Experiment No:06

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

(Reflection)

Description:

Reflection is a 2D transformation in computer graphics that produces a mirror image of an object across a specified axis or line. It is commonly used to create symmetrical designs or simulate the effect of mirrors.

Reflection transformations flip the object over a reference line, such as:

- The **x-axis**
- The **y-axis**
- The **origin**
- An arbitrary line (more advanced)

Reflection changes the **s**ign of one or both coordinates depending on the axis of reflection, while preserving the shape and size of the object.

Algorithm:

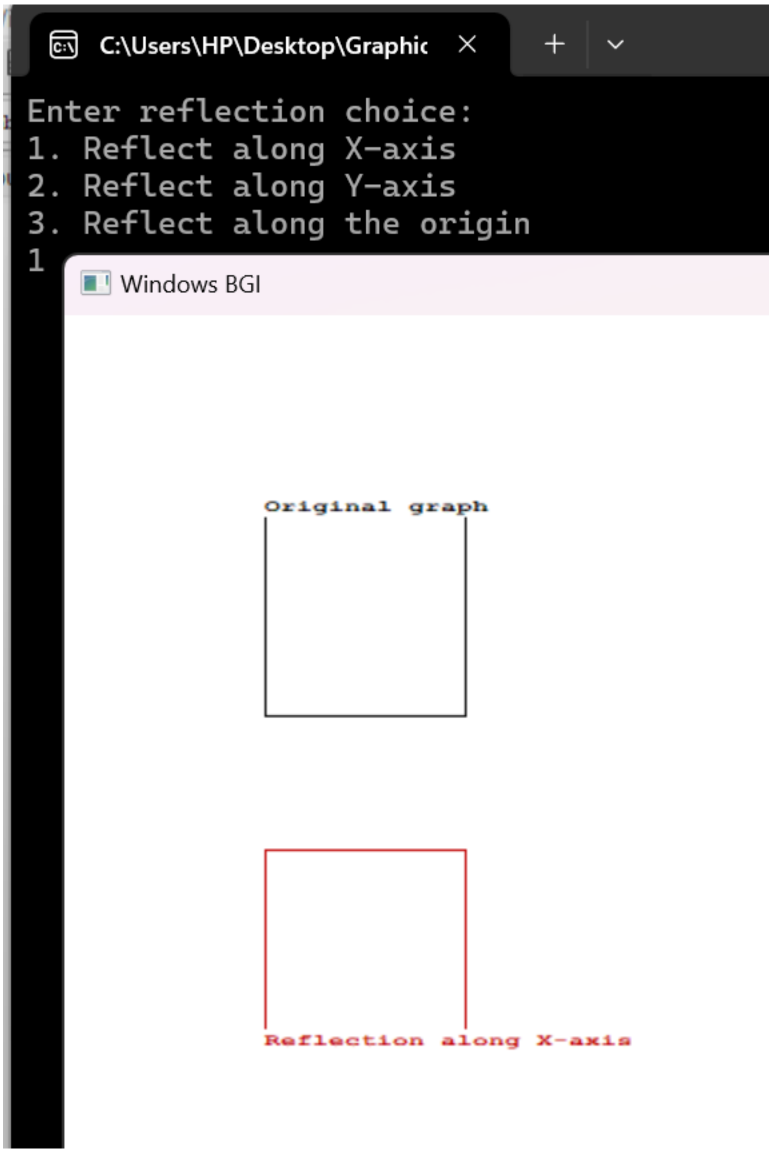
1. **Start**
2. Define the original coordinates of the object (e.g., triangle, square).
3. Ask the user to choose the axis or line of reflection:
    - X-axis
    - Y-axis
    - Origin
    - y = x
4. Based on the user's choice, apply the corresponding reflection transformation:
    - For X-axis: x′=x,y′=−yx' = x, y' = -yx′=x,y′=−y
    - For Y-axis: x′=−x,y′=yx' = -x, y' = yx′=−x,y′=y
    - For Origin: x′=−x,y′=−yx' = -x, y' = -yx′=−x,y′=−y
    - For y = x: x′=y,y′=xx' = y, y' = xx′=y,y′=x
5. Store the new coordinates.
6. Display the original and the reflected object.
7. **End**

Source Code:

```
#include <graphics.h>

#include <iostream>

using namespace std;
```

```
int main() {

    int gd = DETECT, gm;

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

     setbkcolor(WHITE);  // Set background color to white

    cleardevice();

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

cout << "Enter reflection choice:\n1. Reflect along X-axis\n2. Reflect along Y-axis\n3. Reflect along the origin\n";

    cin >> choice;// Original shape

    setcolor(BLACK);

    rectangle(x1, y1, x2, y2);

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

  // Reflection

    setcolor(RED); if (choice == 1) {

        // Reflecting along the X-axis (invert Y-coordinates)

        rectangle(x1, -y1 + getmaxy(), x2, -y2 + getmaxy());

        outtextxy(x1, -y1 + getmaxy() - 10, "Reflection along X-axis");

      else if (choice == 2) {

        // Reflecting along the Y-axis (invert X-coordinates)

        rectangle(-x1 + getmaxx(), y1, -x2 + getmaxx(), y2);

        outtextxy(-x1 + getmaxx(), y1 - 10, "Reflection along Y-axis");

    }  else if (choice == 3) {

        // Reflecting along the origin (invert both X and Y coordinates)

        rectangle(-x1 + getmaxx(), -y1 + getmaxy(), -x2 + getmaxx(), -y2 + getmaxy());

        outtextxy(-x1 + getmaxx(), -y1 + getmaxy() - 10, "Reflection along Origin");

    }

 getch();

    closegraph();
return 0}
```

Output:



```
Enter reflection choice:
1. Reflect along X-axis
2. Reflect along Y-axis
3. Reflect along the origin
1
```

Figure name: Reflecting of Square.

Conclusion:

In this experiment, we successfully implemented 2D reflection transformations in computer graphics. We demonstrated how objects can be mirrored across various axes or lines using simple matrix operations. This transformation preserves the size and shape of the object while changing its orientation. Understanding reflection is essential in graphics programming for achieving visual symmetry and creating mirror-like effects in animations, games, and simulations.