Experiment No:06

Experiment Name: Implementation of 2D Transformation in Computer Graphics.

(Scaling)

Description

Scaling is a fundamental 2D transformation technique in computer graphics used to change the size of an object (make it larger or smaller) in a two-dimensional plane. This transformation is achieved by multiplying the original coordinates of the object by scaling factors along the x-axis and y-axis. Scaling can be done with respect to the origin or the center (centroid) of the object.

The scaling operation is governed by the following equations:

- **$x' = x \times sx$**
- **$y' = y \times sy$**

Where:

- (x, y) are the original coordinates
- (x', y') are the scaled coordinates
- sx is the scaling factor along the x-axis
- sy is the scaling factor along the y-axis

If the scaling is done around the centroid, the object is first translated to the origin, scaled, and then translated back to its original position.

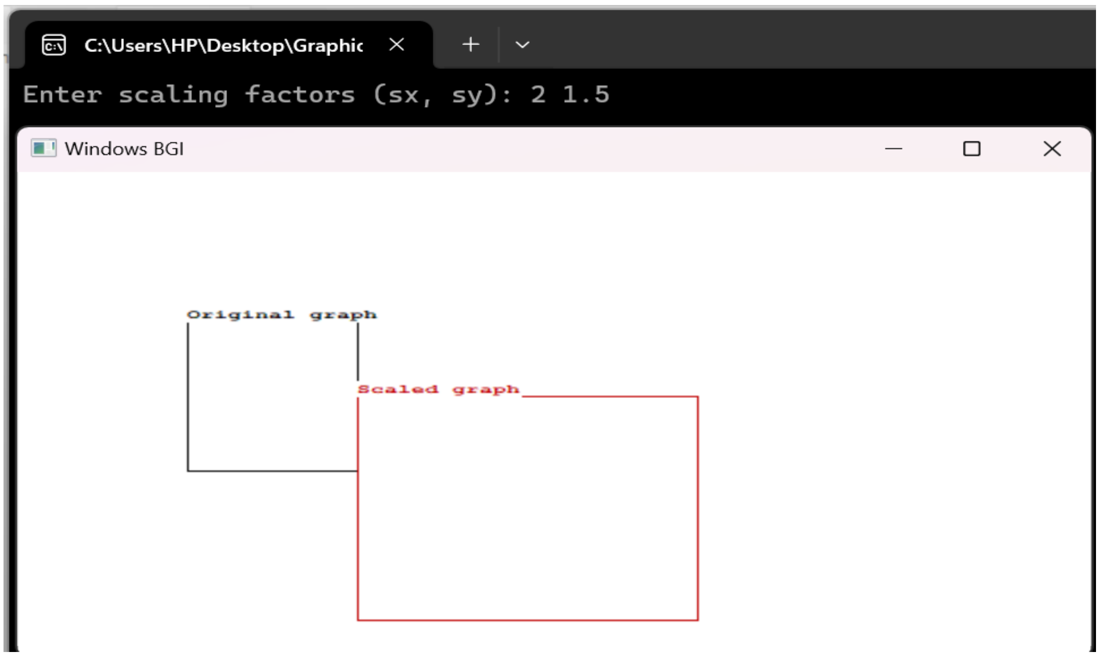
Algorithm for 2D Scaling:

```
1. Start the graphics mode using initgraph().
2. Define the coordinates of the shape (triangle or rectangle).
3. Calculate the centroid (if scaling around the center of the shape).
4. Ask the user to enter the scaling factors sx and sy.
5. For each vertex of the shape:
    o Translate the point to the origin (if scaling around the centroid).
    o Apply scaling using the formulas:
        ▪ $x' = x \times sx$
        ▪ $y' = y \times sy$
    o Translate the point back (if required).
6. Draw the original and the scaled shapes using different colors.
7. Wait for a key press using getch() and then close the graphics window using closegraph().
```

Source Code:

```
#include <graphics.h>

#include <iostream>

using namespace std;

int main() {

    int gd = DETECT, gm;

    initgraph(&gd, &gm, "");

        setbkcolor(WHITE);

    cleardevice();

 // Original rectangle

    int x1 = 100, y1 = 100, x2 = 200, y2 = 200;

 float sx, sy;

    cout << "Enter scaling factors (sx, sy): ";

    cin >> sx >> sy;

// Original shape

    setcolor(BLACK);

    rectangle(x1, y1, x2, y2);

    outtextxy(x1, y1 - 10, "Original graph");

// Scaled shape

    setcolor(RED);

    rectangle(x1 * sx, y1 * sy, x2 * sx, y2 * sy);

    outtextxy(x1 * sx, y1 * sy - 10, "Scaled graph");

 getch();

    closegraph();

    return 0;

}
```

Output:



```
C:\Users\HP\Desktop\Graphic
Enter scaling factors (sx, sy): 2 1.5
```

Figure name: Scaling.

Conclusion (Scaling):

In this experiment, we successfully implemented 2D scaling using C++ and the graphics.h library. By applying the basic mathematical formulas for scaling, we observed how the dimensions and positions of a shape change based on user-defined scaling factors. This experiment enhanced our understanding of how scaling transformations affect objects in a 2D plane, both with respect to the origin and the centroid. It demonstrated the practical use of geometric transformations in various computer graphics applications, including resizing objects in design tools, zooming in simulations, and dynamic scaling in animations and games.