Experiment 5

Student Name: Gaganjot Singh **UID: 22BCS14843**
Branch: B.E. CSE III Yr **Section:** 22BCS-IOT-612-B
Semester: 6th
Subject Name: Computer Graphics with Lab **Subject Code:** 22CSH-352

1. **Aim:** Implement clockwise and Anti-Clockwise rotation of a triangle about a point and evaluate results.

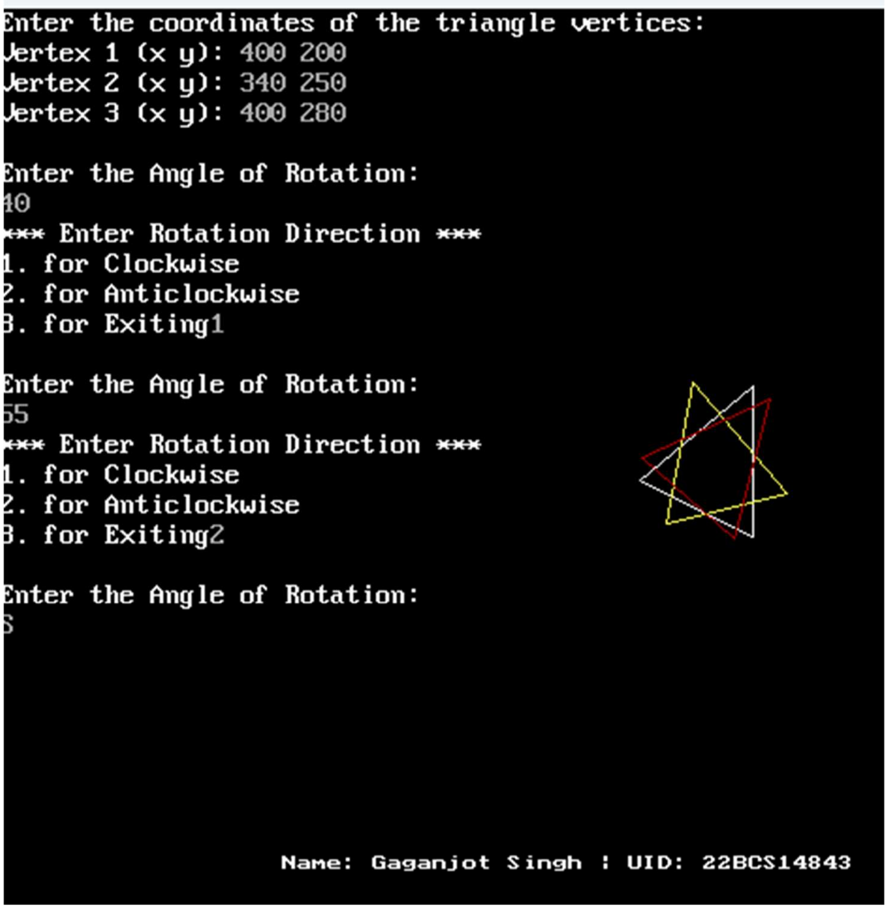
2. **Objective:** To perform and visualize clockwise and anti-clockwise rotation s of a triangle about a specified point.

3. **Code:**

```
#include <iostream.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>

void Draw(int x1, int y1, int x2, int y2, int x3, int y3, int color)
{   setcolor(color);
    line(x1, y1, x2, y2);
    line(x2, y2, x3, y3);
    line(x3, y3, x1, y1);}

void Rotate(int &x1, int &y1, int &x2, int &y2, int &x3, int &y3, float angle)
{
    float radian = angle * M_PI / 180;

    float cx = (x1 + x2 + x3) / 3.0; // Centroid calcualtion
    float cy = (y1 + y2 + y3) / 3.0;

    x1 -= cx;    y1 -= cy; // trnalste to Origin
    x2 -= cx;    y2 -= cy;
    x3 -= cx;    y3 -= cy;

    // Rotate around the origin (0, 0)
    int x1_new = int(x1 * cos(radian) - y1 * sin(radian));
    int y1_new = int(x1 * sin(radian) + y1 * cos(radian));

    int x2_new = int(x2 * cos(radian) - y2 * sin(radian));
    int y2_new = int(x2 * sin(radian) + y2 * cos(radian));

    int x3_new = int(x3 * cos(radian) - y3 * sin(radian));
    int y3_new = int(x3 * sin(radian) + y3 * cos(radian));

    x1 = x1_new + cx;
    y1 = y1_new + cy;
    x2 = x2_new + cx;
    y2 = y2_new + cy;

    x3 = x3_new + cx;
    y3 = y3_new + cy;}

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\Turboc3\\BGI");

    outtextxy(150, 450, "Name: Gaganjot Singh | UID: 22BCS14843");

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

    // Input triangle vertices
    cout << "Enter the coordinates of the triangle vertices:\n";
    cout << "Vertex 1 (x y): ";
    cin >> x1 >> y1;
    cout << "Vertex 2 (x y): ";
    cin >> x2 >> y2;
    cout << "Vertex 3 (x y): ";
    cin >> x3 >> y3;

    Draw(x1, y1, x2, y2, x3, y3, WHITE);

    float angle;
    int direction;

    int c;
    do
    {
        cout << "\nEnter the Angle of Rotation: \n";
        cin >> angle;
        cout << "\n*** Enter Rotation Direction ***\n1. for Clockwise\n2. for
Anticlockwise\n3. for Exiting";
        cin >> c;

        switch (c)
        {
        case 1:
            angle = -angle; // Clockwise rotation
            Rotate(x1, y1, x2, y2, x3, y3, angle);
            Draw(x1, y1, x2, y2, x3, y3, YELLOW);
            break;
        case 2:
            Rotate(x1, y1, x2, y2, x3, y3, angle); // Anticlockwise rotation
            Draw(x1, y1, x2, y2, x3, y3, RED);
            break;
        case 3:
            cout << "Exiting program.";
            break;
        default:
            cout << "Invalid choice!";
            break;
        }
    } while (c != 3);

    getch();    closegraph();    return 0;}
```

4. Output:

a) Red Triangle indicate ‘Anti-Clockwise’ rotation.
b) Yellow Triangle indicate ‘Clockwise’ rotation.
c) White triangle is the Original Triangle.



```
Enter the coordinates of the triangle vertices:
Vertex 1 (x y): 400 200
Vertex 2 (x y): 340 250
Vertex 3 (x y): 400 280

Enter the Angle of Rotation:
40
*** Enter Rotation Direction ***
1. for Clockwise
2. for Anticlockwise
3. for Exiting1

Enter the Angle of Rotation:
55
*** Enter Rotation Direction ***
1. for Clockwise
2. for Anticlockwise
3. for Exiting2

Enter the Angle of Rotation:
5




                         Name: Gaganjot Singh : UID: 22BCS14843
```

5. Learning Outcome:

i. Learnt to use mathematical functions from the <math.h> library (e.g., sin(), cos(), M_PI).

ii. Learnt how to apply a rotation matrix to rotate points around the origin using trigonometric functions (sin() and cos()).

iii. Learnt to calculate the centroid of a triangle, which is crucial for rotation around the center.

iv. Translation of the points from origin to the centroid.

v. Use colors for line plotting.