#include <GL/glut.h>

#include<math.h>

#include<stdio.h>

void scaling(float , float);

typedef struct Point

{

float x, y;

};

Point p[3]={ {50, 50}, {150, 50}, {100, 125}};

void drawTriangle(Point p[3])

{

glColor3f(0.0, 0.0, 0.0);

glBegin(GL\_LINE\_LOOP);

glVertex2i(p[0].x, p[0].y);

glVertex2i(p[1].x, p[1].y);

glVertex2i(p[2].x, p[2].y);

glEnd();

glFlush();

}

void display()

{

int opt;

float sx, sy;

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

drawTriangle(p);

printf("\*\*\*\*\*\*\*\*\*\*\*\* Traingle Scaling \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n1. Scale along x-axis \n 2. Scale along y-axis \n 3. Scale along both x-axis and y-axis \n");

printf("Enter your option:");

scanf("%d", & opt);

switch(opt)

{

case 1: printf("\n Enter value for sx: ");

scanf("%f", &sx);

scaling(sx, 0);

break;

case 2: printf("\n Enter value for sy: ");

scanf("%f", &sy);

scaling(0, sy);

break;

case 3: printf("\n Enter value for sx : ");

scanf("%f", &sx);

printf("\n Enter value for sy : ");

scanf("%f", &sy);

scaling(sx, sy);

break;

default: return;

}

glFlush();

}

void scaling(float sx, float sy)

{ int i;

Point new\_p[3];

for(i=0;i<3;i++)

{

if(sx !=0)

new\_p[i].x = p[i].x \* sx;

else

new\_p[i].x=p[i].x;

if(sy!=0)

new\_p[i].y = p[i].y \* sy;

else

new\_p[i].y=p[i].y;

}

drawTriangle(new\_p);

}

void init(void)

{

glClearColor(1.0,1.0,1.0,0.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0.0,400.0,0.0,400.0);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(400,400);

glutInitWindowPosition(500,100);

glutCreateWindow("2d Scaling");

init();

glutDisplayFunc(display);

glutMainLoop();

}

