PRACTICAL 7

NAME: SHREYA SANJAY CHAUDHARI

ROLL NO: 04 BATCH: A DIV: COMPS 3

Experiment No 7

Aim: Implement 2D Transformations: Translation, Scaling, Rotation.

Introduction:

Transformation means changing some graphics into something else by applying rules. We can have various types of transformations such as translation, scaling up or down, rotation, shearing, etc. When a transformation takes place on a 2D plane, it is called 2D transformation. Transformations play an important role in computer graphics to reposition the graphics on the screen and change their size or orientation.

Theory:

Translation

A translation moves an object to a different position on the screen. You can translate a point in 2D by adding translation coordinate (tx, ty) to the original coordinate (X, Y) to get the new coordinate (X', Y').

$$X' = X + tx$$
$$Y' = Y + ty$$

The pair (tx, ty) is called the translation vector or shift vector. The above equations can also be represented using the column vectors.

$$P=[X][Y]\ p' = [X'][Y']T = [tx][ty]$$

We can write it as −

$$P' = P + T$$

Rotation

In rotation, we rotate the object at particular angle θ (theta) from its origin. From the following figure, we can see that the point P(X, Y) is located at angle φ from the horizontal X coordinate with distance r from the origin. Let us suppose you want to rotate it at the angle θ. After rotating it to a new location, you will get a new point P' (X', Y').

Scaling

To change the size of an object, scaling transformation is used. In the scaling process, you either expand or compress the dimensions of the object. Scaling can be achieved by multiplying the original coordinates of the object with the scaling factor to get the desired result. Let us assume that the original coordinates are (X, Y), the scaling factors are (SX, SY), and the produced coordinates are (X', Y'). This can be mathematically represented as shown below −

X' = X . SX and Y' = Y . SY

Procedure:

Name: Class:
Batch: Roll No.:

Step 1: Start
Step 2: Initialize the graphics mode.
Step 3: Construct a 2D object (use Drawpoly()) e.g. (x,y)
Step 4: A) Translation
a. Get the translation value tx, ty
b. Move the 2d object with tx, ty (x'=x+tx,y'=y+ty)
c. Plot (x',y')
Step 5: B) Scaling
a. Get the scaling value Sx,Sy
b. Resize the object with Sx,Sy (x'=x*Sx,y'=y*Sy)
c. Plot (x',y')
Step 6: C) Rotation
a. Get the Rotation angle
b. Rotate the object by the angle ϕ
x'=x cos ϕ - y sin ϕ
y'=x sin ϕ - y cosϕ
c. Plot (x',y')

