

# Đồ họa máy tính

TS. Ngô Ngọc Thành

[thanhnn\\_cntt@epu.edu.vn](mailto:thanhnn_cntt@epu.edu.vn)

# Giới thiệu về đồ họa máy tính

- Đồ họa máy tính là một ngành khoa học Tin học chuyên nghiên cứu về các phương pháp và kỹ thuật để có thể mô tả và thao tác trên các đối tượng của thế giới thực bằng máy tính
- Về bản chất: đó là một quá trình xây dựng và phát triển các công cụ trên cả hai lĩnh vực phần cứng và phần mềm hỗ trợ cho các lập trình viên thiết kế các chương trình có khả năng đồ họa cao.
- Với việc mô tả dữ liệu thông qua các hình ảnh và màu sắc đa dạng của nó, các chương trình đồ họa thường thu hút người sử dụng bởi tính thân thiện, dễ dùng,... kích thích khả năng sáng tạo và nâng cao năng suất làm việc.

# Các kỹ thuật đồ họa

## □ Kỹ thuật đồ họa điểm:

- Các hình ảnh được hiển thị thông qua từng pixel (từng mẫu rời rạc).
- Với kỹ thuật này, chúng ta có thể tạo ra, xóa hoặc thay đổi thuộc tính của từng pixel của các đối tượng.
- Các hình ảnh được hiển thị như một lưới điểm rời rạc (grid), từng điểm đều có vị trí xác định được hiển thị với một giá trị nguyên biểu thị màu sắc hoặc độ sáng của điểm đó. Tập hợp tất cả các pixel của grid tạo nên hình ảnh của đối tượng mà ta muốn biểu diễn.

# Các kỹ thuật đồ họa

## □ Kỹ thuật đồ họa vector:

- Xây dựng mô hình hình học cho hình ảnh đối tượng, xác định các thuộc tính của mô hình hình học, sau đó dựa trên mô hình này để thực hiện quá trình tô trát để hiển thị từng điểm của mô hình, hình ảnh của đối tượng.
- Kỹ thuật này chỉ lưu trữ mô hình toán học của các thành phần trong mô hình hình học cùng với các thuộc tính tương ứng mà không cần lưu lại toàn bộ tất cả các pixel của hình ảnh đối tượng.

# Ví dụ đồ họa

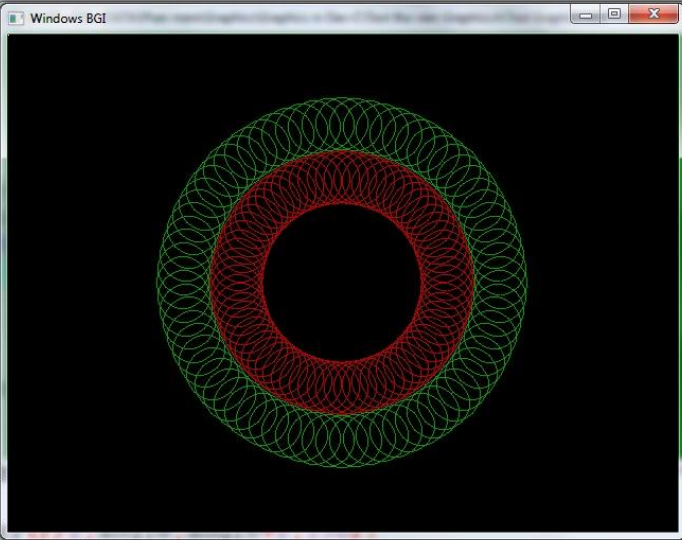
Test Graphics - [Test Graphics.dev] - [Executing] - Dev-C++ 5.6.3

File Edit Search View Project Execute Tools CVS Window Help

Test Graphics.cpp

```
1 #include<graphics.h>
2 #include<conio.h>
3 #include<dos.h>
4
5 main()
6 {
7     int gd = DETECT, gm, x,
8     struct arccoordstype a,
9     initgraph(&gd, &gm, "C:\
10     delay(2000);
11     while(angle<=360)
12     {
13         setcolor(BLACK);
14         arc(getmaxx()/2,getma
15         setcolor(RED);
16         getarccoords(&a);
17         circle(a.xstart,a.ysta
18         setcolor(BLACK);
19         arc(getmaxx()/2,getma
20         getarccoords(&a);
21         setcolor(GREEN);
22         circle(a.xstart,a.ystart,25);
23         angle = angle+5;
24         delay(50);
25     }
26     getch();
27     closegraph();
28 }
```

