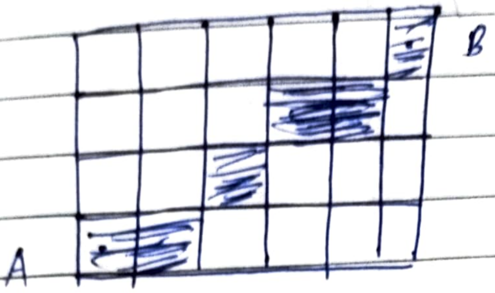
Computer Graphics

→ Our aim is to develop an algorithm to draw best fit line between two points on screen



* Academic Definition

→ Computer Graphics is a discipline that involves drawing/displaying graphical objects on the graphic devices like VDU or a computer or a workstation

* Commercial Definition

→ Computer Graphics refers to tools used to make the pictures or objects. These tools can be both Hardware & Software.

Hardware tools:-

Video Monitor, Graphics Card, Printer, Mouse, Pen, Trackball, Joy Stick

Software tools:-

PLs have a collection of graphic entities that produce the pictures themselves or there are dedicated PLs for graphics.

Graphical Objects are generated by using points, lines, curves, surfaces, solids displayed on VDU/workstation.

A point in CG is the smallest displayable object or a pixel.

* Open GL

→ It is an open source graphics library. It is a device independent application programming interface.

→ Main libraries in open GL :-

1) Basic GL
2) GLU (Utility Library)
3) GLUT (Utility Toolkit)
4) GLUI (User interface)

→ Basic GL provides functions that are a permanent part of open GL. Every function in open GL starts with GL.

→ GLU manages windows, events, full ~~screens~~ screen rendering.

→ GLUT manages high level processing required for drawing of quadratic surfaces. These involve matrix operations.

→ GLUI provides sophisticated controls to open GL.

Plotting a Line

* DDA (Digital Differential Analyzer)

→ This is a scan-conversion line algorithm where the line is sampled at unit intervals in one co-ordinate and determine the closest corresponding integer value to the line for the other co-ordinate.

$$m = \frac{\Delta y}{\Delta x} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$

If we take unit interval in x direction, $x_{k+1} - x_k = 1$

$$\therefore y_{k+1} = y_k + m$$

If we take unit interval in y direction, $y_{k+1} - y_k = 1$

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

To avoid jumpping or skipping a pixel in either x or y co-ordinate, we take unit intervals of that co-ordinate that grows quickly.

Thus if $|m| > 1$ then, we take unit intervals in y

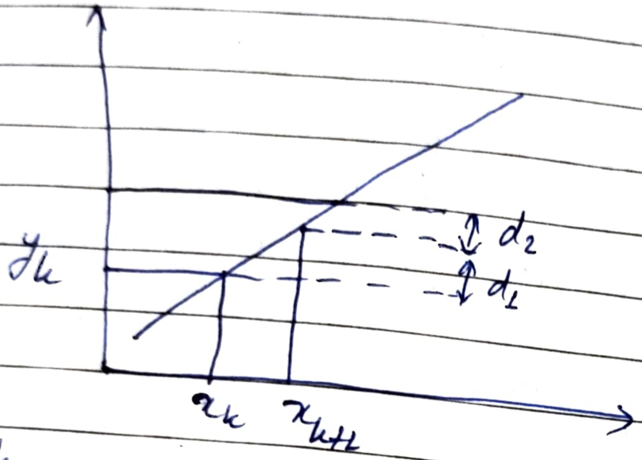
$$y_{k+1} = y_k + 1$$
$$x_{k+1} = x_k + 1/m$$

if $|m| < 1$ then, we take unit intervals in x

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

* ~~Bres~~ Bresenham's Line Algorithm

→ For a line with $|m| < 1$, assuming (x_k, y_k) is plotted correctly, we can determine either to plot (x_{k+1}, y_k) or (x_{k+1}, y_{k+1}) based on the distance of the two pixels from the actual line path.



We know,

$$y = mx + b$$

Thus, ~~at~~ at x_{k+1} $y = m x_{k+1} + b = m(x_k + 1) + b$

$$d_1 = y - y_k$$
$$= m(x_k + 1) + b - y_k$$

$$d_2 = y_{k+1} - y$$
$$= y_k + 1 - m(x_k + 1) - b$$
$$=$$

$$\therefore d_1 - d_2 = m(x_k + 1) + b - y_k - y_k - 1 + m(x_k + 1) + b$$
$$= 2(m(x_k + 1) - y_k) + 2b - 1$$

Putting $m = \Delta y / \Delta x$

$$= 2 \frac{\Delta y}{\Delta x} (x_k + 1) - 2y_k + 2b - 1$$

$\therefore$ ~~$\Delta x (d_1 - d_2) = 2 \Delta y (x_k + 1) - 2 y_k + 2b - 1$~~

$$\therefore \Delta x (d_1 - d_2) = 2 \Delta y (x_k + 1) - 2 \Delta x y_k + \Delta x (2b - 1)$$
$$= 2 \Delta y x_k - 2 \Delta x y_k + \Delta x (2b - 1) + 2 \Delta y$$
$$= 2 \Delta y \cdot x_k - 2 \Delta x \cdot y_k + c \text{ where,}$$

$$c = \Delta x (2b - 1) + 2 \Delta y$$