



Experiment 3

Student Name: Abhiraj Patel

Branch: CSE

Semester: 6

Subject Name: Computer Graphics Lab

UID: 22BCS11329

Section/Group: KRG IOT-1 A

Date of Performance: 30/01/25

Subject Code: 22CSH-352

1. Aim:

Apply translation, scaling, and rotation transformations on a given triangle and observe the changes.

2. Objective:

To apply geometric transformations such as translation, scaling, and rotation on a given triangle.

3. Algorithm:

a) Translation:

- Initialize Graphics Mode.
- Take Input for Triangle Coordinates
- Draw the Original Triangle.
- Use the line() function to draw three lines connecting the three given points.
- Prompt the user to enter translation values tx and ty.
- Update the coordinates:

$$x1' = x1 + tx, y1' = y1 + ty$$

$$x2' = x2 + tx, y2' = y2 + ty$$

$$x3' = x3 + tx, y3' = y3 + ty$$

b) Scaling:

- Initialize Graphics Mode.
- Take Input for Triangle Coordinates

- Draw the Original Triangle
- Prompt the user to enter scaling factors s_x and s_y
- Update the coordinates of each vertex by multiplying them with the respective scaling factors:

$$x_1' = x_1 \times s_x, y_1' = y_1 \times s_y$$

$$x_2' = x_2 \times s_x, y_2' = y_2 \times s_y$$

$$x_3' = x_3 \times s_x, y_3' = y_3 \times s_y$$

Rotation:

- Initialize Graphics Mode.
- Take Input for Triangle Coordinates
- Draw the Original Triangle.
- Take Input for Rotation Angle.
- Use the rotation transformation formulas

$$x' = x \cos(\theta) - y \sin(\theta)$$

$$y' = x \sin(\theta) + y \cos(\theta)$$

4. Implementation/Code:

a) Translation:

```
#include<iostream.h>
```

```
#include<conio.h>
```

```
#include<graphics.h>
```

```
void main() {
```

```
    clrscr();
```

```
    int gd = DETECT, gm;
```

```
    initgraph(&gd, &gm, "c://turboc3//bgi");
```

```
int x1, y1, x2, y2, x3, y3, tx, ty;
cout << "Enter x1, y1: ";
cin >> x1 >> y1;
cout << "Enter x2, y2: ";
cin >> x2 >> y2;
cout << "Enter x3, y3: ";
cin >> x3 >> y3;
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x1, y1);
cout << "Enter translation value among x-axis: ";
cin >> tx;
cout << "Enter translation value among y-axis: ";
cin >> ty;
x1 += tx; x2 += tx; x3 += tx;
y1 += ty; y2 += ty; y3 += ty;
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x1, y1);
getch();
closegraph();
}
```

```
Enter x1, y1: 100 100
Enter x2, y2: 80 150
Enter x3, y3: 120 150
Enter translation value among x-axis: 30 30
Enter translation value among y-axis:
```



Fig 1: Translation

b) Scaling:

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
void main() {
    clrscr();
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "c://turboc3//bgi");
    int x1, y1, x2, y2, x3, y3, sx, sy;
    cout << "Enter x1, y1: ";
    cin >> x1 >> y1;
    cout << "Enter x2, y2: ";
    cin >> x2 >> y2;
    cout << "Enter x3, y3: ";
    cin >> x3 >> y3;
    line(x1, y1, x2, y2);
    line(x2, y2, x3, y3);
```

```
line(x3, y3, x1, y1);  
cout << "Enter scaling value among x-axis: ";  
cin >> sx;  
cout << "Enter scaling value among y-axis: ";  
cin >> sy;  
x1 *= sx; x2 *= sx; x3 *= sx;  
y1 *= sy; y2 *= sy; y3 *= sy;  
line(x1, y1, x2, y2);  
line(x2, y2, x3, y3);  
line(x3, y3, x1, y1);  
getch();  
closegraph();  
}
```

