

# Bitmapy - format i zastosowanie.

---

Podstawowy format plików  
bitmapowych, dogodność  
zastosowania bitmap w prostych  
animacjach 2D.

Format BMP został zaprojektowany przez firmę Microsoft, do przechowywania obrazów tzw. grafiki rastrowej, czyli zdjęć, wysokiej jakości tekstur i innych skomplikowanych obrazów.

### Zalety formatu BMP:

- ➔ prostota formatu
- ➔ duża szybkość przetwarzania skomplikowanych obrazów
- ➔ powszechność formatu na wielu platformach sprzętowych

### Wady formatu BMP:

- ➔ duża pamięciożerność
- ➔ mała lub brak kompresji obrazu wewnątrz pliku
- ➔ format BMP jest nierozpoznawalny przez HTML

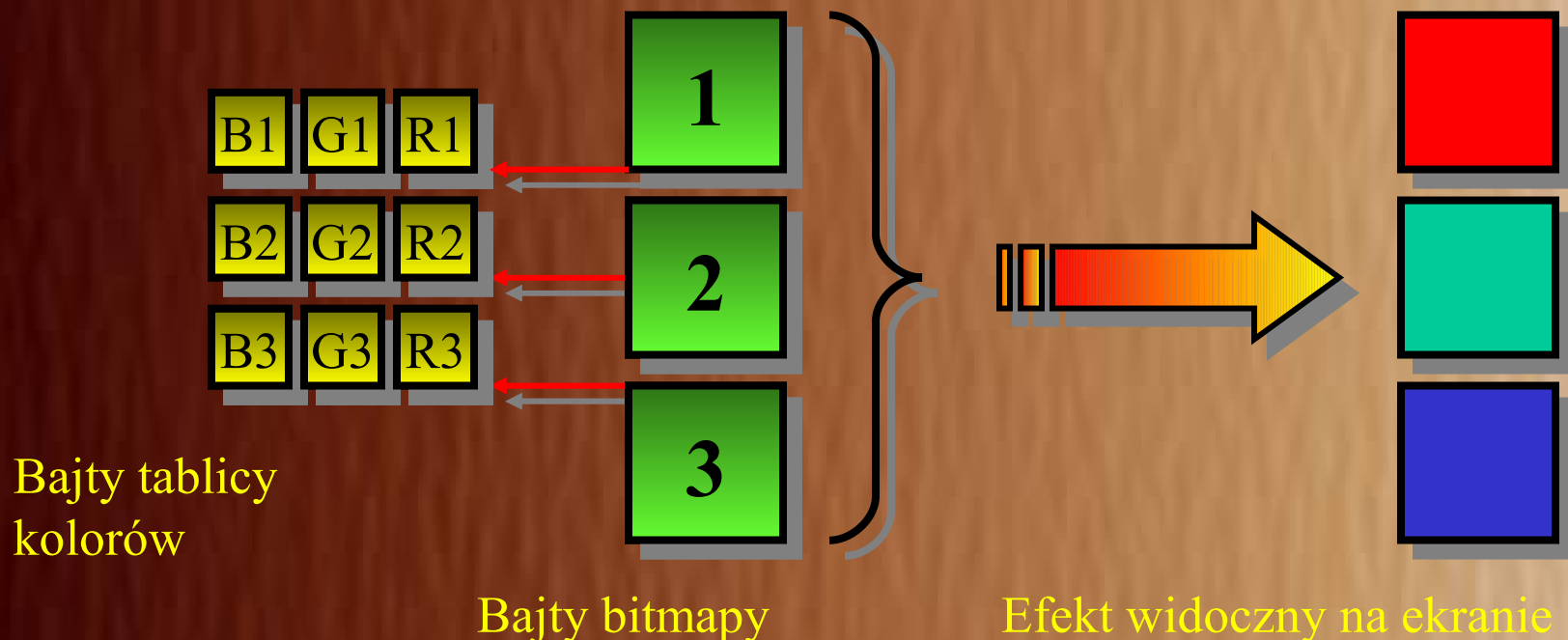


Istnieją cztery rodzaje plików BMP: pliki 1, 4, 8 i 24 bitowe.