Windows BGI

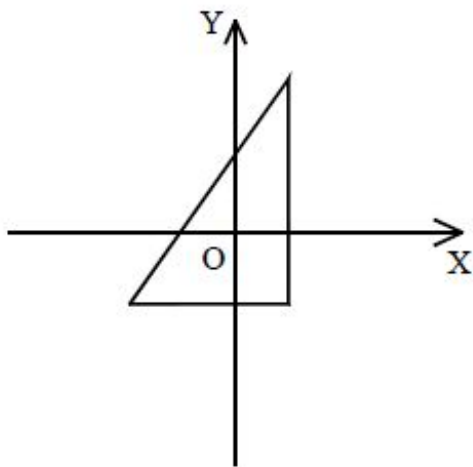


Line: 10 Col: 36 Sel: 0 Lines: 28 Length: 710 Insert Done parsing in 0.109 seconds

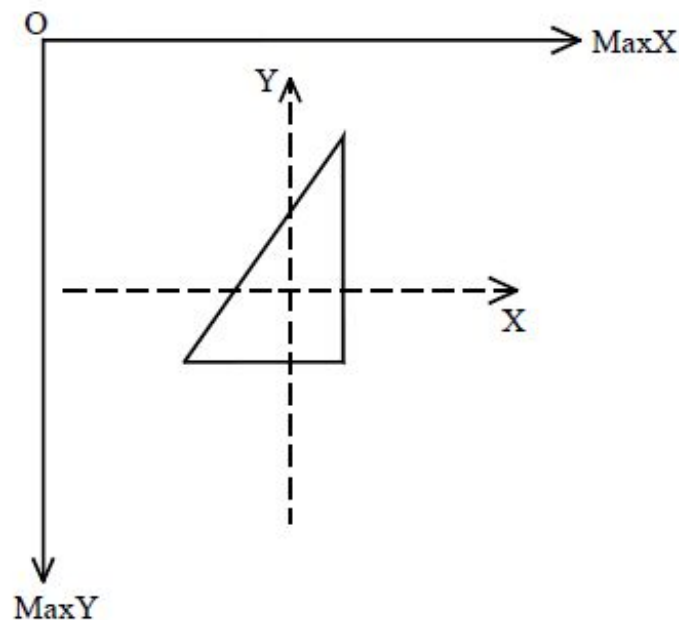
# Màn hình đồ họa

- Mỗi máy tính đều có một CARD dùng để quản lý màn hình, gọi là Video Adapter hay Graphics Adapter.
- Có nhiều loại adapter như: CGA, MCGA, EGA, VGA,
- Các adapter có thể làm việc ở hai chế độ: văn bản (Text Mode) và đồ họa (Graphics Mode)

# Biểu diễn tọa độ



Tọa độ thế giới thực



Tọa độ thiết bị màn hình.

# Vẽ điểm

- Trong các hệ thống đồ họa, một điểm (**pixel**) được biểu thị bởi các tọa độ bằng số.
  - Ví dụ: Trong mặt phẳng, một điểm là một cặp  $(x,y)$ .  
Trong không gian ba chiều, một điểm là bộ ba  $(x,y,z)$ .
- Trên màn hình của máy tính, mỗi điểm là một vị trí trong vùng nhớ màn hình dùng để lưu trữ các thông tin về độ sáng của điểm tương ứng trên màn hình.
- Số điểm vẽ trên màn hình được gọi là **độ phân giải của màn hình** (320x200, 480x640, 1024x1024,...)



# Các thuật toán vẽ đường thẳng

- Thuật toán DDA
- Thuật toán Bresenham
- Thuật toán MidPoint

# Đường thẳng

- Phương trình đường thẳng có thể phát biểu dưới dạng:

$$y = m.x + b \quad (1)$$

- Phương trình đường thẳng qua 2 điểm  $A(x_1, y_1)$ ,  $B(x_2, y_2)$ :

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$$

$$\text{Đặt } \Delta x = x_2 - x_1, \Delta y = y_2 - y_1$$

$$(*) \Leftrightarrow y = x \cdot \frac{\Delta y}{\Delta x} + y_1 - x_1 \cdot \frac{\Delta y}{\Delta x}$$

Suy ra

$$m = \frac{\Delta y}{\Delta x} \quad \text{nên } \Delta y = m \cdot \Delta x \text{ và } b = y_1 - m \cdot x_1$$

