

30 Января 2019г.21:36:45

Поиск

Powered by [На Главную](#)[Исходники](#)[Форум](#)[Информер](#)[Страны мира](#)

Статьи о Паскале >> Формат Bmp-файла

Формат Bmp-файла

 Сергей Андрианов

09.11.2001

Вторая статья из цикла "Основы спрайтовой анимации". Подробное описание структуры BMP файла и разработка программы считывания изображений (спрайтов, фона и элементов интерфейса) из BMP-файлов.



16k

[Мир ПК, #10/2001](#)

Постоянный адрес статьи: <http://www.osp.ru/pcworld/2001/10/099.htm>

Формат Bmp-файла

Сергей Андрианов

09.11.2001

В статье [«Основы спрайтовой анимации»](#) была рассмотрена небольшая программа, перемещающая спрайт по экрану, но, к сожалению, он при этом выглядел не так, как хотелось бы. В этой статье мы попробуем «привести» спрайт в порядок.

Изображение спрайта мы получили из Bmp-файла, из таких же файлов можно брать изображение фона, курсора мыши и элементов интерфейса. Однако на экране мы видим не совсем то, что ожидали: изображение оказалось перевернутым и к тому же с иными, нежели требовалось, цветами. Итак, научимся правильно считывать Bmp-файлы и перевернем картинку «с головы на ноги».

По решению разработчиков формат Bmp-файла не привязан к конкретной аппаратной платформе. Этот файл состоит из четырех частей: заголовка, информационного заголовка, таблицы цветов (палитры) и данных изображения. Если в файле хранится изображение с глубиной цвета 24 бита (16 млн. цветов), то таблица цветов может отсутствовать, однако в нашем, 256-цветном случае она есть. Структура каждой из частей файла, хранящего 256-цветное изображение, дана в [таблице](#), а соответствующие типы записей приведены в [листинге 1](#).

Заголовок файла начинается с *сигнатуры* «BM», а затем идет длина файла, выраженная в байтах. Следующие 4 байта зарезервированы для дальнейших расширений формата, а заканчивается этот заголовок *смещением* от начала файла до записанных в нем данных изображения. При 256 цветах это смещение составляет 1078 — именно столько и пришлось пропустить в нашей прошлой программе, чтобы добраться до данных.

Информационный заголовок начинается с собственной длины (она может изменяться, но для 256-цветного файла составляет 40 байт) и содержит размеры изображения, разрешение, характеристики представления цвета и другие параметры.

Ширина и высота изображения задаются в точках растра и пояснений, пожалуй, не требуют.

Количество плоскостей могло применяться в файлах, имеющих небольшую глубину цвета. При числе цветов 256 и больше оно всегда равно 1, поэтому сейчас это поле уже можно считать устаревшим, но для совместимости оно сохраняется.

Глубина цвета считается важнейшей характеристикой способа представления цвета в файле и измеряется в битах на точку. В данном случае она равна 8.

Компрессия. В Bmp-файлах обычно не используется, но поле в заголовке для нее предусмотрено. Обычно она равна 0, и это означает, что изображение не сжато. В дальнейшем будем использовать только такие файлы.

Размер изображения — количество байт памяти, требующихся для хранения этого изображения, не считая данных палитры.

Горизонтальное и вертикальное разрешения измеряются в точках растра на метр. Они особенно важны для сохранения масштаба отсканированных картинок. Изображения, созданные с помощью графических редакторов, как правило, имеют в этих полях нули.

Число цветов позволяет сократить размер таблицы палитры, если в изображении реально присутствует меньше цветов, чем это допускает выбранная глубина цвета. Однако на практике такие файлы почти не встречаются. Если число цветов принимает значение, максимально допустимое глубиной цвета, например 256 цветов при 8 битах, поле обнуляют.