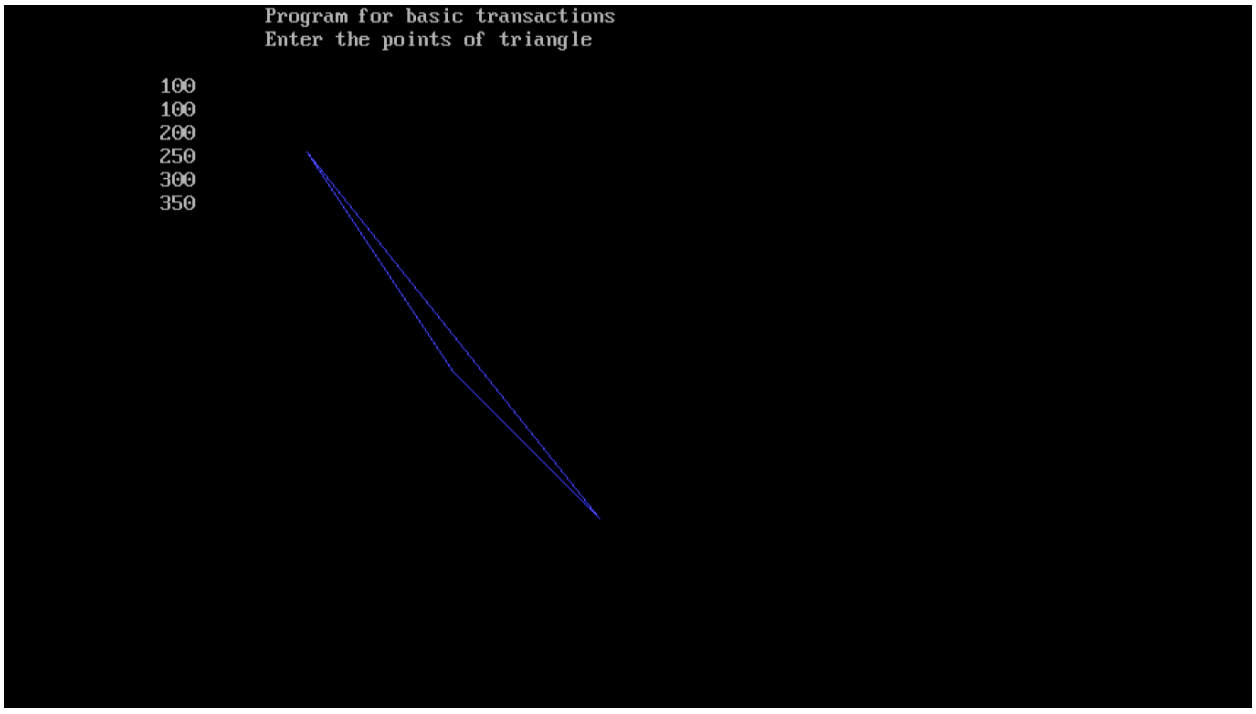
Program:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include<math.h>
void main()
{
        int gm,gd;
        int x1,x2,x3,y1,y2,y3,nx1,nx2,nx3,ny1,ny2,ny3,c;
        int sx,sy,xt,yt,r;
        float t;
        detectgraph((&gd,&gm);
        initgraph(&gd,&gm,"..//BGI");
        printf("\t Program for basic transactions");
        printf("\n\t Enter the points of triangle");
        setcolor(1);

scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3);
        line(x1,y1,x2,y2);
        line(x2,y2,x3,y3);
        line(x3,y3,x1,y1);
        getch();
        printf("\n 1.Transaction\n 2.Rotation\n
3.Scalling\n 4.exit");
        printf("Enter your choice:");
        scanf("%d",&c);
case 2:
                printf("\n Enter the angle
of rotation");
                scanf("%d",&r);
                t=3.14*r/180;

nx1=abs(x1*cos(t)-y1*sin(t));

ny1=abs(x1*sin(t)+y1*cos(t));

nx2=abs(x2*cos(t)-y2*sin(t));

ny2=abs(x2*sin(t)+y2*cos(t));

nx3=abs(x3*cos(t)-y3*sin(t));

ny3=abs(x3*sin(t)+y3*cos(t));
                line(nx1,ny1,nx2,ny2);
                line(nx2,ny2,nx3,ny3);
                line(nx3,ny3,nx1,ny1);
                getch();

        case 3:
                printf("\n Enter the
scalling factor");
                scanf("%d%d",&sx,&sy);
```

Name: Class:
Batch: Roll No.:

```
        switch(c)
        {
                case 1:
                        printf("\n Enter the translation
factor");
                        scanf("%d%d",&xt,&yt);
                        nx1=x1+xt;
                        ny1=y1+yt;
                        nx2=x2+xt;
                        ny2=y2+yt;
                        nx3=x3+xt;
                        ny3=y3+yt;
                        line(nx1,ny1,nx2,ny2);
                        line(nx2,ny2,nx3,ny3);
                        line(nx3,ny3,nx1,ny1);
                        getch();
                        nx1=x1*sx;
                        ny1=y2*sy;
                        nx2=x2*sx;
                        ny2=y2*sy;
                        nx3=x3*sx;
                        ny3=y3*sy;
                        line(nx1,ny1,nx2,ny2);
                        line(nx2,ny2,nx3,ny3);
                        line(nx3,ny3,nx1,ny1);
                        getch();
                case 4:
                        break;
                default:
                        printf("Enter the correct
choice");
                        }
                        closegraph();
                        }
```

Output:



Name: Class:

Batch: Roll No.: