CSE 423 – Computer Graphics
Bresenham Line Drawing Algorithm

We know screen of a monitor is the collection of pixels. So, if you represent the pixels in World Coordinate System (2D approach) then it will similar as that.
We know, the line equation is

$$y = mx + c \text{..........(1)}$$

Where, m is a slope and calculated by –

$$m = \frac{\Delta y}{\Delta x}$$

If you draw a line in the origin of the 2D coordinate (the indigo line) (mid line) then the different between $y - axis$ and difference between $x - axis$ will be equal and $m = 1$.
On the other hand, $m < 1$, if $\Delta x > \Delta y$, and $m > 1$, if $\Delta x < \Delta y$.
Therefore, we are getting 3 cases for slope m.

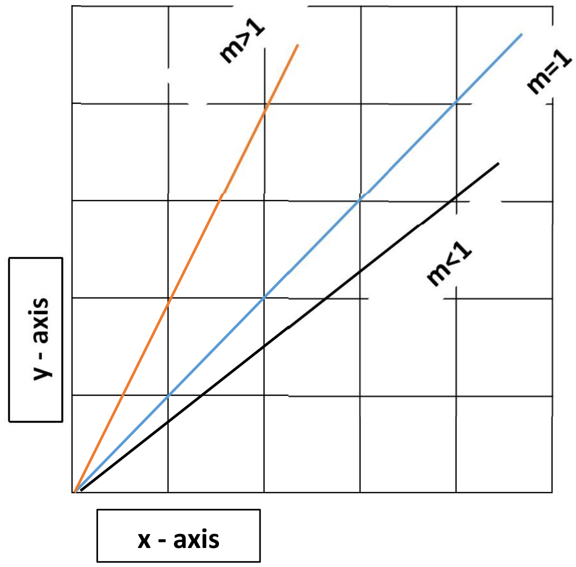

Fig 01:These are the collection of pixels

Consider the line equation (1), if we draw any line in the coordinate system is called Vector. So, if we get any pixel's position in the coordinate system such as (x,y) is called Raster. The process is to convert Vector to Raster called Rasterization. Now, we have to draw a line means we have to rasterize a line.

In 2D coordinate system, we have to find coordinates. From the equation (1) of line, if we assume the value of x then we may get the value of y, reversely, we can get the value of x if we assume the value y, which called Sampling.

However, if you have $m < 1$, which means $\Delta x > \Delta y$. In that case you have to assume (sampling) the value of x and for $\Delta x < \Delta y$ and $m > 1$, you have to sampling the value of y.

Nevertheless, for the indigo line (mid line) he value of x and y remain same.

N:B: as we have got a float value for any coordinate (x or y) for DDA algorithm and as if the value float consume so much time than a integer value in that case the Bresenham algorithm for you.

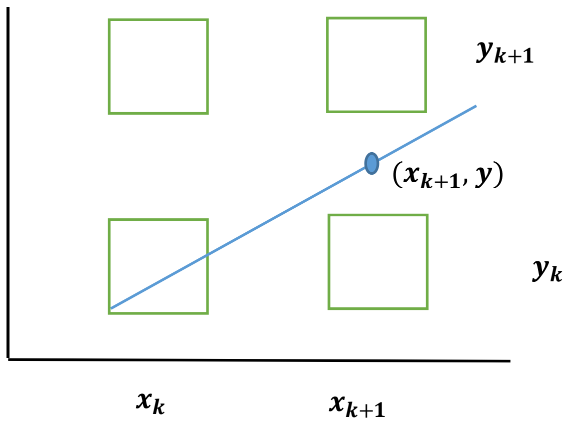

Fig 02: Consider the pixels to find the distance between two pixels.

N:B: it should be remember that, any line always goes in between to the pixels where is no any gap.

