#include<windows.h>

#include<stdio.h>

#include<GL/gl.h>

#include<GL/glu.h>

#include<GL/glut.h>

#include<math.h>

int tx,ty,deg,sx,sy,ch,i,j,k;

void init()

{

glClearColor(0.0,0.0,0.0,0.0);

gluOrtho2D(0,700.0,0,700.0);

}

void translation()

{

int a[3][3]={{1,0,tx},{0,1,ty},{0,0,1}};

int b[3][4]={{100,200,200,100},{100,100,200,200},{1,1,1,1}};

int c[3][4]={0};

glColor3f(1.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(b[0][i],b[1][i]);

glEnd();

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

{

for(j=0;j<4;j++)

{

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

{

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

glColor3f(2.0,0.0,0.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(c[0][i],c[1][i]);

glEnd();

glFlush();

}

void rotation()

{

//int deg=10;

double theta=(deg\*3.14159)/180;

double cos1=cos(theta),sin1=sin(theta);

double a[3][3]={{cos1,-sin1,0},{sin1,cos1,0},{0,0,1}};

double b[3][4]={{100,200,200,100},{100,100,200,200},{1,1,1,1}};

double c[3][4]={0};

glColor3f(1.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(b[0][i],b[1][i]);

glEnd();

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

{

for(j=0;j<4;j++)

{

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

{

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

glColor3f(2.0,0.0,0.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(c[0][i],c[1][i]);

glEnd();

glFlush();

}

void scaling()

{

double a[3][3]={{sx,0,0},{0,sy,0},{0,0,1}};

double b[3][4]={{100,200,200,100},{100,100,200,200},{1,1,1,1}};

double c[3][4]={0};

glColor3f(1.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(b[0][i],b[1][i]);

glEnd();

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

{

for(j=0;j<4;j++)

{

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

{

c[i][j]=c[i][j]+a[i][k]\*b[k][j];

}

}

}

glColor3f(2.0,0.0,0.0);

glBegin(GL\_LINE\_LOOP);

for(i=0;i<4;i++)

glVertex2d(c[0][i],c[1][i]);

glEnd();

glFlush();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

switch(ch)

{

case 1:

translation();

break;

case 2:

rotation();

break;

case 3:

scaling();

break;

}

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowPosition(10,10);

glutInitWindowSize(500,500);

printf("Enter 1 for Translation\n");

printf("Enter 2 for Rotation\n");

printf("Enter 3 for Scaling\n");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("Enter tx and ty\n");

scanf("%d%d",&tx,&ty);

break;

case 2:

printf("Enter degree\n");

scanf("%d",&deg);

break;

case 3:

printf("Enter sx and sy\n");

scanf("%d%d",&sx,&sy);

break;

default:

printf("Invalid Choice\nrf");

}

glutCreateWindow("Square");

init();

glutDisplayFunc(display);

glutMainLoop();

return EXIT\_SUCCESS;

}