**Assignment no: 06**

**//Aim:Write a program using C/C++ to draw a line with line styles (Thick, Thin, Dotted).**

//Line styles using DDA

#include<graphics.h>

#include<iostream>

#include<stdlib.h>

#include<math.h>

using namespace std;

class pt //base class

{

protected: int xco,yco,color;

public:

pt()

{

xco=0;yco=0;color=15;

}

void setco(int x,int y)

{

xco=x;

yco=y;

}

void setcolor(int c)

{

color=c;

}

void draw()

{ putpixel(xco,yco,color);

}

};

class dline: public pt //derived class

{

private: int x2,y2;

public:

dline():pt()

{

x2=0,y2=0;

}

void setline(int x, int y, int xx, int yy)

{

pt::setco(x,y);

x2=xx;

y2=yy;

}

void drawsi(int colour) //Simple DDA Line

{

float x,y,dx,dy,len;

int i;

pt::setcolor(colour);

dx=abs(x2-xco);

dy=abs(y2-yco);

if(dx >= dy)

{

len=dx;

}

else

{

len=dy;

}

dx=(x2-xco)/len;

dy=(y2-yco)/len;

x = xco + 0.5;

y = yco + 0.5;

i=1;

while(i<=len)

{

pt::setco(x,y);

pt::draw();

x = x + dx;

y = y + dy;

i = i + 1;

}

pt::setco(x,y);

pt::draw();

}

void drawda(int colour) //Dash DDA Line

{

float x,y,dx,dy,len;

int i,dash\_pixel=0, dash\_space=0;

pt::setcolor(colour);

dx=abs(x2-xco);

dy=abs(y2-yco);

if(dx >= dy)

{

len=dx;

}

else

{

len=dy;

}

dx=(x2-xco)/len;

dy=(y2-yco)/len;

x = xco + 0.5;

y = yco + 0.5;

i=1;

while(i<=len)

{

dash\_pixel=0;

while(dash\_pixel<5)

{

pt::setco(x,y);

pt::draw();

x = x + dx;

y = y + dy;

i = i + 1;

dash\_pixel = dash\_pixel +1;

}

dash\_space=0;

while(dash\_space<=2)

{

x = x + dx;

y = y + dy;

i = i + 1;

dash\_space = dash\_space +1;

}

}

}

void drawdo(int colour) //Dotted DDA Line

{ float x,y,dx,dy,len;

int i,dot\_space;

pt::setcolor(colour);

dx=abs(x2-xco);

dy=abs(y2-yco);

if(dx >= dy)

{

len=dx;

}

else

{

len=dy;

}

dx=(x2-xco)/len;

dy=(y2-yco)/len;

x = xco + 0.5;

y = yco + 0.5;

i=1;

while(i<=len)

{

dot\_space=0;

while(dot\_space<=1)

{

x = x + dx;

y = y + dy;

i = i + 1;

dot\_space = dot\_space +1;

}

pt::setco(x,y);

pt::draw();

}

}

void drawth(int x1,int y1, int x2, int y2,int colour ) //Thick DDA Line

{

float x,y,dx,dy,len;

int i;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx >= dy)

{ len=dx;

}

else

{ len=dy;

}

dx=(x2-x1)/len;

dy=(y2-y1)/len;

x = x1 + 0.5;

y = y1 + 0.5;

i=1;

while(i<=len)

{ putpixel(x,y,colour);

x = x + dx;

y = y + dy;

i = i + 1;

}

putpixel(x,y,colour);

}

};

int main()

{

int gd=DETECT,gm=VGAMAX;

int i, ch,x1,y1,x2,y2, dx,dy,xmax,ymax,xmid,ymid,wx,wy,th;

char a;

initgraph(&gd,&gm,NULL);

dline ls;

xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

line(xmid,0,xmid,ymax); //Y co-ordinate

line(0,ymid,xmax,ymid); //X co-ordinate

do

{ xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

cout<<"\nEnter Line Styles";

cout<<"\n1.SIMPLE..";

cout<<"\n2.DASH..";

cout<<"\n3.DOTTED..";

cout<<"\n4.THICK..";

cout<<"\n5.EXIT..";

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawsi(15);

break;

case 2:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawda(15);

break;

case 3:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawdo(15);

break;

case 4:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

cout<<"Enter Thickness: ";

cin>>th;

ls.drawth(x1+xmid,ymid-y1,x2+xmid,ymid-y2,15);

if((y2-y1)/(x2-x1) <1)

{

wy=(th-1)\*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2\*abs(x2-x1));

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

{

ls.drawth(x1+xmid,ymid-y1-i,x2+xmid,ymid-y2-i,15);

ls.drawth(x1+xmid,ymid-y1+i,x2+xmid,ymid-y2+i,15);

}

}

else

{

wx=(th-1)\*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2\*abs(y2-y1));

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

{

ls.drawth(x1+xmid-i,ymid-y1,x2+xmid-i,ymid-y2,15);

ls.drawth(x1+xmid+i,ymid-y1,x2+xmid+i,ymid-y2,15);

}

}

break;

case 5:

exit;

break;

}

cout<<"\nDO U Want To Continue y OR n: ";

cin>>a;

}while(a!='n');

delay(3000);

getch();

closegraph();

return 0;

}

/\*

gescoe@gescoe-VirtualBox:~/Downloads/cgprogram012$ ./a05

Enter Line Styles

1.SIMPLE..

