

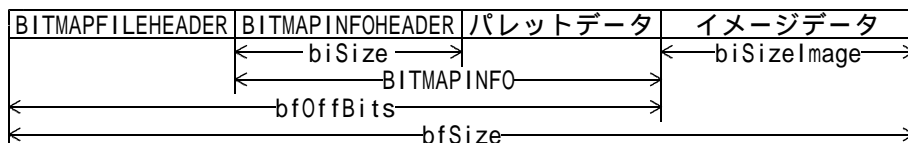
# オブジェクト指向と ゲームプログラミング

## コンポーネント編 - 第1回 ビットマップファイル

### ビットマップ

ビットマップとは、Windowsが標準でサポートしている画像フォーマットです。拡張子は「.bmp」です。ビットマップは、白黒(2色)から16色、256色、フルカラー(16,777,216色)までの色数をサポートしています。基本的にイメージをそのまま保存するので、ファイルサイズが大きくなってしまいます。RLE(ランレングス)という方法で圧縮することもできますが、正しく読み込めない場合もあります。

ファイルに保存されたビットマップは、以下のような形式になります。



ビットマップファイルの先頭には、BITMAPFILEHEADER構造体があります。この構造体には、ファイルの種類、サイズ、データの配置に関する情報が格納されています。先頭のメンバ(bfType)に、ファイルの種類が格納されています。ビットマップファイルの場合は、ここに'BM'の2文字(注:文字列ではないので'¥0'はありません)が格納されます。よって、ファイルの先頭2バイトが'BM'であるかどうかを調べれば、ビットマップファイルかどうかを調べることができます。bfSizeメンバにはファイルサイズ、bfOffBitsメンバにはイメージデータまでのバイト数(オフセット値)が格納されています。

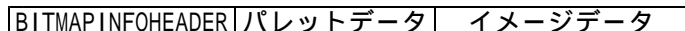
BITMAPFILEHEADER構造体の次に格納されているのがBITMAPINFOHEADER構造体です。この構造体には、カラー形式や寸法などが格納されています。特に重要なメンバがbiWidth, biHeight, biBitCountメンバです。

biWidth, biHeightメンバにはイメージの幅と高さが格納されています。biBitCountメンバには色数が格納されています。この値が1(モノクロ)、4(16色)、8(256色)の場合には、パレットデータを保有し、イメージデータはそのパレットデータのインデックス値となることを表しています。たとえば、4ビットカラービットマップでは、最大16個のパレットがあり、イメージデータは1ピクセルあたり4ビット(0~15の値)で構成されます。

16, 24, 32ビットのビットマップでは、イメージデータがそのままRGB値を表現しています。24ビットカラーのビットマップは、青、緑、赤の順でそれぞれ1バイト、計3バイトで1つのピクセルを表現します。

BITMAPINFOHEADER構造体の後にはRGBQUAD構造体型のパレットデータが続きます。パレットが存在する場合、BITMAPINFOHEADER構造体のbiClrUsedメンバの値がパレットの数を表していますが、この値に0が設定されている場合もあります。その場合には、biBitCountメンバでピクセルあたりのビット数により最大カラー数を求めて、それをパレットデータの数とします。

Windowsでは、ファイルのほかにリソースにもビットマップを保存することができます。リソースに保存した場合は、以下のように、ビットマップファイルからBITMAPFILEHEADER構造体を取り除いた形式になります。



・ BITMAPFILEHEADER構造体  
ファイルの種類、サイズ、データの配置に関する情報を保持します。

- メンバ -

WORD bfType.....'BM'の文字コード  
DWORD bfSize.....ファイルサイズ  
WORD bfReserved1 ... 予約されているので、0になります  
WORD bfReserved2 ... 予約されているので、0になります  
DWORD bfOffBits... ..イメージデータまでのバイト数

・ BITMAPINFO構造体

ビットマップの寸法と色の情報を保持します。

- メンバ -

BITMAPINFOHEADER bmiHeader .....次項を参照してください  
RGBQUAD bmiColors[1]...色を定義するRGBQUAD構造体型の配列

・ BITMAPINFOHEADER構造体

ビットマップの寸法とカラーフォーマットに関する情報を保持します。

- メンバ -

DWORD biSize.....構造体のサイズ  
LONG biWidth.....ビットマップの幅  
LONG biHeight.....ビットマップの高さ  
WORD biPlanes.....プレーンの数(ビットマップ形式では必ず1)  
WORD biBitCount... ..ピクセルあたりのビット数  
DWORD biCompression.....圧縮形式  
DWORD biSizeImage.....イメージのサイズ(バイト)  
LONG biXPelsPerMeter...ターゲットデバイスの水平解像度(ピクセル数/m)  
LONG biYPelsPerMeter...ターゲットデバイスの垂直解像度(ピクセル数/m)  
DWORD biClrUsed.....パレットの数  
DWORD biClrImportant ...表示において重要なパレットの数

- 備考 -

biClrUsedが0 ...biBitCountメンバの値をもとに、最大の色数を使用

biClrUsedが0以外かつbiBitCountが16未満...biClrUsedは実際の色数を示します

biClrUsedが0以外かつbiBitCountが16以上...Windowsのパレットテーブルのサイズを示します

biClrImportantが0 ...すべての色が重要

・ RGBQUAD構造体

赤、緑、青の相対的な輝度で構成される色を表します。

- メンバ -

BYTE rgbBlue.....青の輝度  
BYTE rgbGreen .....緑の輝度  
BYTE rgbRed .....赤の輝度  
BYTE rgbReserved...予約されています。0でなければなりません

256色ビットマップファイルの構造  
- 画像 (320 × 240) -



<p>ビットマップファイルヘッダ BITMAPFILEHEADER</p>	<pre>struct BITMAPFILEHEADER {     WORD    bfType      = 'BM' (0x424D)     DWORD   bfSize     = 77,878     WORD    bfReserved1 = 0     WORD    bfReserved2 = 0     DWORD   bfOffBits  = 1,078 };</pre>
<p>ビットマップインフォヘッダ BITMAPINFOHEADER</p>	<pre>struct BITMAPINFOHEADER{     DWORD   biSize      = 40     LONG    biWidth     = 320     LONG    biHeight    = 240     WORD    biPlanes    = 1     WORD    biBitCount  = 8     DWORD   biCompression = BI_RGB     DWORD   biSizeImage = 0または76,800     LONG    biXPelsPerMeter = 0     LONG    biYPelsPerMeter = 0     DWORD   biClrUsed   = 0または256     DWORD   biClrImportant = 0 };</pre>
<p>カラーテーブル RGBQUAD × 色数</p>	<pre>struct RGBQUAD {     BYTE  rgbBlue;     BYTE  rgbGreen;     BYTE  rgbRed;     BYTE  rgbReserved; };</pre> 
<p>イメージデータ 上下が反転しています 1ラインは4の倍数に合わせられるため、余分なデータが付加される場合があります。 256色以下の場合、右のような色値そのものではなく、色値に対応するパレットの番号が保存されます</p>	