#include "stdafx.h"

#include <iostream>

#include <math.h>

#include <GL/glut.h>

using namespace std;

void Primitives(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

// gluPerspective( 45.0, 1.567, 1.0, 2.0);

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3d(100, 100, 0);

glVertex3d(200, 100, 0);

glVertex3d(200, 200, 0);

glVertex3d(100, 200, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 1.0, 0.0);

glVertex3d(200, 100, 0);

glVertex3d(225, 125, 0);

glVertex3d(225, 225, 0);

glVertex3d(200, 200, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.5, 1.0, 0.0);

glVertex3d(100, 100, 0);

glVertex3d(125, 125, 0);

glVertex3d(125, 225, 0);

glVertex3d(100, 200, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.5, 0.0);

glVertex3d(100, 200, 0);

glVertex3d(125, 225, 0);

glVertex3d(225, 225, 0);

glVertex3d(200, 200, 0);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.0, 0.5, 0.0);

glVertex3d(100, 100, 0);

glVertex3d(125, 125, 0);

glVertex3d(225, 125, 0);

glVertex3d(200, 100, 0);

glEnd();

glFlush();

}

void Init()

{

glClearColor(1.0, 1.0, 1.0, 0);

glColor3f(1.0, 0.0, 0.0); // red

glViewport(0, 0, 640, 480);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0, 480, 0, 480, -100, 500);

}

int main(int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 640);

glutInitWindowPosition(0, 0);

glutCreateWindow("Primitives");

Init();

glutDisplayFunc(Primitives);

glutMainLoop();

return 0;

}



