

kingmoon

本人专注于: .Net领域各类技术: Winform Asp.Net WPF Sliverlight WCF等.同时也关注架构设计,敏捷开发,个人管理,职业发展领域

随笔 - 49, 文章 - 0, 评论 - 148, 阅读 - 48万


导航

[首页](#)  
[新随笔](#)  
[联系](#)  
[订阅](#)   
[管理](#)

|             |    |    |    |    |    |    |
|-------------|----|----|----|----|----|----|
| < 2011年4月 > |    |    |    |    |    |    |
| 日           | 一  | 二  | 三  | 四  | 五  | 六  |
| 27          | 28 | 29 | 30 | 31 | 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 |
| 1           | 2  | 3  | 4  | 5  | 6  | 7  |

公告

沪江小D在线词典

 点击查词

昵称: kingmoon  
园龄: 11年5个月  
粉丝: 194  
关注: 14  
[+加关注](#)

搜索

谷歌搜索

最新随笔

- 1.[原创]神马你那就叫毕设? 看哥的神毕设--插件系统
- 2.[东风哥系列之CLR VIR C#]\_\_CLR执行模型 (重口味不喜勿喷)
- 3.[原创]2011个人年末回顾与总结\_\_一个2B程序员的经历
- 4.[十万个为什么]\_1000条WPF规则教训总结\_持续更新
- 5.[十万个为什么]\_C#的Enum中Flags的用法
- 6.[十万个为什么]\_UNICODE和汉字的各种渊源
- 7.[手把手教你写WPF界面]1.界面布局技巧与原则篇
- 8.[FS1专题]\_FS1简介
- 9.[MVVM专题]\_Caliburn.Micro资料大整合
- 10.[十万个为什么]\_程序版本号

积分与排名

积分 - 55657  
排名 - 25960

随笔分类

- [A].C#基础知识(8)
- [B].MVVM专题(9)
- [C].WPF基础知识(6)
- [D].面向对象设计与分析专题(7)
- [E].软件设计师复习笔记(4)
- [F].数据库(1)

BMP文件结构

【转自网络】

BMP文件存储结构的格式可以在Windows中的WINGDI.h文件中找到定义。

BMP文件总体上由4部分组成，分别是位图文件头、位图信息头、调色板和图像数据，如表5-1所示。

表5-1 BMP文件的组成结构

|                                   |
|-----------------------------------|
| 位图文件头 (bitmap-file header)        |
| 位图信息头 (bitmap-information header) |
| 彩色表/调色板 (color table)             |
| 位图数据 (bitmap-data)                |

下面来详细看一下每个组成部分的细节。

1. 位图文件头 (bitmap-file header)

位图文件头 (bitmap-file header) 包含了图像类型、图像大小、图像数据存放地址和两个保留未使用的字段。

打开WINGDI.h文件，搜索"BITMAPFILEHEADER"就可以定位到BMP文件的位图文件头的数据结构定义。

```
typedef struct tagBITMAPFILEHEADER {  
    WORD    bfType;  
    DWORD   bfSize;  
    WORD    bfReserved1;  
    WORD    bfReserved2;  
    DWORD   bfOffBits;  
} BITMAPFILEHEADER, FAR *LPBITMAPFILEHEADER,  
    *PBITMAPFILEHEADER;
```

表5-2列出了tagBITMAPFILEHEADER中各字段的含义。

表5-2 tagBITMAPFILEHEADER结构

| 字段名    | 大小 (单位: 字节) | 描述                         |
|--------|-------------|----------------------------|
| bfType | 2           | 位图类别，根据不同的操作系统而不同，在Windows |

[H].十万个为什么(4)  
[J].FlexemStudio1(1)  
[K].开发者人生(2)

随笔档案

- 2012年9月(1)
- 2012年7月(1)
- 2012年1月(2)
- 2011年12月(1)
- 2011年9月(2)
- 2011年8月(2)
- 2011年7月(4)
- 2011年6月(4)
- 2011年5月(11)
- 2011年4月(21)

|                 |   |                |
|-----------------|---|----------------|
|                 |   | 中，此字段的值总为 ‘BM’ |
| bfSize          | 4 | BMP图像文件的大小     |
| bfReserve<br>d1 | 2 | 总为0            |
| bfReserve<br>d2 | 2 | 总为0            |
| bfOffBits       | 4 | BMP图像数据的地址     |

2. 位图信息头 (bitmap-information header)

