**Detail report for COSC4370 Fall 20 HW2 - OpenGL**

**Objective**

Objective of this Project practice some of the basics of OpenGL with 3D scene and dealing with camera and Projection. By drawing several 3D scenes with OpenGL using glutSolidTeapot , glutSolidCube and use of keboard events

**Environment**

In this project we will be using the GLUT/freeglut library for our OpenGL code. If you are on Windows, we can find a freeglut 3.0.0 package for Visual Studio (MSVC) at http://www.transmissionzero.co.uk/software/freeglutdevel/. On OS X, GLUT should be installed by default. Finally, if we are on Linux and don’t have GLUT already installed, you should be able to install it by running sudo apt-get install freeglut3-dev or sudo yum install freeglut devel in a terminal. We have written the included Makefile so that the project should compile with it on Linux or OS X. For Windows, we will need to create a project in Visual Studio rather than using the Makefile.

For windows :

Put glut.h inside “C:\Program  
Files(x86)\Microsoft Visual Studio  
10.0\VC\include\**GL**\” (you may need to create the directory **GL with** yourself).  
λ Put glut32.lib inside “C:\Program  
Files\Microsoft Visual Studio 9.0\VC\lib\”.  
**Windows 32-Bit Users**: Put glut32.dll inside  
“C:\Windows\System32\”.  
**Windows 64-Bit Users**: Put glut32.dll inside  
“C:\Windows\SysWOW64\”.

**Project Details**

In this project we will Create three scenes and we will switch between them by using keyboard ( button 1 to display scene one , button 2 to display scene two , button 3 to display scene three and we use q, Q and Escape key to quit from the program).

For using keyboard we implement keyboard(unsigned char key, int x, int y) as the following

void keyboard(unsigned char key, int x, int y) {

// TODO: Your code here! use key to switch between different scene, and set q, Q and Escape key as quit

// SET 1 TO display problem 1

if (key == '1')

curProblem = 1;

// SET 2 TO display problem 2

if (key == '2')

curProblem = 2;

// SET 3 TO display problem 3

if (key == '3')

curProblem = 3;

// set q, Q and Escape key as quitset q, Q and Escape key as quit

// Escape code = 27

// q code = 113

// Q code = 81

if (key == 27 || key == 113 || key == 81)

exit(0);

glutPostRedisplay();

}

**For scene one** we use glutSolidTeapot to draw scene that display a cycle from SolidTeapot on screen and for this scene we implement problem1 function

**In problem1 function**:

We enable Material to enable the object to reflect it’s color by using   
glColorMaterial(GL\_FRONT, GL\_DIFFUSE);

glEnable(GL\_COLOR\_MATERIAL);

and we used glPushMatrix and glPopMatrix to save the original matrix of open gl   
in addition to we use glColor3f to color the object and change the color for each object in the scene

we used glTranslatef to set the position of each object on the scene by specify the x , y and z position on the scene world  
we use glRotatef to rotate each object on the scene by specify rotation angle and which x,y,z axis we rotate around by putting 1 in second argument if we want to rotate around X axis , putting 1 in third argument if we want to rotate around Y axis , putting 1 in fourth argument if we want to rotate around Z axis

we use glutSolidTeapot to draw the object on the scene with size .25

if we need to zoom in our zoom out just we change the value of third argument in glTranslatef function

**For scene two** we use glutSolidTeapot to draw scene that display a triangle from SolidTeapot on screen and for this scene we implement problem2 function

**In problem2 function**:

We enable Material to enable the object to reflect it’s color by using   
glColorMaterial(GL\_FRONT, GL\_DIFFUSE);

glEnable(GL\_COLOR\_MATERIAL);

and we used glPushMatrix and glPopMatrix to save the original matrix of open gl   
in addition to we use glColor3f to color the object and change the color for each object in the scene

we used glTranslatef to set the position of each object on the scene by specify the x , y and z position on the scene world  
we use glutSolidTeapot to draw the object on the scene with size .25

if we need to zoom in our zoom out just we change the value of third argument in glTranslatef function

