**Rohan Nyati**

**500075940**

**R177219148**

**B-5 AI&ML SEM-5**

**Experiment-3**

**Polar Equations Circle**

#include<windows.h>

#include<GL\glew.h>

#include<GL\glut.h>

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

int xc, yc, r;

void putpixel(int x, int y)

{

glPointSize(5.0);

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_POINTS);

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

glEnd();

}

void display()

{

float x, y;

x = 0, y = r;

float theta = 0;

float inc = (float)1 / r;

glColor3f(1.0, 0.0, 0.0); //Quadrant Plot Graph

glBegin(GL\_LINES);

glVertex2i(-50, 0);

glVertex2i(50, 0);

glVertex2i(0, -50);

glVertex2i(0, 50);

glEnd();

float end = 3.14 / 4;

float C = cos(inc);

float S = sin(inc);

while (theta <= end)

{

float xtemp = x;

x = x \* C - y \* S;

y = y \* C + xtemp \* S;

putpixel(x, y);

theta = theta + inc;

}

glFlush();

}

void init()

{

glClearColor(0.7, 0.7, 0.7, 0.7);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-50, 50, -50, 50);

}

int main(int argc, char\* argv[]) {

printf("Enter the coordinates of the circle's centre:");

scanf("%d %d",&xc,&yc);

printf("Enter the radius of the circle:");

scanf("%d",&r);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutInitWindowSize(500, 500);

glutInitWindowPosition(200, 100);

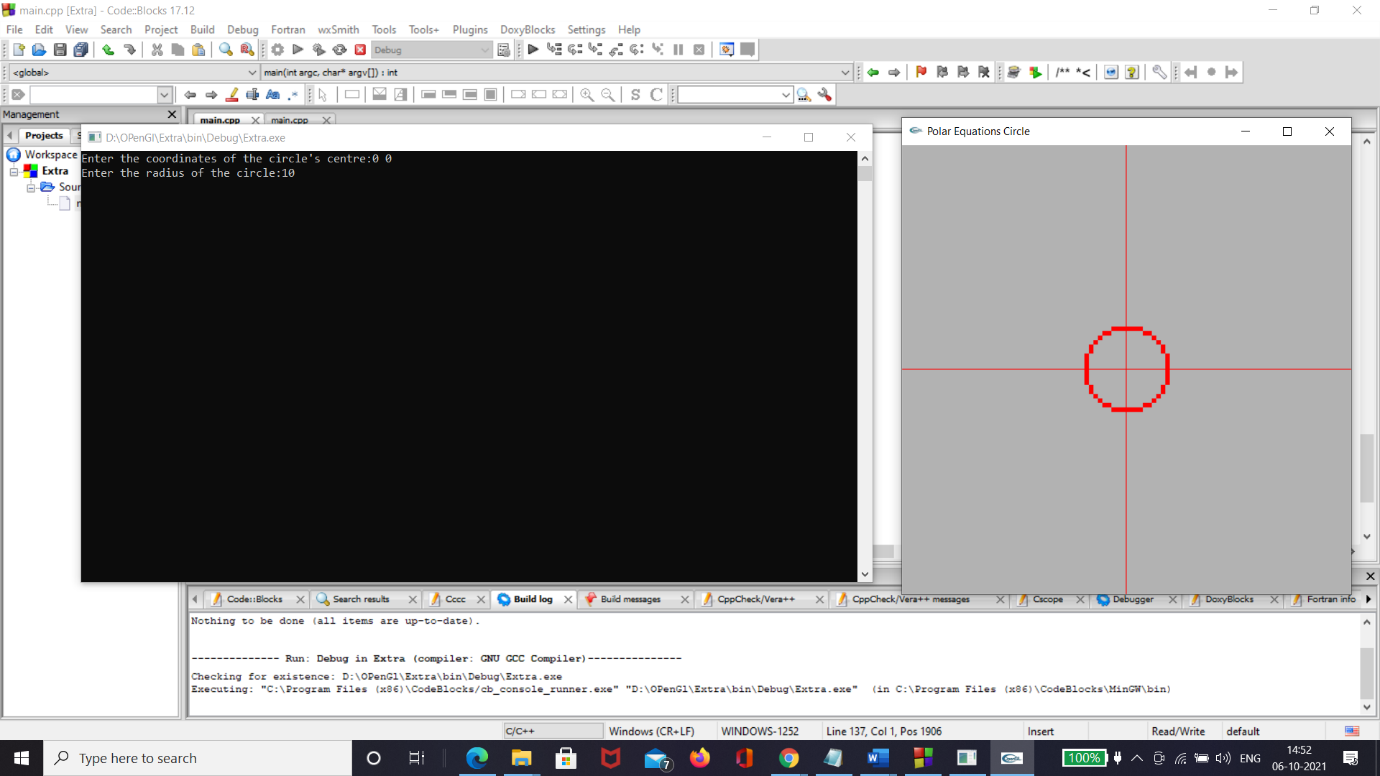
glutCreateWindow("Polar Equations Circle");

init();

glutDisplayFunc(display);

glutMainLoop();

}



**Midpoint Circle**

#include<windows.h>

#include<GL\glu.h>

#include<GL\glut.h>

#include <stdio.h>

#include <stdlib.h>

int x,y,r,xc,yc;

void display()

{

glColor3f(1.0, 0.0, 0.0); //Quadrant Plot Graph

glBegin(GL\_LINES);

glVertex2i(-50, 0);

glVertex2i(50, 0);

glVertex2i(0, -50);

glVertex2i(0, 50);

glEnd();

glPointSize(3.0);

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_POINTS);

int d[r];

d[0]=1-r;

x=0,y=0;

y=r;

if(d[0]<=0)

{

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

d[1]=d[0]+2\*x+1;

x=x+1;

}

else

{

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

d[1]=d[0]+2\*x+3-2\*y;

x=x+1;

y=y-1;

}

int i=1;

for(; i<y; i++)

{

if(d[i]<=0)

{

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

d[i+1]=d[i]+2\*x+1;

x=x+1;

}

else

{

glVertex2i(xc + x, yc + y);

glVertex2i(xc + x, yc - y);

glVertex2i(xc + y, yc + x);

glVertex2i(xc + y, yc - x);

glVertex2i(xc - x, yc - y);

glVertex2i(xc - y, yc - x);

glVertex2i(xc - x, yc + y);

glVertex2i(xc - y, yc + x);

d[i+1]=d[i]+2\*x+3-2\*y;

x=x+1;

y=y-1;

}

}

glEnd();

glFlush();

}

void init()

{

glClearColor(0.7, 0.7, 0.7, 0.7);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-50, 50, -50, 50);

}

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

{

printf("Enter the coordinates of the circle's centre:");

scanf("%d %d",&xc,&yc);

printf("Enter the value of r : ");

scanf("%d",&r);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutInitWindowSize(350, 350);

glutInitWindowPosition(100, 100);

glutCreateWindow("Midpoint Circle");

init();

glutDisplayFunc(display);

glutMainLoop();

}

