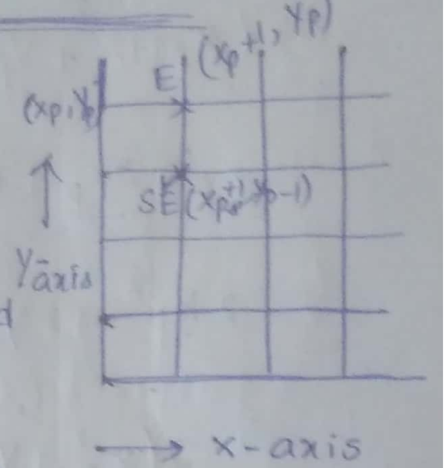
CED15102 Q

BRESSENHAMS CIRCLE DRAWING ALGORITHM

→ Suppose we are standing at the point (x_p, x_p)

→ We know that equation of circle using the cordinates (x, y) and radius r is



i.e we can obtain circle using coordinate axis :-

$$F(x, y) = x^2 + y^2 = r^2$$

$$= x^2 + y^2 - r^2 = 0$$

$$\Rightarrow F_r(x_p, y_p) = x_p^2 + y_p^2 - r^2 = 0$$

Step 2:- Now For selecting one coordinate among (x_p+1, y_p), (x_p+1, y_p-1) we need to consider the concept

$$F(x, y) = \begin{cases} < 0 \Rightarrow (x, y) \text{ is inside the circle} \\ = 0 \Rightarrow (x, y) \text{ is on the circle} \\ > 0 \Rightarrow (x, y) \text{ lies outside circle.} \end{cases}$$

$$d_E = F(x_p+1, y_p) = (x_p+1)^2 + y_p^2 - r^2 \longrightarrow ①$$

$$d_{SE} = F(x_p+1, y_p-1) = (x_p+1)^2 + (y_p-1)^2 - r^2 \longrightarrow ②$$

decision parameter

$$d = d_E + d_{SE}$$

i.e we want less distance

Step 3:- If $(d > 0)$ select SE pt // ∵ Resultant is +ve $(d_E > d_{SE})$

If $(d < 0)$ select E pt // ∵ $d_{SE} > d_E$

Step 4:- Solve 'd'

$$d = (x_p+1)^2 + y_p^2 - r^2 + (x_p+1)^2 + (y_p-1)^2 - r^2$$

$$= x_p^2 + 2x_p + 1 + y_p^2 - r^2 + x_p^2 + 2x_p + 1 + y_p^2 + 1 - 2y_p - r^2$$

$$= 2x_p^2 + 2y_p^2 + 4x_p - 2y_p - 2r^2 + 3$$

$$\Rightarrow 2(x_p^2 + y_p^2 - r^2) + 4x_p - 2y_p + 3$$

$$= 2(0) + 4x_p - 2y_p + 3$$

$$\boxed{d = 4x_p - 2y_p + 3}$$

Step 5:- considering initial point as

$(x_p, y_p) = (0, r)$

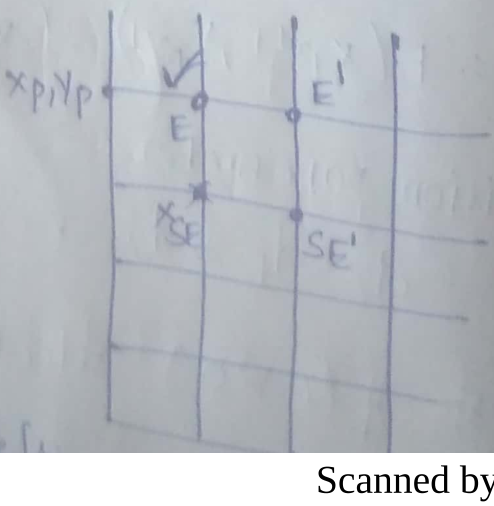
$\Rightarrow d = 0 - 2r + 3$

$d = 3 - 2r$

Step 6:-

Generalisation:-

Suppose E got selected as valid pixel



Next pixel that should be selected is one among E', SE' pixels

E' coordinates (x_p+2, y_p)

SE' cordinates (x_p+2, y_p-1)

$$d_{new} = (x_p+2)^2 + (y_p)^2 - r^2 + (x_p+2)^2 + (y_p-1)^2$$

$$= 9 + 8x_p - 2y_p$$

$$\Delta = d_{new} - d$$

$$\Delta = 6 + 4x_p$$

$$d_{new} = \Delta + d$$

$$\boxed{d_{new} = 6 + 4 * x_p + d}$$

when east gets selected

Ily For SE got selected previously

next pixel to be selected is

$$\boxed{d_{new} = d + 4 * (x_p - y_p) + 10}$$

Bressenhams ~~Line~~ Circle Drawing Algorithm

Pseudo code :-

$x=0$, $y=r$, decn parameter $= 3-2r$

putpixel (x,y)

while $(y>x)$ do

x++

If $(d<0)$

$d = d + 4*x + 6$

else

$d = d + 4(x-y) + 10$; y--;

end

put pixel (x,y)

end while

This is when centre considering $(0,0)$

when centre is (x_c, y_c) ⇒

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

Scanned by CamScanner

Bressenhams Line Drawing

→ Drawback of DDA algorithm:-

using the round function

→ Here we will avoid rounding..