位图信息头 (bitmap-information header) 包含了位图信息头的大小、图像的宽高、图像的色深、压缩说明图像数据的大小和其他一些参数。

打开WINGDI.h文件，搜索"tagBITMAPINFOHEADER"就可以定位到BMP文件的位图信息头的数据结构定义。

```
typedef struct tagBITMAPINFOHEADER{
    DWORD    biSize;
    LONG     biWidth;
    LONG     biHeight;
    WORD     biPlanes;
    WORD     biBitCount;
    DWORD    biCompression;
    DWORD    biSizelImage;
    LONG     biXPelsPerMeter;
    LONG     biYPelsPerMeter;
    DWORD    biClrUsed;
    DWORD    biClrImportant;
} BITMAPINFOHEADER, FAR *LPBITMAPINFOHEADER,
*PBITMAPINFOHEADER;
```

表5-3列出了tagBITMAPFILEHEADER中各字段的含义。

表5-3 tagBITMAPFILEHEADER结构

| 字 段 名   | 大小<br>(单<br>位:<br>字节) | 描 述   |
|---------|-----------------------|---|
| biSize  | 4                     | 本结构的大小，根据不同的操作系统而不同，在Windows中，此字段的值总为28h字节=40字节 |
| biWidth | 4                     | BMP图像的宽度，单位像素                                   |
|         |                       |   |

|                 |   |  |
|-----------------|---|--|
| biHeight        | 4 | 总为0  |
| biPlanes        | 2 | 总为0  |
| biBitCount      | 2 | BMP图像的色深，即一个像素用多少位表示，常见有1、4、8、16、24和32，分别对应单色、16色、256色、16位高彩色、24位真彩色和32位增强型真彩色 |
| biCompression   | 4 | 压缩方式，0表示不压缩，1表示RLE8压缩，2表示RLE4压缩，3表示每个像素值由指定的掩码决定                               |
| biSizeImage     | 4 | BMP图像数据大小，必须是4的倍数，图像数据大小不是4的倍数时用0填充补足  |
| biXPelsPerMeter | 4 | 水平分辨率，单位像素/m   |
| biYPelsPerMeter | 4 | 垂直分辨率，单位像素/m   |
| biClrUsed       | 4 | BMP图像使用的颜色，0表示使用全部颜色，对于256色位图来说，此值为100h=256                                    |
| biClrImportant  | 4 | 重要的颜色数，此值为0时所有颜色都重要，对于使用调色板的BMP图像来说，当显卡不能够显示所有颜色时，此值将辅助驱动程序显示颜色                |

3. 彩色表/调色板 (color table)

彩色表/调色板 (color table) 是单色、16色和256色图像文件所特有的，相对应的调色板大小是2、16和256，调色板以4字节为单位，每4个字节存放一个颜色值，图像 的数据是指向调色板的索引。

可以将调色板想象成一个数组，每个数组元素的大小为4字节，假设有一256色的BMP图像的调色板数据为：

调色板[0]=黑、调色板[1]=白、调色板[2]=红、调色板[3]=蓝...调色板[255]=黄

图像数据01 00 02 FF表示调用调色板[1]、调色板[0]、调色板[2]和调色板[255]中的数据来显示图像颜色。

在早期的计算机中，显卡相对比较落后，不一定能保证显示所有颜色，所以在调色板中的颜色数据应尽可能将图像中主要的颜色按顺序排列在前面，位图信息 头的biClrImportant字段指出了有多少种颜色是重要的。

每个调色板的大小为4字节，按蓝、绿、红存储一个颜色值。

打开WINGDI.h文件，搜索"tagRGBTRIPLE"就可以定位到BMP文件的调色板的数据结构定义。

```
typedef struct tagRGBQUAD {
    BYTE  rgbBlue;
    BYTE  rgbGreen;
    BYTE  rgbRed;
    BYTE  rgbReserved;
} RGBQUAD;
```

表5-4列出了tagRGBTRIPLE中各字段的含义。

表5-4 tagRGBTRIPLE结构

| 字 段 名           | 大小（单位：字节） | 描 述    |
|-----------------|-----------|--------|
| rgbBlue         | 1         | 蓝色值    |
| rgbGreen        | 1         | 绿色值    |
| rgbRed          | 1         | 红色值    |
| rgbReserv<br>ed | 1         | 保留，总为0 |

4. 位图数据 (bitmap-data)

如果图像是单色、16色和256色，则紧跟着调色板的是位图数据，位图数据是指向调色板的索引序号。

如果位图是16位、24位和32位色，则图像文件中不保留调色板，即不存在调色板，图像的颜色直接在位图数据中给出。

16位图像使用2字节保存颜色值，常见有两种格式：5位红5位绿5位蓝和5位红6位绿5位蓝，即555格式和565格式。555格式只使用了15 位，最后一位保留，设为0。