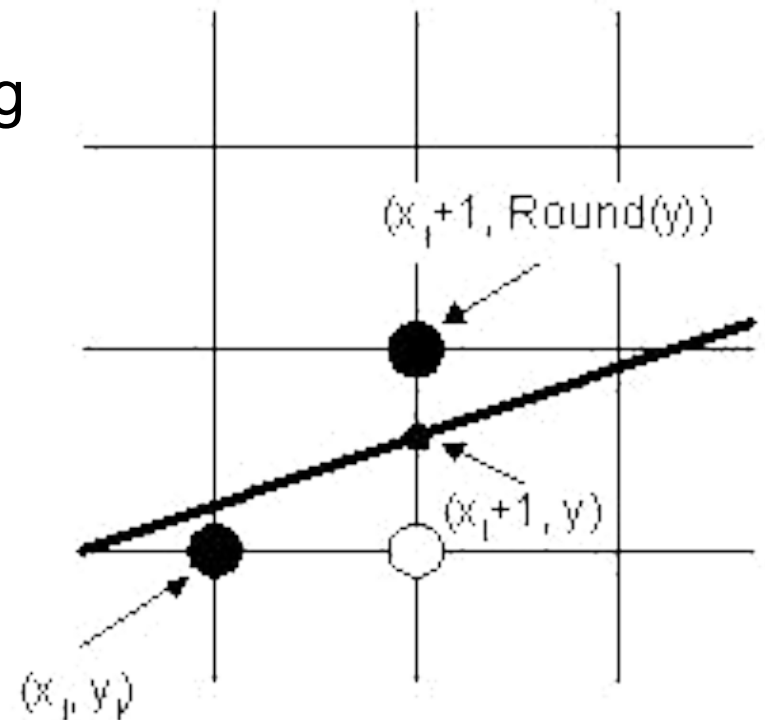
# Thuật toán DDA

□ Cho phương trình đường thẳng

d:  **$y = mx + b$**

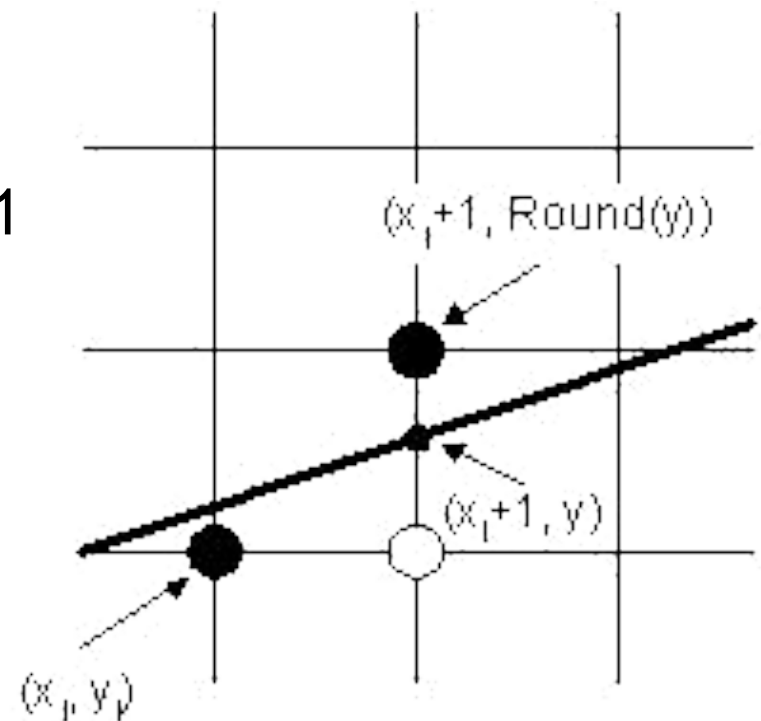
$$m = (y_2 - y_1) / (x_2 - x_1)$$

$$b = y_1 - mx_1$$



# Thuật toán DDA

- Giả sử vẽ được  $(x_i, y_i)$
- Tiếp theo, chọn  $y_{i+1}$  là  $y_i$  hay  $y_i+1$  dựa vào phương trình của đường thẳng  $d$ .



