**BÀI 3: THUẬT TOÁN VẼ ĐƯỜNG THẲNG DDA**

Trong phần đồ họa máy tính, ta sẽ được làm quen với các cách tạo hình khối, đường nét, màu sắc trong giao diện đồ họa của máy tính. Và đầu tiên, ta sẽ đi tìm hiểu cách vẽ 1 đường thẳng đơn giản nhất theo thuật toán DDA.

DDA là tên viết tắt của **Digital Differential Analyzer.**

Thực ra, chúng ta có thể hiểu thuật toán DDA là thuật toán vẽ đường thẳng theo cách làm tròn tọa độ các điểm mà đường thẳng đó đi qua.

**Thuật toán DDA**

**1) Đặt vấn đề:** Cho 2 điểm A(x1,y1) và B(x2,y2). Hãy vẽ đường thẳng đi qua 2 điểm đó.

**2) Xây dựng thuật toán:**

Phương trình đường thẳng có thể phát biểu dưới dạng:

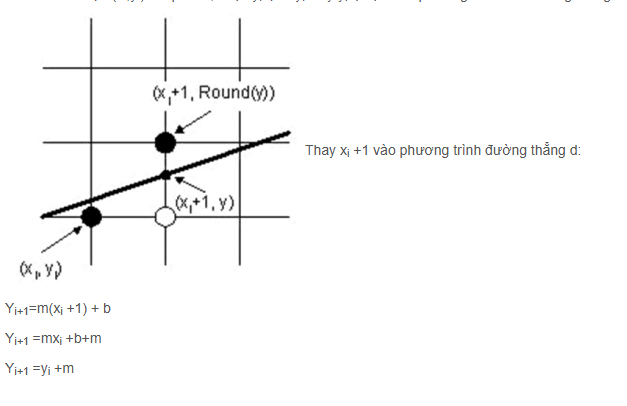
(d) y=m.x +b

m=(y2-y1)/(x2-x1)

b=y1-mx1

Trong đó: m là hệ số góc (slope)

-Giả sử vẽ được (xi,yi). Tiếp theo, chọn yi+1 là yi hay yi+1 dựa vào phương trình của đường thẳng d.



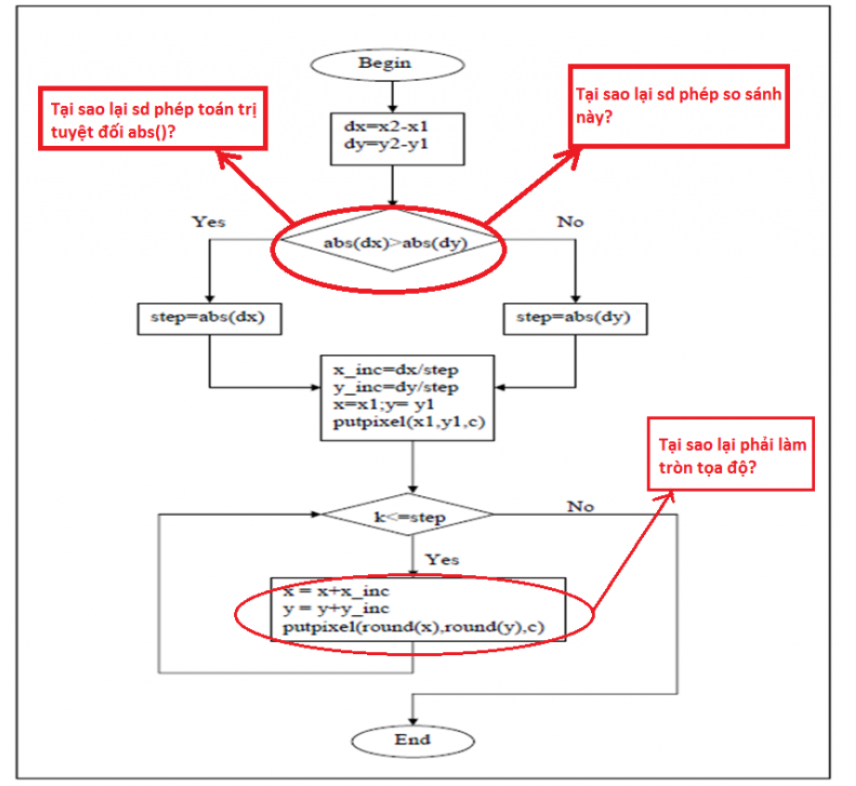
Thay xi +1 vào phương trình đường thẳng d:

Yi+1=m(xi +1) + b

Yi+1 =mxi +b+m

Yi+1 =yi +m

**3) Lưu đồ của thuật toán DDA:**



*Biểu diễn thuật toán bằng lưu đồ*

1. Tại sao lại sử dụng abs() trong phép so sánh?

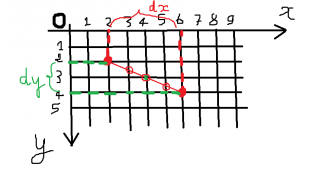
Gợi ý: Vì bản chất của phép tính abs(dx) là đi tính khoảng cách giữa 2 điểm x2, x1 . Tương tự abs(dy) là khoảng cách giữa y2,y1. Mà khoảng cách thì luôn dương.