2.DASH..

3.DOTTED..

4.THICK..

5.EXIT..

Enter your choice:1

Enter x1: 10

Enter y1: 10

Enter x2: 60

Enter y2: 60

DO U Want To Continue y OR n: y

Enter Line Styles

1.SIMPLE..

2.DASH..

3.DOTTED..

4.THICK..

5.EXIT..

Enter your choice: 2

Enter x1: -10

Enter y1: 10

Enter x2: -80

Enter y2: 80

DO U Want To Continue y OR n: y

Enter Line Styles

1.SIMPLE..

2.DASH..

3.DOTTED..

4.THICK..

5.EXIT..

Enter your choice: 3

Enter x1: -10

Enter y1: -10

Enter x2: -100

Enter y2: -100

DO U Want To Continue y OR n: y

Enter Line Styles

1.SIMPLE..

2.DASH..

3.DOTTED..

4.THICK..

5.EXIT..

Enter your choice: 4

Enter x1: 10

Enter y1: -10

Enter x2: 100

Enter y2: -100

Enter Thickness: 10

DO U Want To Continue y OR n: n \*/

---------------------------------------------------------------------------------------------------

**//line styles using Bresenham's algo**

#include<graphics.h>

#include<iostream>

#include<stdlib.h>

#include<math.h>

using namespace std;

class pt //base class

{

protected: int xco,yco,color;

public:

pt()

{

xco=0;yco=0;color=15;

}

void setco(int x,int y)

{

xco=x;

yco=y;

}

void setcolor(int c)

{

color=c;

}

void draw()

{ putpixel(xco,yco,color);

}

};

class dline: public pt //derived class

{

private: int x2,y2;

public:

dline():pt()

{

x2=0,y2=0;

}

void setline(int x, int y, int xx, int yy)

{

pt::setco(x,y);

x2=xx;

y2=yy;

}

void drawsi() //Bresenham's Line

{

float x,y,dx,dy,e,temp;

int i,s1,s2,ex;

dx=abs(x2-xco);

dy=abs(y2-yco);

x=xco;

y=yco;

pt::setco(x,y);

pt::draw();

if(x2 > xco) //sign() function

{

s1=1;

}

if(x2 < xco)

{

s1=-1;

}

if(x2 == xco)

{

s1=0;

}

if(y2 > yco)

{

s2=1;

}

if(y2 < yco)

{

s2=-1;

}

if(y2 == yco)

{

s2=0;

}

if(dy > dx)

{

temp = dx;

dx = dy;

dy = temp;

ex = 1;

}

else

{

ex=0;

}

e=2\*dy-dx; //decision variable

i=1;

do

{

while(e>=0)

{

if(ex==1)

{

x = x + s1;

}

else

{

y = y + s2;

}

e = e + 2\*dy - 2\*dx;

}

if(ex==1)

{

y = y + s2;

}

else

{

x = x + s1;

}

e = e + 2\*dy;

pt::setco(x,y);

pt::draw();

i = i + 1;

}while(i<=dx);

}

void drawda() //Dash Line

{

float x,y,dx,dy,e,temp;

int i,s1,s2,ex, dash=0;

dx=abs(x2-xco);

dy=abs(y2-yco);

x=xco;

y=yco;

pt::setco(x,y);

pt::draw();

if(x2 > xco) //sign() function

{

s1=1;

}

if(x2 < xco)

{

s1=-1;

}

if(x2 == xco)

{

s1=0;

}

if(y2 > yco)

{

s2=1;

}

if(y2 < yco)

{

s2=-1;

}

if(y2 == yco)

{

s2=0;

}

if(dy > dx)

{

temp = dx;

dx = dy;

dy = temp;

ex = 1;

}

else

{

ex=0;

}

e=2\*dy-dx; //decision variable

i=1;

do

{

dash=0;

if(dash<5)

{

while(e>=0)

{

if(ex==1)

{

x = x + s1;

}

else

{

y = y + s2;

}

e = e + 2\*dy - 2\*dx;

}

if(ex==1)

{

y = y + s2;

}

else

{

x = x + s1;

}

e = e + 2\*dy;

pt::setco(x,y);

pt::draw();

i = i + 1;

dash = dash +1;

