运用淡入淡出和动画技术制作动态软件封面

孙　乔　(北京理工大学)　许　悦　(河南建设银行)

摘要:本文介绍采用淡入淡出和动画技术制作带有动感软件封面的方法。

一、引 言

美观生动的封面能提高软件使用者的兴趣和工作效率,但若软件画面生成速度慢则会影响图面的美观,而静止的画面又会给人以生硬的感觉。笔者从游戏软件中得到启发,在本文介绍运用淡入淡出和动画技术制作具有动感软件封面的方法。

二、显示模式的选择

高分辨率下固然可以得到清晰的画面,但是,常用的12H(640x480 16色)模式因为颜色的表现力较弱,不能满足要求。因此,我们采用低分辨率(320x200)下的256色模式。这种13H模式是VGA(TVGA,SVGA)所特有的,虽然分辨率较低,但其丰富的色彩表现力可以弥补这一不足。

与12H的位面式结构不同,13H模式下显示内存被线性地映射,屏幕上一个象素对应着显示内存中的一个字节,计算公式如下:

$$字节地址=Y\times 320+X \qquad (2-1)$$

其中,$Y\in[0,199]$ $X\in[0,319]$

三、开、关屏幕技术

许多软件在生成图象时,能看到画图的整个过程,从而影响了视觉效果,许多特技也无法实现。因此,我们希望在绘图的过程中,屏幕不要显示,就象看演出时幕布要等后台准备好才拉开一样,而采用关闭屏幕显示技术即可达到这一效果。

BIOS的10H中断提供了实现这一技术的功能,其入口参数为:AH=12H, BL=36H, AL=0(允许显示),AL=1(禁止显示),返回值AL=12H。C语言函数如下:

```
/*关闭屏幕显示*/
void crtclose()
{
    union REGS r;
    r.h.ah=0x12;
    r.h.bl=0x36;
    r.h.al=1;
    int86(0x10,&r,&r);
}
/*打开屏幕显示*/
void crtopen ()
{
    union REGS r;
    r.h.ah=0x12;
    r.h.bl=0x36;
    r.h.al=0;
    int86(0x10,&r,&r);
}
```

四、淡入淡出技术

在游戏当中,经常可以看到图象渐渐显示出来,或渐渐暗下去,直到消失,给人一种柔缓的感觉,这就是淡入淡出(Fade-in, Fade-out),它实际上是影视技巧中的一种。

VGA视频DAC由R(红)、G(绿)、B(蓝)三个视频DAC组成,每个视频DAC将6位二值彩色信息转换成模拟电压驱动VGA显示器,彩色查找表(Color Look-up Table)将8位VGA属性控制器输出值转换成18位(每个DAC 6位),从而VGA可以同时显示256K颜色中的256种。VGA的256个彩色寄存器均没有独立的输入输出口地址,其索引号从00到FF。当需要修改某一彩色寄存器的值时,首先把该寄存器的索引号送到彩

色查找表的地址寄存器，这时，彩色查找表处于等待写入的状态，依次将三个分量的值写入彩色表数据寄存器则完成了修改。每写三个字节后，寄存器自动加 1，所以，不必重复设置就可以给一组彩色寄存器赋值。

对 VGA 卡，有 64 级灰度，其定义为：

灰度＝0.3＊R+0.59＊G+0.11＊B

所以，只要修改 R、G、B 分量的值就可以调节颜色的灰度，从而改变整幅图的亮度。这就是淡入淡出技术实现的基础。

在对灰度进行调整时，要求各分量(R,G,B)的比例保持不变，从而图在处理过程中颜色不会发生失真。

为了使淡入淡出的过程平滑，必须采用直接写口地址的方式(采用其他方法，如调用中断设置调色板，在有的机子上会产生强烈的抖动感)，这样速度快，人眼感觉不到颜色调节的中间过程。所用到的寄存器包括 0x3C8,0x3C9，首先向 0x3C8 写入要修改的查找表序号，然后向 0x3C9 连续三次分别写入 R、G、B 分量的值，即可完成对该颜色号的修改。

在淡入过程中，获得图象的调色板后将调色板置零，然后调节各分量值，使灰度逐渐增加，直到达到图的实际数值。淡出过程则相反。淡入过程的子程序如下：

```
void fadein()
{
  / * 对调色板数据进行判断 * /
  outp(0x3c8,j); / * j是调色板入口索引号 * /
  outp(0x3c9,bufpal[i]); / * bufpal[]是调色板数据 * /
  outp(0x3c9,bufpal[i+1]);
  outp(0x3c9,bufpal[i+2]);
}
```

