

COMPUTER GRAPHICS

THIRD CLASS

University of Diyala/ College of education for pure
science / Computer science department

Dr. Adil I. KHALIL
10/12/2017

2. DRAWING ELEMENTARY FIGURES (PART 3-3)

PLOTTING CIRCLE

1- Circle equation method

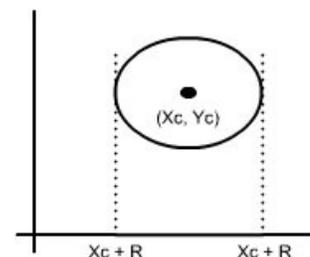
In the circle equation method, a circle is drawn using the equation:

$$(x-x_c)^2 + (y-y_c)^2 = r^2 \quad \dots\dots\dots (1)$$

In this equation, (x_c, y_c) is the center of the circle and r is the radius of the circle. To draw a circle using the equation method:

1. Enter the value of the center of the circle, (x_c, y_c) and its radius r .
2. From equation (1), derive a relationship for y using the equation

$$\text{Thus, } y = y_c \pm \sqrt{r^2 - (x - x_c)^2}$$



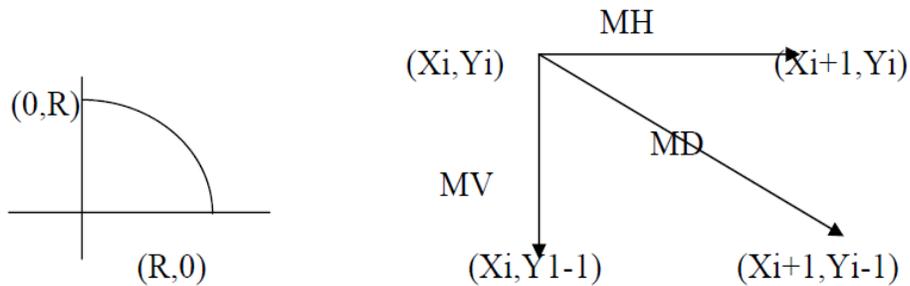
3. Iterate the value of x from $(x - r)$ to $(x + r)$, as shown in the figure above. Then obtain the corresponding values of y from equation given in step 2.
4. Plot the values of (x, y) as pixels on the circumference of the circle using the function `putpixel()`.

2- Bresenham's circle algorithm

To begin with circle drawing, note that only one octant of the circle need to be generated. The other parts can be obtained by successive reflections. To drive Bresenham circle generation algorithm , consider the first quadrant of the origin centered circle. Notice that, if the algorithm begins at $X=0$ and $Y=R$ (R is the radius) , then for clockwise generation of the circle Y is a monotonically decreasing function of X in the first quadrant. Similarly if the algorithm begins at $Y=0$ and $X=R$ then for counterclockwise generation of the circle, X is a decreasing function of Y . For clockwise generation of the circle there are only three possible selections for the next pixel which best represent the circle:

- 1- Horizontal to the right
- 2- Diagonally downward to the right
- 3- Vertically downward

These are labeled MH , MD , MV



The algorithm chose the pixel (the movement) which minimize the square of the distance between one of these pixels and the true circle

The distance in the three cases are measured by :

$$\text{Case 1 : } MH = | (Xi+1)^2 + (Yi)^2 - R^2 |$$

$$\text{Case 2 : } MD = | (Xi+1)^2 + (Yi-1)^2 - R^2 |$$

$$\text{Case 3 : } MV = | (Xi)^2 + (Yi-1)^2 - R^2 |$$

The difference between the square of the distance from the center of the circle to the diagonal pixel at (Xi+1 , Yi-1) and the distance to a point on the circle R^2 is

$$Di = (Xi+1)^2 + (Yi-1)^2 - R^2$$

1- If $Di < 0$ then the diagonal point (Xi+1, Yi-1) is *inside* the actual circle i.e. we use case 1 or case 2.

It is clear that either the pixel at (Xi+1, Yi) == MH or that the pixel at (Xi+1, Yi-1) == MD must be chosen.

$$\epsilon = | (Xi+1)^2 + (Yi)^2 - R^2 | - | (Xi+1)^2 + (Yi-1)^2 - R^2 |$$

if $\epsilon < 0$ then the difference from the actual circle to the diagonal pixel (MD) is greater than that to the horizontal pixel (MH).

If $\epsilon < 0$ choose MH (Xi+1, Yi)

If $\epsilon > 0$ choose MD (Xi+1, Yi-1)

The horizontal move has been selected when $\epsilon = 0$; i.e. when the distance are equal.

