Nafiz Fahad

20-43416-1

CODE

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

/\* clear all pixels \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

/\* draw basics shapes

\*

\*/

glColor3f (1.0, 1.0, 1.0);

glPointSize(5.0);

// draw stuff

// draw some line. task 1 draw a point in the middle of given four points..

glBegin(GL\_POINTS);

glColor3f (0.2, 0.0, 0.9); // 0.2/0/0.9

glVertex2f(0.2f,0.88f);

glVertex2f(0.5f,0.88f);

glVertex2f(0.2f,0.92f);

glVertex2f(0.5f,0.92f);

glVertex2f(0.35f,0.90f);

glEnd();

// draw 2 lines.task 2: draw 2 more lines along with the given lines

glBegin(GL\_LINES);

glColor3f (1.0, 0.2, 1.0);

glVertex2f(0.1f,0.7f);

glVertex2f(0.9f,0.7f);

glVertex2f(0.1f,0.77f);

glVertex2f(0.9f,0.77f);

glVertex2f(0.1f,0.74f);

glVertex2f(0.9f,0.74f);

glVertex2f(0.1f,0.81f);

glVertex2f(0.9f,0.81f);

glEnd();

glBegin(GL\_TRIANGLES); //Begin triangle coordinates

//Triangle.task 3: draw one more triangle

glColor3f (0.0, 0.5, 1.0);

glVertex3f(0.30f, 0.35f,0.0f);

glVertex3f(0.60f,0.35f, 0.0f);

glVertex3f(0.45f,0.55f, 0.0f);

glEnd();

glBegin(GL\_TRIANGLES);

glColor3f (0.0, 0.0, 1.0);

glVertex3f(0.20f, 0.220f,0.0f);

glVertex3f(0.50f,0.220f, 0.0f);

glVertex3f(0.35f,0.30f, 0.0f);

glEnd();//End triangle coordinates

//Task 4 draw a square

glBegin(GL\_QUADS);

glColor3f(0.0, 1.0, 0.0);

glVertex3f(.6, .090, 0.0);

glVertex3f(.8, .090, 0.0);

glVertex3f(.8, .010, 0.0);

glVertex3f(.6, .010, 0.0);

glEnd();

/\* don't wait!

\* start processing buffered OpenGL routines

\*/

glFlush ();

}

void init (void)

{

/\* select clearing (background) color \*/

glClearColor (0.0, 0.0, 0.0, 0.0);

/\* initialize viewing values \*/

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();// replaces current matrix with identity matrix

glOrtho(0.0, 1.0, 0.0, 1.0, -10.0, 10.0); // defines the size of graph paper

}

/\*

\* Declare initial window size, position, and display mode

\* (single buffer and RGBA). Open window with "Shapes"

\* in its title bar. Call initialization routines.

\* Register callback function to display graphics.

\* Enter main loop and process events.

\*/

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (600, 600);

glutInitWindowPosition (100, 100);

glutCreateWindow ("Basic Shapes");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0; /\* ISO C requires main to return int. \*/

}

**Output**

