**Practical No. 3**

***Title:-***Implement **DDA Algorithm** for Line Drawing.

***Course outcome*** :- Apply the algorithms to draw lines, circle and polygons.

***Resources Required (Hardware & Softwares):-***

1. A Desktop PC/ Laptop
2. Ansi C/ Turbo C/ (Any distribution) installed

***Theory:-***

***DDA Line Drawing Algorithm***

DDA Algorithm generates a line from differential equation of line

The slope(m) of a straight line is given by

*m= y2-y1 = Δy*

*x2-x1 Δx*

where (x1,y1) and (x2,y2) are two end points of a line.

For any given x interval(Δx) we can calculate corresponding y interval(Δy)

*Δy=y2-y1 Δx*

*x2-x1*

Similarly for given Δy we can calculate corresponding Δx as

*Δx= x2-x1 Δy*

*y2-y1*

Once the intervals are known the values for next x and next y on straight line can be obtained as

*xi+1=xi+1+Δx -eq 1*

and *yi+1=yi+1+Δy -eq 2*

For simple DDA either Δx or Δy whichever is larger is chosen as one raster unit i.e.

*if |Δx|>=|Δy| then*

*length=Δx*

*Else*

*length=Δy*

***Algorithms:-***

1. **DDA Algorithm:-**
2. Read the line end points(x1,y1) and (x2,y2) such that they are not equal[if equal then plot that point and exit]

2. Δx=|x2-x1| and Δy=|y2-y1|

3. If(Δx>=y) then

length=Δx

else

length=Δy

end if

4. Δx=(x2-x1) /length Δy=(y2-y2)/length

5. x=x1+0.5\* sign(Δx)

y=y1+0.5 \* sign(Δy)

[Here sign function makes algorithm works in all quadrants returns -1,0,1 when argument is <0,=0, >0 resp.]

6. i=1

while (i<=length)

{

plot(integer(x),integer(y))

x=x+Δx

y=y+Δy

i=i+1

}

7. Stop.

**Program for DDA Line drawing Algorithm**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

void main()

{

float x,y,x1,y1,x2,y2,dx,dy,length;

inti,gd=DETECT,gm;

clrscr();

printf("Enter the value of x1 :\t");

scanf("%f",&x1);

printf("Enter the value of y1 :\t");

scanf("%f",&y1);

printf("Enter the value of x2 :\t");

scanf("%f",&x2);

printf("Enter the value of y2 :\t");

scanf("%f",&y2);

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

{

length=dx;

}

else

{

length=dy;

}

dx=(x2-x1)/length;

dy=(y2-y1)/length;

x=x1+0.5;

y=y1+0.5;

i=1;

while(i<=length)

{

putpixel(x,y,15);

x=x+dx;

y=y+dy;

i=i+1;

delay(100);

}

getch();

closegraph();

}

**Output : - ( Paste your own Output )**

***Conclusion:-***

Thus, we have implemented **DDA Algorithm** for Line Drawing.