

Lab Report 2

Title: To use DDA algorithm to draw a line between given points.

Theory: The Digital Differential Analyzer(DDA) is a algorithm widely used method in computer graphics for drawing line between two specified points. The algorithm works by calculating difference in x and y coordinates (i.e dx and dy). Slope is calculated by using the difference dy and dx (i.e slope = dy/dx). This slope is now used to draw line between two points, and the initial x and y point are incremented according to it.

DDA involves floating point arithmetic to calculate the increment of x and y coordinates. This can lead to rounding errors and precision issues. This thing can make it less efficient compared to Bresenham's line algorithm. Although DDA is straightforward for linear path , the implementation becomes more complex when extended to non linear par

DDA Algorithm:

- Step 1: Start
- Step 2: Take the coordinates of the initial point (x_initial, y_initial) and the final point (x_final, y_final).
- Step 3: Find the difference between the points by using the formula:
 $dx = x_final - x_initial;$
 $dy = y_final - y_initial;$
- Step 4: Calculate the slope of the line
 $slope = dy/dx$
- Step 5: Delay the drawing for visibility
 $delay(100);$
- Step 6: Begin drawing the line using the following logic:
 - If slope < 1:
 $x_initial = x_initial + 1;$
 $y_initial = y_initial + slope;$
 - If slope > 1:
 $x_initial = x_initial + (1/slope);$
 $y_initial = y_initial + 1;$
 - If slope == 1:
 $x_initial = x_initial + 1;$
 $y_initial = y_initial + 1;$
- Step 7: Put the pixel at the current coordinates:
 $putpixel(x_initial, y_initial, RED);$
- Step 8: Repeat step 6 & 7 until (x_initial, y_initial) reaches (x_final, y_final).
- Step 9: Stop

The code:

The code for the asked program is given below:

```
// DDA algorithm to draw a line. And using the application of the line, draw the letter KSHITIJ
//Including preprocessor directives
#include <graphics.h>
#include <conio.h>
void DDA(int x_initial , int y_initial , int x_final, int y_final){
    int dx = x_final - x_initial;
    int dy = y_final - y_initial;
    float slope = (dy)/(dx);
    delay(100);
    while(x_initial != x_final && y_initial != y_final)
    {

        if(slope < 1)
        {
            x_initial = x_initial + 1;
            y_initial = y_initial + slope;
        }
        else if (slope > 1)
        {
            y_initial = y_initial + 1;
            x_initial = x_initial + (1/slope);
        }
        else if (slope == 1)
        {
            x_initial = x_initial + 1;
            y_initial = y_initial + 1;
        }
        putpixel(x_initial, y_initial, RED);
    }
}
//main function
int main()
{
    int gd = DETECT, gm;

    // Initialize graphics function
    initgraph(&gd, &gm, "");
    initwindow(1100, 900 , "Graphics Lab");
    // Draw 'K'
```

```

DDA(50, 50, 51, 250);
DDA(50, 150, 100, 50);
DDA(50, 150, 101, 250);

// Draw 'S'
DDA(170, 50, 270, 51);
DDA(170, 150, 270, 151);
DDA(170, 250, 270, 251);
DDA(170, 50, 171, 150);
DDA(270, 150, 271, 250);

// Draw 'H'
DDA(290, 50, 291, 250);
DDA(390, 50, 391, 250);
DDA(290, 150, 390, 151);

// Draw 'I'
DDA(410, 50, 510, 51);
DDA(460, 50, 461, 250);
DDA(410, 250, 510, 251);

// Draw 'T'
DDA(530, 50, 630, 51);
DDA(580, 50, 581, 250);

// Draw 'I' again
DDA(650, 50, 750, 51);
DDA(700, 50, 701, 250);
DDA(650, 250, 750, 251);

// Draw 'J'
DDA(770, 50, 970, 51); // Top horizontal line
DDA(870, 50, 871, 250); // Vertical line
DDA(770, 250, 870, 251); // Bottom horizontal line

//Concluding the code
getch();
closegraph();
return 0;
}

```

The output:

The output of the asked program is given below:



Figure 1: Letter “KSHITIJ” drawn using application of DDA algorithm.

Conclusion:

The DDA algorithm effectively demonstrates how to draw lines between given points. The code includes basic conditional cases to draw line. This program efficiently plots multiple characters, using the line drawing algorithm , and some basic coordinate math involved, which finally results in forming entire word using DDA. This showcases the versatility and practical application of the DDA algorithm in computer graphics.