

Assignment-3 - B1

Title : To draw lines with different patterns.

Problem: Write a C++ program for line drawing using DDA & Bresenham algorithms with patterns such as solid, dotted, dash, dash dot and thick.

Objective:
 1) To understand concept of different line styles eg: thin, thick, dotted.
 2) To implement DDA/Bresenham for drawing lines.

Outcome: Student should be able to draw a line with different patterns.

Theory: Draw a dashed and dotted line using DDA algorithm, take the line coordinate from the user to plot the desired line. The program starts with one given point and then calculate each successive pixel that lies on the line using the line using DDA algorithm.

1) Solid line → Using simple DDA line drawing algorithm (Refer ass 1)

2) Dashed line →

Dashed (float, float, float, float)

begin

while (i ≤ len)

if (i % 5 != 0) then

setpixel (x, y, color)

```

x = x + xinc
y = y + yinc
i++
end while
end dashed

```

Dotted line →

```

dotted(float, float, float, float)
begin
    i = 0
    while (i ≤ len)
        if (i % 5 == 0)
            setPixel(x, y, color)
            x = x + dx
            y = y + dy
        end while
    end dotted

```

4) Dash-dot

```

dash_dot(float, float, float, float)
begin
    i = 0
    j = 0
    while (i ≤ len)
        if (j ≤ 10)
            setPixel(x, y, color)
            if (j == 15)
                setPixel(x, y, color)
            end if
            x = x + xinc
            y = y + yinc
            i++, j++, j = j % 20
        end if
    end while
end dash_dot

```


end white
end ~~dotted~~ dash dash dot

5] Thick line \rightarrow also w : width as input
float $(dy)^2 = (y_2 - y_1) * (y_2 - y_1)$

$$(dx)^2 = (x_2 - x_1) * (x_2 - x_1)$$

$$temp = \text{sqrt}((dy)^2 + (dx)^2)$$

if $((y_2 - y_1) < (x_2 - x_1))$

{ float temp1 = temp / abs(x2 - x1)

int wy = ((w - 1) * temp1) / 2

for (int i = 0; i < wy; i++)

{

DDA (x1, y1 + i, x2, y2 + i)

DDA (x1, y1 - i, x2, y2 - i)

}

}

else

{

temp1 = temp / abs(y2 - y1)

int wx = ((w - 1) * temp1) / 2

for (int i = 0; i < wx; i++)

{

DDA (x1 + i, y, x2 + i, y)






DDA (x1 - i, y, x2 - i, y)

}

}

end Thick-

Test Cases →

I/P	Exp/O/P	actual O/P	result
Solid	Solid line		Pass
dotted line	dotted line		Pass
dashed dot	dash-dotted line		Pass
Dashed	dash line		Pass
Thick line	Thick line		Pass

Conclusion: We have successfully drawn various types of lines using DDA and Bresenham algorithm.

©
my