```
Enter x1, y1: 100 100  
Enter x2, y2: 80 150  
Enter x3, y3: 120 150  
Enter scaling value among x-axis: 3  
Enter scaling value among y-axis: 2
```

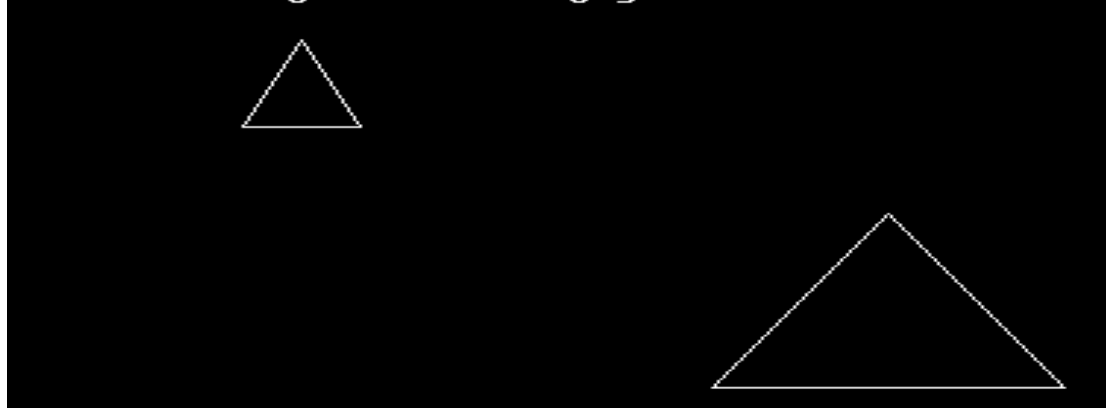


Fig 2: Scaling

c) Rotation

```
#include<iostream.h>
```

```
#include<conio.h>

#include<math.h>

#include<graphics.h>

void main() {

    clrscr();

    int gd = DETECT, gm;

    initgraph(&gd, &gm, "c://turboc3//bgi");

    int x1, y1, x2, y2, x3, y3;

    float angle;

    cout << "Enter x1, y1: ";

    cin >> x1 >> y1;

    cout << "Enter x2, y2: ";

    cin >> x2 >> y2;

    cout << "Enter x3, y3: ";

    cin >> x3 >> y3;

    line(x1, y1, x2, y2);

    line(x2, y2, x3, y3);

    line(x3, y3, x1, y1);

    cout << "Enter the rotation angle: ";

    cin >> angle;

    angle = angle * 3.1428 / 180;

    int tempX, tempY;

    tempX = x1; tempY = y1;

    x1 = tempX * cos(angle) - tempY * sin(angle);

    y1 = tempX * sin(angle) + tempY * cos(angle);
```

```
tempX = x2; tempY = y2;  
x2 = tempX * cos(angle) - tempY * sin(angle);  
y2 = tempX * sin(angle) + tempY * cos(angle);  
tempX = x3; tempY = y3;  
x3 = tempX * cos(angle) - tempY * sin(angle);  
y3 = tempX * sin(angle) + tempY * cos(angle);  
line(x1, y1, x2, y2);  
line(x2, y2, x3, y3);  
line(x3, y3, x1, y1);  
getch();  
closegraph();
```

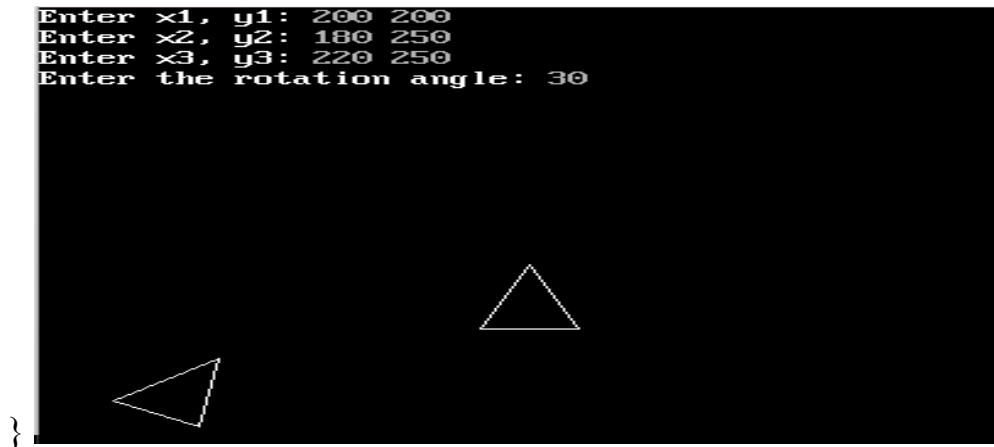


Fig 3: Rotation

5. Learning Outcome:

- Understanding Basic Graphics Programming.
- Understanding 2D Transformations.
- Understood the concept of coordinate transformation using trigonometric functions