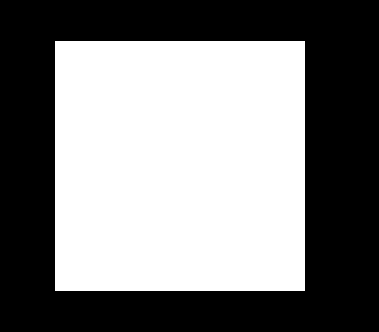
**CS4610 Assignment 2a Report**

By: Nathaniel Callahan

**CS4610 Assignment 2a Report**

1. Read an obj mesh file containing a geometric model represented as triangle meshes and display it centered in the display window.

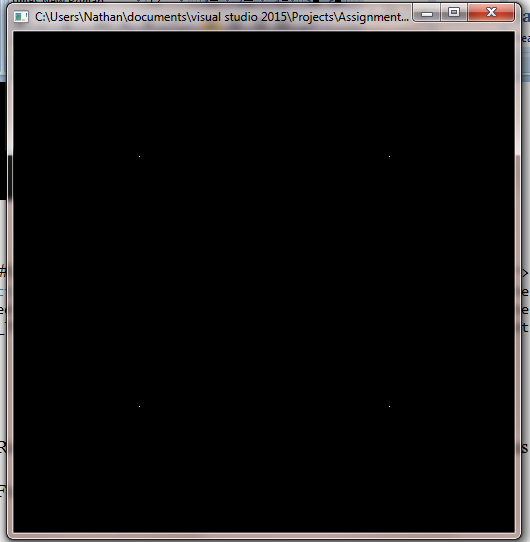
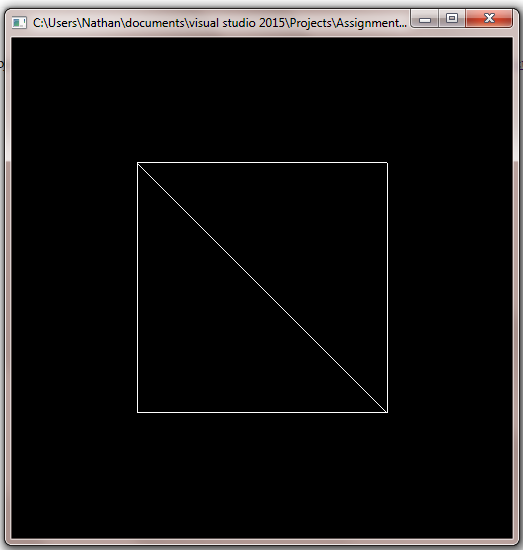
I followed the Q&A slides provided by the TA to get the cube.obj file to display when running the project.

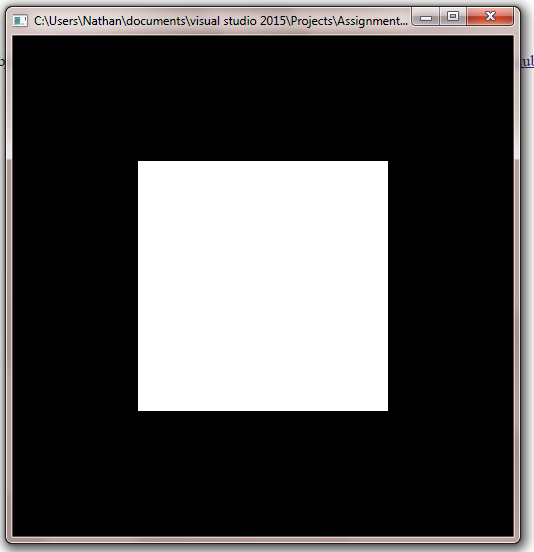


I added #include<vector> to the project so I could add std::vector<GLfloat\*> vertices and std::vector<GLint\*> faces to store the values from the .obj file. I then opened the file and added all the all the vertices and face values in the file. For the surface display I used GL\_TRIANGLES in glBegin(). This produced a solid cube that is centered to the middle.

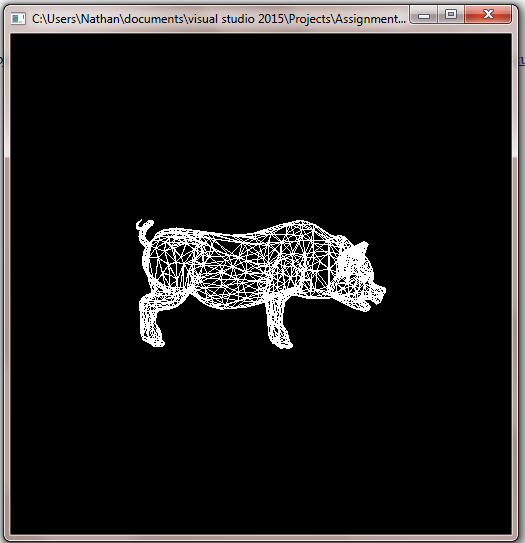
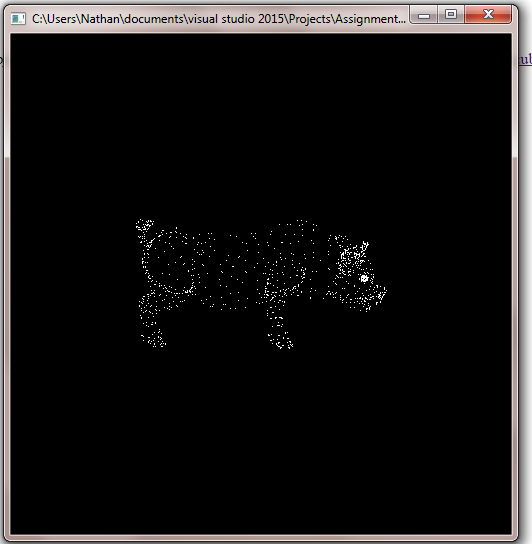
1. Render an object using points, wireframe and surface representations.