24位图像使用3字节保存颜色值，每一个字节代表一种颜色，按红、绿、蓝排列。

32位图像使用4字节保存颜色值，每一个字节代表一种颜色，除了原来的红、绿、蓝，还有Alpha通道，即透明色。

如果图像带有调色板，则位图数据可以根据需要选择压缩与不压缩，如果选择压缩，则根据BMP图像是16色或256色，采用RLE4或RLE8压缩算 法压缩。

RLE4是压缩16色图像数据的，RLE4采用表5-5所示方式压缩数据。

表5-5 RLE4压缩方法

| 方 案 | 1字节      | 2字节  | 3字节 | 4字节 | N字节 |
|-----|----------|------|-----|-----|-----|
| A   | 重复次<br>数 | 颜色索引 |     |     |     |

|   |     |                |      |      |         |
|---|-----|----------------|------|------|---------|
| B | 设为0 | 后面有效的<br>颜色索引数 | 颜色索引 | 颜色索引 | 颜色索引... |
|---|-----|----------------|------|------|---------|

假设有如下16色位图数据，共20字节，数据使用了RLE4压缩：

```
05 00 04 05 00 08 09 05 04 00 04 05 08 09 04 08 07 01 00 00
```

数据解压时首先读取05，因为05不等于0，所以选择A方案，根据A方案，05表示后面数据重复的次数，接着读取00，00表示有两个颜色索引，每个索引占4位，第一个像素在高4位，第二个像素在低4位，即在一个字节中低像素在高位，高像素在低位。05 00解压后等于00 00 0。

读取04，选择A方案，按照上面的操作解析，04是后面数据重复的次数，05是两个颜色索引，第3个颜色索引为5，第4个颜色索引为0。04 05解压后等于05 05。

读取00，选择B方案，读取08，08表示后面有效的颜色索引数。00 08解压后等于09 05 04 00。

读取04，选择A方案，按照上面的操作解析，04是后面数据重复的次数，05是两个颜色索引。04 05解压后等于05 05。

读取08，选择A方案，按照上面的操作解析，08是后面数据重复的次数，09是两个颜色索引。08 09解压后等于09 09 09 09。

读取04，选择A方案，按照上面的操作解析，04是后面数据重复的次数，08是两个颜色索引。04 08解压后等于08 08。

读取07，选择A方案，按照上面的操作解析，07是后面数据重复的次数，01是两个颜色索引。07 01解压后等于01 01 01 0。

读取00，选择B方案，读取00，00表示后面有效的颜色索引数，0表示无，即解压完一行数据。

综合上面的操作，解压后的数据为：

```
00 00 00 50 50 90 50 40 00 50 50 90 90 90 90 80 80 10 10 10
```

看上去和原来的数据大小一样，没有体现到压缩效果，这是因为上面的例子只选择了20字节数据，而且这20字节数据中重复的数据不多，使用RLE压缩 重复数据不多的数据时，有时可能压缩后的大小反而比原来的数据还大。其实一般情况下当数据比较多而且重复的时候，使用RLE压缩效果还是比较理想的。

RLE8的压缩方式可以参考上面的RLE4解压方法，惟一的区别是RLE8使用1个字节存放颜色索引，而RLE4使用4位存放颜色索引。

结合上面对BMP文件的分析，下面分别对256色和24位色的BMP图像进行十六进制分析，通过在十六进制编辑器中分析文件结构，能够增加分析文件的经验。

如图5-1和图5-2所示，分别为256色BMP图像cat2.bmp和24位色BMP图像cat1.bmp。其中cat2.bmp图像的分辨率为 200×153，文件大小为31 680字节。cat1.bmp图像的分辨率为200×150，文件大小为90 056字节。

