#include <GL\glut.h>

#include<GL\GL.h>

#include<GL\GL.H>

float fXPos=0.0;

float fYPos=0.0;

float fZPos=0.0;

float fRot = 90.0;

void SetTransformations()

{

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45,

(double)800/ 600,

0.01,

1000);

glEnable(GL\_DEPTH\_TEST);

// set the camera here

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(

//cam pos

30, 40, 20,

//looking at

0, 0, 0,

//up dir

0,1, 0);

}

void DrawSquare()

{

glBegin(GL\_QUADS);

glVertex2f(-.5f, -.5f);

glVertex2f(.5f, -.5f);

glVertex2f(.5f, .5f);

glVertex2f(-.5f, .5f);

glEnd();

}

void DrawCube()

{

//+z square

glPushMatrix();

{

glColor3f(1, 0, 0);

glTranslatef(0, 0, .5f);

DrawSquare();

}

glPopMatrix();

//-z square

glPushMatrix();

{

glColor3f(0, 1, 0);

glTranslatef(0, 0, -.5f);

DrawSquare();

}

glPopMatrix();

//+x square

glPushMatrix();

{

glColor3f(1, .5f, 0);

glTranslatef(.5f, 0, 0);

glRotatef(90, 0, 1, 0);

DrawSquare();

}

glPopMatrix();

//-x square

glPushMatrix();

{

glColor3f(0, 1, .5f);

glTranslatef(-.5f, 0, 0);

glRotatef(90, 0, 1, 0);

DrawSquare();

}

glPopMatrix();

//+y square

glPushMatrix();

{

glColor3f(.5f, 0, 1);

glTranslatef(0, .5f, 0);

glRotatef(90, 1, 0, 0);

DrawSquare();

}

glPopMatrix();

//-y square

glPushMatrix();

{

glColor3f(1, 0, .5f);

glTranslatef(0, -.5f, 0);

glRotatef(90, 1, 0, 0);

DrawSquare();

}

glPopMatrix();

}

/// <summary>

/// Draws a 1x1 square with center at (0, 0).

/// </summary>

void OnDisplay()

{

glClearColor(1, 1, 1, 1);

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glPushMatrix();

{

glScalef(10, 10, 10);

// Inside OnPaint

//glRotatef(fRotAroundZ, 0, 1, 0);

glTranslatef(fXPos,fYPos,0);

glRotatef(fRot, 1, 0, 0);

DrawCube();

glutSolidTeapot(1);//size = 2

}

glPopMatrix();

glutSwapBuffers();

}

void OnKeyPress(unsigned char key, int x, int y)

{

if (key == 27)

exit(0);

switch(key)

{

case 'a':// a key

case 'A':

fXPos -= 0.5;

break;

case 'd':// d key

case 'D':

fXPos += 0.5;

break;

case 'w':// w key

case 'W':

fYPos += 0.5;

break;

case 's':// s key

case 'S':

fYPos -= 0.5;

break;

case 'e':

case 'E':

fRot += 5;

break;

case 'q':

case 'Q':

fRot -= 5;

break;

};

}

/\*\*

Handles the special key press. This event is whenever

a special key is being pressed.

\*/

void OnSpecialKeyPress(int key, int x, int y)

{

switch(key)

{

case GLUT\_KEY\_LEFT:// Left function key

fXPos -= 0.5;

break;

case GLUT\_KEY\_RIGHT:// Right function key

fXPos += 0.5;

break;

case GLUT\_KEY\_UP:// Up function key

fYPos += 0.5;

break;

case GLUT\_KEY\_DOWN:// Down function key

fYPos -= 0.5;

break;

};

}

void main(int argc, char \*argv[]) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGBA);

//Create an 800x600 window with its top-left corner at pixel (100, 100)

glutInitWindowPosition(100, 100); //pass (-1, -1) for Window-Manager defaults

glutInitWindowSize(800, 600);

glutCreateWindow("OpenGL Lab");

//OnDisplay will handle the paint event

glutDisplayFunc(OnDisplay);

// here is the setting of the idle function

glutIdleFunc(OnDisplay);

// here is the setting of the key function

glutKeyboardFunc(OnKeyPress);

glutSpecialFunc(OnSpecialKeyPress);

SetTransformations();

glutMainLoop();

}