For the Cube:

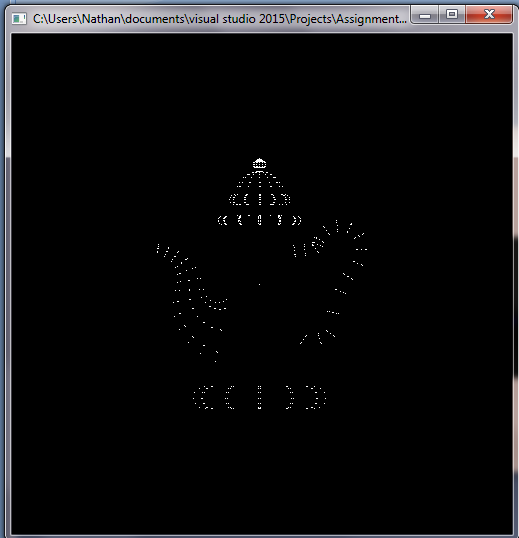
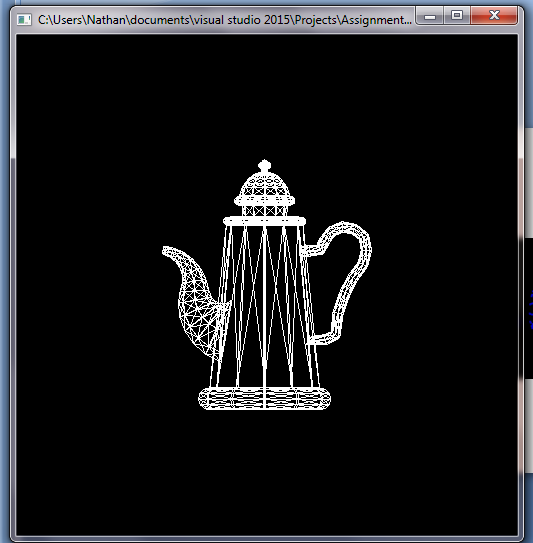
 

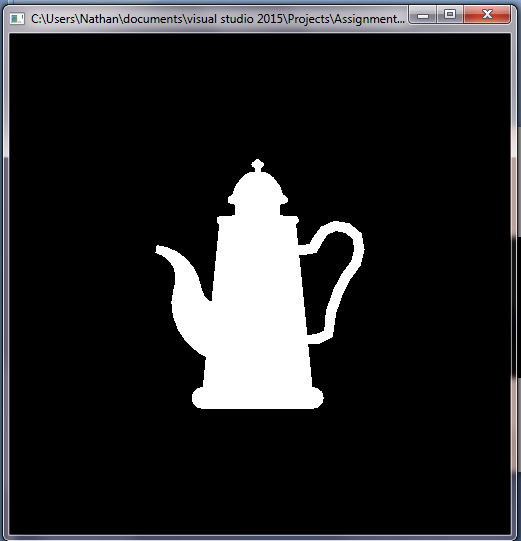


For the Pig:



For the Teapot:



To choose between the three objects to load in and display, the user must click ‘q’, ‘w’, or ‘e’. Clicking on ‘q’ will load in and display the cube. Clicking on ‘w’ will load in and display the pig. Clicking on the ‘e’ will load in and display the teapot. Code wise, I called a method called key(). Key uses a switch statement that checks to see if any of those buttons are pressed, and if they are load in the object from its file and display it on the screen.

To choose between the three display types of the object (points, wireframe, and surface), the user must click ‘a’, ‘s’, or ‘d’. Clicking on ‘a’ will display the object using only points. Clicking on ‘s’ will display the object using a wireframe. Clicking on the ‘d’ will display the object using the surface of the object. Code wise, I also called the key() method to switch between display types. Key uses a switch statement that checks to see if any of those buttons are pressed, and if they are display the object in the specified form.

There were two chalenging parts in the assignment: moving the object to the frame so you can see the object and scaling down the objects size so it fits the frame.

To move the object into the frame you must find the max and min x, y, and z values and find the midpoints of each diminsion of the object (ex: centerX = (max x + min x)/2). Once all the centers are found, in the display function do gltranslatef(-centerX, -centerY, -centerZ), this will center the object in the center of the screen to view.

To resize the object so that it fits inside the frame I needed to find a scale value to shrink the object by. To do this you take the dimensions max value and subtract its center value (ex: maxY – centerY). Do this for x, y, and z, the largest of these values will be what you use to scale. To shrink the object, in the display function do a glScalef(1/scale, 1/scale, 1/scale) (learned this from class Q & A).

Also, to get a better angle of the objects I used glRoatef() so when displayed the objects look a little nicer.

One remaining issue is that I must call glRoatef() twice to rotate the object the way I wanted instead of once. I rotate is -90 degrees in the x diminision and 90 degrees in the z diminsion.

glRotatef(-90.0, 1.0, 0.0, 0.0);

glRotatef(90.0, 0.0, 0.0, 1.0);

A solution may be to not just use one for the deminsions values. I may be able to get away with using a value other than one like glRotatef(90.0, -1.0, 0.0, 1.0); and use only one glRotatef() instead of two