Oznacza to, że kolor piksela obrazu w danym rodzaju pliku kodowany jest za pomocą wymienionej liczby bitów.

W przypadku bitmap 24 bitowych, każdy piksel kodowany jest trzema bajtami, odpowiadającymi bezpośrednio składowym czerwonej, zielonej i niebieskiej koloru piksela.

Dla pozostałych rodzajów bitmap każdy piksel bitmapy jest indeksem w tabeli kolorów będącej częścią pliku BMP. Dopiero wartość tabeli wskazywana przez ten indeks jest kolorem piksela widzianym na ekranie.



**Organizacja bitmapy w modelu 1, 4, 8 bitowym.**

Plik BMP składa się z czterech (lub trzech, dla bitmap 24 bitowych) części, zawierających informacje o parametrach bitmapy, używanych kolorach i wreszcie reprezentacji bitowej samego obrazu.



*Organizacja pliku BMP w modelu 1, 4, 8 bitowym.*

- Implementacja struktury pliku BMP może być wykonana w języku C++ w następujący sposób:
1. BITMAPFILEHEADER - struktura obejmująca parametry nagłówka pliku BMP
  2. BITMAPINFOHEADER - struktura obejmująca parametry bitmapy
  3. tablica kolorów zorganizowana jako tablica struktur RGBQUAD (tylko dla bitmap 1, 4, 8 bit)
  4. tablica bajtów odwzorowująca obraz w pikselach.

Definicje powyższych struktur wyglądają następująco:

```
struct BITMAPFILEHEADER
```

```
{  
    UINT    bfType;  
    DWORD   bfSize;  
    UINT    bfReserved1;  
    UINT    bfReserved2;  
    DWORD   bfOffBits;  
};
```

Dwa bajty identyfikujące typ pliku, wartość: "BM"

Całkowity rozmiar pliku w bajtach

Bajt zarezerwowany, zawsze zero

Bajt zarezerwowany, zawsze zero

Liczba bajtów od początku struktury BITMAPFILEHEADER do początku tablicy pikseli.

```
struct BITMAPINFOHEADER
{
    DWORD biSize;
    LONG   biWidth;
    LONG   biHeight;
    WORD   biPlanes;
    WORD   biBitCount;
    DWORD  biCompression;
    DWORD  biSizeImage;
    LONG   biXPelsPerMeter;
    LONG   biYPelsPerMeter;
    DWORD  biClrUsed;
    DWORD  biClrImportant;
};
```

Rozmiar struktury BITMAPINFOHEADER,  
powinno być zawsze 40

Szerokość bitmapy w pikselach

Wysokość bitmapy w pikselach

Wartość zawsze równa 1, można zignorować

Liczba bitów kodujących piksel : 1, 4, 8 lub 24

0 - brak wewnętrznej kompresji, w przeciwnym razie  
zastosowano wewnętrzną kompresję pliku BMP

Rozmiar obrazu bitmapowego w bajtach

Można zignorować

Można zignorować

Liczba indeksów w tablicy kolorów

Można zignorować

```
struct RGBQUAD
```

```
{
```

```
  BYTE rgbBlue;
```

```
  BYTE rgbGreen;
```

```
  BYTE rgbRed;
```

```
  BYTE rgbReserved;
```

```
};
```

