Bresenham's integer line generation algorithm

» Bresenham's integer algorithm for the first octant.

1. The line end-pts are (x_1, y_1) $(x_2\ y_2)$ assume not equal
2. All variables are assumed integer.
3. Initialize variables

   $x = x_1$

   $y = y_1$

   $\Delta x = x_2 - x_1$

   $\Delta y = y_2 - y_1$

4. The error term $\bar{e}$ is calculated as $\bar{e} = 2(\Delta y) - \Delta x$
5. Begin the main loop
6. For $i = 1 \longrightarrow \Delta x$
7. Set pixel (x, y)
8. While $\bar{e} > 0$, $y = y+1$

   $\bar{e} = \bar{e} - 2\Delta x$

9. End while.

   $x = x+1$

   $\bar{e} = \bar{e} + 2\Delta y$.

10. Next i
11. Finish.

practice

» Bresenham's generalized integer algorithm for all quadrant's

1. The line end-pts are (x_1, y_1) & $(x_2 y_2)$ assume not equal
2. All variables are assumed integer
3. The sign function returns $-1, 0, 1$ as its argument is <0, $=0$, >0
4. Initialize variables

   $x = x_1$

   $y = y_1$

   Δx = absolute i.e. abs $(x_2 - x_1)$

   Δy = abs $(y_2 - y_1)$

$s_1 = sign\ (x_2 - x_1)$

$s_2 = sign\ (y_2 - y_1)$

5. Interchange Δx & Δy depending on the slope of line
6. If $\Delta y > \Delta x$,

   $temp = \Delta x$

   $\Delta x = \Delta y$

   $\Delta y = temp$

7. Interchange = 1
8. Else Interchange = 0
9. End if
10. initialize the error term as
11. $\bar{e} = 2\Delta y - \Delta x$
11. Main loop
12. For $i = 1 \longrightarrow \Delta x$
13. Set pixel (x, y)
14. While $\bar{e} > 0$
15. If interchange = 1

    then $x = x + s_1$

    else $y = y + s_2$

16. End if

    $\bar{e} = \bar{e} - 2 \cdot \Delta x$

17. End while
18. if Interchange = 1

    then $y = y + s_2$

    else $x = x + s_1$

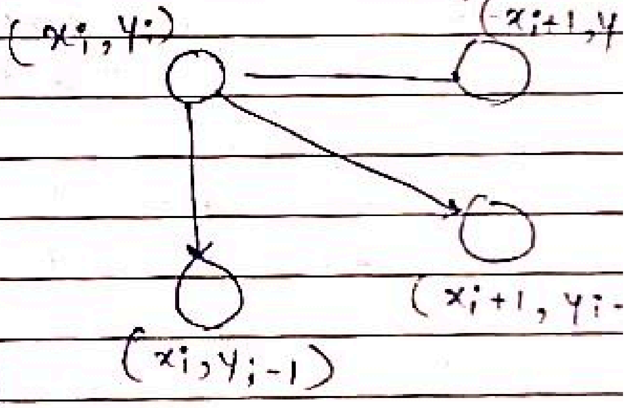
19. End if

    $\bar{e} = \bar{e} + 2\Delta y$.

20. Next i
21. Finish.

Bresenham's circle generation algorithm.

1) One of the most efficient and easiest algo. to derive the circle is bresenham's circle generation algo.
2) Here assume the clockwise generation of the circle. So, for any given point on the circle there are 3 possible selection for next pixel which best represents the circle.
3) The 3 possible selection are horizontal to the right, diagonally downward to the right and vertically downward.
4) These moments are labelled as M_H, M_D, M_V resp
5) The algo chooses the pixel which minimizes the squares of the distance b/w one of these pixels and the true circle



6) $M_H = |(x_i+1)^2 + (y_i)^2 - (R)^2|$

$M_D = |(x_i+1)^2 + (y_i-1)^2 - (R)^2|$

$M_V = |(x_i)^2 + (y_i-1)^2 - (R)^2|$

7) As std equ^n of circle with origin centre is ~~i.e~~
$R^2 = x^2 + y^2$
$\therefore$ function $f(x,y) = x^2 + y^2 - R^2$

8) The difference b/w sq. of the distance from the cent of the circle to the diagonal pixel at (x_i+1, y_i-1) and distance to a point on the circle R^2 is
$\Delta_i = (x_i+1)^2 + (y_i-1)^2 - R^2$.

diagonal pixel is inside true circle.

