#include<stdio.h>

#include<GL/glut.h> // Header File For The GLUT Library

#include<math.h>

int ww=1200,wh=800;

int xi,yi,xf,yf,r;

float theta=2.0933;

int sign(int x)

{

if(x > 0) return 1;

if(x < 0) return -1;

return 0;

}

void putpixel(int x,int y)

{

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

}

void Bresenham\_circle(int r)

{

int x=0,y=r;

int d=3-2\*r;

do{

putpixel(x, y);

putpixel(y, x);

putpixel(-x, y);

putpixel(-x, -y);

putpixel(-y, x);

putpixel(-y, -x);

putpixel(y, -x);

putpixel(x, -y);

if(d<0)

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

else{

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

y--;

}

x++;

} while(x<=y);

}

void bresenhamAlg(int X1,int Y1, int X2,int Y2)

{

float x,y,dx,dy,length;

int i;

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\*sign(X1);

y=Y1 + 0.5\*sign(Y1);

i=1;

while(i<=length)

{

glBegin(GL\_POINTS);

glVertex2i(x,y);

glEnd();

glFlush();

x=x+dx;

y=y+dy;

i=i+1;

}

}

void triangle ()

{

int x1,x2,y1,y2;

bresenhamAlg (0, r, (-1\*sqrt(3)\*r)/2,-r/2);

bresenhamAlg (0, r, (1\*sqrt(3)\*r)/2,-r/2);

bresenhamAlg ((-1\*sqrt(3)\*r)/2,-r/2,(1\*sqrt(3)\*r)/2,-r/2);

}

void display()

{

// glClearColor(0.4,0.7,0.2,1.0);

glColor3f(0.0,1.0,1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Bresenham\_circle(r);

Bresenham\_circle(r/2);

triangle();

glFlush();

}

void myinit()

{

glViewport(0,0,ww,wh);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-1200.0,(GLdouble)ww,-800.0,(GLdouble)wh);

glMatrixMode(GL\_MODELVIEW);

}

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

{

printf("enter centre of the circle");

scanf("%d%d",&xi,&yi);

printf("\nenter radius of circle");

scanf("%d",&r);

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(ww,wh);

glutCreateWindow("Bresenham-Circle");

myinit(); /\* Initialize window. \*/

glutDisplayFunc(display);

glutMainLoop();

return 0;

}