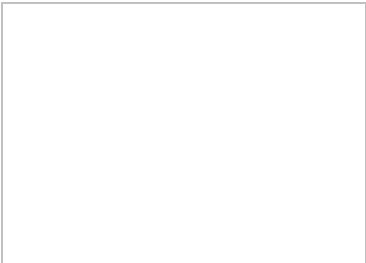


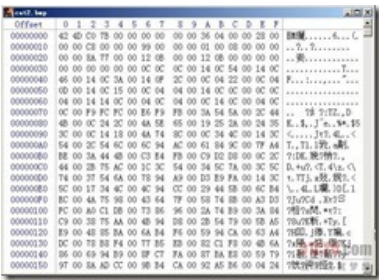


图5-1 cat2.bmp图像



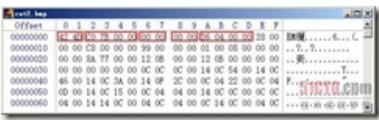
图5-2 cat1.bmp图像

现在来分析cat2.bmp的图像文件，在Winhex中打开cat2.bmp，如图5-3所示。



(点击查看大图) 图5-3 在Winhex中打开cat2.bmp图像文件

首先分析位图文件头的结构，如图5-4所示。根据 BMP文件的位图文件头结构定义分析出cat2.bmp图像的位图文件头中各字段的含义，如表5-6所示。



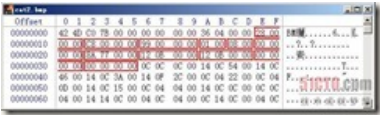
(点击查看大图) 图5-4 cat2.bmp图像文件的位图文件头

表5-6 cat2.bmp图像文件中位图文件头各字段的含义

| 十六进制值       | 描述                           |
|-------------|------------------------------|
| 42 4D:      | BM的ASCII值，在Windows中的BMP文件标识符 |
| C0 7B 00 00 | 7B C0h=31680，是cat2文件的大小      |

|             |                                    |
|-------------|------------------------------------|
| 00 00 00 00 | 保留值，总为0                            |
| 36 04 00    | 436h=1078，是图像数据的地址，即文件头+信息头+调色板的长度 |

继续分析接下来的数据，根据BMP文件结构的定义，接下来的数据是位图信息头，cat2.bmp图像文件的位图信息头的内容如图5-5所示。



(点击查看大图) 图5-5 cat2.bmp图像的位图信息头

表5-7所示为cat2.bmp图像文件中位图信息头各字段的含义。

表5-7 cat2.bmp图像文件中位图信息头各字段的含义

| 十六进制值        | 描 述                                     |
|--------------|---|
| 28 00 00 00: | cat2.bmp图像的位图信息头大小                      |
| C8 00 00 00  | 00 00 00 C8 = 200，是cat2图像的宽度，单位像素       |
| 99 00 00 00  | 00 00 00 99 = 153，是cat2图像的高度，单位像素       |
| 01 00        | 总是1                                     |
| 08 00        | 00 08 = 8，cat2图像的色深，即2的8次幂等于256色        |
| 00 00 00 00  | 压缩方式，0表示不压缩                             |
| 8A 77 00 00  | 00 00 77 8A = 30602，是cat2图像的图像数据大小，单位字节 |
| 12 0B 00 00  | 00 00 0B 12 = 2834，cat2图像的水平分辨率，单位像素/m  |
| 12 0B 00 00  | 00 00 0B 12 = 2834，cat2图像的垂直分辨率，单位像素/m  |
| 00 00 00 00  | cat2图像使用的颜色数，0表示使用全部颜色                  |
| 00 00 00 00  | cat2图像中重要的颜色数，0表示所有颜色都重要                |

继续分析接下来的数据，根据BMP文件结构的定义，因为cat2.bmp图像是256色的位图，所以应该有256个调色板，每个调色板占4字节，整个调色板一共1024字节大小。cat2.bmp图像文件的调色板数据如图5-6和图5-7所示。



(点击查看大图) 图5-6 cat2.bmp图像的调色板地址从00000036h开始存储



(点击查看大图) 图5-7 cat2.bmp图像的调色板数据结束地址是00000435h

从图5-6和图5-7中可以看出，cat2.bmp图像的调色板地址从00000036h开始到00000435h结束，即 $00000435h - 00000036h + 1 = 400h = 1024$ 。

如果想查看cat2图像的调色板对应的实际显示颜色，可以使用Adobe Photoshop CS打开cat2.bmp，在Adobe Photoshop CS的菜单栏中选择"图像"→"模式"→"颜色表"，即可观看cat2的调色板，如图5-8所示。

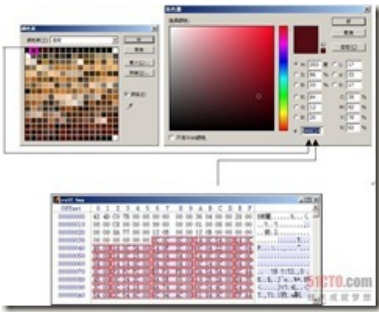


图5-8 在Adobe Photoshop CS中查看cat2的调色板

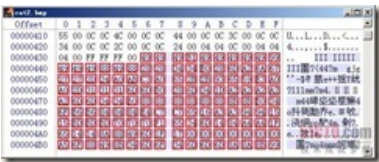
图5-8所示cat2.bmp的调色板颜色和图5-6中的十六进制数据是一一对应的。在Adobe Photoshop CS的调色板上单击任何一个像素的颜色即可弹出一个拾色器对话框显示该像素颜色的详细组成信息。cat2.bmp调色板和cat2.bmp的十六进制数据的对应关系如图5-9所示。

