Name → Ameya Barapatre
Roll No - 6

DS / [illegible]

10

SARASWATI Education Society's
SARASWATI College of Engineering

PAGE NO. :
DATE : 31/07/24

Experiment No - 2

Aim → To implement Bresenhamis line algorithm for having a line segment between two given end points (x_1, y_1) & (x_2, y_2).

Resource required → Torbo C, Pointer, pointout, stationary.

Theory → Bresenhamis line ~~algo~~ algorithm is an efficient method for drawing a straight line between two points in a rader graphics content developed by Jack Bresenham in 1962, it was integer arithmatic instead of floating point arithmatic, making it both fast and suitable for use in real-time graphics system and embedded systems.

Principle → The basic principle of Bresenhamis algorithm is to determine which pixel should be plotted to best approximate the line between two given points (x_1, y_1) and (x_2, y_2). The algorithm incrementally determines the next pixel along the line by comparing the error term, ensuring that the plotted pixels are as close as possible to the theoretical line.

Bresenham's line drawing algooithm →

Step 1 → Read the end points (x_1, y_1) and (x_2, y_2)

Step 2 → $\Delta x = x_2 - x_1$ & $\Delta y = y_2 - y_1$

$$m = \Delta y / \Delta x = \frac{y_2 - y_1}{x_2 - x_1}$$

Step 3 → Find initial decision Parameter (P).

$$P = 2\Delta y - \Delta x.$$

Step 4 → $N = \max(\Delta x, \Delta y)$

Repeat step 5 onwarid to N is toue.

Step 5 → If $|m| < 1$, then, if $P < 0$ then,

$$x_{n+1} = x_n + 1$$
$$y_{n+1} = y_n$$
$$P_{n+1} = P_n + 2\Delta y$$

else

$P \geq 0$ then

$$x_{n+1} = x_n + 1$$
$$y_{n+1} = y_n + 1$$
$$P_{n+1} = P_n + 2(\Delta y - \Delta x)$$

else $m \geq 1$

if $P < 0$

$$y_{n+1} = y_n + 1$$
$$x_{n+1} = x_n$$
$$P_{n+1} = P_n + 2\Delta x$$

else $P \geq 0$

$x_{n+1} = x_n + 1$

$y_{n+1} = y_n + 1$

$P_{n+1} = P_n + 2(\Delta x - \Delta y)$

Conclusion → Bresenham's line algorithm is an efficient way to draw lines in computer graphics by leveraging integer arithmetic and simple decision - making, it minimizes computational overhead while accurately approximating a ~~subject~~ straight line between two points. This makes the algorithm suitable for real time applications in computer graphics and embedded systems.

7/8