2- if $Di > 0$ then the diagonal point (Xi+1, Yi-1) is *outside* the actual circle i.e. we use case 2 or case 3 .

$$\beta = | (X_{i+1})^2 + (Y_{i-1})^2 - R^2 | - | (X_i)^2 + (Y_{i-1})^2 + R^2 |$$

if $\beta \leq 0$ choose MD (X_{i+1}, Y_{i-1})

if $\beta > 0$ choose MV (X_i, Y_{i-1})

3- if $D_i=0$ then we choose the pixel at (X_{i+1}, Y_{i-1}) i.e. MD

Summery

1- $D_i < 0$

$\epsilon \leq 0$, then choose pixel at (X_{i+1}, Y_i) i.e. MH

$\epsilon > 0$, then choose pixel at (X_{i+1}, Y_{i-1}) i.e. MD

2- $D_i > 0$

$\beta \leq 0$, then choose pixel at (X_{i+1}, Y_{i-1}) i.e. MD

$\beta > 0$, then choose pixel at (X_i, Y_{i-1}) i.e. MV

3- $D_i = 0$

choose pixel at (X_{i+1}, Y_{i-1}) i.e. MD

Bresenham circle algorithm (for the first quadrant)

$X_i=0$

$Y_i=R$

$D_i= 2(1-R)$

$Limit = 0$

1: Plot (X_i, Y_i)

If $Y_i \leq Limit$ then goto 4

If $D_i < 0$ then goto 2

If $D_i > 0$ then goto 3

If $D_i = 0$ then goto 20

2: $\epsilon = 2D_i + 2Y_i - 1$

If $\epsilon \leq 0$ then goto 10

If $\epsilon > 0$ then goto 20

3: $\beta = 2D_i - 2X_i - 1$

If $\beta \leq 0$ then goto 20

If $\beta > 0$ then goto 30

10: $X_i = X_i + 1$ { MH }

$D_i = D_i + 2X_i + 1$

Goto 1

20: $X_i = X_i + 1$ { MD }

$Y_i = Y_i - 1$

$D_i = D_i + 2X_i - 2Y_i + 2$

Goto 1

30: $Y_i = Y_i - 1$ { MV }

$D_i = D_i - 2Y_i + 1$

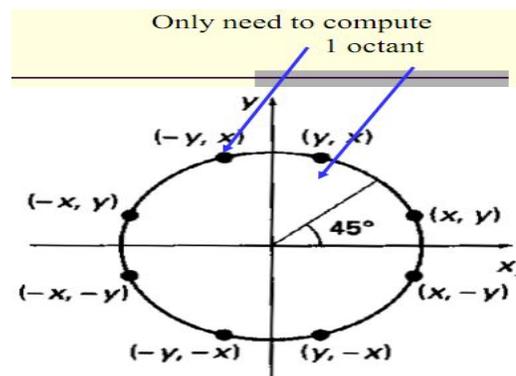
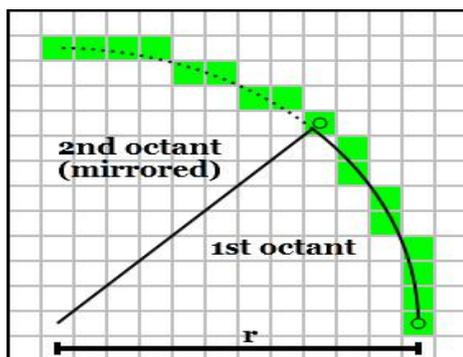
Goto 1

4: Finish

Example : Draw a circle with $R=4$

Plot (X,Y)	D_i	α	β	X_i	Y_i
	-6	-	-	0	4
(0,4)	-3	-5	-	1	4
(1,4)	-3	1	-	2	3
(2,3)	4	-1	-	3	3
(3,3)	1	-	1	3	2
(3,2)	9	-	-5	4	1
(4,1)	10	-	9	4	0
(4,0)	-3				

This algorithm draws one quadrant, and if we want to draw another quadrants we should use symmetry property on the circle that shows below .



Thus, at every stage, we can generate 8 points, as shown below

$$(x_c+x, y_c+y) , (x_c-x, y_c+y) , (x_c+x, y_c-y) , (x_c-x, y_c-y)$$

$$(x_c+y, y_c+x) , (x_c+y, y_c-x) , (x_c-y, y_c-x) , (x_c-y, y_c+x)$$

3- Parametric polar representation of circle:

The two polar coordinates r and θ can be converted to the Cartesian coordinates x and y by using the trigonometric functions **sine** and **cosine**:

$$X = X_c + R * \cos(\theta)$$

$$Y = Y_c + R * \sin(\theta)$$

Where θ is measured in radians.

