Experiment No. 6

Title: Implement following 2D transformations on the object with respect to axis

a. Translation
b. Scaling
c. Rotation about arbitrary point

Name : Samruddhi Sangram Yadav

Roll No. S564.

Code:

```
#include<GL/glut.h>
#include<math.h>

double parr[8];

void init()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glClearColor(0,0,0,1);
   glColor3f(1,0,1);
   gluOrtho2D(-500,500,-500,500); // Left,right,bottom,top

   // Polygon Default (square)
   parr[0] = 60;  parr[1] = 60;
   parr[2] = 250; parr[3] = 60;
   parr[4] = 250; parr[5] = 250;
   parr[6] = 60;  parr[7] = 250;
}

double degreeToRad(double deg)
{
   return 3.14 * (deg / 180.0);
}

void polygon()
{
   glColor3f(1,0,0);
   glBegin(GL_LINE_LOOP);
      glVertex2f(parr[0],parr[1]);
      glVertex2f(parr[2],parr[3]);
      glVertex2f(parr[4],parr[5]);
      glVertex2f(parr[6],parr[7]);
   glEnd();

   glFlush();
}

void drawCorodinates()
{
   glClear(GL_COLOR_BUFFER_BIT);

   glColor3f(1,1,1);
   glPointSize(4);

   glBegin(GL_LINES);
      glVertex2f(-500,0);
```

```
        glVertex2f(500,0);

        glVertex2f(0,500);
        glVertex2f(0,-500);
    glEnd();

    glColor3f(1,0,0);

    glBegin(GL_POINTS);
        glVertex2f(0,0);
    glEnd();

    glFlush();
}

// Translation
void translate()
{
    int x = 50, y = 30;
    for(int i = 0; i < 8; i += 2)
        parr[i] += x;

    for(int i = 1; i < 8; i += 2)
        parr[i] += y;

    polygon();
}

// Rotation about a pivot point
void rotation()
{
    double angle = 45;
    double rad = degreeToRad(angle);

    // Set your pivot point here
    double pivotX = 100;
    double pivotY = 100;

    double x, y;
    for(int i = 0; i < 8; i += 2)
    {
        // Translate point to pivot origin
        double dx = parr[i] - pivotX;
        double dy = parr[i+1] - pivotY;

        // Apply rotation
        x = dx * cos(rad) - dy * sin(rad);
        y = dx * sin(rad) + dy * cos(rad);

        // Translate back
        parr[i] = x + pivotX;
        parr[i+1] = y + pivotY;
    }
```

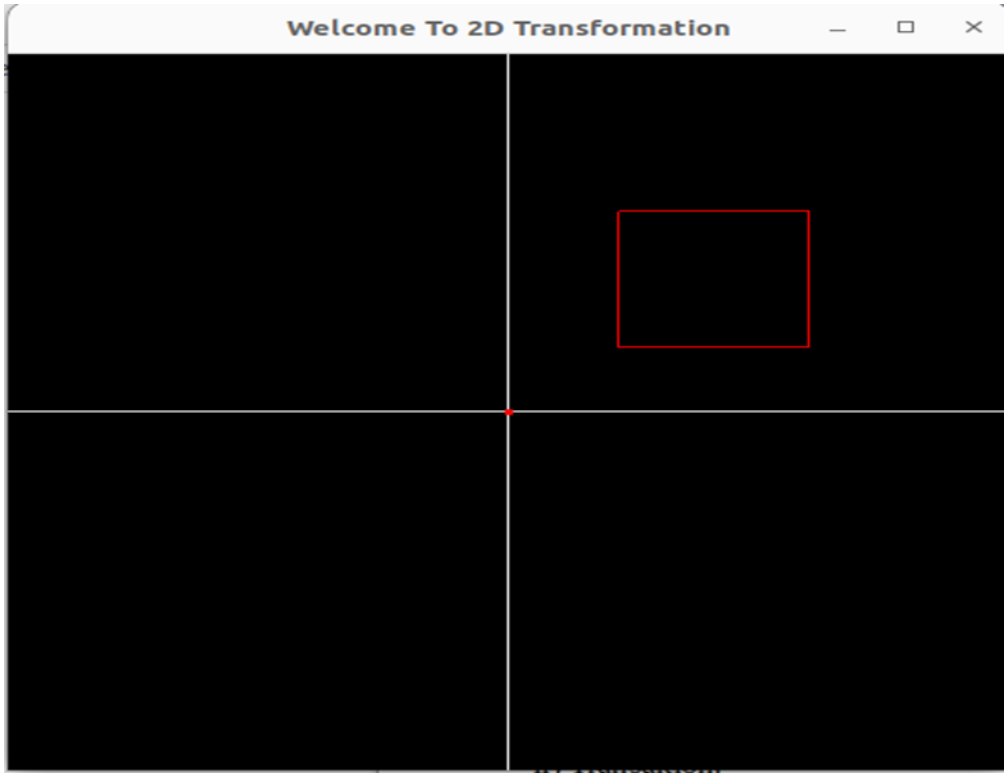
```
    polygon();
}

// Scaling
void scaling()
{
    double sx = 2, sy = 2;

    for(int i = 0; i < 8; i += 2)
    {
        parr[i] *= sx;
        parr[i+1] *= sy;
    }

    polygon();
}

void menu(int ch)
{
    drawCorodinates();
    switch(ch)
    {
        case 1: polygon(); break;
        case 2: translate(); break;
        case 3: scaling(); break;
        case 4: rotation(); break;
    }
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(100, 100);

    glutCreateWindow("Welcome To 2D Transformation");
    init();
    glutDisplayFunc(drawCorodinates);

    glutCreateMenu(menu);
        glutAddMenuEntry("1 Display Polygon", 1);
        glutAddMenuEntry("2 You're Translating", 2);
        glutAddMenuEntry("3 You're Scaling", 3);
        glutAddMenuEntry("4 You're Rotating (About Pivot)", 4);
    glutAttachMenu(GLUT_RIGHT_BUTTON);

    glutMainLoop();
    return 0;
}
```

