**Computer graphics Assignment 3**

2014004411 김시완

**Implemented function**

Function of manipulate camera include zoom in/out.

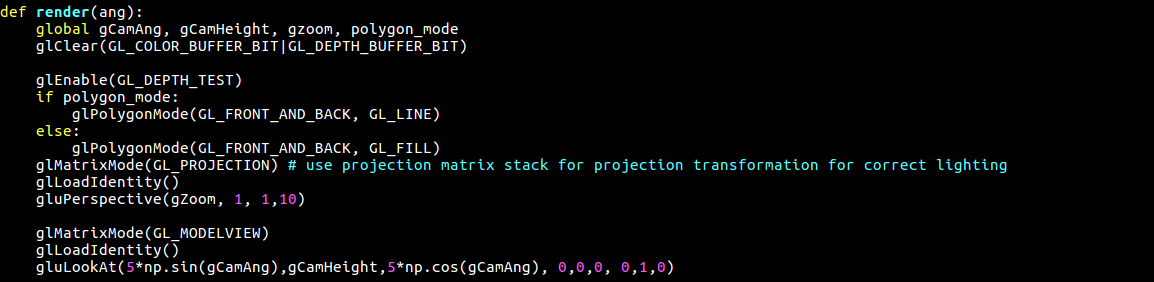
Load an obj file and render it using by glDrawArrays function(include various number of polygons).

Switch the mode between toggle wireframe / solid mode.

Print out the information of obj file face.

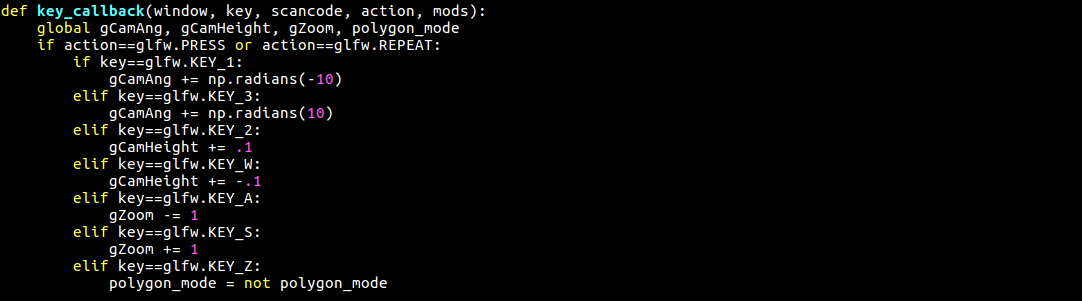
Lighting.

**Implementation Description**

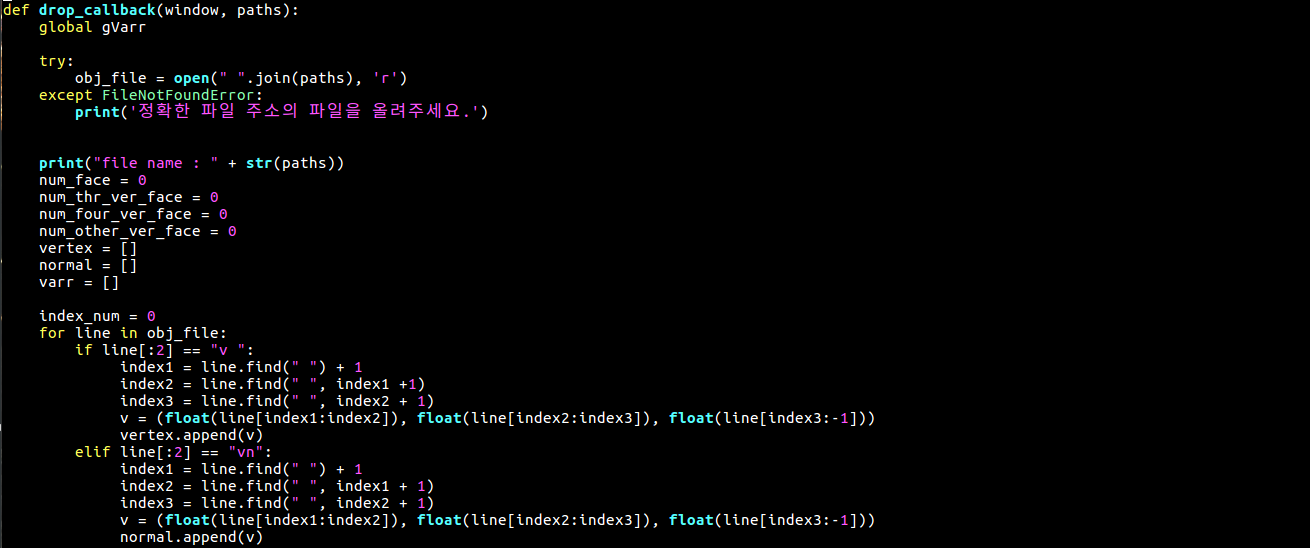


To implement the zoom in / out function, added the ‘gZoom’ variable which is first argument on ‘gluPerspective’ function. It is the field of view angle, in degrees, in the y-direction. So if it is increased, it would seem like an object is getting closer to a camera. Conversely, reducing the size of the ‘gZoom’ variable will reduce the angle of view in the y-axis direction, which will make the object appear to move away from the camera. To manipulate the value of the ‘gZoom’ variable by press A and S key, added the code on ‘key\_callback’ function like below capture.