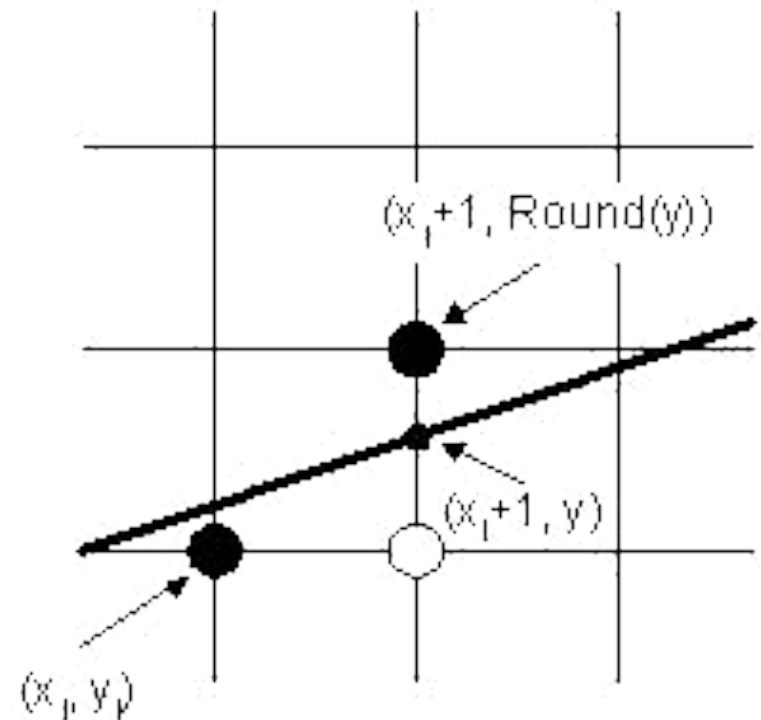
# Thuật toán DDA

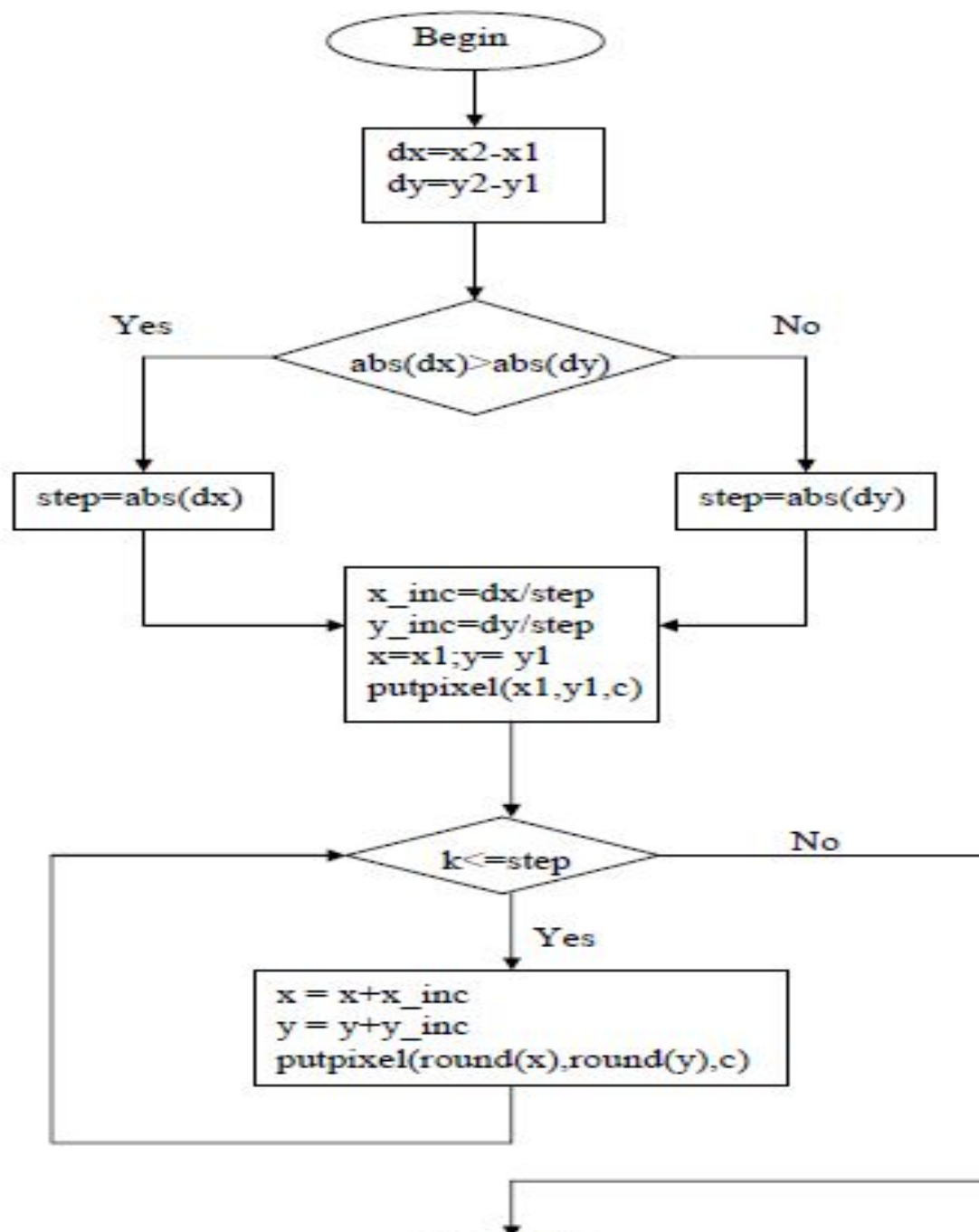
- Thay  $x_i+1$  vào phương trình đường thẳng d:

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

$$y_{i+1} = mx_i + b + m$$

$$\mathbf{y_{i+1} = y_i + m}$$





# Thuật toán DDA – Ngôn ngữ tự nhiên

- Ta có thuật toán vẽ đường thẳng DDA như sau:
  - Nhập  $A(x_1, y_1)$   $B(x_2, y_2)$
  - Tính  $\Delta x = x_2 - x_1$ ,  $\Delta y = y_2 - y_1$  và  $Step = \text{Max}(|\Delta x|, |\Delta y|)$
  - Khởi tạo các giá trị:  
 $IncX = \Delta x / Step$ ;  $IncY = \Delta y / Step$ ; {bước tăng khi vẽ}  
 $x = x_1$ ;  $y = y_1$ ; {Chọn điểm vẽ đầu tiên}  
Vẽ điểm  $(x, y)$ ;
  - Cho  $i$  chạy từ 1 đến  $Step$ :  
 $x = x + IncX$ ;  $y = y + IncY$ ;  
Vẽ điểm  $(\text{Round}(x), \text{Round}(y))$

# Thuật toán DDA - Code

```
#include <iostream>
#include <winbgim.h>
#define Round(a) (int)(a+0.5) // lam tron so
#define DELAY 10
using namespace std ;
int color = RED;

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
    setbkcolor(WHITE);
    //lineDDA(50, 150, 300, 200); // ve duong thang
    line(50, 150, 400, 400);
    delay(50000);
    return 0;
}
```



# Cài đặt đồ hoạ trong DevC++

□ Link

download:

<https://drive.google.com/file/d/189xUSvY4IkFxOmN7y0PAaGNzitJKFfR8/view?usp=sharing>

□ Bước 1: Cài đặt “**Dev-Cpp 5.6.1 MinGW 4.8.1 Setup.exe**”

□ Bước 2: Lần lượt tải 2 tập tin winbgim.h và conio.h và cho vào 2 thư mục trong thư mục cài đặt của DevC++: (Giả sử mình cài đặt DevC++ trong ổ đĩa C )

C:\Dev-Cpp\minGW32\include

C:\Dev-Cpp\minGW32\mingw32\include

# Cài đặt đồ hoạ trong DevC++

- **Bước 3:** Tiếp tục tải 2 tập tin libbgi.a và libconio.a cho vào 2 thư mục:

C:\Dev-Cpp\minGW32\lib

C:\Dev-Cpp\minGW32\mingw32\lib

- **Bước 4:** tải tập tin 6-ConsoleAppGraphics.template và cho vào thư mục :

C:\Dev-Cpp\Templates

# Sử dụng

- Tạo một project loại Console Graphics Application và chọn ngôn ngữ là C++ để có tập tin lưu dưới dạng main.cpp.
- Bước 1: Khởi động Dev C++
- Bước 2: File → New→Project...→Console Graphics Application.

Phần ngôn ngữ chọn C++ project.

# Ví dụ

```
#include <iostream>
#include <winbgim.h>
using namespace std ;

int main()
{
    initwindow(400,500) ;    //khởi tạo window có chiều rộng x=400
                             và chiều cao y = 500
    setwindowtitle("Giao diện đồ họa DevC++") ;
                             //thiết lập tiêu đề cho windows
    circle(200,250,50) ;    //vẽ đường tròn tâm có tọa độ
                             (200,250) và bán kính 50px
    delay(50000) ;          //cho chương trình ngủ trong 5 giây
    return 0 ;
}
```