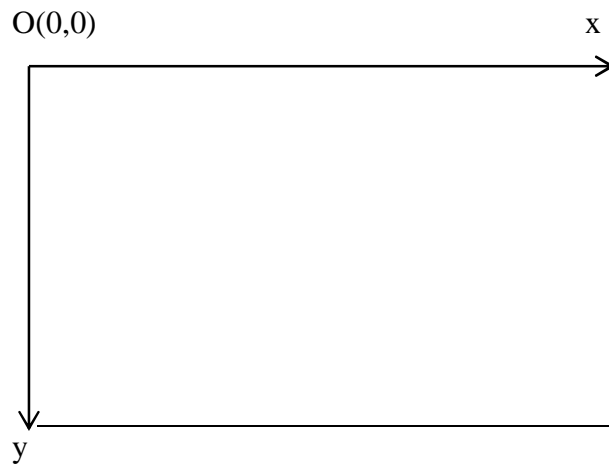


ĐỒ HỌA VÀ ÂM THANH TRONG C++

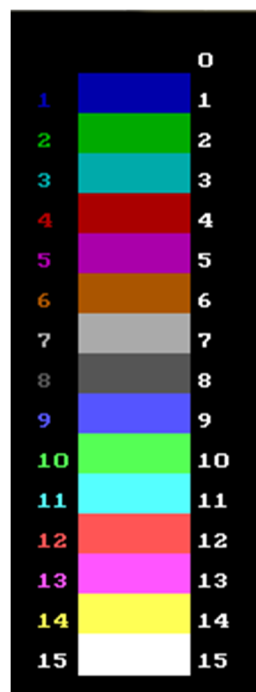
1. Đồ họa

- Màn hình đồ họa



Kích thước màn hình đồ họa trong TurboC thường là (640,480) và 16 màu. Vậy x chạy từ 0 -> 639, y chạy từ 0 -> 479

- Bảng 16 màu:



- Khởi động chế độ đồ họa:
 - + Trong TurboC, BorlandC: khai báo tự động tìm mode màn hình


```
int manhin,mode;
manhin = mode = 0; // tự động tìm mode màn hình
initgraph(&manhin,&mode,"đường dẫn tới thư mục chứa file EGAVGA.BGI");
```

 Trong TurboC file **EGAVGA.BGI** ở thư mục: **"*\\TC\\BGI"**
 - + Trong Devcpp: trước khi lập trình đồ họa trên devcpp bạn phải cài thư viện đồ họa, hướng dẫn chi tiết trên google search


```
initwindow(int x,int y);// kích thước màn hình đồ họa dài rộng bao nhiêu
          VD: initwindow(900,700);
```
- Kiểm tra xem có lỗi gì khi tạo màn hình đồ họa:


```
graphresult();
```

 hàm trả lại 0: không có lỗi gì, từ 1 -> 18 tương ứng với các lỗi
- Xóa tất cả những hình đã vẽ trên màn hình đồ họa:


```
cleardevice();
```
- Kết thúc chế độ đồ họa:


```
closegraph();
```
- Các hàm thường dùng

<code>getmaxx()</code>	Tọa độ x lớn nhất
<code>getmaxy()</code>	Tọa độ y lớn nhất
<code>getx()</code>	Vị trí x hiện tại của con trỏ
<code>gety()</code>	Vị trí y hiện tại của con trỏ
<code>setbkcolor(<màu nền>)</code>	Đặt màu nền: 0 -> 15
<code>setcolor(<màu vẽ>)</code>	Đặt màu vẽ: 0 -> 15
<code>getbkcolor()</code>	Màu nền hiện tại
<code>getmaxcolor()</code>	Số màu tối đa của màn hình
<code>getcolor()</code>	Màu vẽ hiện tại
<code>putpixel(x,y,c)</code>	Vẽ điểm có tọa độ (x,y) với màu c
<code>getpixel(x,y)</code>	Trả lại màu tại điểm (x,y)
<code>line(x,y,x1,y1)</code>	Vẽ đoạn thẳng đi qua (x,y) và (x1,y1)
<code>lineto(x,y)</code>	Vẽ đoạn thẳng đi qua vị trí hiện tại của con trỏ tới điểm (x,y)
<code>linrel(x,y)</code>	Giống với line(x,y)

