PORT CITY INTERNATIONAL UNIVERSITY

Report 02

Course Title : Computer Graphics Sessional

Course Code : CSE 422

Submitted to	Submitted by
Mumtahina Ahmed Mumu Lecturer Department of Computer Science & Engineering Port City International University	Irfan Uz Zaman Hridoy **ID** : CSE 01806750 **Batch** : 18-C

Date of Submission : 05-10-2021

Experiment No : 02

Experiment Name : Programs on scan conversion of a circle and ellipse using Midpoint algorithm.

Mid-Point Circle Drawing Algorithm :

(1) Aim : The mid-point circle drawing algorithm is an algorithm used to determine the points needed for rasterizing a circle. We use the mid-point algorithm to calculate all the perimeter points of the circle in the first octant and then print them along with their mirror points in the other octants.

(2) Necessary Apparatus :

1. Hardware : Mouse, Monitor/Screen, RAM (Random Access Memory), CPU (Central Processing Unit), Hard Disk Drive (HDD), CD Rom.

2. Software: Codeblocks.

3. Compiler : Mingw

4. Software Interface : OpenGL (Open Graphics Library).

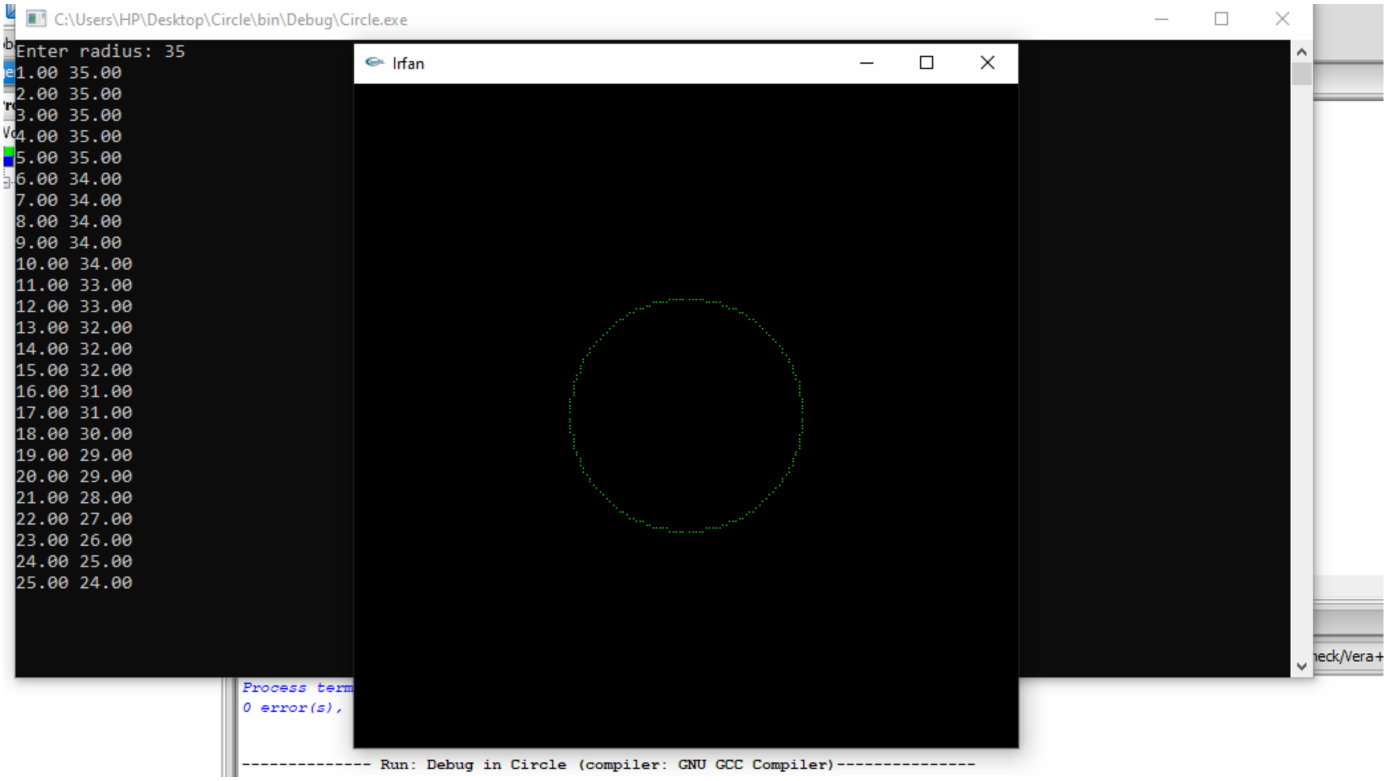
5 . The OpenGL Utility Toolkit : Glut.

