

Algoritmo para el trazo de líneas

La ecuación de la una recta puede anunciarse de la forma:

$$y = m * x + b \quad \text{Ec. 1}$$

Con m como pendiente de la recta y b como intercepción de y . Dado que los 2 extremos de un segmento rectilíneo se especifica como (x_1, y_1) y (x_2, y_2) , podemos determinar valores de la pendiente m y de la intersección y, b , con los siguientes cálculos:

Algoritmo para el trazo de líneas

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{Ec. 2}$$

$$b = y_1 - m * x_1 \quad \text{Ec. 3}$$

Los algoritmos para desplegar líneas rectas se basan en la ecuación Ec. 1 y los cálculos que se dan en las ecuaciones Ec. 2 y Ec. 3

Algoritmo para el trazo de líneas

Para cualquier intervalo Dx de x a lo largo de una recta, podemos calcular el intervalo Dy de y correspondiente de la ecuación Ec. 2 como:

$$Dy = m * Dx \quad \text{Ec. 4}$$

Algoritmo para el trazo de líneas

ALGORITMO DDA

El analizador diferencial digital (DDA) es un algoritmo que sirve para calcular posiciones de píxeles a lo largo de una línea, mediante el uso de la ecuación Ec. 4.

$$Dy = m * Dx \quad \text{Ec. 4}$$

Esto se lleva a cabo realizando pasos unitarios con una coordenada y calculando valores correspondientes de la otra coordenada.

Algoritmo para el trazo de líneas

Primero consideremos una línea con pendiente positiva, si la pendiente es menor que o igual a 1, se toma la variación en los valores de la coordenada x como 1 y calcula cada valor sucesivo de la coordenada y como

$$Y_{i+1} = y_i + m \quad \text{Ec. 5}$$

El subíndice i toma valores enteros comenzando desde 1, para el primer punto, y aumenta de 1 en 1 hasta alcanzar el extremo final. Como m puede ser un número real cualquiera, los valores calculados de y deben redondearse al entero más próximo

Algoritmo para el trazo de líneas

Para rectas con pendientes positiva mayor que 1, se invierten los papeles de x y de y . Es decir, nos movemos en etapas de y unitarias y calculamos cada valor sucesivo de x como:

$$X_{i+1} = X_i + 1/m \quad \text{Ec.6}$$

Las Ecuaciones Ec. 5 y Ec. 6 suponen que procedemos a lo largo de la línea desde el extremo izquierdo hasta el derecho. Si estos 2 puntos extremos se invierten de manera que el punto extremo inicial esté del lado derecho, entonces tenemos $Dx = -1y$

$$Y_{i+1} = Y_i - m \quad \text{Ec. 7}$$

Algoritmo para el trazo de líneas

O bien (cuando la pendiente es mayor que 1) se tiene

$Dy = -1$ con:

$$X_{i+1} = X_i - 1/m \quad Ec\ 8$$

Algoritmo en Pseudocódigo DDA

Inicio

Leer X_1, Y_1, X_2, Y_2

$Dx = X_2 - X_1$

$Dy = Y_2 - Y_1$

Algoritmo para el trazo de líneas

Si $\text{Abs}(Dy) > \text{Abs}(Dx)$ entonces

 longitud = $\text{Abs}(Dy)$

si no

 longitud = $\text{Abs}(Dx)$

 inc_x = $Dx / \text{longitud}$ /* Redondear al entero más próximo

 inc_y = $Dy / \text{longitud}$ /* Redondear al entero más próximo

 X = X1

 Y = Y1

Algoritmo para el trazo de líneas

Para $k = 1$ hasta longitud hacer

 pintapunto(x, y)

$X = X + inc_x$

$Y = Y + inc_y$

Fin de Para

Fin del Algoritmo

/* Pintar Pixel por Pixel

```
pintapunto(Graphics g, int x, int y)
{
    g.drawLine(x,y,x,y)
}
```