<code>rectangle(x1,y2,x2,y2)</code>	Vẽ hình chữ nhật rỗng
<code>bar(x1,y2,x2,y2)</code>	Vẽ hình chữ nhật đặc
<code>bar3d(x1,y2,x2,y2,h,top)</code>	h là chiều cao, top = 1 có nắp
<code>setlinestyle(kiểu đường, mẫu tô, độ đậm)</code>	+ kiểu đường từ 0 -> 4 0: đường đặc 1: đường chấm 2: đường gạch 3: đường gạch dài 4: đường tự tạo + mẫu tô: chỉ có tác dụng khi kiểu đường là 4, ta dùng 2byte để định nghĩa + độ đậm là 1 hoặc 3 1: nét vẽ bình thường 3: nét vẽ đậm
<code>setfillstyle(mẫu tô, màu tô)</code>	+ mẫu tô: từ 0 -> 12 + màu tô: từ 0 -> 15
<code>setfillpattern(mẫu tô, màu tô)</code>	Định nghĩa mẫu tô
<code>getfillsettings(struct fillsettingstype *info)</code>	Lấy mẫu tô hiện tại
<code>getfillpattern(mẫu tô)</code>	Trả lại mẫu tô hiện tại
<code>drawpoly(mảng số nguyên chứa tọa độ các điểm,số cặp điểm)</code>	Vẽ đa giác rỗng
<code>fillpoly(mảng số nguyên chứa tọa độ các điểm,số cặp điểm)</code>	Vẽ đa giác đặc
<code>arc(x, y, góc đầu, góc cuối, bán kính)</code>	Vẽ cung tròn có tâm (x, y) với các góc và bán kính tương ứng
<code>circle(x, y, bán kính)</code>	Vẽ đường tròn có tâm tại (x, y)
<code>pieslice(x, y, góc đầu, góc cuối, bán kính)</code>	Vẽ hình quạt tròn đặc với mẫu hiện tại
<code>ellipse(x, y, góc đầu, góc cuối, a, b)</code>	Vẽ cung elip với tâm, các góc và các bán kính theo hoành độ và tung độ tương ứng
<code>fillellipse(x, y, a, b)</code>	Vẽ hình elip đặc
<code>sector(x, y, góc đầu, góc cuối, a, b)</code>	Vẽ hình quạt elip
<code>floodfill(x, y, c)</code>	Tô màu một hình kín chứa điểm x, y và màu c, màu c phải trùng với <code>setfillstyle(mẫu tô, c);</code>

- Viết văn bản trong màn hình đồ họa
`outtextxy(x, y, s)`: viết văn bản tại vị trí (x,y)
`moveto(x, y)`: chuyển con trỏ chuột đến vị trí (x,y)
- Điều chỉnh Font chữ, hướng cỡ chữ

`settextstyle(Font, hướng, cỡ chữ)`

+ Font

<code>DEFAULT_FONT</code>	<code>0</code>
<code>SMALL_FONT</code>	<code>1</code>
<code>TRIPLEX_FONT</code>	<code>2</code>
<code>SANS_SERIF_FONT</code>	<code>3</code>
<code>GOTHIC_FONT</code>	<code>4</code>

+ Hướng

`HOIRIZ_DIR 0` // nằm hàng ngang

`VERT_DIR 1` // nằm theo đường thẳng đứng

+ Cỡ chữ bắt đầu từ 1

- Điều chỉnh cách viết

`settextjustify`(theo hướng ngang, theo hướng dọc)

+ Theo hướng nằm ngang:

`LEFT_TEXT = 0` : Viết từ trái sang phải.

`CENTER_TEXT = 1` : Viết từ vị trí con trỏ sang hai bên.

`RIGHT_TEXE = 2` : Viết từ phải sang trái.

+ Theo hướng thẳng đứng:

`BOTTOM_TEXT = 0` : Viết từ dưới lên.

`CENTER_TEXT = 1` : Viết từ vị trí con trỏ lên trên và xuống dưới.

`TOP_TEXT = 2` : Viết từ trên xuống

Ví dụ: viết chương trình hiện thị bảng 16 màu

```
#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
```

```
int main()
{
    int mh,mode,i,x,y;
```

```

char a[3];
mh = mode = 0;
initgraph(&mh,&mode,"C:\\TC\\BGI");
x = 20;
y = 10;
for(i=0;i<16;i++)
{
    setfillstyle(1,i);
    bar(x+15,y,x+70,y+20);
    itoa(i,a,10);
    setcolor(i);
    outtextxy(x-5,(2*y+20)/2,a);
    setcolor(15);
    outtextxy(x+75,(2*y+20)/2,a);
    y = y + 20;
}
getche();
closegraph();
return 0;
}

```

- Chuyển động: xóa hình cũ bằng cách vẽ hình đó có màu trùng với màu nền rồi vẽ hình mới với màu cũ.

VD: vẽ hình tròn đặc chuyển động từ trái qua phải

```

#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
#include<stdlib.h>

//ve hình tron đặc chuyển động từ trái qua phải
void HT()
{
    int x,y,n;
    y = getmaxy()/2;
    n = getmaxx() - 10;
    x = 20;
    while(!kbhit())
    {
        if(x>n-10) x = 20;
        setcolor(4);
        circle(x,y,20);
    }
}

```

```

        setfillstyle(1,4);
        floodfill(x,y,4);
        delay(10);
        //xoa hình tron vua ve
        setcolor(0);
        circle(x,y,20);
        setfillstyle(1,0);
        floodfill(x,y,0);
        x = x + 1;
    }
}

//ham chinh
int main()
{
    int mh,mode,loi;
    mh = mode = 0;
    initgraph(&mh,&mode,"C:\\TC\\BGI");
    loi = graphresult();
    if(lois)
    {
        cout<<"\n - Co loi kho khoi tao do hoa";
        cout<<" - Ma loi: "<<loi;
        getch();
        return 0;
    }
    HT();
    getch();
    closegraph();
    return 0;
}

