#include<stdio.h>

#include<math.h>

#include<GL/glut.h>

using namespace std;

int xc,yc,r;

void dda\_circle(int xc,int yc,int r)

{

float xc1,xc2,yc1,yc2,eps,sx,sy;

int val,i;

xc1=r;

yc1=0;

sx=xc1;

sy=yc1;

i=0;

do{

val=pow(2,i);

i++;

}while(val<r);

eps = 1/pow(2,i-1);

do{

xc2 = xc1 + yc1\*eps;

yc2 = yc1 - eps\*xc2;

glBegin(GL\_POINTS);

glVertex2f(xc+xc2, yc-yc2);

xc1=xc2;

yc1=yc2;

glEnd();

}while((yc1-sy)<eps || (sx-xc1)>eps);

glFlush();

}

void Init()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0.0,0.0,0.0,0);

glColor3f(1.0,0.0,0.0);

gluOrtho2D(0,640,480,0);

glFlush();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

dda\_circle(xc,yc,r);

glFlush();

}

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

{

printf(" \n Enter the centre points ");

scanf("%d", &xc);

scanf("%d", &yc);

printf("\nEnter the radius");

scanf("%d", &r);

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(0,0);

glutInitWindowSize(640, 480);

glutCreateWindow("DDA\_Circle");;

glutDisplayFunc(display);

Init();

glutMainLoop();

return 0;

}

