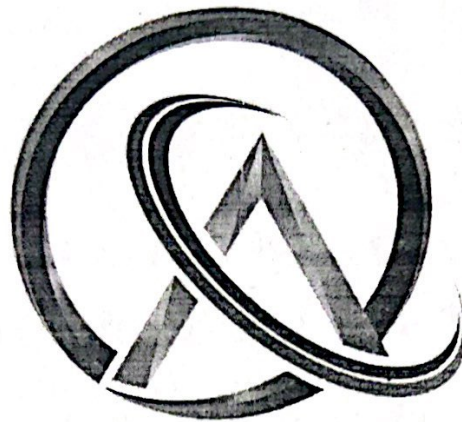


INSTITUTE OF ENGINEERING
ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT
KALANKI, KATHMANDU
(AFFILIATED TO TRIBHUVAN UNIVERSITY)



**ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT**

**ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT**
Affiliated to Tribhuvan University (T.U.)

LAB REPORT
SUBJECT : *Computer Graphics*
LAB NO : *14*

SUBMITTED BY:

NAME : *Aayush Basnet*
ROLL NO: *02*
DATE :

SUBMITTED TO:

*Department of Computer
and Electronics*

Objectives

- To learn about mid-point circle algorithm and implement it using C programming

Theory

We can define a circle function with centre $(0,0)$ and radius as

$$f_{\text{circle}}(x,y) = x^2 + y^2 - r^2$$

Any point (x,y) on boundary of circle with radius r satisfies equation $f_{\text{circle}}(x,y) = 0$

If point is inside circle, $f(x,y) < 0$ else $f(x,y) > 0$ i.e.

$$f_{\text{circle}}(x,y) \begin{cases} < 0, & \text{if } (x,y) \text{ is inside boundary} \\ = 0, & \text{if } (x,y) \text{ is on boundary} \\ > 0, & \text{if } (x,y) \text{ is outside boundary} \end{cases}$$

Assume (x_k, y_k) is plotted, we next need to determine whether pixel at position (x_{k+1}, y_k) or (x_{k+1}, y_{k+1}) is closer to the circle.

Now, decision parameter is,

$$P_k = f_{\text{circle}}(x_{k+1}, y_k - 1/2) \dots \text{--- (I)}$$

For successive decision parameter,

$$P_{k+1} = f_{\text{circle}}(x_{k+1} + 1, y_{k+1} - 1/2) \dots \text{--- (II)}$$

Subtracting (I) from (II)

$$P_{k+1} - P_k = f_{\text{circle}}(x_{k+1} + 1, y_{k+1} - 1/2) - f_{\text{circle}}(x_{k+1} + 1, y_k - 1/2)$$

$$= [(x_{k+1} + 1)^2 + (y_{k+1} - 1/2)^2 - r^2] - [(x_{k+1} + 1)^2 + (y_k - 1/2)^2 - r^2]$$

$$= x_{k+1}^2 + 2x_{k+1} + 1 + y_{k+1}^2 - y_{k+1} + 1/4 - [x_{k+1}^2 + 2x_{k+1} + 1 + y_k^2 - y_k + 1/4]$$

$$= 2x_{k+1} + 1 + y_{k+1}^2 - y_{k+1} - y_k^2 + y_k$$

$$= 2x_{k+1} + 1 + y_{k+1}^2 - y_k^2 - (y_{k+1} - y_k) \dots \text{--- (III)}$$

If $P_k < 0$

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k$$

$$\therefore P_{k+1} = P_k + 2x_{k+1} + 1$$

$$p_k \geq 0$$

$$x_{k+1} = x_k + 1$$

$$y_{k+1} = y_k - 1$$

$$\begin{aligned} p_{k+1} &= p_k + 2x_{k+1} + 1 + (y_k - 1)^2 - y_k^2 + 1 \\ &= p_k + 2x_{k+1} + 1 + y_k^2 - 2y_k + 1 - y_k^2 + 1 \\ &= p_k + 2x_{k+1} + 1 - 2(y_k - 1) \\ &= p_k + 2x_{k+1} - 2y_{k+1} + 1 \end{aligned}$$

Now, initial decision parameter is

$$p_0 = f_{\text{circle}}(x_0 + 1, y_0 - 1/2)$$

$$= f_{\text{circle}}(1, r - 1/2)$$

$$= 1^2 + (r - 1/2)^2 - r^2$$

$$= 5/4 - r$$

$$= 1 - r$$

Algorithm

1. Input centre (x_k, y_k) and radius r and set $(x_0, y_0) = (0, r)$

2. Calculate initial decision parameter as

$$p_0 = 5/4 - r \approx 1 - r$$

3. At each x_k position, starting at $k=0$, perform following test:

if $(p_k < 0)$, next point is (x_{k+1}, y_k)

$$\& p_{k+1} = p_k + 2x_{k+1} + 1$$

else, next point is (x_{k+1}, y_{k+1})

$$p_{k+1} = p_k + 2x_{k+1} - 2y_{k+1} + 1$$

where,

$$2x_{k+1} = 2x_k + 2 \& 2y_{k+1} = 2y_k - 2$$

4. Determine symmetry points in other seven octants.

5. Move each calculated pixel position (x, y) onto the circular path centered on (x_k, y_k) and plot the co-ordinate values.

$$x = x + x_c$$

$$y = y + y_c$$

6. Repeat step 3 through 5 until $x \geq y$.