Lab Report No: 01

Problem Name: Implementation of DDA Algorithm.

Description: The Digital Differential Analyzer (DDA) Algorithm is a line-drawing algorithm used in computer graphics to generate a straight line between two points. It is an incremental method that efficiently determines the intermediate pixel positions along the line path based on the equation of a straight line. The DDA algorithm works by calculating small increments in the x and y coordinates from the starting point to the ending point. Instead of using complex floating-point arithmetic, the algorithm utilizes simple addition operations, making it efficient and fast.

Algorithm:

Given-

Starting coordinates = (X0, Y0)

Ending coordinates = (Xn, Yn)

And Xp=X0, Yp=Y0

Step: 01

Calculate ΔX, ΔY and M from the given input.

These parameters are calculated as-

ΔX = Xn – X0

ΔY =Yn – Y0

M = ΔY / ΔX

Step:02

Find the number of steps or points in between the starting and ending coordinates.

if (absolute (ΔX) > absolute (ΔY))

Steps = absolute (ΔX);

else

Steps = absolute (ΔY);

Step: 03:

Suppose the current point is (Xp, Yp) and the next point is (Xp+1, Yp+1).

Find the next point by following the below three cases-

Case:01: If M<1

- Xp+1=round off (1+Xp)
- Yp+1=round off (M+Yp)

Case:02: If M=1

- Xp+1=round off (1+Xp)
- Yp+1=round off (1+Yp)

Case:03: If M>1

- Xp+1=round off (1/M+Xp)
- Yp+1=round off (1+Yp)

Step-04:

Keep repeating Step-03 until the end point is reached or the number of generated new points (including the starting and ending points) equals to the steps count.

Code:

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

void drawLineDDA(int x1, int y1, int x2, int y2) {
    int dx = x2 - x1;
    int dy = y2 - y1;

    int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);
```

```
    float Xinc = dx / (float)steps;
    float Yinc = dy / (float)steps;

    float X = x1;
    float Y = y1;
      // Draw pixels
    for (int i = 0; i <= steps; i++) {
        putpixel(round(X), round(Y), BLACK); // Draw pixel in Yellow
        X += Xinc;
        Y += Yinc;
    }
}

int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, ""); // Initialize graphics mode

    setbkcolor(WHITE); // Set background color to White
    cleardevice();     // Apply the background color

    int x1 = 100, y1 = 100, x2 = 200, y2 = 210;
    drawLineDDA(x1, y1, x2, y2);

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

Output:

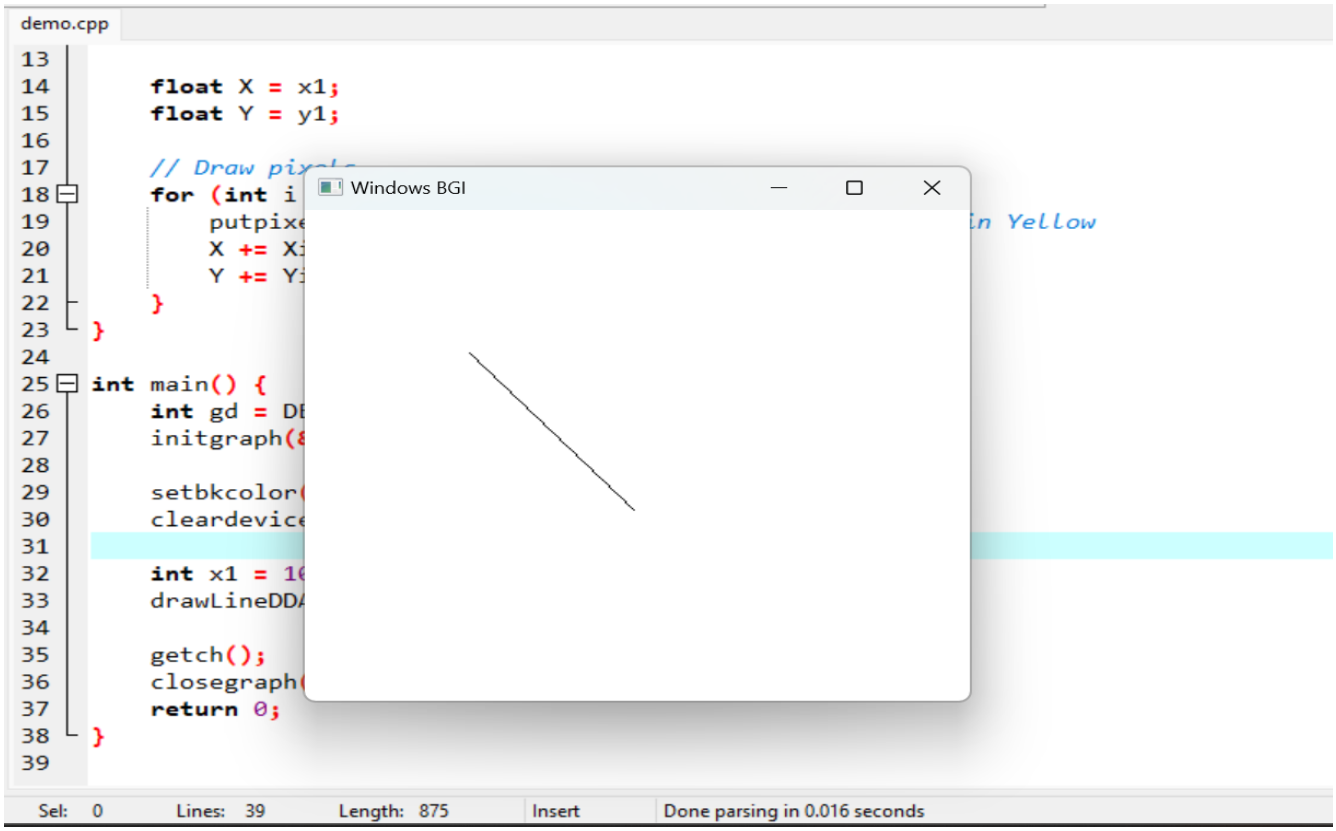

Figure Name: System Output.

Conclusion:

The DDA (Digital Differential Analyzer) Algorithm is a simple and efficient method for drawing a straight line by calculating and plotting points between the start and end coordinates. In this lab, we observed that DDA produces smooth and accurate lines by incrementally updating pixel positions. It effectively handles different slopes and provides a clear understanding of how lines are drawn on a computer screen. This experiment helped us learn an important technique used in computer graphics for line rendering.

Remarks: Using App:

DEV C++