Step 1:-

Calculate the slope from the start point and end point

$m < 1$	$m > 1$	$m = 1$
$x_{k+1} = x_k + 1$	$y_{k+1} = y_k + 1$	$x_{k+1} = x_k + 1$
$y_k = ?$	$x_{k+1} = ?$	$y_{k+1} = y_k + 1$

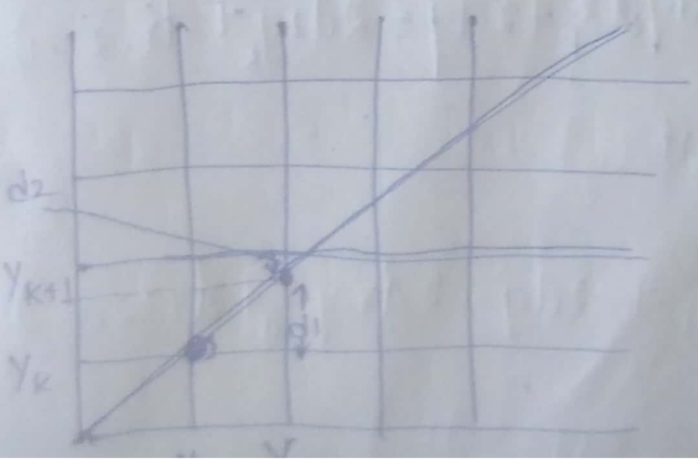
Step 2:-

Bressenhams $(m < 1)$:- m: Slope

$$y = mx + c$$

$$\boxed{y = m(x_k + 1) + c} \longrightarrow (1)$$

i.e we have to find y at x_{k+1}



at: x_{k+1} Now we need to select y_k (or) y_{k+1} = ?

Step3: To select y_k, y_{k+1} we need to calculate Decision parameter (d_p)

$$d_p = (x_{k+1} - x_k)(d_1 - d_2)$$

$$\boxed{d_p = dx\,(d_1 - d_2)}$$

From graph

$$d_1 = y - y_k$$

$$d_1 = m(x_k + 1) - y_k \quad [\because \text{From } ①]$$

$$d_2 = y_{k+1} - y$$

$$d_2 = y_k + 1 - m(x_k + 1) - c$$

$$\boxed{d_1 - d_2 = 2m(x_k + 1) - 2y_k + 2c - 1}$$

$$d_p = dx\,(d_1 - d_2) = dx\left(2 \cdot \frac{dy}{dx}(x_k + 1) - 2y_k + 2c - 1\right)$$

$$\Rightarrow (d_p)_k = P_k = 2dy\,x_k + 2dy - 2dx\,y_k + dx(2c-1)$$

d_p at next point.

$$P_{k+1} = 2dy\,x_{k+1} + 2dy - 2dx\,y_{k+1} + dx(2c-1)$$

$$(P_{k+1} - P_k) = 2dy(x_{k+1} - x_k) - 2dx(y_{k+1} - y_k)$$

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

$y_{k+1} - y_k$ we need to decide

$$= 2dy(\cancel{x_k} + 1 - \cancel{x_k}) - 2dx(y_{k+1} - y_k)$$

$$\boxed{P_{k+1} = P_k + 2dy - 2dx(y_{k+1} - y_k)} \rightarrow ②$$

We

$$P_k = d_1 = 2dyx_k + 2dy - 2dxy_k + dx(2c-1)$$

we have $y = mx + c$

$$= 2dyx_k + 2dy - 2dxy_k + dx(2(y - mx) - 1)$$

$$= 2dyx_k + 2dy - 2dxy_k + 2dxy - 2\cancel{dx}\frac{dy}{\cancel{dx}}x - dx$$

$$= 2dyx_k + 2dy - 2dxy_k + 2dxy - 2dyx - dx$$

at pt (x_k, y_k)

$$\Rightarrow 2dy\cancel{x_k} + 2dy - \cancel{2dxy_k} + \cancel{2dxy_k} - \cancel{2dyx_k} - dx$$

If $P_k \geq 0$	$p < 0$
$x_{k+1} = x_k + 1$	$x_{k+1} = x_k$
$y_{k+1} = y_k + 1$	$y_{k+1} = y_k$

∴ We donot require rounding function bcz we are selecting either (y_k, y_{k+1})

<u>Pseudocode</u> :- For $(x_1, y_1), (x_2, y_2)$ to draw line

$dx = (x_2 - x_1)$

$dy = (y_2 - y_1)$

$\boxed{d = 2 * dy - dx}$ // Decn parameter

x = x1

y = y1

putpixel (x, y)

while(x <= x2)

{

if (d<0)

{

x = x+1;

y = y

d = d + 2*(dy);

}

else {

x = x+1;

y = y+1

d = p + 2*(dy - dx); }

Scanned by CamScanner

DDA:- Pseudocode

→ coordinates (x_1, y_1) , (x_2, y_2)

for this coordinates if we want to draw a line.

slope $m = \dfrac{y_2 - y_1}{x_2 - x_1}$

$x = x_1$, $y = y_1$

while ($y <= y_2$ || $x \leq x_2$)

{

put pixel(round (x), round (y))

$x = x + 1$

$y = y + m$

}