

## 2 D transformation

```
#include<stdio.h>
#include<math.h>
#include<GL/glut.h>
int ch;
float x1=0.5,x2=0.8,x3=0.8,x4=0.5,y=0.5,y2=0.5,y3=0.8,y4=0.8;
float X1,X2,X3,X4,Y,Y2,Y3,Y4;
void display(void)
{
    float tx,ty;
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.78,9.33,0.37);
    glPointSize(10.0);
    glBegin(GL_POLYGON);
    glVertex2f(x1,y);
    glVertex2f(x2,y2);
    glVertex2f(x3,y3);
    glVertex2f(x4,y4);
    glEnd();
    glColor3f(8080,0.0,0.0);
    glBegin(GL_POLYGON);
    glVertex2f(X1,Y);
    glVertex2f(X2,Y2);
    glVertex2f(X3,Y3);
    glVertex2f(X4,Y4);
    glEnd();
    glFlush();
}
void translate()
{
    float tx,ty;
    printf("ENTER tx AND ty VALUE\n");
    scanf("%f%f",&tx,&ty);
    X1=x1+tx;X2=x2+tx;X3=x3+tx;X4=x4+tx;
    Y=y+ty;Y2=y2+ty;Y3=y3+ty;Y4=y4+ty;
}
void rotate()
{
    int theta;
    printf("ENTER AN ANGLE\n");
    scanf("%d",&theta);
    X1=x1*cos(theta)-y*sin(theta);
    X2=x2*cos(theta)-y2*sin(theta);
    X3=x3*cos(theta)-y3*sin(theta);
```

```

        X4=x4*cos(theta)-y4*sin(theta);
        Y=x1*sin(theta)+y*cos(theta);
        Y2=x2*sin(theta)+y2*cos(theta);
        Y3=x3*sin(theta)+y3*cos(theta);
        Y4=x4*sin(theta)+y4*cos(theta);
    }
void scale()
{
    floatsx,sy;
    printf("ENTER sx AND sy VALUE\n");
    scanf("%f%f",&sx,&sy);
    X1=x1*sx;X2=x2*sx;X3=x3*sx;X4=x4*sx;
    Y=y*sy;Y2=y2*sy;Y3=y3*sy;Y4=y4*sy;
}
void init(void)
{
    glClearColor(0.0,0.0,0.0,0.0);
}
int main(int argc,char **argv)
{
    printf("2D TRANSFORMATION OPERATIONS\n");
    printf("1:TRANSLATION\n");
    printf("2:ROTATION\n");
    printf("3:SCALING\n");
    printf("ENTER UR CHOICE\n");
    scanf("%d",&ch);
    switch(ch)
    {
        case 1: translate();
                break;
        case 2: rotate();
                break;
        case 3: scale();
                break;
    }
    glutInit(&argc,argv);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutCreateWindow("2D TRANS");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

### 3D transformation

```
#include<stdio.h>

#include<math.h>

#include<GL/glut.h>

int ch;

float

x1=0.5,x2=0.8,x3=0.8,x4=0.5,y=0.5,y2=0.5,y3=0.8,y4=0.8,z1=0.6,z2=0.4,z3=0.7,z4=0.2;

float X1,X2,X3,X4,Y,Y2,Y3,Y4,Z1,Z2,Z3,Z4;

void display(void)

{

    float tx,ty;

    glClear(GL_COLOR_BUFFER_BIT);

    glColor3f(0.78,9.33,0.37);

    glPointSize(10.0);

    glBegin(GL_POLYGON);

    glVertex3f(x1,y,z1);

    glVertex3f(x2,y2,z2);

    glVertex3f(x3,y3,z3);

    glVertex3f(x4,y4,z4);

    glEnd();

    glColor3f(8080,0.0,0.0);

    glBegin(GL_POLYGON);

    glVertex3f(X1,Y,Z1);

    glVertex3f(X2,Y2,Z2);

    glVertex3f(X3,Y3,Z3);

    glVertex3f(X4,Y4,Z4);
```

```

        glEnd();

        glFlush();
    }

void translate()
{
    float tx,ty,tz;

    printf("ENTER tx ty AND tz VALUE\n");

    scanf("%f%f%f",&tx,&ty,&tz);

    X1=x1+tx;X2=x2+tx;X3=x3+tx;X4=x4+tx;

    Y=y+ty;Y2=y2+ty;Y3=y3+ty;Y4=y4+ty;

    Z1=z1+tz;Z2=z2+tz;Z3=z3+tz;Z4=z4+tz;
}

void rotate()
{
    int theta;

    printf("ENTER AN ANGLE\n");

    scanf("%d",&theta);

    X1=x1*cos(theta)-y*sin(theta);

    X2=x2*cos(theta)-y2*sin(theta);

    X3=x3*cos(theta)-y3*sin(theta);

    X4=x4*cos(theta)-y4*sin(theta);

    Y=x1*sin(theta)+y*cos(theta);

    Y2=x2*sin(theta)+y2*cos(theta);

    Y3=x3*sin(theta)+y3*cos(theta);

    Y4=x4*sin(theta)+y4*cos(theta);

    Z1=z1*cos(theta)-z1*sin(theta);

```

```

        Z2=z2*cos(theta)-z2*sin(theta);

        Z3=z3*cos(theta)-z3*sin(theta);

        Z4=z4*cos(theta)-z4*sin(theta);

    }

void scale()

{

    float sx,sy,sz;

    printf("ENTER sx ,sy AND sz VALUE\n");

    scanf("%f%f%f",&sx,&sy,&sz);

    X1=x1*sx;X2=x2*sx;X3=x3*sx;X4=x4*sx;

    Y=y*sy;Y2=y2*sy;Y3=y3*sy;Y4=y4*sy;

    Z1=z1*sz;Z2=z2*sz;Z3=z3*sz;Z4=z4*sz;

}

void init(void)

{

    glClearColor(0.0,0.0,0.0,0.0);

}

int main(int argc,char **argv)

{

    printf("3D TRANSFORMATION OPERATIONS\n");

        printf("1:TRANSLATION\n");

        printf("2:ROTATION\n");

        printf("3:SCALING\n");

        printf("ENTER UR CHOICE\n");

    scanf("%d",&ch);

    switch(ch)

```

```
{  
    case 1: translate();  
    break;  
    case 2: rotate();  
    break;  
    case 3: scale();  
    break;  
}  
glutInit(&argc,argv);  
glutInitWindowSize(500,500);  
glutInitWindowPosition(0,0);  
glutCreateWindow("3D TRANS");  
init();  
glutDisplayFunc(display);  
glutMainLoop();  
return 0;  
}
```