

ALGORITMO PARA LA GENERACION DE UNA CIRCUNFERENCIA

Como la circunferencia es una componente común de muchos tipos de imágenes y gráficas, los procedimientos para generar circunferencias se incluyen a menudo en paquetes de gráficas. Los parámetros básicos que definen una circunferencia son las coordenadas del centro (x_c , y_c) y el radio r . Podemos expresar la ecuación de una circunferencia en varias formas mediante parámetros de coordenadas cartesianas o polares.

Ecuaciones de Circunferencias.

Una forma estándar de la ecuación de la circunferencia es el teorema de Pitágoras:

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

Ec. 9

Esta ecuación podría usarse para trazar una circunferencia recorriendo el eje x en pasos unitarios de $x_c - r$ hasta $x_c + r$ y calculando los valores de y correspondientes en cada posición como:

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

Ec. 10

Obviamente, este punto de vista implica una tarea de cálculo considerable en cada etapa y el espaciamiento entre las posiciones de los píxeles trazados no sería uniforme. Podría ajustar el espaciamiento intercambiando x y y (probando valores de y y calculando valores de x) siempre que el valor absoluto de la pendiente de la circunferencia se hiciera mayor que 1. Pero esto agrega cálculos y la verificación en el algoritmo.

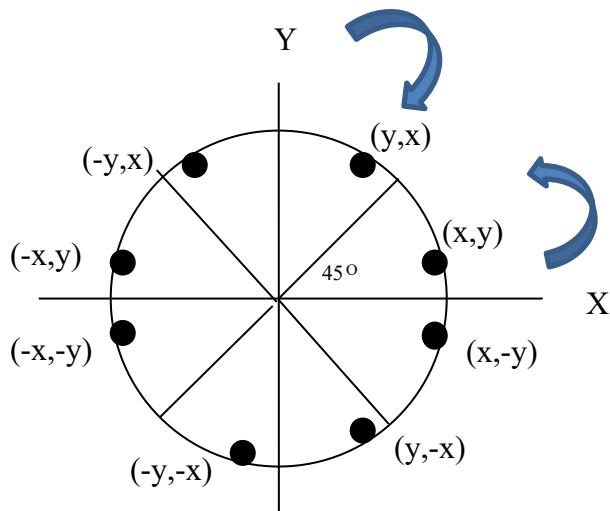
Una manera de eliminar el espaciamiento desigual asociado con la ecuación 9 consiste en calcular puntos situados en la frontera circular mediante el uso de coordenadas polares. Si se expresa la ecuación de la circunferencia en forma polar paramétrica se obtiene el par de ecuaciones

$$\begin{aligned} x &= x_c + r * \cos \theta \\ y &= y_c + r * \sin \theta \end{aligned}$$

Ec. 11

Cuando se despliega se genera con estas ecuaciones mediante el uso de un valor angular fijo de θ , se traza una circunferencia con puntos igualmente espaciados a lo largo de la circunferencia. El tamaño de la etapa seleccionada de θ depende de la aplicación. Este tamaño de etapa nos da píxeles que están separados aproximadamente una unidad.

Pueden mejorarse estos métodos aprovechando la simetría de las circunferencias. Un punto dado de la circunferencia puede trazarse en otros puntos de la circunferencia intercambiando coordenadas y alterando el signo de los valores coordinados si es necesario. Es decir, un punto en la posición (x, y) en un octavo de una circunferencia puede utilizarse para graficar los otros siete puntos que se muestran. Mediante el uso de este enfoque, podría generarse todas las posiciones de píxeles alrededor de una circunferencia de un despliegue calculando solamente los puntos dentro del sector de $x = 0$ a $x = y$.



```

void circulo (int cx, int cy, int radio)
para ángulo = 0 hasta 45 hacer
    x = radio * cos(ángulo) /*Convertir ángulo a radianes
    y = radio * sen(ángulo) /*Convertir ángulo a radianes
    pintapunto(cx+x, cy+y)
    pintapunto(cx+y, cy+x)
    pintapunto(cx-y, cy+x)
    pintapunto(cx-x, cy+y)
    pintapunto(cx-x, cy-y)
    pintapunto(cx-y, cy-x)
    pintapunto(cx+y, cy-x)
    pintapunto(cx+x, cy-y)
fin para

```