



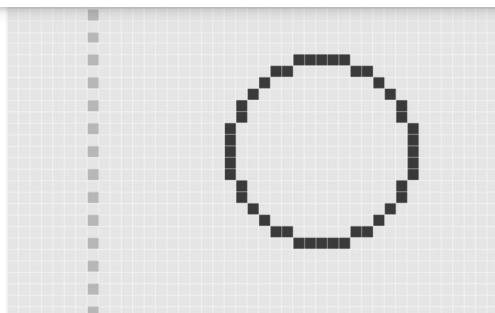
Apps

Contact

Blog

Search...




 PROGRAMMING

Thuật Toán Midpoint Vẽ Đường Tròn

NGUYỄN
NGHĨA

20/09/2020

3

• MIN
READ

Giới thiệu thuật toán Midpoint để vẽ đường tròn và hướng dẫn hiện thực bằng ngôn ngữ C++.

Tìm hiểu thuật toán Midpoint

Đường tròn có tâm $O(x_c, y_c) = (0, 0)$, bán kính r có phương trình:

$$x^2 + y^2 = r^2 \Rightarrow x^2 + y^2 - r^2 = 0$$

Đặt $f(x, y) = x^2 + y^2 - r^2$

Với mọi điểm $P(x, y)$ nằm

CONTENTS



Tìm hiểu thuật toán Midpoint

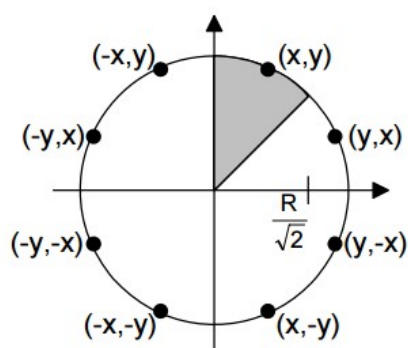
Tính F_i

Tính F_{i+1} theo F_i

Tính giá trị F đầu tiên

Hiện thực thuật toán Midpoint

- $P(x, y)$ nằm trên đường tròn O nếu $f(x, y) = 0$
- $P(x, y)$ nằm ngoài đường tròn O nếu $f(x, y) > 0$
- $P(x, y)$ nằm trong đường tròn O nếu $f(x, y) < 0$



Do đường tròn có tính đối xứng qua các cung $1/8$, nghĩa là ứng với một điểm có tọa độ (x, y) thuộc 1 cung bất kỳ, có thể hoàn toàn xác định được tọa độ 7 điểm còn lại bằng cách lấy đối xứng qua các cung.

Từ tính chất đó nên chỉ cần vẽ $1/8$ đường tròn là đủ, sau đó sẽ lấy đối xứng để được đường tròn hoàn chỉnh.

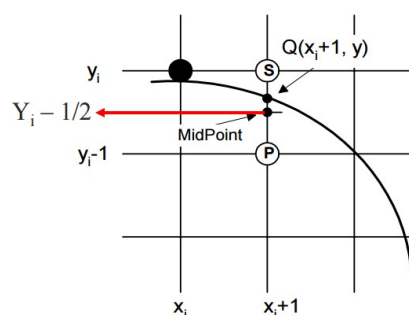
Điểm đầu tiên vẽ là điểm $(x = 0, y = R)$

Trong cung $1/8$ thứ nhất do khoảng biến thiên của x lớn hơn khoảng biến thiên

Giả sử đã vẽ được (X_i, Y_i) ở bước thứ i , cần xác định (X_{i+1}, Y_{i+1}) ở bước thứ $i + 1$.

$$X_{i+1} = X_i + 1$$

$$Y_{i+1} \in \{Y_i, Y_i - 1\}$$



Tính F_i

Đặt $F_i = F(X, Y - 1/2)$:

$$F(X_i + 1, Y_i - 1/2) = (X_i + 1)^2 + (Y_i - 1/2)^2 - R^2$$

$$F_i = X_i^2 + 2X_i + Y_i^2 - Y_i + 5/4 - R^2$$

Nếu $F_i < 0 \Leftrightarrow (X_{i+1}, Y)$ gần với $Y_i \Rightarrow Y_{i+1} = Y_i$

Nếu $F_i \geq 0 \Leftrightarrow (X_{i+1}, Y)$ gần với $Y_{i-1} \Rightarrow Y_{i+1} = Y_{i-1}$

