// Name – Dahiwal Satyam Santoshkumar

// Roll Number – 207019

#include<iostream>

using namespace std;

#include<math.h>

#include<GL/glut.h>

#define PI 3.142

int x,y,r,choice;

void myInit();

void display();

void ddacircle(int x,int y,int r);

void bcircle(int x,int y,int r);

void mcircle(int x,int y,int r);

void octant(int x,int y);

void ddaline(double x1,double yy1,double x2,double y2);

int main(int argc,char \*\*argv)

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(0,0);

glutInitWindowSize(640,480);

cout<<"\n--------------------Circle--------------------\n";

cout<<"\n\tEnter Centre Co-ordinates(X,Y) : ";

cin>>x>>y;

cout<<"\n\tEnter the radius : ";

cin>>r;

do

{

cout<<"\n\tDraw Circles with ";

cout<<"\n\t1:DDA\n\t2:Bresenhams\n\t3:MidPoint >>";

cin>>choice;

}while(choice<1 || choice>3);

glutCreateWindow("Circle");

myInit();

glutDisplayFunc(display);

glutMainLoop();

}

// FUNCTIONS :

void myInit()

{

glClearColor(1,1,1,0);

glColor3f(0,0,0);

glPointSize(2.0);

gluOrtho2D(0,640,0,480);

}

void display()

{

double base=r/tan(PI/6), newr=r/sin(PI/6); //newr=sqrt(pow(r,2)+pow(base,2))

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

switch(choice)

{

case 1:

ddacircle(x,y,r);

ddacircle(x,y,newr);

break;

case 2:

bcircle(x,y,r);

bcircle(x,y,newr);

break;

case 3:

mcircle(x,y,r);

mcircle(x,y,newr);

break;

}

ddaline(x,y-r,x+base,y-r);

ddaline(x+base,y-r,x,y+newr);

ddaline(x,y+newr,x-base,y-r);

ddaline(x-base,y-r,x,y-r);

}

void ddacircle(int x,int y,int r)

{

double n=0,sum,xn=0,yn=r;

while(1)

{

if(r<=pow(2,n))

break;

n++;

}

sum=1/(pow(2,n));

while(xn<=yn)

{

octant(xn,yn);

xn=xn+sum\*yn;

yn=yn-sum\*xn;

}

}

void bcircle(int x,int y,int r)

{

double sum,xn=0,yn=r;

sum=3-2\*r;

while(xn<=yn)

{

if(sum<=0)

{

xn=xn+1;

yn=yn;

sum=sum+4\*xn+6;

}

else

{

xn=xn+1;

yn=yn-1;

sum=sum+4\*(xn-yn)+10;

}

octant(xn,yn);

}

}

void mcircle(int x,int y,int r)

{

double p,xn=0,yn=r;

p=pow(xn+1,2)+pow(yn-1/2,2)-pow(r,2);

while(xn<=yn)

{

if(p<=0)

{

xn=xn+1;

yn=yn;

p=p+(2\*xn+1);

}

else

{

xn=xn+1;

yn=yn-1;

p=p+(2\*xn-2\*yn);

}

octant(xn,yn);

}

}

void octant(int xn,int yn)

{

glBegin(GL\_POINTS);

glVertex2d(x+xn,y+yn);

glVertex2d(x-xn,y+yn);

glVertex2d(x-xn,y-yn);

glVertex2d(x+xn,y-yn);

glVertex2d(x+yn,y+xn);

glVertex2d(x+yn,y-xn);

glVertex2d(x-yn,y-xn);

glVertex2d(x-yn,y+xn);

glEnd();

glFlush();

}

void ddaline(double x1,double yy1,double x2,double y2)

{

double dx=(x2-x1),dy=(y2-yy1),steps,xInc,yInc;

if(abs(dx)>=abs(dy))

steps=abs(dx);

else

steps=abs(dy);

xInc=dx/steps;

yInc=dy/steps;

glBegin(GL\_POINTS);

for(int k=0 ; k<=steps ; k++,x1=x1+xInc,yy1=yy1+yInc)

glVertex2d(x1,yy1);

glEnd();

glFlush();

}

Output

satyam@ubuntu:~$ g++ three.cpp -lglut -lGL -lGLEW -lGLU -o three

satyam@ubuntu:~$ ./three

--------------------Circle--------------------

Enter Centre Co-ordinates(X,Y) : 250

250

Enter the radius : 70

Draw Circles with

1:DDA

2:Bresenhams

3:MidPoint >>1

satyam@ubuntu:~$ g++ three.cpp -lglut -lGL -lGLEW -lGLU -o three

satyam@ubuntu:~$ ./three

--------------------Circle--------------------

Enter Centre Co-ordinates(X,Y) : 250

250

Enter the radius : 70

Draw Circles with

1:DDA

2:Bresenhams

3:MidPoint >>2

satyam@ubuntu:~$ g++ three.cpp -lglut -lGL -lGLEW -lGLU -o three

satyam@ubuntu:~$ ./three

--------------------Circle--------------------

Enter Centre Co-ordinates(X,Y) : 250

250

Enter the radius : 70

Draw Circles with

1:DDA

2:Bresenhams

3:MidPoint >>3

