

CHƯƠNG 6

ĐỒ HOẠ VÀ ÂM THANH

Đồ họa
Âm thanh

I. ĐỒ HOẠ

1. Khái niệm đồ họa

a. Điểm ảnh và độ phân giải

Màn hình ở chế độ đồ họa là tập hợp các điểm (pixel-picture elements) ảnh. Số điểm ảnh và cách bố trí theo chiều ngang, dọc của màn hình được gọi là độ phân giải (resolution). Vì vậy độ phân giải thường được đặc trưng bởi một cặp số chỉ định số điểm ảnh theo chiều ngang và chiều dọc của màn hình. Ví dụ màn hình VGA ở mode 2 có độ phân giải là 640x480, tức trên mỗi dòng ngang của màn hình có thể vẽ được 640 điểm ảnh và trên mỗi cột dọc vẽ được 480 điểm ảnh. Các cột và dòng được đánh số từ 0, theo chiều từ trái sang phải (đối với cột) và từ trên xuống dưới (đối với dòng). Một điểm ảnh hay còn gọi là pixel là giao điểm của một cột và một dòng nào đó trên màn hình và vị trí của nó được thể hiện bởi cặp tọa độ (x,y) với x biểu diễn cho cột và y biểu diễn cho dòng. Ví dụ với màn hình trên điểm ảnh “đầu tiên” nằm ở góc trên bên trái của màn hình có tọa độ (0,0) và điểm “cuối cùng” ở góc dưới bên phải có tọa độ (639,479). Điểm có tọa độ (150,200) là giao điểm của cột thứ 150 và dòng 200.

b. Trình điều khiển đồ họa

Màn hình đồ họa có nhiều loại khác nhau. Mỗi loại màn hình cần có trình điều khiển tương ứng. Cung cấp các trình điều khiển màn hình trong thư mục BGI đặt dưới thư mục gốc của C (TC hoặc BC) gồm có:

Tên trình điều khiển	Kiểu màn hình đồ họa
ATT.BGI	ATT & T6300 (400 dòng)
CGA.BGI	IBMCGA, MCGA và các máy tương thích
EGAVGA.BGI	IBM EGA, VGA và các máy tương thích

HERC.BGI	Hercules mono và các máy tương thích
IBM8514.BGI	IBM 8514 và các máy tương thích
PC3270.BGI	IBM 3270 PC

Ngoài các trình điều khiển trong thư mục BGI còn chứa các file font chữ có đuôi CHR gồm:

GOTH.CHR

LITT.CHR

SANS.CHR

TRIP.CHR

c. Một (mode) đồ họa

Mỗi màn hình đồ họa có thể hoạt động dưới nhiều một khác nhau. Độ phân giải của màn hình phụ thuộc vào từng một. Ví dụ màn hình VGA có thể hoạt động dưới các một 0 (VGALO: độ phân giải thấp 640x200), 1 (VGAMED: độ phân giải trung bình 640x350), 2 (VGAHI: độ phân giải cao 640x480).

2. Vào/ra chế độ đồ họa

Trong C++ các hàm liên quan đến đồ họa được khai báo trong tệp <graphics.h>

a. Khởi động chế độ đồ họa

void initgraph(int *graphdriver, int *graphmode, char *drivepath)

- drivepath: đường dẫn của thư mục chứa các trình điều khiển đồ họa. Nếu rỗng sẽ tìm trong thư mục hiện tại.
- graphdriver, graphmode: Chỉ định trình quản lý và một màn hình cần sử dụng. Trong đó graphdriver có thể nhận 1 trong các giá trị sau:

DETECT	0
CGA	1
EGA	3
EGA64	4
EGAMONO	5
VGA	9
.....	..

Hiển nhiên việc chọn giá trị của graphdriver phải tương ứng với màn hình thực tế. Trong trường hợp ta không biết chủng loại thực tế của màn hình có thể sử dụng giá trị DETECT (hoặc 0) là giá trị chỉ định cho chương trình tự tìm hiểu về màn hình và gọi trình điều khiển tương ứng. Trong trường hợp này graphmode sẽ được gán giá trị tự động với mode có độ phân giải cao nhất có thể. Về graphmode có thể nhận các giá trị sau:

CGAC0	0	320 x 200	
CGAC1	1	320 x 200	
CGAC2	2	320 x 200	
CGAC3	3	320 x 200	
CGAHI	4	640 x 200	2 color
EGALO	0	640 x 200	16 color
EGAHI	1	640 x 350	16 color
EGA64LO	0	640 x 200	16 color
EGA64HI	1	640 x 350	4 color
VGALO	0	640 x 200	16 color
VGAMED	0	640 x 350	16 color
VGAHI	0	640 x 480	16 color

Trong quá trình sử dụng để xoá màn hình đồ hoạ ta dùng hàm **cleardevice()**;

b. Kết thúc chế độ đồ hoạ

Để kết thúc chế độ đồ hoạ về lại chế độ văn bản ta sử dụng hàm **closegraph()**;

c. Lỗi đồ hoạ

- Sau mỗi thao tác đồ hoạ, hàm **graphresult()** sẽ cho giá trị 0 nếu không có lỗi, hoặc các giá trị âm (-1 .. -18) tương ứng với lỗi. Hàm **grapherrormsg(n)** trả lại nội dung lỗi và mã lỗi.

Mã lỗi	Hàng lỗi (graphresult())	Nội dung lỗi (grapherrormsg())
0	grOk	No error
-1	grNoInitGraph	(BGI) Không có BGI
-2	grNotDetected	Graphics hardware not detected

-3

grFileNotFound

Device driver file not found

Ví dụ 1 :

Ví dụ sau đây khởi tạo chế độ đồ họa với `graphdriver = 0` (DETECT) và thông báo lỗi nếu không thành công hoặc thông báo chế độ đồ họa cũng như mode màn hình. Để biết độ phân giải của màn hình có thể dùng các hàm **getmaxx()** (số cột) và **getmaxy()** (số dòng)

```
void main()
{
    int gd = DETECT, gm, maloi;
    initgraph(&gd, &gm, "C:\\BC\\BGI");
    maloi = graphresult();
    if (maloi != grOk)
    {
        cout << "Lỗi: " << grapherrormsg(maloi) << endl;
        cout << "An phím bất kỳ để dừng "; getch();
        exit(1);
    } else {
        cout << "Chế độ màn hình = " << gd << endl;
        cout << "Mode màn hình = " << gm << endl;
        cout << "Độ phân giải: " << getmaxx() << ", " << getmaxy() << endl;
        getch();
    }
    closegraph();
}
```

Các phần tiếp theo sau đây sẽ cung cấp các câu lệnh để vẽ trong chế độ đồ họa.

3. Vẽ điểm, đường, khối, màu sắc

a. Màu sắc

- **getmaxcolor()**: Trả lại số hiệu (hằng) tương ứng với màu tối đa của màn hình hiện tại. Do các hằng màu được tính từ 0 nên số màu sẽ bằng hằng trả lại cộng

thêm 1.

- **setbkcolor(màu):** Đặt màu nền. Có tác dụng với văn bản và các nét vẽ.
- **setcolor(màu):** Đặt màu vẽ. Có tác dụng với văn bản và các nét vẽ.
- **getbkcolor():** Trả lại màu nền hiện tại.
- **getcolor():** Trả lại màu vẽ hiện tại (từ 0 đến getmaxcolor()).

Ví dụ: In tọa độ tại vị trí hiện tại của con trỏ màn hình. Trong ví dụ này chúng ta sử dụng câu lệnh `sprintf(xâu s, "dòng đk", các biểu thức cần in)`, câu lệnh này sẽ thay việc in các biểu thức ra màn hình thành in ra xâu s (tức tạo xâu s từ các biểu thức). Ngoài ra hàm `outtextxy(x, y, s)` sẽ in xâu s tại vị trí (x,y).

```
void intoado(dx, dy, color)
{
    int oldcolor;
    oldcolor = getcolor();
    setcolor(color);
    char td[10];
    sprintf(td, "(%d, %d)", getx(), gety());
    outtextxy(getx()+dx, gety()+dy, td);
    setcolor(oldcolor);
}
```

b. Vẽ điểm

- **putpixel(x, y, c):** Vẽ điểm (x, y) với màu c.
- **getpixel(x, y):** Trả lại màu tại điểm (x, y).

Ví dụ 2 : Vẽ bầu trời sao

```
void sky()
{
    int maxx, maxy, maxc;
    int i, xarr[3001], yarr[3001];
    maxx = getmaxx();
    maxy = getmaxy();
    maxc = getmaxcolor();
}
```