Tính F_{i+1} theo F_i

$$F_{i+1} - F_i = 2X_i + 3 + (Y_i + 1/2 - Y_i^2) + (Y_{i+1} - Y_i) \quad (*)$$

$2X_i + 5$, do thay thế $Y_{i+1} = Y_i$ vào (*)

Nếu $F_i \geq 0$ thì $F_{i+1} = F_i + 2(X_i - Y_i) + 5$, do thay thế $Y_{i+1} = Y_{i-1}$ vào (*)

Tính giá trị F đầu tiên

$$F(X_i + 1, Y_i - 1/2) = (X_i + 1)^2 + (Y_i - 1/2)^2 - R^2$$

$$F_i = X_i^2 + 2X_i + Y_i^2 - Y_i + 5/4 - R^2$$

Thay $X_i = 0$ và $Y_i = R$ trong công thức trên:

$$F = 5/4 - R$$

Hiện thực thuật toán Midpoint

Hàm vẽ 8 điểm đối xứng nhau

```
void put8pixel(int xc, int yc, int x, int y, int color)
{
    putpixel(x + xc, y + yc, color);
    putpixel(-x + xc, y + yc, color);
    putpixel(x + xc, -y + yc, color);
    putpixel(-x + xc, -y + yc, color);
    putpixel(y + xc, x + yc, color);
    putpixel(-y + xc, x + yc, color);
    putpixel(x + xc, y - yc, color);
    putpixel(-x + xc, y - yc, color);
    putpixel(x + xc, -y - yc, color);
    putpixel(-x + xc, -y - yc, color);
}
```

```

        putpixel(y + xc, x +
yc, color);
        putpixel(-y + xc, -x +
yc, color);
    }

```

Hàm vẽ đường tròn

```

void drawCircleMidpoint(int
t xc, int yc, int r, int c
olor)
{
    int x = 0, y = r;
    int f = 1 - r;

    put8pixel(xc, yc, x,
y, color);

    while (x < y)
    {
        if (f < 0) f += (x
<< 1) + 3;
        else
        {
            y--;
            f += ((x - y)
<< 1) + 5;
        }
        x++;
        put8pixel(xc, yc,
x, y, color);
    }
}

```

Chương trình chính

```

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "
c:\\tc\\bgi");

    drawCircleMidpoint(20
0, 200, 100, colors::BLU
E);

    Sleep(3000);
    closegraph();
    return 0;
}

```

RELATED POSTS



Thư viện trong Arduino chứa mã nguồn có những đặc điểm chung, được xây dựng thành một gói bao gồm thư mục, file: examples, .h, .cpp, ... nhằm giúp việc phát triển nhanh chóng hơn. Bài viết sử dụng thư viện AFMotor làm ví dụ, đây là thư viện hỗ trợ các hàm điều khiển động cơ.

Cài Đặt Thư Viện trên Arduino IDE



Memory Leak – rò rỉ bộ nhớ xảy ra khi ta cấp phát một vùng nhớ và không thu hồi lại chúng sau khi sử dụng. Để tránh việc này xảy ra, theo lý thuyết là bất kể khi nào ta gọi toán tử new thì ngay sau đó ta cần gọi toán tử delete (tương ứng với cặp đôi malloc(...) - free(...)). Bài viết hướng dẫn bạn đọc sử dụng thư viện VLD để phát hiện memory leak.

Phát Hiện Memory Leak với VLD



Giới thiệu và hướng dẫn sử dụng thư viện std::string trong C++ xử lý chuỗi.

std::string và Xử Lý Chuỗi trong C++



Giới thiệu về thư viện Simple DirectMedia Layers và làm quen 1 số thao tác cơ bản với thư viện này.

Khởi Tạo Môi Trường Lập Trình Game Sử Dụng Thư Viện SDL (Simple DirectMedia Layer)



Apps

Contact

Blog



Home

Apps

Contact

Blog



FEATURED APPS

Pixel Ruler
Pissa Ruler
Pixel
Colour

COMMUNITY

Twitter
Facebook

LEGAL

Privacy
policy
Term of
use

STDIO CO., LTD

30 Trinh Dinh Thao, Hoa
Thanh ward, district Tan Phu,
Ho Chi Minh city, Vietnam
+84 28 22 00 11 12
developer@stdio.vn

383/1 Quang Trung, ward
10, district Go Vap, Ho Chi
Minh city

Business license number:
0311563559 issued by the
Department of Planning and
Investment of Ho Chi Minh
City on February 23, 2012

©STDIO, 2013 - 2022



Apps

Contact

Blog



Apps

Contact

Blog
