Experiment No:03

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

(Translation)

Description:

2D transformation is a fundamental concept in computer graphics used to modify the position, size, or orientation of objects. Translation is one of the simplest transformations that moves an object from one location to another in the 2D plane by adding translation distances tx and ty to the original coordinates. In this experiment, a rectangle is drawn and then translated based on user input.

Algorithm:

1. **Start the graphics mode** using initgraph().
2. **Set background color** (e.g., white) and clear the screen.
3. Define the coordinates of the **original rectangle** (x1, y1, x2, y2).
4. Ask the user to **input translation values** tx and ty.
5. Draw the **original rectangle** and label it.
6. Apply translation:
    - New coordinates:
        - x1' = x1 + tx
        - y1' = y1 + ty
        - x2' = x2 + tx
        - y2' = y2 + ty
7. Draw the **translated rectangle** using the new coordinates and label it.
8. **Wait for user input**, then close the graphics window.

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;
```

```
int tx, ty;
    cout << "Enter translation (tx, ty): ";
    cin >> tx >> ty;
 // Original shape
    setcolor(BLACK);
    rectangle(x1, y1, x2, y2);
    outtextxy(x1, y1 - 10, "Original graph");
// Translated shape
    setcolor(RED);
    rectangle(x1 + tx, y1 + ty, x2 + tx, y2 + ty);
    outtextxy(x1 + tx, y1 + ty - 10, "Translated graph");
    getch();
    closegraph();
    return 0;
}
```

Output:

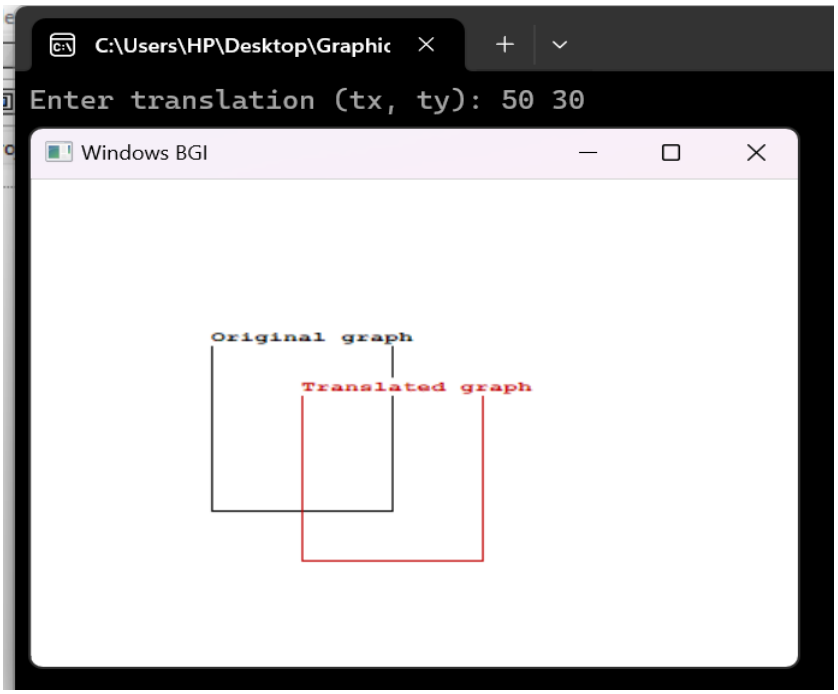

Figure name: Translation of Square.

Conclusion:

Through this experiment, we successfully implemented the 2D translation transformation in computer graphics using C++. The translation operation was clearly visualized by shifting the position of the original rectangle based on user-provided translation values. This experiment helped in understanding how geometric transformations work in computer graphics, which is essential for rendering, animation, and interactive applications. It also introduced practical usage of the graphics.h library for visual output.

Required Tools:

- **Dev-C++ IDE**
- **WinBGIm Graphics Library** (included with or added to Dev-C++)
- **C++ Compiler** (such as TDM-GCC that comes with Dev-C++)
- **Windows Operating System** (Graphics.h is platform-specific)