**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**EX NO: 4 – Midpoint Circle Drawing Algorithm**

**Name: Nachammai Devi Pooja S**

**RollNo:185001096**

**Date:12/08/2020**

**Aim:**

To Write a C++ program using OpenGL to implement Midpoint Circle drawing algorithm with radius and a centre given as user input.

**Algorithm:**

Step 1: Read three values

* The xc,yc values for the midpoint coordinates in the window
* The r value is the radius of the circle

Step 2: Set x=0,y=r

Step 3: Calculate decision parameter p using formula 5/4 -r

Step 4: While(y>x)

Step 5: If (p<0)

* Increment x
* Increment p by 2\*x+1

Step 6: Else

* Decrement y
* Increment x
* Increment p by 2\*(x-y)+1

Step 7:

* Plot(x+xc,y+yc)
* Plot(x+xc,-y+yc)
* Plot(-x+xc,y+yc)
* Plot(-x+xc,-y+yc)

Similarly plot the lower half of the circle

**Code:**

#include <stdio.h>

#include <iostream>

#include<gl/glut.h>

using namespace std;

int xc, yc, r;

void plot(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(x + xc, y + yc);

glEnd();

}

void myInit(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0f, 0.0f, 2.0f);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 500.0, 0.0, 500.0);

}

void midPointCircleAlgo(int xc, int yc, int r)

{

glBegin(GL\_POINTS);

//glPointSize(6.0);

int x = 0;

int y = r;

float p = 5 / 4 - r;

glVertex2i(x + xc, y + yc);

while (y > x)

{

if (p < 0)

{

x++;

p += 2 \* x + 1;

}

else

{

y--;

x++;

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

}

glVertex2i(x + xc, y + yc);

glVertex2i(x + xc, -y + yc);

glVertex2i(-x + xc, y + yc);

glVertex2i(-x + xc, -y + yc);

glVertex2i(y + xc, x + yc);

glVertex2i(-y + xc, x + yc);

glVertex2i(y + xc, -x + yc);

glVertex2i(-y + xc, -x + yc);

}

glEnd();

}

void myDisplay(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

//4a

glColor3f(1.7, 0.4, 1.1);

glPointSize(4.0);

midPointCircleAlgo(240, 250, 150);

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 150);

glutCreateWindow("Midpoint Circle Drawing!");

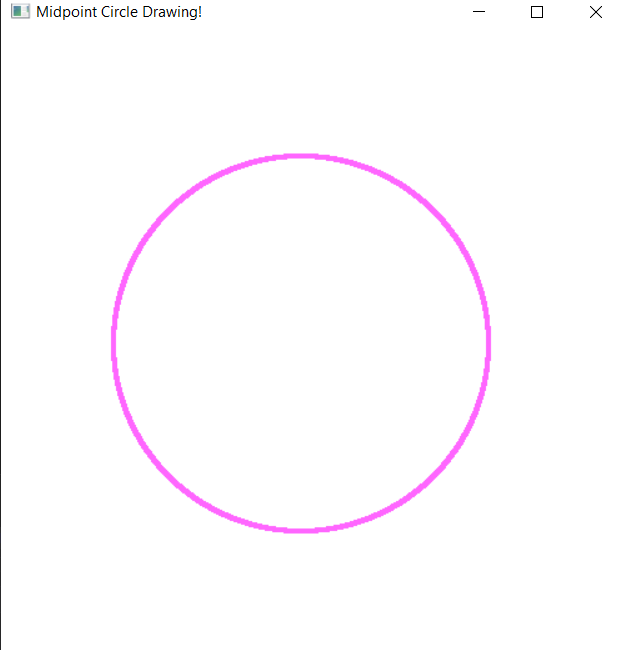
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

}

**OUTPUT:**

****

**4b) Aim:**

To Write a C++ program using OPENGL to replicate any circular object with the help of Midpoint Circle algorithm.

**Code:**

#include <stdio.h>

#include <iostream>

#include<gl/glut.h>

using namespace std;

int xc, yc, r;

void plot(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(x + xc, y + yc);

glEnd();

}

void myInit(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glColor3f(0.0f, 0.0f, 2.0f);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 500.0, 0.0, 500.0);

}

void midPointCircleAlgo(int xc, int yc, int r)

{

glBegin(GL\_POINTS);

//glPointSize(6.0);

int x = 0;

int y = r;

float p = 5 / 4 - r;

glVertex2i(x + xc, y + yc);

while (y > x)

{

if (p < 0)

{

x++;

p += 2 \* x + 1;

}

else

{

y--;

x++;

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

}

glVertex2i(x + xc, y + yc);

glVertex2i(x + xc, -y + yc);

glVertex2i(-x + xc, y + yc);

glVertex2i(-x + xc, -y + yc);

glVertex2i(y + xc, x + yc);

glVertex2i(-y + xc, x + yc);

glVertex2i(y + xc, -x + yc);

glVertex2i(-y + xc, -x + yc);

}

glEnd();

}

void myDisplay(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

//4b

glPointSize(5.0);

midPointCircleAlgo(230, 230, 15);

midPointCircleAlgo(190, 200, 15);

midPointCircleAlgo(190, 280, 15);

midPointCircleAlgo(280, 190, 15);

midPointCircleAlgo(280, 240, 15);

midPointCircleAlgo(299, 270, 15);

midPointCircleAlgo(260, 260, 15);

glColor3f(0.7, 0.4, 0.1);

glPointSize(8.0);

midPointCircleAlgo(240, 250, 100);

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(100, 150);

glutCreateWindow("Midpoint Circle Drawing!");

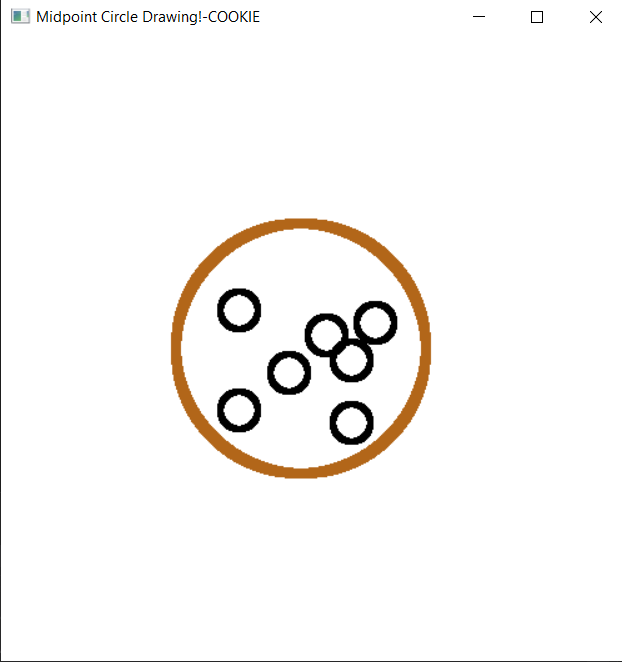
glutDisplayFunc(myDisplay);

myInit();

glutMainLoop();

}

**OUTPUT:**

****

**Result:**

The Midpoint Circle drawing algorithm is compiled and got the output successfully.