If any line go to the gap, we have to take decision which pixel we may consider. In Bresenham line drawing algorithm, we should take the nearest pixel so, how do we the nearest pixel?
The answer: we may find two distances $d_1$and d_2 between two pixels from the exact point.
$x - axis$ should be increased by always 1 $(x_k + 1)$
$d_1 = y - y_k$……..(2) and
$d_2 = y_{k+1} - y$………(3)
We should always take the nearest pixel's coordinate.
Now, if $d_1 - d_2 > 0$, or $d_1 > d_2$, so the next coordinate of $y = y_k + 1$
If $d_1 - d_2 < 0$, or $d_1 < d_2$, so the next coordinate of $y = y_k$
These conditions actually known as Decision Parameter.
Consider the equation (1) at actual point $(x_k + 1, y)$—
$y = m(x_k + 1) + c$……(4)

Consider the equation (2) –
$d_1 = y - y_k$
$=> \ d_1 = m(x_k + 1) + c - y_k$ ……… (5)
Consider the equation (3) –
$d_2 = y_{k+1} - y$
$=> d_2 = y_{k+1} - m(x_k + 1) - c$ ……(6)

Now, assume
$d_1 - d_2 = \ m(x_k + 1) + c - y_k - y_k - 1 + m(x_k + 1) + c$
$= 2m(x_k + 1) - 2y_k + 2c - 1$
$= 2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2c - 1$

$=> \Delta x(d_1 - d_2)$
$= \Delta x[2\frac{\Delta y}{\Delta x}(x_k + 1) - 2y_k + 2c - 1]$

$\Delta x(d_1 - d_2)$ is Decision Parameter and represented by P_k

$P_k = 2\Delta y(x_k + 1) - 2\Delta x y_k + 2c\Delta x - \Delta x$
$= 2x_k\Delta y + 2\Delta y - 2y_k\Delta x + 2c\Delta x - \Delta x$
$= 2x_k\Delta y - 2y_k\Delta x + \mathbf{2\Delta y + 2c\Delta x - \Delta x}$ ……………….(7)

Now,
$P_{k+1} = 2x_{k+1}\Delta y - 2y_{k+1}\Delta x + \mathbf{2\Delta y + 2c\Delta x - \Delta x}$……………………..(8)

Now, perform equation (8) – equation (7)
$P_{k+1} - P_k = 2\Delta y(x_{k+1}) - 2\Delta x(y_{k+1}) + A - 2\Delta y(x_k) - 2\Delta x y_k - A$

$P_{k+1} = P_k + 2\Delta y(x_{next} - x_k) - 2\Delta x(y_{next} - y_k)$

If $P_{k+1} - P_k < 0$, then –
$P_{k+1} = P_k + 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_k - y_k)$
$P_{k+1} = P_k + 2\Delta y$ ………..(9)

If $P_{k+1} - P_k \geq 0$, then –
$P_{k+1} = P_k + 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k)$
$P_{k+1} = P_k + 2\Delta y - 2\Delta x$ ………..(10)

Now, find the initial decision parameter P_k
$P_k = 2x_k\Delta y - 2y_k\Delta x + 2\Delta y + \boldsymbol{2c\Delta x - \Delta x}$
$= \ldots + 2\Delta x(y_k - mx_k) - \Delta x$ [$c = y_k - mx_k$]
$= \ldots + 2\Delta x y_k - 2\Delta x \frac{\Delta y}{\Delta x} x_k - \Delta x$
$= \ldots + 2\Delta x y_k - 2\Delta y x_k - \Delta x$
$=> P_k = 2\Delta y - \Delta x$

Now, the decisions are –

if $P_k < 0$ means $d_1 < d_2$
the next coordinate = $(x_k + 1, y_k)$
and the equation (9) will be followed to calculate P_{k+1}

if $P_k \geq 0$ means $d_1 > d_2$
the next coordinate = $(x_k + 1, y_k + 1)$
and the equation (10) will be followed to calculate P_{k+1}

Concept:

$\boldsymbol{m < 1}$	$\boldsymbol{m \geq 1}$
$P_k = 2\Delta y - \Delta x$	$P_k = 2\Delta x - \Delta y$
if $P_k < 0$ means $d_1 < d_2$ the next coordinate = $(x_k + 1, y_k)$ $P_{k+1} = P_k + 2\Delta y$	if $P_k < 0$ means $d_1 < d_2$ the next coordinate = $(x_k + 1, y_k)$ $P_{k+1} = P_k + 2\Delta x$
if $P_k \geq 0$ means $d_1 > d_2$ the next coordinate = $(x_k + 1, y_k + 1)$ $P_{k+1} = P_k + 2\Delta y - 2\Delta x$	if $P_k \geq 0$ means $d_1 > d_2$ the next coordinate = $(x_k + 1, y_k + 1)$ $P_{k+1} = P_k + 2\Delta x - 2\Delta y$