**For scene three** we use glVertex3f to draw 3D Quad and for this scene we implement problem3 function

**In problem3 function**:

We enable Material to enable the object to reflect it’s color by using   
glColorMaterial(GL\_FRONT, GL\_DIFFUSE);

glEnable(GL\_COLOR\_MATERIAL);

and we used glPushMatrix and glPopMatrix to save the original matrix of open gl   
in addition to we use glColor3f to color the object and change the color for each object in the scene

we used glTranslatef to set the position of each object on the scene by specify the x , y and z position on the scene world

we use glRotatef to rotate each object on the scene by specify rotation angle and which x,y,z axis we rotate around by putting 1 in second argument if we want to rotate around X axis , putting 1 in third argument if we want to rotate around Y axis , putting 1 in fourth argument if we want to rotate around Z axis

we use glVertex3f to draw 3D Quad on the scene

if we need to zoom in our zoom out just we change the value of third argument in glTranslatef function

To setup 3D scene Projection and camera position we use

glViewport(0, 0, windowWidth, windowHeight);

to set the whole window for drawing the scene

gluPerspective(45, (windowWidth /(float)windowHeight), 1, 10);

to setup the field of view to 45 degree and the nearest object at z = 1 and forest object at z = 10

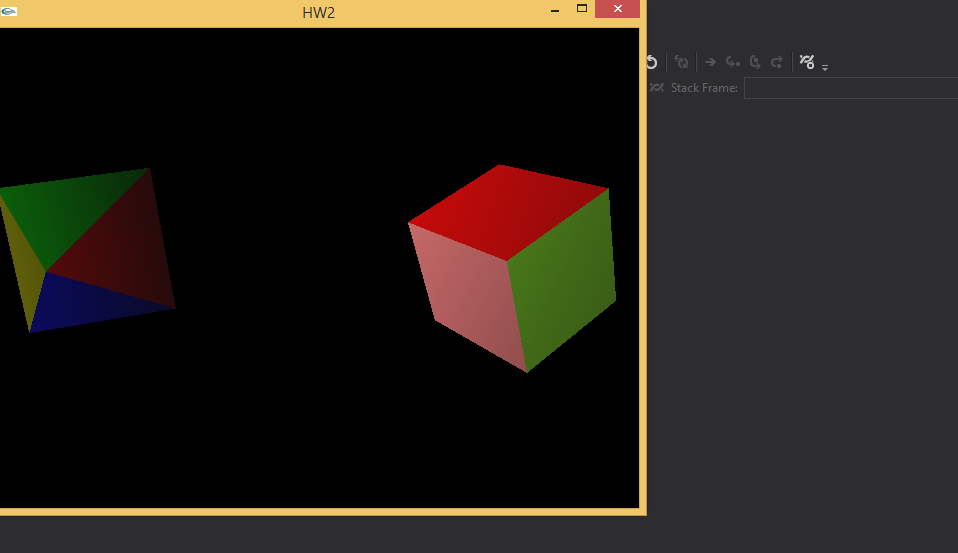
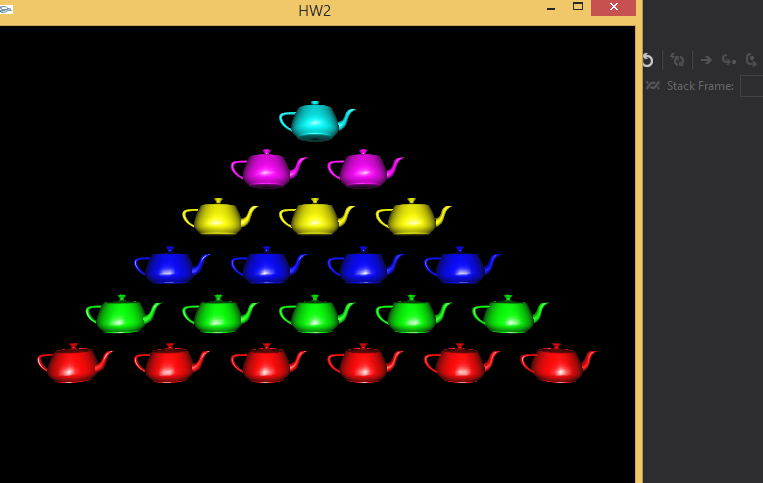
and setup camera position through

gluLookAt(cameraPos[0], cameraPos[1], cameraPos[2],

0, cameraPos[1], -10,

0, 1, 0);

**Project Result**



Reference:

Most of the syntax used in the required function were new to me. I did not know how to implement them. To learn about syntax I went to <https://stackoverflow.com/> and geeksforgeeks.com. Once I was able to figure out how the different syntax work I used so many data to figure out the correct position for the object.