

Graficas elementales 2D

Ivan Olmos Pineda

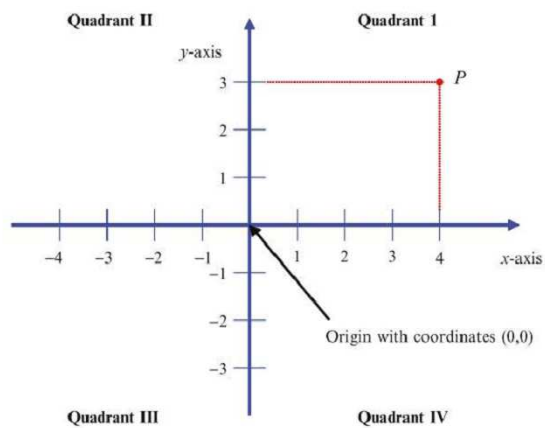
Introducción

- ▶ **Para graficar en 2D, se requieren básicamente**
 - ▶ Un sistema de coordenadas
 - ▶ Ecuaciones que definan las curvas que se desean trazar en la pantalla

- ▶ **Con respecto al sistema de coordenadas, se tienen:**
 - ▶ Sistema de coordenadas cartesianas
 - ▶ Sistema de coordenadas polares
 - ▶ Sistema de coordenadas homogéneas
 - ▶ Sistema de coordenadas cilíndricas
 - ▶ Sistema de coordenadas esféricas
 - ▶ ...



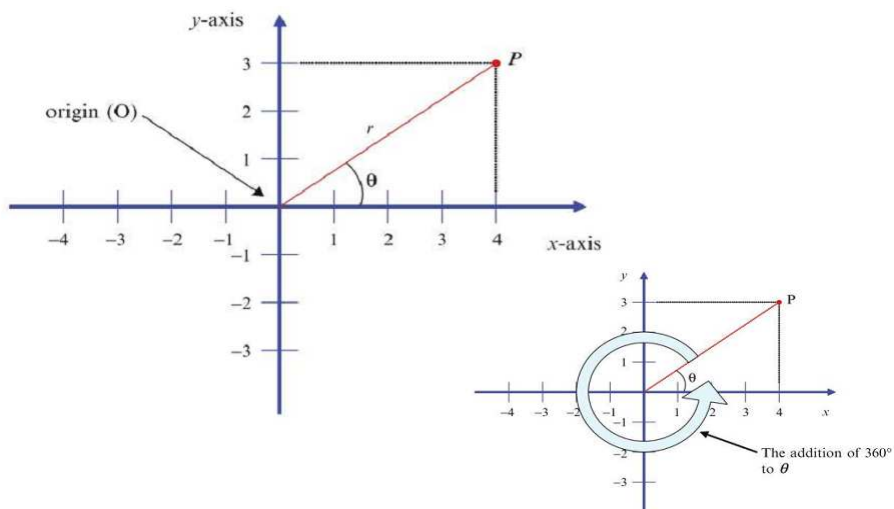
Sistema de coordenadas 2D (cartesiano)



Nota: el origen puede ser modificado (origen relativo)



Sistema de coordenadas polares



Ecuaciones básicas de líneas

- ▶ El algoritmo básico para trazar líneas entre dos puntos (x_1, y_1) , (x_2, y_2) es:

```
double m, y;
m = (y2 - y1)/(x2 - x1);
for x = x1 : x2 do
    y = (x - x1)m + y1;
    drawPixel( x, Round(y) );
end for
```

- ▶ donde:
 - ▶ Round: función de redondeo (piso o techo)
 - ▶ Número de operaciones de punto flotante: 4



Algoritmo Incremental para dibujar líneas

- ▶ Se puede agilizar el proceso de dibujado de líneas a través del **algoritmo incremental**
- ▶ Un punto se calcula a través de la expresión:

$$y_i = x_i m + B$$

- ▶ A partir de esta expresión, el punto “i+1” es:

$$y_{i+1} = x_{i+1} m + B,$$

$$y_{i+1} = (x_i + \Delta x) m + B,$$

- ▶ Como el incremento $\Delta x = 1$, se tiene que:

$$y_{i+1} = (x_i + 1) m + B,$$

$$y_{i+1} = x_i m + B + m,$$

$$y_{i+1} = y_i + m.$$



Algoritmo Incremental para dibujar líneas

```

double m, y;
m = (double)(y2 - y1)/(double)(x2 - x1);
y = y1;
for x = x1 : x2 do
    drawPixel( x, Round(y) );
    y+ = m;
end for

```

Notas:

1. Este algoritmo requiere 2 operaciones de punto flotante
2. Dado que las variables reales tienen precisión limitada, sumar un valor "m" de forma iterada introducirá un error acumulado para trazar líneas largas



Reescribiendo la ecuación de la recta

- ▶ La ecuación de una recta, en general se puede ver como:

$$y = \frac{dy}{dx}x + B$$

- ▶ Reescribiendo la ecuación, se obtiene:

$$dx \cdot y = dy \cdot x + dx \cdot B,$$

$$f(x, y) = dy \cdot x - dx \cdot y + dx \cdot B = 0.$$

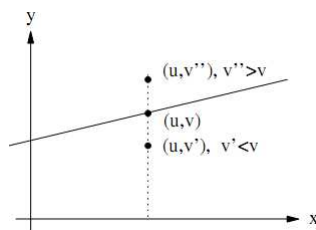
$$f(x, y) = ax + by + c = 0.$$



Reescribiendo la ecuación de la recta

- ▶ Es claro que se tienen los siguientes comportamientos de la ecuación:

- ▶ Para el punto (u,v) : $f(u,v) = 0$
- ▶ Para el punto (u, v') : $f(u,v') < 0$
- ▶ Para el punto (u,v'') : $f(u,v'') > 0$



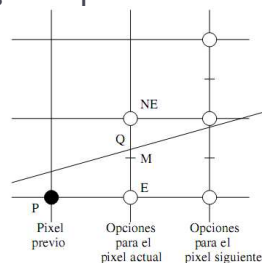
Algoritmo Punto Medio

- ▶ Sea $P = (x_p, y_p)$ un píxel, a través de "f" es posible seleccionar el siguiente píxel (punto medio)

$$M = (x_p + 1, y_p + \frac{1}{2})$$

- ▶ Notemos que:

- ▶ Si $f(M) > 0$, se escogerá el píxel NE
- ▶ Si $f(M) < 0$, se escogerá el píxel E



Algoritmo Punto Medio

- ▶ Se aplica un algoritmo incremental en $f(M)$
- ▶ Se pueden generar 2 casos:

Si ha escogido el punto E, M se incrementa un paso en la dirección x . Entonces,

$$d_{\text{act}} = f\left(x_p + 2, y_p + \frac{1}{2}\right),$$

$$= a(x_p + 2) + b\left(y_p + \frac{1}{2}\right) + c,$$

pero

$$d_{\text{ant}} = a(x_p + 1) + b\left(y_p + \frac{1}{2}\right) + c.$$

Y calculando $d_{\text{act}} - d_{\text{ant}}$ para obtener la diferencia incremental, se obtiene $\Delta_E = a = dy$.

Si se escoge el pixel NE, el siguiente punto a evaluar es

$$d_{\text{act}} = f\left(x_p + 2, y_p + \frac{3}{2}\right),$$

$$= a(x_p + 2) + b\left(y_p + \frac{3}{2}\right) + c.$$

Substrayendo d_{ant} de d_{act} obtenemos

$$d_{\text{act}} - d_{\text{ant}} = \Delta_{NE} = a + b = dy - dx;$$

Algoritmo Punto Medio

- ▶ El algoritmo inicia en $P = (x_0, y_0)$, donde el primer punto medio es $M_0 = (x_0 + 1, y_0 + 1/2)$
- ▶ De lo anterior se obtiene que:

$$f\left(x_0 + 1, y_0 + \frac{1}{2}\right) = a(x_0 + 1) + b\left(y_0 + \frac{1}{2}\right) + c$$

$$= ax_0 + by_0 + c + a + b/2$$

$$= f(x_0, y_0) + a + b/2.$$

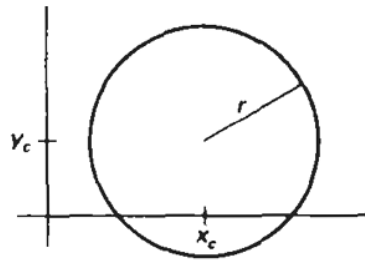
- ▶ Dado que $f(x_0, y_0) = 0$, entonces:

$$d_{\text{inicial}} = dy - dx/2$$

Círculos

- ▶ Para graficar círculos en coordenadas rectangulares, se utiliza la siguiente ecuación:

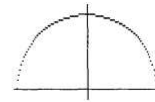
$$(x - x_c)^2 + (y - y_c)^2 = r^2$$



Círculos

- ▶ Despejando de la ecuación anterior, se puede obtener el valor para “y”

$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2}$$

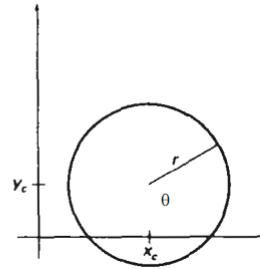


Círculos

- Una alternativa para graficar círculos es utilizar coordenadas polares

$$x = x_c + r \cos \theta$$

$$y = y_c + r \sin \theta$$



Nota: no es necesario calcular todos los puntos a través de la ecuación, basta con obtener los valores para un cuadrante

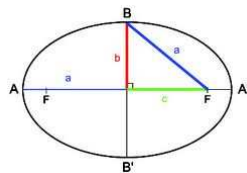


Elipse

- El cálculo de una elipse es a través de la siguiente ecuación:

$$\sqrt{(x - x_1)^2 + (y - y_1)^2} + \sqrt{(x - x_2)^2 + (y - y_2)^2} = \text{constant}$$

$$Ax^2 + By^2 + Cxy + Dx + Ey + F = 0$$

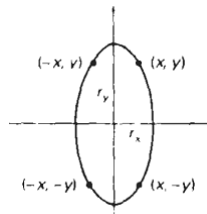


$$\left(\frac{x - x_c}{r_x} \right)^2 + \left(\frac{y - y_c}{r_y} \right)^2 = 1$$



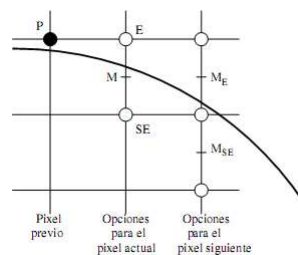
Elipse

- ▶ Al igual que para el caso del círculo, la simetría apoya para evitar el cálculo innecesario de puntos



Algoritmos punto medio

- ▶ Para graficar tanto el círculo como la elipse, es recomendable utilizar la técnica del punto medio
- ▶ A partir de dos valores fijos, determinar los valores de los puntos intermedios



- ▶ ¿Cómo calcularlos?

