**Assignment no: 05**

**//Aim:Write a program in C/C++ to draw a circle of desired radius.**

#include<graphics.h>

#include<iostream>

#include<stdlib.h>

#include<math.h>

using namespace std;

class dcircle

{

private: int x0,y0;

public:

dcircle()

{

x0=0;y0=0;

}

void setoff(int xx,int yy)

{

x0=xx;

y0=yy;

}

void drawdc(float x, float y, int r) //DDA Circle

{

float x1,y1,x2,y2,startx,starty,ep;

int i,val;

x1=r\*cos(0); //Initialize starting point

y1=r\*sin(0);

startx = x1;

starty = y1;

i=0;

do

{

val = pow(2,i);

i++;

}while(val<r);

ep = 1/pow(2,i-1); //calculation of epsilon

do

{

x2 = x1 + y1\*ep;

y2 = y1 - x2\*ep;

putpixel(x0+x+x2, y0-(y+y2),15);

x1 = x2;

y1 = y2;

}while((y1 - starty) < ep || (startx - x1) > ep);

}

void drawbc(int x1, int y1, int r) //Bresenham's Circle

{

int i, x, y;

float d;

x=0, y=r;

d = 3 - 2\*r; //decision variable

do

{

putpixel(x1+x0+x, y0-y1+y,15);

putpixel(x1+x0+y, y0-y1+x,15);

putpixel(x1+x0+y, y0-y1-x,15);

putpixel(x1+x0+x, y0-y1-y,15);

putpixel(x1+x0-x, y0-y1-y,15);

putpixel(x1+x0-y, y0-y1-x,15);

putpixel(x1+x0-y, y0-y1+x,15);

putpixel(x1+x0-x, y0-y1+y,15);

if(d<=0)

{

x = x + 1;

d = d + (4\*x) + 6;

}

else

{

x = x + 1;

y = y - 1;

d = d + (4\*x-4\*y) + 10;

}

}while(x<=y);

}

void drawmc(float x1, float y1, int r) // Mid point Circle

{

int i, x, y;

float d;

x=0, y=r;

d = 1.25 - r; //decision variable

do

{

putpixel(x1+x0+x, y0-y1+y,15);

putpixel(x1+x0+y, y0-y1+x,15);

putpixel(x1+x0+y, y0-y1-x,15);

putpixel(x1+x0+x, y0-y1-y,15);

putpixel(x1+x0-x, y0-y1-y,15);

putpixel(x1+x0-y, y0-y1-x,15);

putpixel(x1+x0-y, y0-y1+x,15);

putpixel(x1+x0-x, y0-y1+y,15);

if(d<0)

{

x = x + 1;

d = d + (2\*x) + 3;

}

else

{

x = x + 1;

y = y - 1;

d = d + (2\*x-2\*y) + 5;

}

}while(x<=y);

}

};

int main()

{

int gd=DETECT,gm=VGAMAX;

int i, x, y, r,ch, xmax,ymax,xmid,ymid;

float a,b;

char ans;

initgraph(&gd,& gm, NULL);

dcircle c1;

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

{

cout<<"\nEnter Cricle Drwaing algorithm";

cout<<"\n1.DDA..";

cout<<"\n2.BRESENHAM'S..";

cout<<"\n3.MID POINT..";

cout<<"\n4.EXIT..";

cout<<"\nEnter your choice: ";

cin>>ch;

switch(ch)

{

case 1:

{

cout<<"\n Enter x: "; cin>>a;

cout<<"\n Enter y: "; cin>>b;

cout<<"\n Enter radius: "; cin>>r;

c1.setoff(xmid, ymid);

setcolor(15);

c1.drawdc(a,b,r);

break;

}

case 2:

{

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

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

cout<<"\n Enter radius: "; cin>>r;

c1.setoff(xmid, ymid);

setcolor(15);

c1.drawbc(x,y,r);

break;

}

case 3:

{

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

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

cout<<"\n Enter radius: "; cin>>r;

c1.setoff(xmid, ymid);

setcolor(15);

c1.drawmc(x,y,r);

break;

}

case 4:

exit;

break;

}

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

cin>>ans;

}while(ans!='n');

delay(3000);

getch();

closegraph();

return 0;

}

/\*

gescoe@gescoe-VirtualBox:~/Downloads/cgprogram012$ g++ 5.cpp -o 5 -lgraph

gescoe@gescoe-VirtualBox:~/Downloads/cgprogram012$ ./5

Enter Cricle Drwaing algorithm

1.DDA..

2.BRESENHAM'S..

3.MID POINT..

4.EXIT..

Enter your choice: [xcb] Unknown sequence number while processing queue

[xcb] Most likely this is a multi-threaded client and XInitThreads has not been called

[xcb] Aborting, sorry about that.

5: ../../src/xcb\_io.c:274: poll\_for\_event: Assertion `!xcb\_xlib\_threads\_sequence\_lost' failed.

1

Enter x: 100

Enter y: 100

Enter radius: 50

DO U Want To Continue y OR n: y

Enter Cricle Drwaing algorithm

1.DDA..

2.BRESENHAM'S..

3.MID POINT..

4.EXIT..

Enter your choice: 2

Enter x: -100

Enter y: -100

Enter radius: 60

DO U Want To Continue y OR n: y

Enter Cricle Drwaing algorithm

1.DDA..

2.BRESENHAM'S..

3.MID POINT..

4.EXIT..

Enter your choice: 3

Enter x: -100

Enter y: 100

Enter radius: 80

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