Output:

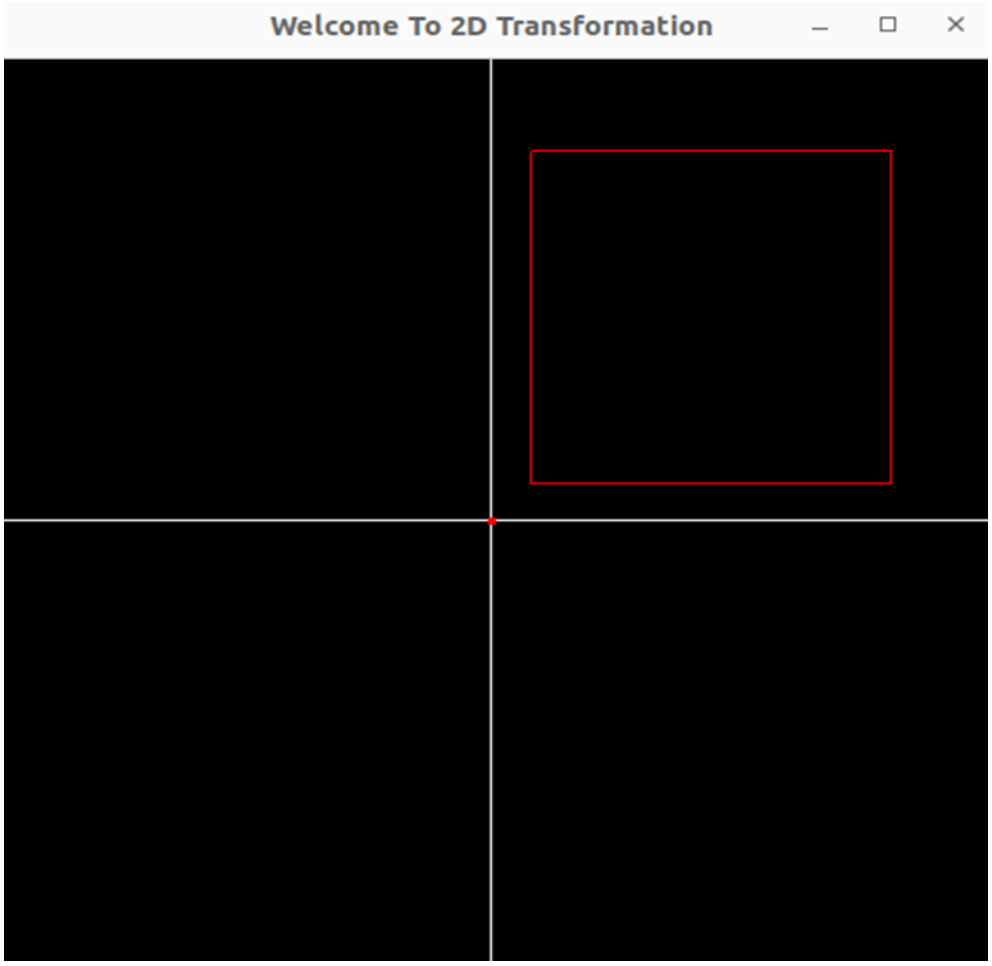
```
it@it-HP-EliteDesk-800-G2-SFF:~$ g++ 2TD.cpp -lGL -lGLU -lglut
it@it-HP-EliteDesk-800-G2-SFF:~$ ./a.out
```

i) Display polygon

ii) Transaltion:



iii)Scaling



iii) rotation with pivot point.