Число основных цветов — идет с начала палитры, и его желательно выводить без искажений. Данное поле бывает важно тогда, когда максимальное число цветов дисплея было меньше, чем в палитре Bmp-файла. При разработке формата, очевидно, принималось, что наиболее часто встречающиеся цвета будут располагаться в начале таблицы. Сейчас этого требования практически не придерживаются, т. е. цвета не упорядочиваются по частоте, с которой они встречаются в файле. Это очень важно, поскольку палитры двух разных файлов, даже составленных из одних и тех же цветов, содержали бы их (цвета) в разном порядке, что могло существенно осложнить одновременный вывод таких изображений на экран.

За информационным заголовком следует таблица цветов, представляющая собой массив из 256 (по числу цветов) 4-байтовых полей. Каждое поле соответствует своему цвету в палитре, а три байта из четырех — компонентам синей, зеленой и красной составляющих для этого цвета. Последний, самый старший байт каждого поля зарезервирован и равен 0.

После таблицы цветов находятся данные изображения, которое по строкам растра записано снизу вверх, а внутри строки — слева направо. Так как на некоторых платформах невозможно считать единицу данных, которая меньше 4 байт, длина каждой строки выровнена на границу в 4 байта, т. е. при длине строки, некратной четырем, она дополняется нулями. Это обстоятельство обязательно надо учитывать при считывании файла, хотя, возможно, лучше заранее позаботиться, чтобы горизонтальные размеры всех изображений были кратны 4.

Как мы уже говорили, формат файла был разработан универсальным для различных платформ, поэтому нет ничего удивительного в том, что цвета палитры хранятся в нем иначе, чем принято для VGA. Во время выполнения процедуры чтения производится необходимая перекодировка. (О том, что представляет собой палитра VGA и как с ней работать, мы поговорим в следующих статьях.)

Модуль для чтения 256-цветных Bmp-файлов имеет всего две процедуры. Как видно из листинга, в процедуру чтения файла ReadBMP необходимо передать размеры изображения. Это удобно, если картинку нужно считывать не полностью. Когда заранее известны размеры, это не вызывает проблем, однако было бы хорошо, если бы с помощью нашего модуля можно было читать любые изображения, в том числе и такие, размер которых заранее неизвестен. Для этого предусмотрена процедура ReadBMPheader, считывающая только заголовок файла. Вызвав ее, можно проверить, записано ли изображение в выбранном 256-цветном формате, узнать его размеры и только потом выделять для него память и помещать в отведенный буфер.

Теперь подключим к нашей программе новый модуль. Для этого пропишем его имя в директиве uses, а также предусмотрим массив для хранения данных о палитре, который может быть описан так:

```
p: array[0..767] of byte;
```

Процедура CreateSprite, вызывающая операцию чтения файла из нового модуля, упростилась (см. [листинг 2](#)).

Что же изменилось на экране? Спрайт перевернулся, но по-прежнему имеет неестественные цвета. О том, как придать ему первоначальный вид, а также о некоторых возможностях, предоставляемых палитрой, читайте в [следующей статье](#).

Структура Bmp-файла

Имя	Длина	Смещение	Описание
-----	-------	----------	----------

Заголовок файла (BitmapFileHeader)

Type	2	0	Сигнатура "BM"
Size	4	2	Размер файла
Reserved 1	2	6	Зарезервировано
Reserved 2	2	8	Зарезервировано
OffsetBits	4	10	Смещение изображения от начала файла

Информационный заголовок (BitmapInfoHeader)

Size	4	14	Длина заголовка
Width	4	18	Ширина изображения, точки
Height	4	22	Высота изображения, точки
Planes	2	26	Число плоскостей
BitCount	2	28	Глубина цвета, бит на точку
Compression	4	30	Тип компрессии (0 - несжатое изображение)
SizeImage	4	34	Размер изображения, байт
XpelsPerMeter	4	38	Горизонтальное разрешение, точки на метр
YpelsPerMeter	4	42	Вертикальное разрешение, точки на метр
ColorsUsed	4	46	Число используемых цветов (0 - максимально возможное для данной глубины цвета)
ColorsImportant	4	50	Число основных цветов

Таблица цветов (палитра) (ColorTable)

ColorTable	1024	54	256 элементов по 4 байта
------------	------	----	--------------------------

Данные изображения (Bitmap Array)

Image	Size	1078	Изображение, записанное по строкам слева направо и снизу вверх
-------	------	------	--

