**SSN COLLEGE OF ENGINEERING, KALAVAKKAM  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
UCS1712 – GRAPHICS AND MULTIMEDIA LAB ------------------------------------------------------------------------------------------------------------**

**Lab Exercise 4 : Midpoint Circle Drawing Algorithm in C++ using OpenGL**

**Aim:**

To plot points that make up the circle with center (xc,yc) and radius r using Midpoint circle drawing algorithm

**Algorithm:**

1. Input radius r and circle center (xc, yc ). set the first point (x0 , y0 ) = (0, r ).
2. Calculate the initial value of the decision parameter as p0 = 1 – r.
3. At each xk position, starting at k = 0, perform the following test:  
   If pk <0,  
   plot(xk +1,yk )and pk+1 =pk +2xk+1 +1,

Else,

1. plot(xk +1,yk -1)and pk+1 =pk +2xk+1 +1-2yk+1,

Where 2xk+1 =2xk +2and2yk+1 =2yk –2.

**Source Code:**

#include<GLUT/glut.h>

#include<iostream>

#include<string>

#include<cmath>

using namespace std;

void drawString(float x, float y, const char \*string) {

glRasterPos2f(x, y);

for (const char \*c = string; \*c != '\0'; c++) {

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_12, \*c);

}

}

void myInit() {

glClearColor(1.0,0.6,0.5,0.0);

glPointSize(2);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-240.0,240.0,-240.0,240.0);

}

void plototherwc(int x,int y,int c1,int c2){

glBegin(GL\_POINTS);

glVertex2d(-x+c1,y+c2);

glVertex2d(-x+c1,-y+c2);

glVertex2d(x+c1,-y+c2);

glVertex2d(y+c1,x+c2);

glVertex2d(-y+c1,x+c2);

glVertex2d(-y+c1,-x+c2);

glVertex2d(y+c1,-x+c2);

glEnd();

}

void myDisplay() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3b(0, 0, 0);

glBegin(GL\_LINES);

glVertex2f(0,240);

glVertex2f(0,-240);

glVertex2f(240,0);

glVertex2f(-240,0);

glEnd();

int c1,c2,r,p,x,y;

cout<<"Enter co-ordinates of centre : ";

cin >> c1 >> c2;

cout<<"Enter radius: ";

cin>>r;

p=1-r;

x=0;

y=r;

glBegin(GL\_POINTS);

glVertex2d(x+c1,y+c2);

glVertex2d(x+c1,-y+c2);

glVertex2d(-y+c1,x+c2);

glVertex2d(y+c1,x+c2);

glEnd();

while(x<=y){//initially condition was x<y changed to x<=y

if(p>=0){

x=x+1;

y=y-1;

p=p+(2\*x)-(2\*y)+1;

glBegin(GL\_POINTS);

glVertex2d(x+c1,y+c2);

glEnd();

plototherwc(x,y,c1,c2);

}

else{

x=x+1;

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

glBegin(GL\_POINTS);

glVertex2d(x+c1,y+c2);

glEnd();

plototherwc(x,y,c1,c2);

}

}

glFlush();

}

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

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(480,480);

glutCreateWindow("Midpoint Circle Drawing Algorithm");

glutDisplayFunc(myDisplay);

myInit();

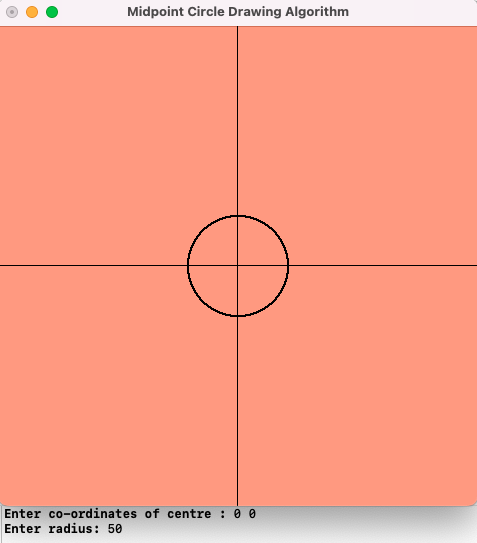
glutMainLoop();

return 1;

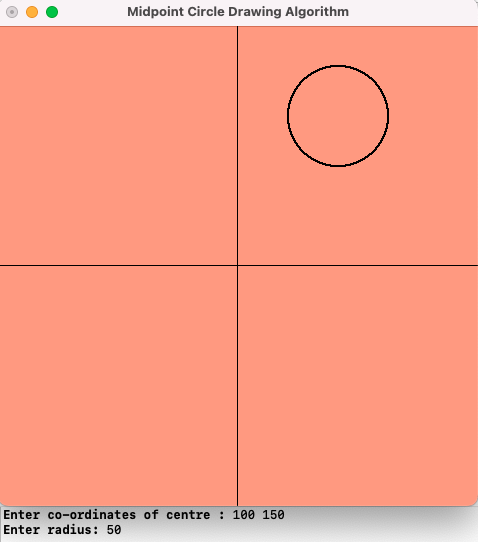
}

**Output:**

**Case (i) : Circle center as origin i.e (0,0)**

****

**Case (ii) : any arbitrary point as a center**

****

**Learning Outcome:**

Learnt how to perform Midpoint circle drawing algorithm in C++ using OpenGL