Case I): If $\Delta_i < 0$ then

- The diagonal pt. (x_i+1, y_i-1) is <u>inside</u> the actual circle so here it is clear that either the pixel at (x_i+1, y_i) i.e. M_H or the pixel at (x_i+1, y_i-1) i.e. M_D must be chosen.
- To decide which, first consider case-I) by examining the difference b/w sq. of the distance from actual circle to pixel at M_H and distance from actual circle to pixel at M_D

$$\delta = M_H - M_D$$

$$\delta = \left|(x_i+1)^2 + (y_i)^2 - (R)^2\right| - \left|(x_i+1)^2 + (y_i-1)^2 - R^2\right|$$

- If $\delta \le 0$, then distance from actual circle to diagonal pixel M_D is greater than that to horizantal pixel M_H and if $\delta > 0$ the cond^n is opposite.

- $\delta \le 0$, then always choose M_H at (x_i+1, y_i)
- $\delta > 0$, then always " M_D at (x_i+1, y_i-1)

- The horizontal move is selected when $\delta = 0$, in this case always select M_H.
- The δ is evaluated as $\left|(x_i+1)^2 + (y_i)^2 - (R)^2\right| \ge 0$ $\left|(x_i+1)^2 + (y_i-1)^2 - R^2\right| < 0$.

Now, δ can be evaluated as,

$$\delta = (x_i+1)^2 + (y_i)^2 - (R^2) + (x_i+1)^2 + (y_i-1)^2 - (R)^2$$

Solving equ^n by adding and subtracting $(-2y_i+1)$ gives,

$$\delta = 2\left[(x_i+1)^2 + (y_i-1)^2 - (R)^2\right] + 2y_i - 1$$

Using the def^n of ~~δ~~ Δi gives,

$$\boxed{\delta = 2(\Delta_i + y_i) - 1}$$

c) Case II) : If $\Delta_i > 0$ then

- For this $cond^n$ the diagonal pixel (x_i+1, y_i-1) is outside the actual circle.
- Here it is clear that either the pixel at (x_i+1, y_i-1) i.e. M_D or the pixel at (x_i, y_i-1) i.e. M_V must be chosen.
- So again to decide which pixel we have to calculate δ', $\delta' = M_D - M_V$

$$= \left|(x_i+1)^2 + (y_i-1)^2 - R^2\right| - \left|x_i^2 + (y_i-1)^2 - R^2\right|$$

- If $\delta' \leq 0$, then distance from actual circle to vertical pixel at (x_i, y_i-1) is larger and so the diagonal move M to the pixel at (x_i+1, y_i-1) is chosen.
- If $\delta' > 0$, then case is opposite.

$$\boxed{\begin{array}{l} \bullet\ \delta' \leq 0\ ;\ \text{choose } M_D \text{ at } (x_i+1, y_i-1) \\ \bullet\ \delta' > 0\ ,\ \text{choose } M_V \text{ at } (x_i, y_i-1) \end{array}}$$

δ' is evaluated as $|x$

Q. Apply bresenham's generalized integer line genera[illegible] algorithm for the line segment from $P_1(0,0)$ to [illegible] find out the intermediate values of pixels & show these all values in tabular form.

Ans: Here, $(x_1, y_1) = (0,0)$
$(x_2, y_2) = (-8, -4)$

$x = x_1 = 0$
$y = y_1 = 0$
$\Delta x = 8$
$\Delta y = 4$
$s_1 = -1$
$s_2 = -1$
Interchange $= 0$
$\bar{e} = 2\Delta y - \Delta x$
$\bar{e} = 0$

i	set pixel	$\bar{e}$	x	y
		0	0	0
1	(0,0)	8	-1	0
2	(-1,0)	-8 0	-2	-1
3	(-2,-1)	8	-3	-1
4	(-3,-1)	-8 0	-4	-2
5	(-4,-2)	8	-5	-2
6	(-5,-2)	-8 0	-6	-3
7	(-6,-3)	8	-7	-3
8	(-7,-3)	-8 0	-8	-4
~~9~~	~~(-8,-4)~~			