```

2. Âm thanh

- Trong TurboC:

sound(n): phát âm ra loa máy tính với tần số n

delay(ms): kéo dài trong ms miligiây

nosound(): tắt âm thanh đã phát

+ Các hàm này nằm ở thư viện **dos.h**

- Tần số âm thanh của một nốt nhạc:

Đô 1	66	Đô thăng	70
Rê	73	Rê thăng	78
Mi	82	Pha	86
Pha thăng	91	Son	96
Son thăng	102	La	108
La thăng	115	Si	122
-----	-----	-----	-----
Đô 2	130	Đô thăng	139
Rê	148	Rê thăng	156
Mi	164	Pha	176
Pha thăng	188	Son	196
Son thăng	209	La	220
La thăng	233	Si	247
-----	-----	-----	-----
Đô 3	264	Đô thăng	281
Rê	297	Rê thăng	313
Mi	330	Pha	352
Pha thăng	374	Đô thăng	396
Son	415	La	440
La thăng	468	Si	495
-----	-----	-----	-----
Đô 4	528	Đô thăng	565
Rê	594	Rê thăng	625
Mi	660	Pha	704
Pha thăng	748	Son	792
Son thăng	836	La	880
La thăng	935	Si	990
-----	-----	-----	-----
Đô 5	1050	Siêu âm	30000

3. Bài tập

Bài 1. Viết chương trình mô phỏng chuyển động của kim giây

```
#include<graphics.h>
#include<conio.h>
#include<math.h>
#include<dos.h>

void KimGiay()
{
```

```

int a,b,r,goc,xoa;
float x,y;
a = getmaxx()/2;
b = getmaxy()/2;
setlinestyle(0,0,1);
circle(a,b,4);
//floodfill(x,y,4);
setcolor(4);
circle(a,b,b-10);
r = b - 18;
x = y = 0;
setcolor(2);
goc = 0;
while(!kbhit())
{
    x = r*cos((goc*3.14)/180);
    if(x<0) x = a + x;
    else x = a + x;
    y = r*sin((goc*3.14)/180);
    if(y<0) y = b + y;
    else y = b + y;
    setcolor(2);
    line(a,b,(int)ceil(x),(int)ceil(y));
    delay(200);
    goc++;

    //xoa duong vua ve
    xoa = goc - 1;
    x = r*cos((xoa*3.14)/180);
    if(x<0) x = a + x;
    else x = a + x;
    y = r*sin((xoa*3.14)/180);
    if(y<0) y = b + y;
    else y = b + y;
    setcolor(0);
    line(a,b,(int)ceil(x),(int)ceil(y));
    if(goc>360) goc = 0;
    delay(50);
}
}

//ham chinh
int main()
{
    int mh,mode;
    mh = mode = 0;

```



```

initgraph(&mh,&mode,"C:\\TC\\BGI");
KimGiay();
getch();
closegraph();
return 0;
}

```

Bài 2. Vẽ chuyển động theo hình xoắn ốc

```

#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
#include<process.h>

//hình xoắn ốc
void XoanOc()
{
    int x,y;
    x = getmaxx()/2;
    y = getmaxy()/2;
    int trai,phai,tren,duoi;
    trai = phai = x;
    tren = duoi = y;
    while((x>0&&x<getmaxx())&&(y>0&&y<getmaxy()))
    {
        //ve mot nua tu duoi len tren
        tren = tren - 6;
        while(y>=tren)
        {
            putpixel(x,y,4);
            y--;
            delay(2);
        }
        tren = tren - 16;
        //ve tu tu trai qua phai
        phai = phai + 10;
        while(x<=phai)
        {
            putpixel(x,y,4);
            x++;
            delay(2);
        }
        phai = phai + 12;
        //ve tu tren xuong duoi
    }
}

```

```

    duoi = duoi + 12;
    while(y<=duoi)
    {
        putpixel(x,y,4);
        y++;
        delay(2);
    }
    duoi = duoi + 12;
    //ve tu phai qua trai
    trai = trai - 10;
    while(x>=trai)
    {
        putpixel(x,y,4);
        x--;
        delay(2);
    }
    trai = trai - 12;
    //ve not mot nua con lai tu duoi len tren
    while(y>=getmaxy()/2)
    {
        putpixel(x,y,4);
        y--;
        delay(2);
    }
}

int main()
{
    int mh,mode;
    mh = mode = 0;
    initgraph(&mh,&mode,"C:\\TC\\BGI");
    if(graphresult()!=0)
    {
        cout<<"Co loi";
        exit(1);
    }
    XoanOc();
    getch();
    closegraph();
    return 0;
}

```