表 1 颜色查找表

红色分量	绿色分量	蓝色分量
100011	011000	100100
111000	001001	100111
⋮	⋮	⋮
010010	101010	000011
110001	101100	000111

五动画技术

动画有局部动画和全屏动画两种，本文介绍局部动画。高分辨率下的动画因为是在位面结构下，实现起来过程复杂，速度慢，效果较差。而 13H 模式下屏幕与显存是线性的映射关系，可以较方便地采用直接写屏技术从而大大提高了处理速度，动画效果很好。

13H 下的显示内存起始地址是 A000:0000，为了在屏幕某处画出某种颜色的点，只需按公式(2-1)计算出该点在内存中相应的地址，并写入其颜色索引号即可。按特定的规律在显示内存中写入并移动数据就能实现所需要的动画效果。以下的子程序实现局部图象从上向下平移的效果。

```
void blockmove()
{
fseek(fp,0x12+768+65078,SEEK SET); / * 动画部分图形的宽度 * /
fread(s1,sizeof(char),2,fp);
width＝s1[1]＊256+s1[0];
fseek(fp,0x16+768+65078,SEEK SET); / * 动画部分图形的高度 * /
fread(s2,sizeof(char),2,fp);
height＝s2[1]＊256+s2[0];
buff＝(char far ＊)farmalloc((unsigned long)height＊(unsigned long)width);
if (buff＝＝NULL)
    {
    printf ("Error to alloc memory.");
    exit(1);
    }
fseek (fp,0x36+256＊4+65078+768,SEEK SET);
fread (buff,sizeof(char),(long)height＊(long)width,fp);
pp＝buff;
for (h＝0;h< height;h+＝1)
    {
    pp＝buff;
    y++;
     for(i＝0;i< h;i++)
        {
        movedata(FP SEG(pp),FP OFF(pp),0xa000,(y-i)＊32(
+x,width);
        pp+＝width;
        }
    }
}
```

六、应 用

笔者采用本文所述方法为多个软件制作了封面，效

果良好。

1.图片的准备

找到所需要的背景图并将其备份，因为需要在图上加汉字，而自行生成的汉字在低分辨率下的效果很差，为此，我们在 3D Studio 中借用矢量字库构造出汉字，然后放样得到立体汉字，赋予其材质并加上灯光等场景，以所选的图为背景进行着色，得到将来的最终效果图。为了使汉字部分从屏幕顶部徐徐拉下，将需要进行动画处理的部分从该图中裁剪下来，我们将使用此动画部分图和原来的背景图生成动画效果的软件封面。为了编程的方便，文中将数据重新整理成如下的结构：调色板数据+背景宽度+背景高度+背景图形数据+动画部分宽度+动画部分高度+动画部分图形数据。

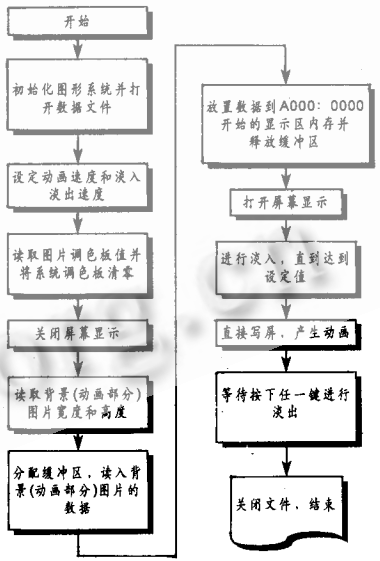
2.程序流程图

软件采用 Borland C 编程，显示卡为 TVGA，在 PC386 和 HP486 等机型上通过了测试。篇幅所限，本文不给出全部源程序，仅给出详细的程序实现流程图。

3.说 明

(1)背景和动画部分的图调色板必须一致，否则画面颜色会发生混乱。

(2)本文采用了 256 色的图形驱动程序对系统进行初始化。由于 Borland C／C ++本身没有这一驱动程序，因此可以通过中断调用将系统初始化为 13H 模式，而程序中因为采用了直接写屏技术，所以并不需要与图形有关的函数支持。



流 程 图