Листинг 1

```
unit bmpread; {процедуры для работы с Bmp}
interface
type
  artype = array[0..0]of byte;
  arptr = ^artype;
  bmFileHeader = record {заголовок файла}
    Typef : word;      {сигнатура }
    Size : longint;    {длина файла в байтах}
```

```

    Res1 : word;      {зарезервировано}
    Res2 : word;      {зарезервировано}
    OfBm : longint;   {смещение изображения в байтах (1078)}
end;
bmInfoHeader = record {информационный заголовок}
    Size : longint;    {длина заголовка в байтах (40)}
    Widt : longint;    {ширина изображения (в точках)}
    Heig : longint;    {высота изображения (в точках)}
    Plan : word;       {число плоскостей (1)}
    BitC : word;       {глубина цвета (бит на точку) (8)}
    Comp : longint;    {тип компрессии (0 - нет)}
    SizI : longint;    {размер изображения в байтах}
    XppM : longint;    {горизонтальное разрешение}
                    {(точек на метр - обычно 0)}
    YppM : longint;    {вертикальное разрешение}
                    {(точек на метр - обычно 0)}
    NCoL : longint;    {число цветов}
                    {(если максимально допустимое - 0)}
    NCoI : longint;    {число основных цветов}
end;
bmHeader = record    {полный заголовок файла}
    f : bmFileHeader; {заголовок файла}
    i : bmInfoHeader; {информационный заголовок}
    p : array[0..255,0..3] of byte; {таблица палитры}
end;

bmhptr = ^bmHeader;

{чтение изображения из Bmp-файла}
procedure ReadBMP(image:arptr;      {массив с изображением}
                  xim,yim:word;      {размеры}
                  pal:arptr;         {палитра}
                  filename:string); {имя файла}

{чтение заголовка Bmp-файла}
procedure ReadBMPheader(header:bmhptr;filename:string);

implementation

{$R-}

{чтение изображения из Bmp-файла}
procedure ReadBMP(image:arptr; xim,yim:word;
                  pal:arptr; filename:string);
var
    h      : bmHeader;
    i      : integer;
    bmpfile : file;
    s      : longint;
begin
    assign(bmpfile,filename);
    reset(bmpfile,1);
    blockread(bmpfile,h,sizeof(h)); {чтение заголовка}
    for i := 0 to yim-1 do begin    {построчное чтение}
        blockread(bmpfile,image^[(yim-i-1)*xim],xim);
        if (xim mod 4) <> 0 then
            blockread(bmpfile,s,4 - (xim mod 4));
    end;
    close(bmpfile);
    for i ^:= 0 to 255 do begin      {преобразование палитры}
        pal^[i*3+2] := h.p[i,0] shr 2; {синий}
        pal^[i*3+1] := h.p[i,1] shr 2; {зеленый}
        pal^[i*3+0] := h.p[i,2] shr 2; {красный}
    end;
end;

{чтение заголовка Bmp-файла}
procedure ReadBMPheader(header:bmhptr;filename:string);
var
    bmpfile:file;
begin
    assign(bmpfile,filename);

```

```
    reset(bmpfile,1);  
    blockread(bmpfile,header^,sizeof(header^));  
    close(bmpfile);  
end;  
  
end.
```

Листинг 2

```
{<создание> спрайта}  
procedure CreateSprite(s:string; x,y,dx,dy:integer);  
var  
    f : file;      {файл с изображением спрайта}  
begin  
    getmem(Sprt.Img,sizeof(SpriteArrayType));  
    {выделяем память для спрайта}  
    getmem(Sprt.Back,sizeof(SpriteArrayType));  
    {выделяем память для буфера}  
    Readbmp(@(Sprt.Img^),Xsize,Ysize,@p,s);  
    Sprt.x := x;  
    Sprt.y := y;    {задаем начальные значения}  
    Sprt.dx := dx; {координат и приращений}  
    Sprt.dy := dy;  
end;
```

Мир ПК, #10/2001

Постоянный адрес статьи: <http://www.osp.ru/pcworld/2001/10/099.htm>