继续分析接下来的数据，根据BMP文件结构的定义，如果一个图像有调色板，那么紧跟在调色板后面的是图像的数据，这些数据不是实际的颜色值，而是指向调色板数组的索引，根据索引来获取调色板中的颜色，如图5-10所示。





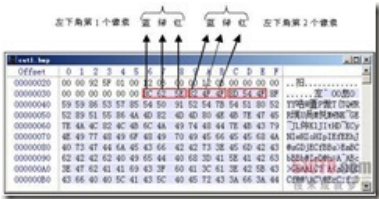
(点击查看大图) 图5-9 cat2.bmp调色板和cat2.bmp的十六进制数据的对应关系



(点击查看大图) 图5-10 cat2.bmp的图像数据

因为cat2.bmp是256色的位图，即采用了8位色深作为指向调色板数组的索引，所以根据图5-10中显示的数据可以得知：49 49 49 B1 49 49 49 49 99表示cat2.bmp位图左下角第1个像素的颜色等于调色板[49]，第2个像素的颜色等于调色板[49]，第3个像素的颜色等于调色板[49]，第4个像素的颜色等于调色板[B1].....依此类推。分析完cat2.bmp图像之后，接下来分析的是cat1.bmp。

cat1.bmp图像是24位色图像，根据BMP文件结构定义得知，cat1.bmp图像没有调色板，图像数据存储的是实际的颜色数据，每个像素用3字节表示，分别是红绿蓝。由于cat1.bmp和cat2.bmp的位图文件头和位图信息头结构一样，所以cat1.bmp的位图文件头和位图信息头可以参考上面对cat2.bmp的分析，下面从cat1.bmp的位图信息头结束的位置开始分析，如图5-11所示。



(点击查看大图) 图5-11 cat1.bmp图像的图像数据

从图5-11可以看到表示每个像素的红绿蓝三色的值，实际存放的时候是倒过来存放的，在分析BMP图像格式时需要注意这点。

通过上面对BMP文件存储结构的分析发现，BMP文件的位图文件头和位图信息头存在着大量的重复数据。如果存储大量同一色深的BMP位图，必然会浪费大量存储空间，所以很多时候游戏编程人员都会去掉BMP文件头和信息头，只保留几个必要的信息和图像数据，那么BMP文件头和信息头中哪几个字段是必须保留的呢？

使用Winhex的文件比较功能比较两个24位色深的BMP图像文件，观察两个文件的文件头和信息头有什么不同的地方，如图5-12所示。



(点击查看大图) 图5-12 使用Winhex比较两个24位色深的BMP图像文件

从图5-12可以看出，两个色深相同的BMP图像的文件头和信息头一共有4处不同的地方，分别是文件头的文件大小、信息头的图像宽度、图像高度和图像数据大小。

所以很多时候，游戏编程人员只保留图像文件的文件大小、图像宽度、图像高度和图像数据大小信息，甚至有时不需要保留文件大小这个数值，使用图像数据大小数值即可。

在分析未知文件存储格式时，如果遇到去掉了文件头的文件时，如上面所说的BMP文件，会给分析未知文件格式带来一定的困难。这时需要使用十六进制编辑器的文件比较功能，观察两个同类的未知文件格式寻找某些潜在的规律，如果实在观察不出规律的，那只能使用白盒分析方法，对调用此未知文件格式的程序进行反汇编跟踪调试了。当然，有时灵感和运气也很重要。

标签: [图像处理](#)

好文要顶

关注我

收藏该文

kingmoon

粉丝 - 194 关注 - 14

160

[+加关注](#)

« 上一篇: [参考文献格式\\_项目需要](#)  
» 下一篇: [论文格式控制系统](#)

posted on 2011-04-18 18:18 kingmoon 阅读(73800) 评论(8) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

- 编辑推荐:**
- [不安装运行时运行 .NET 程序 - NativeAOT](#)
  - [从 C# 崩溃异常 中研究页堆布局](#)
  - [C# 那些短命线程都在干什么?](#)
  - [我操作 MySQL 的惊险一幕](#)
  - [C# 内存泄漏之 Internal 关键词代表什么?](#)

- 最新新闻:**
- [背调下的“科技与狠活”：数据造假、角色演绎、控制员工](#)
  - [小鹏入局，Kittyhawk关停，飞行汽车是伪命题？](#)
  - [股价腰斩，零跑上市未脱险](#)
  - [抖音容不下辛吉飞吗？](#)
  - [国庆立的Flag，“倒”了吗？](#)
- » [更多新闻...](#)