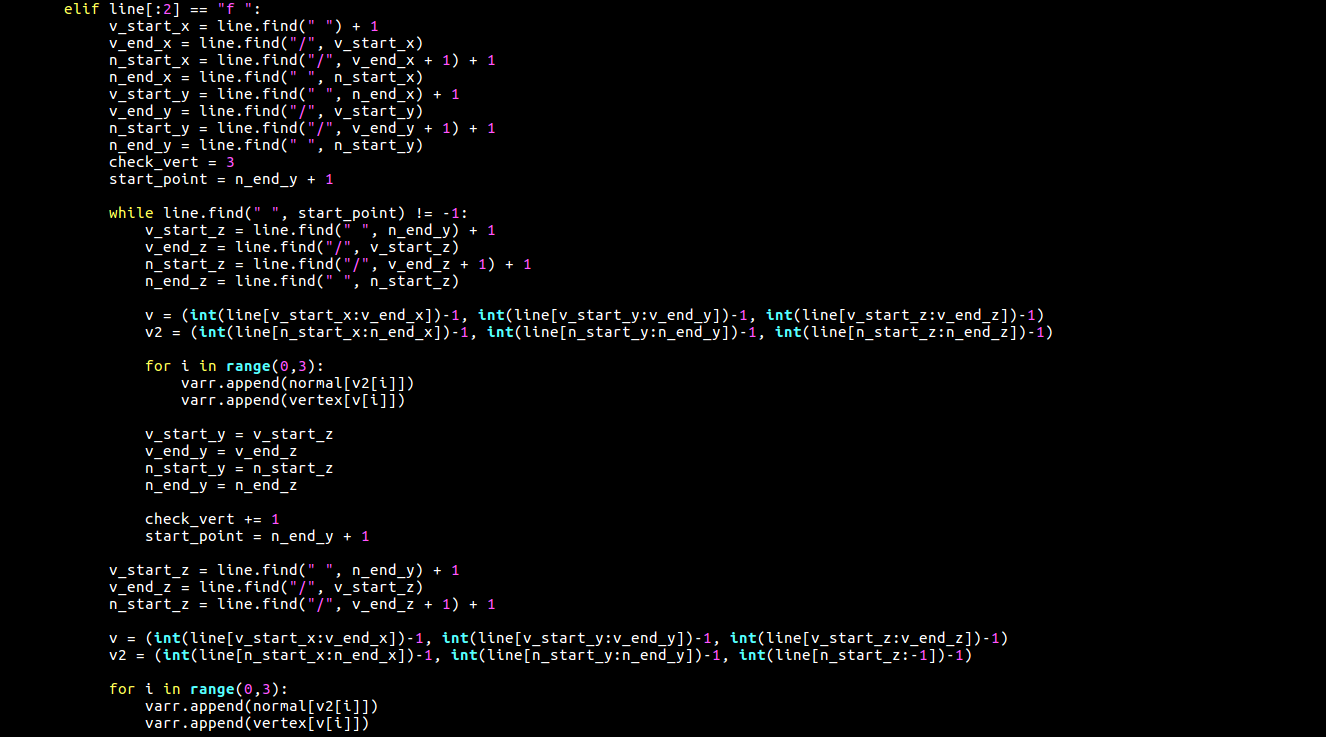
If the variable ‘polygon\_mode’ is true, the object will be rendering in a line, and if false it will be rendering in fill state by using glPolygonMode’ function. This mode is controlled by the key ‘Z’.



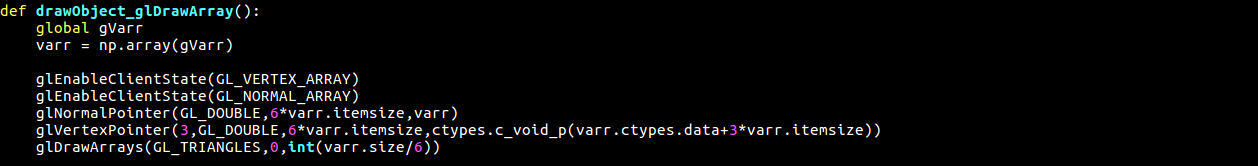
The function ‘key\_callback’ is added to switch mode between wireframe and solid mode and manipulate the camera in the window. Key ‘1’ and key ‘3’ are using to control the camera’s angle and key ‘2’ and key ‘4’ are using to control the camera’s height. Key ‘A’ and key ‘Z’ is change the value of ‘gZoom’ variable and this has the effect of making the distance to the object appear to be getting closer or farther. Lastly, key ‘Z’ is change the mode between wireframe and solid mode on drawing object.