}

else

{

x = x + 2\*s1;

y = y + 2\*s2;

}

}while(i<=dx);

}

void drawdo() //Dotted Line

{

float x,y,dx,dy,e,temp;

int i,s1,s2,ex, dot=0;

dx=abs(x2-xco);

dy=abs(y2-yco);

x=xco;

y=yco;

pt::setco(x,y);

pt::draw();

if(x2 > xco) //sign() function

{

s1=1;

}

if(x2 < xco)

{

s1=-1;

}

if(x2 == xco)

{

s1=0;

}

if(y2 > yco)

{

s2=1;

}

if(y2 < yco)

{

s2=-1;

}

if(y2 == yco)

{

s2=0;

}

if(dy > dx)

{

temp = dx;

dx = dy;

dy = temp;

ex = 1;

}

else

{

ex=0;

}

e=2\*dy-dx; //decision variable

i=1;

do

{

dot=0;

while(dot<5)

{

while(e>=0)

{

if(ex==1)

{

x = x + s1;

}

else

{

y = y + s2;

}

e = e + 2\*dy - 2\*dx;

}

if(ex==1)

{

y = y + s2;

}

else

{

x = x + s1;

}

e = e + 2\*dy;

pt::setco(x,y);

pt::draw();

i = i + 1;

dot = dot +1;

}

}while(i<=dx);

}

void drawth(int x1,int y1, int x2, int y2,int colour ) //Thick Line

{

float x,y,dx,dy,e,temp;

int i,s1,s2,ex;

dx=abs(x2-xco);

dy=abs(y2-yco);

x=xco;

y=yco;

pt::setco(x,y);

pt::draw();

if(x2 > xco) //sign() function

{

s1=1;

}

if(x2 < xco)

{

s1=-1;

}

if(x2 == xco)

{

s1=0;

}

if(y2 > yco)

{

s2=1;

}

if(y2 < yco)

{

s2=-1;

}

if(y2 == yco)

{

s2=0;

}

if(dy > dx)

{

temp = dx;

dx = dy;

dy = temp;

ex = 1;

}

else

{

ex=0;

}

e=2\*dy-dx; //decision variable

i=1;

do

{

while(e>=0)

{

if(ex==1)

{

x = x + s1;

}

else

{

y = y + s2;

}

e = e + 2\*dy - 2\*dx;

}

if(ex==1)

{

y = y + s2;

}

else

{

x = x + s1;

}

e = e + 2\*dy;

pt::setco(x,y);

pt::draw();

i = i + 1;

}while(i<=dx); }

};

int main()

{

int gd=DETECT,gm=VGAMAX;

int i, ch,x1,y1,x2,y2, dx,dy,xmax,ymax,xmid,ymid,wx,wy,th;

char a;

initgraph(&gd,&gm,NULL);

dline ls;

xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

line(xmid,0,xmid,ymax); //Y co-ordinate

line(0,ymid,xmax,ymid); //X co-ordinate

do

{ xmax = getmaxx();

ymax = getmaxy();

xmid = xmax /2;

ymid = ymax /2;

cout<<"\nEnter Line Styles";

cout<<"\n1.SIMPLE..";

cout<<"\n2.DASH..";

cout<<"\n3.DOTTED..";

cout<<"\n4.THICK..";

cout<<"\n5.EXIT..";

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawsi();

break;

case 2:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawda();

break;

case 3:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

ls.setline(x1+xmid,ymid-y1,x2+xmid,ymid-y2);

ls.drawdo();

break;

case 4:

cout<<"\n Enter x1: "; cin>>x1;

cout<<"\n Enter y1: "; cin>>y1;

cout<<"\n Enter x2: "; cin>>x2;

cout<<"\n Enter y2: "; cin>>y2;

cout<<"Enter Thickness: ";

cin>>th;

ls.drawth(x1+xmid,ymid-y1,x2+xmid,ymid-y2,15);

if((y2-y1)/(x2-x1) <1)

{

wy=(th-1)\*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2\*abs(x2-x1));

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

{

ls.drawth(x1+xmid,ymid-y1-i,x2+xmid,ymid-y2-i,15);

ls.drawth(x1+xmid,ymid-y1+i,x2+xmid,ymid-y2+i,15);

}

}

else

{

wx=(th-1)\*sqrt(pow((x2-x1),2)+pow((y2-y1),2))/(2\*abs(y2-y1));

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

{

ls.drawth(x1+xmid-i,ymid-y1,x2+xmid-i,ymid-y2,15);

ls.drawth(x1+xmid+i,ymid-y1,x2+xmid+i,ymid-y2,15);

}

}

break;

case 5:

exit;

break;

}

cout<<"\nDO U Want To Continue y OR n: ";

cin>>a;

}while(a!='n');

delay(3000);

getch();

closegraph();

return 0;

}