1. Tại sao lại sử dụng phép làm tròn số ?

Gợi ý: Vì bản chất của việc vẽ đường thẳng đi qua 2 điểm A, B là nối các điểm pixel trung gian nằm giữa 2 điểm A,B. Nếu tính toán theo PT đường thẳng y=mx+b thì sẽ thu được các điểm pixel trung gian có tọa độ lẻ ( số có phần thập phân). Mà mỗi điểm pixel trên màn hình đồ họa luôn có tọa độ nguyên. Do đó, ta phải làm tròn tọa độ lẻ về số nguyên gần nhất.

1. Tại sao lại sử dụng phép so sánh abs(dx) > abs(dy) ?

Gợi ý:



Thuật toán DDA là vẽ đường thẳng theo việc làm tròn tọa độ các điểm . Điều này khiến đường thẳng được vẽ ra bị gấp khúc, không mịn . Chúng ta vẽ được càng nhiều điểm thì đường thẳng sẽ càng đẹp, càng mịn. Chính vì lý do này mà ta phải so sánh khoảng cách abs(dx) và abs(dy). Khoảng cách nào lớn hơn thì ta sẽ vẽ được nhiều điểm hơn, đường thẳng vẽ ra sẽ đẹp hơn. Xét vào ví dụ trên, nếu vẽ theo x thì ta sẽ vẽ được 5 điểm , làm cho đoạn thẳng mịn hơn so với việc vẽ theo y (chỉ có 3 điểm).

**4) Code minh họa giải thuật:**

\*chú ý: thư viện winbgim sẽ không hỗ trợ màu bằng tên như thư viện graphics

vd: int color=RED;  //hay int color=4;

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | #include <iostream>  #include <graphics.h>  #include <math.h>  #define Round(a) (int)(a+0.5)   // lam tron so  #define max(a,b) (a>b)?a:b  #define DELAY 10  #include <conio.h>  **using** **namespace** std ;  **int** color = WHITE;    **void** lineDDA(**int** x1, **int** y1, **int** x2, **int** y2){       // thuat toan DDA  **int**  Dx = x2 - x1, Dy = y2 - y1;  **float** x\_inc , y\_inc;  **float** step=max(**abs**(Dx),**abs**(Dy));      x\_inc=Dx/step;      y\_inc=Dy/step;  **float** x=x1, y=y1;// Khoi tao cac gia tri ban dau      putpixel(x, y, color);    **int** k=1;  **while**(k <=step){          k++;          delay(DELAY);  // thoi gian tre khi ve 1 diem anh          x += x\_inc;          y += y\_inc;          cout<<"x="<<x<<"\ty="<<y<<endl;          putpixel(Round(x),Round(y),color);        }  }  **int** main(){  **int** gd,gm;      gd=DETECT;      initgraph(&gd,&gm,NULL);        // khoi tao cua so do hoa      setcolor(5);      settextstyle(5,0,4);      outtextxy(250,20, a);      lineDDA(50,100,500,250);      // ve duong thang  //  line(50,100,500,250); //ham ve dg thang trong thu vien graphics / winbgim      getch();  **return** 0;  } |
| --- | --- |
|  |  |

Từ thuật toán trên, các bạn cũng có thể code theo 2 hướng : tính y theo x hoặc tính x theo y để vẽ được đường thẳng.  
Code minh họa cho việc tính y theo x:

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | #include <iostream>  #include <winbgim.h>  #define Round(a) (int)(a+0.5)   // lam tron so  #define DELAY 10  **using** **namespace** std ;  **int** color = 15;    **void** lineDDA(**int** x1, **int** y1, **int** x2, **int** y2){       // thuat toan DDA  **int** x\_unit = 1, Dx = x2 - x1, Dy = y2 - y1;     // Khoi tao cac gia tri ban dau  **int** x = x1;  **float** y = y1;  **float** m = (**float**)Dy/Dx;     // he so goc m      putpixel(x, Round(y), color);    **while**(x < x2){          delay(10);  // thoi gian tre khi ve 1 diem anh          x += x\_unit;          y += m;          putpixel(x,Round(y), color);      }  }  **int** main(){  **int** gd,gm=VGAMAX; gd=DETECT;      initgraph(&gd,&gm,NULL);        // khoi tao cua so do hoa      setcolor(5);      settextstyle(5,0,4);      outtextxy(250,20, a);      lineDDA(50,150, 300, 200);      // ve duong thang  //    line(50,150,400,400);      delay(50000);  **return** 0;  } |
| --- | --- |