scene. I attempted to calculate the different attributes needed for the projection matrix and then apply it to the scene. I must have been missing a few steps because nothing happens, leaving us with the original ortho projection you get just by creating an object at the center of clip space. I know that the final scene should be made up of Model * View * Projection.

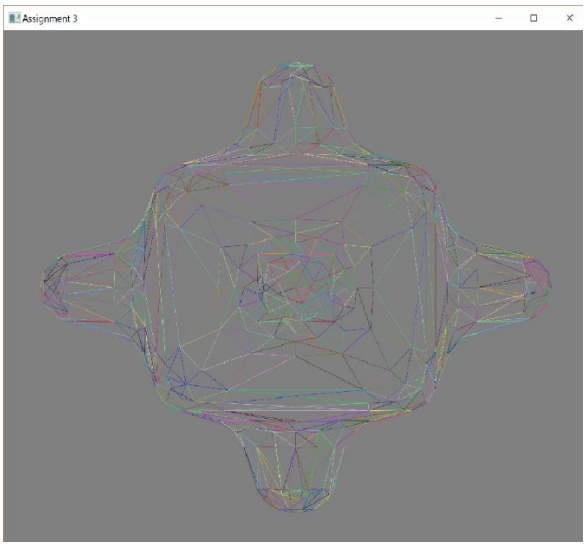
The Cube





The Bumpy Cube





The Bunny