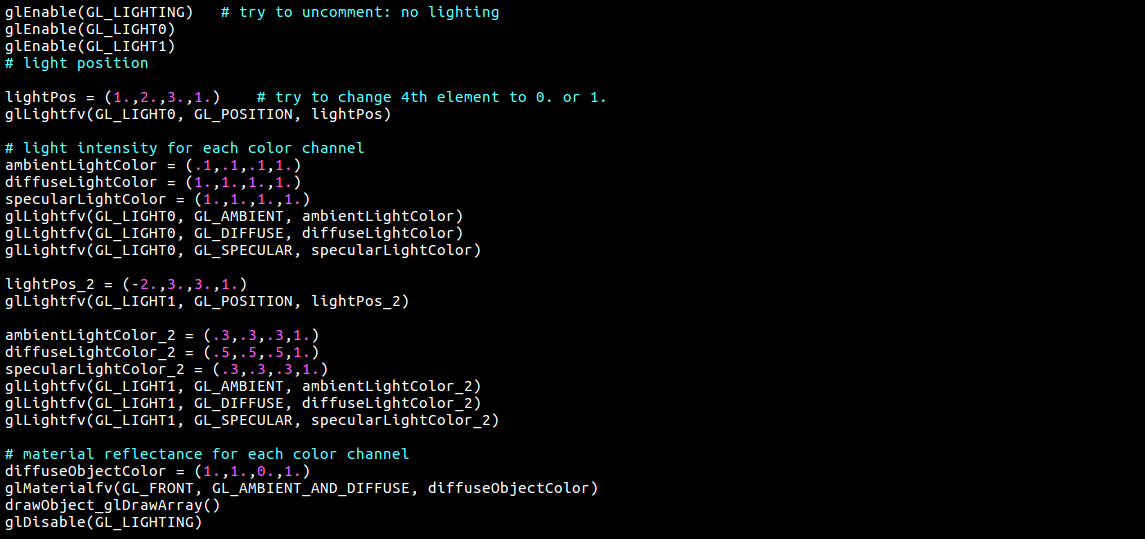
When the file is dropped to the window, the ‘drop\_callback’ function is called. The ‘obj\_file’ variable is the file, which is opened by the file’s path. As reading each of line in the file, separate each line indicate whether vertex or normal. By the type of the line, store it’s information in each array.



And if the line meaning the face information, parsing that line to get the index of vertex and normal array which is contained information used on draw faces. After reading the vertex and normal index information, access that index of each array to obtain the information and store it in a single array in normal, vertex order. If face consists of more than three points, store the information of those points in order likes (1, 2, 3), (1, 3, 4), and (1, 4, 5). And finally, it will be the array consist of the normal and vertex, which is used to consist the faces.



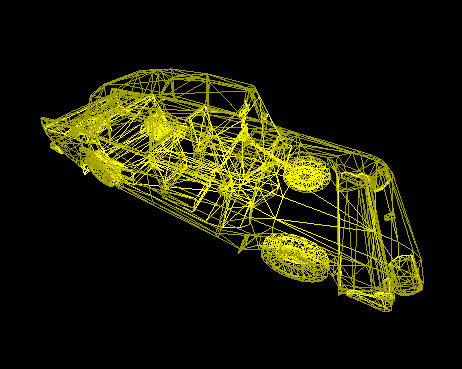
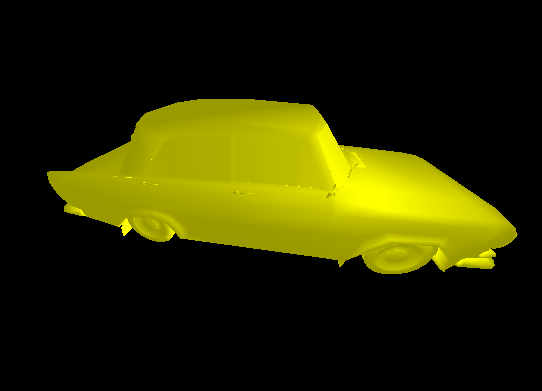
The array previously created is used for specifying the point by reading the vertex and normal information through the functions ‘glVertexPointer’ and ‘glNormalPointer’, respectively, and for rendering through the functions ‘glDrawArrays’.



It is implemented with two light sources in the function, and placed the light sources at position (1.,2.,3.,1.) and (-2.,3.,3.,1.) , respectively. All of the sources are consist of ambient light, diffuse light and specular light and lastly setting the material reflectance by using the ‘glMaterialfv’ function.

**Screenshot images**

**[GL\_LINE MODE] [GL\_FILL MODE]**

**1.**  

**2.** 