θ step size is often set to be $1/r$.

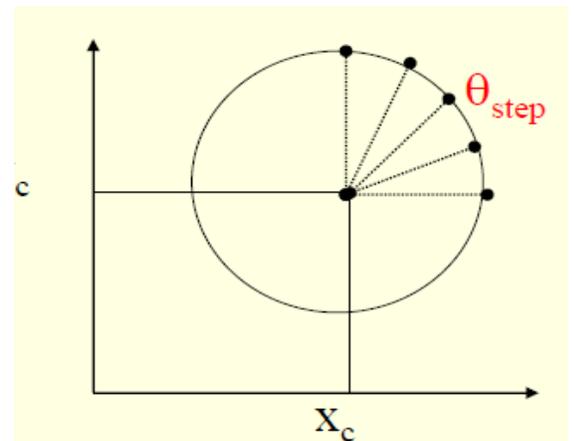
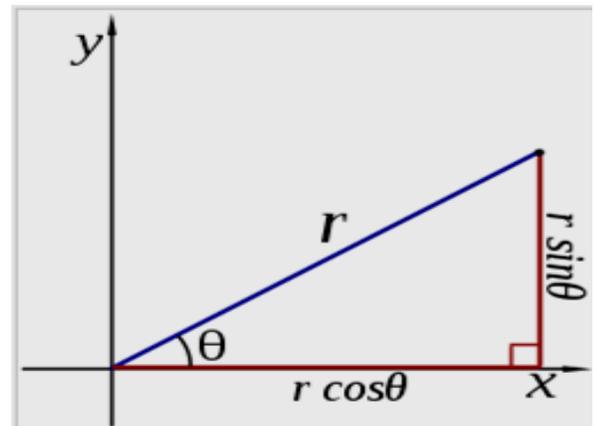
Note: When the program is designed to draw the circle, we must convert degrees to radians. This is done by the following equation.

$$\text{radians} = \text{degrees} \times \pi/180$$

Question : Write a procedure to draw circle by using polar representation.

Spiral:

What is a spiral? A spiral is a curve in the plane or in the space, which runs around a center in a special way.



You can add half circles growing step by step to get spirals.

Spiral can be plotted by gradually increasing

Radius where plotting a circle .

Q: Write procedure to draw spiral ?

Procedure spiral (xc , yc: integer ; r:real);

Var

n,x,y:real; i:integer;

Begin

 i:=-1;

 While (r < 200) do

 Begin

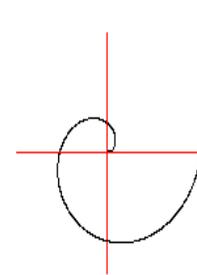
 i:=i+1;n:=i*pi/180 ; R:=R+0.02;

 x:=xc+r*cos(n); y:=yc+r*sin(n);

 putpixel(trunc(x),trunc(y),red); delay(1000);

 End;

 End;

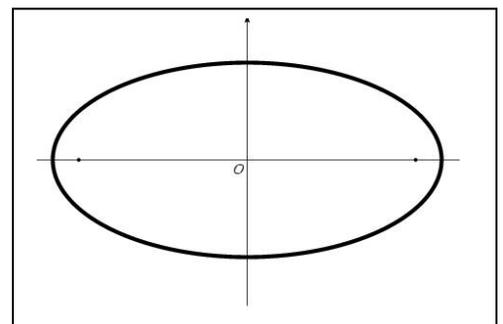


Ellipse: An ellipse is a smooth closed curve which is symmetric about its horizontal and vertical axes.

An ellipse can be represented *parametrically* by the equations $x = a \cos \theta$ and $y = b \sin \theta$, where x and y are the rectangular coordinates of any point on the ellipse, and the parameter θ is the angle at the center measured from the x -axis anticlockwise. This is one method of drawing an ellipse, called the *concentric circle method*

$$x = xc + A \cos \theta$$

$$y = yc + B \sin \theta$$



Q: write procedure to draw ellipse ?

```
procedure spiral(xc,yc,a,b:integer);
```

```
  Var
```

```
    n,x,y:real;    i:integer;
```

```
  Begin
```

```
    i:=-1;
```

```
    While(i< 360) do
```

```
      Begin
```

```
        i:=i+1;n:=i*pi/180 ;
```

```
        x:=xc+a*cos(n);          y:=yc+b*sin(n);
```

```
        putpixel(trunc( x),trunc( y),red);
```

```
        delay(1000);
```

```
      end;
```

```
end;
```