(3) Implementation :

```
#include <stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

float x=0,y,x2,y2,m,i,j,p;

int dx=0,dy=0,r;

void display(void)

{

glClear (GL_COLOR_BUFFER_BIT);

glEnd();

    glColor3f (0.0, 1.0, 0.0);

    glBegin(GL_POINTS);

    p=1-r;

    while((x<=y)){

        if(p<0){

          x=x+1;

          y=y;

            printf("%0.2f %0.2f\n",x,y);

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

            }

            else{

            x=x+1;

            y=y-1;

            printf("%0.2f %0.2f\n",x,y);

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

            }

            glVertex3f (((x/100)), ((y/100)), 0.0);

            glVertex3f (((y/100)), ((x/100)), 0.0);
        glVertex3f ((-(x/100)), (-(y/100)), 0.0);
        glVertex3f ((-(x/100)), ((y/100)), 0.0);
        glVertex3f (((x/100)), (-(y/100)), 0.0);
        glVertex3f (((y/100)), (-(x/100)), 0.0);
        glVertex3f ((-(y/100)), (-(x/100)), 0.0);
        glVertex3f ((-(y/100)), ((x/100)), 0.0);
      }
glEnd();
glFlush ();
}
void init (void)
{
glClearColor (0.0, 0.0, 0.0, 0.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
}

int main(int argc, char** argv)
{
   printf("Enter radius: ");
   scanf("%d",&r);
   y=r;
glutInit(&argc, argv);
glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize (500, 500);
glutInitWindowPosition (100, 100);
glutCreateWindow ("Irfan");
```

```
init ();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}
```

(4) Output :



```
Enter radius: 35
1.00 35.00
2.00 35.00
3.00 35.00
4.00 35.00
5.00 35.00
6.00 34.00
7.00 34.00
8.00 34.00
9.00 34.00
10.00 34.00
11.00 33.00
12.00 33.00
13.00 32.00
14.00 32.00
15.00 32.00
16.00 31.00
17.00 31.00
18.00 30.00
19.00 29.00
20.00 29.00
21.00 28.00
22.00 27.00
23.00 26.00
24.00 25.00
25.00 24.00
```

```
Process term
0 error(s),

-------------- Run: Debug in Circle (compiler: GNU GCC Compiler)---------------
```

Mid-Point Ellipse Drawing Algorithm :

(1) Aim : Mid-point Ellipse algorithm is used to draw an ellipse in computer graphics. Midpoint ellipse algorithm finds points of an ellipse on the first quadrant by dividing the quadrant into two regions.

(2) Necessary Apparatus :

1. Hardware : Mouse, Monitor/Screen, RAM (Random Access Memory), CPU (Central Processing Unit), Hard Disk Drive (HDD), CD Rom.

2. Software: Codeblocks.

3. Compiler : Mingw

4. Software Interface : OpenGL (Open Graphics Library).

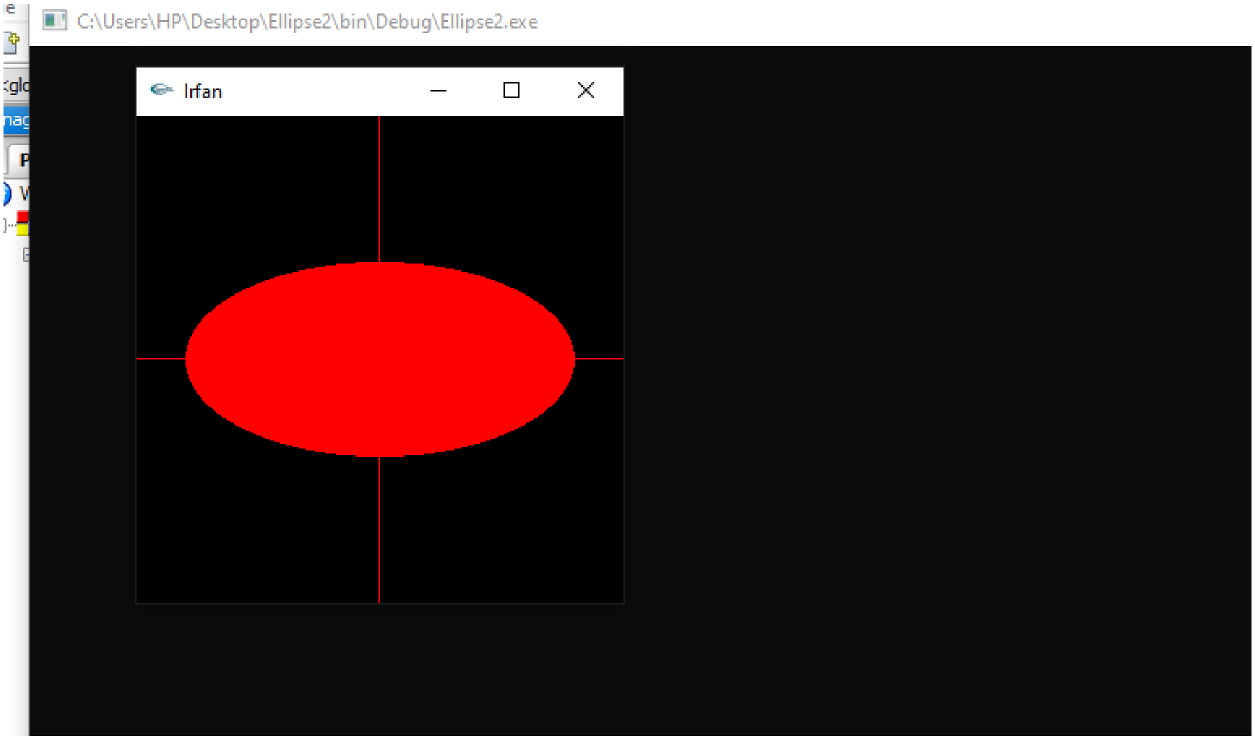
5 . The OpenGL Utility Toolkit : Glut.

(3) Implementation :

```
#include <windows.h>

#include <GL/glut.h>

#include <math.h>


#define PI 3.14


void display_ellipse() {


        glClear(GL_COLOR_BUFFER_BIT);

        glBegin(GL_POINTS);


        glBegin(GL_TRIANGLE_FAN);


        glVertex2f(0.0, 0.0);


      for (int i = 0; i < 370; ++i) {

                glVertex2f(2.0f*cos(PI*i / 180), sin(PI*i / 180));

        }

        glEnd();


        glColor3f(1.0, 0.0, 0.0);

        glBegin(GL_LINES);


        glVertex3f(0.0, -15, 0.0);


        glVertex3f(0.0, 15, 0.0);
```

```
	glVertex3f(-15, 0.0, 0.0);


	glVertex3f(15, 0.0, 0.0);

	glEnd();


	glFlush();


}

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

	glutInit(&argc, argv);

	glutInitDisplayMode(GLUT_SINGLE|GLUT_RGBA);

	glutInitWindowSize(300, 300);

	glutInitWindowPosition(100, 100);

	glutCreateWindow("Irfan");

	glutDisplayFunc(display_ellipse);


	glClearColor(0.0, 0.0, 0.0, 0.0);

	glColor3f(1.0, 0.0, 0.0);

	glMatrixMode(GL_PROJECTION);

	glLoadIdentity();

	gluOrtho2D(-2.5, 2.5, -2.5, 2.5);

	glutMainLoop();

	return 0;

}
```

(4)Output :



C:\Users\HP\Desktop\Ellipse2\bin\Debug\Ellipse2.exe
Irfan