**Składowa niebieska koloru**

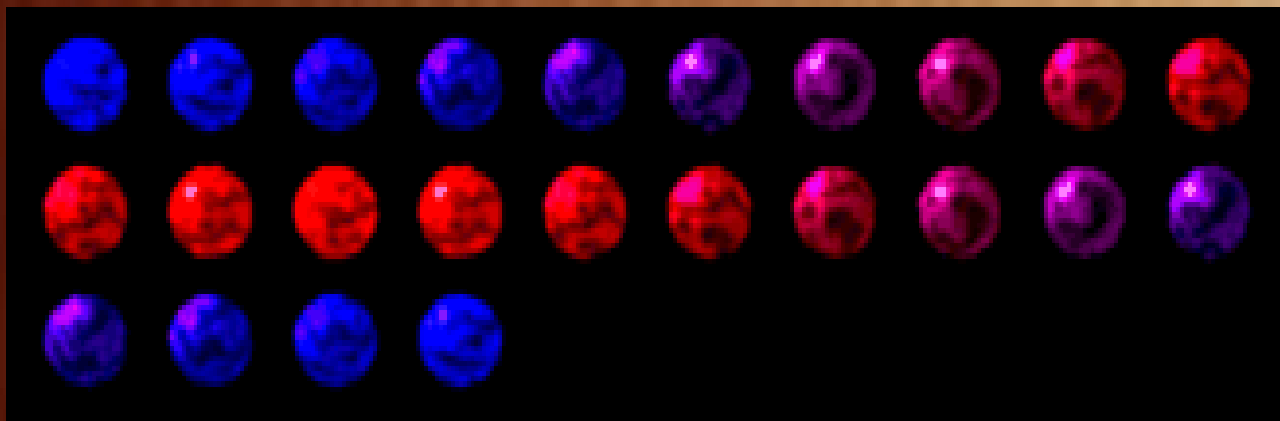
**Składowa zielona koloru**

**Składowa czerwona koloru**

**Bajt zarezerwowany**



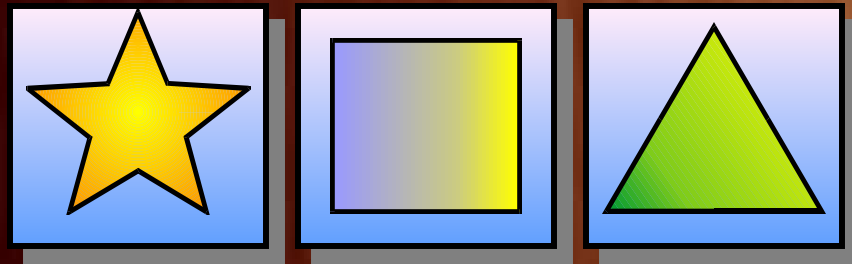
Proste animacje 2D można z powodzeniem realizować przy pomocy wcześniej przygotowanych faz animacji obiektów zapisanych w bitmapach. Bardzo popularną techniką animacji jest w tym przypadku technika podwójnego bufora (ang. Double Buffer). Polega ona na wcześniejszym przygotowaniu w pamięci pełnej ramki animacji i szybkim przepisaniu tak przygotowanej ramki do pamięci obrazu. Jeśli dysponujemy wcześniej przygotowanymi fazami animacji obiektów można nie tylko uzyskać efekt ruchu obiektu na ekranie, ale także zmiany jego kształtu.



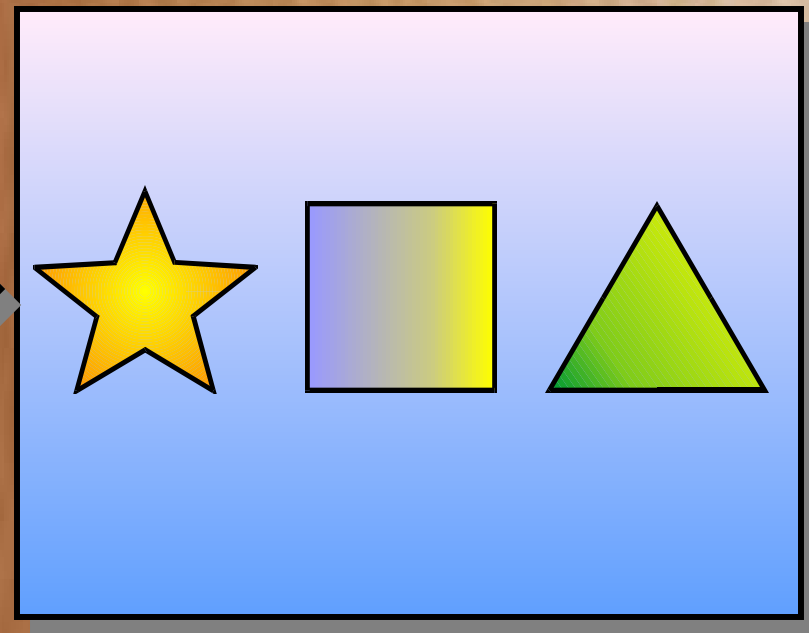
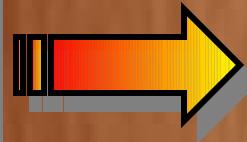
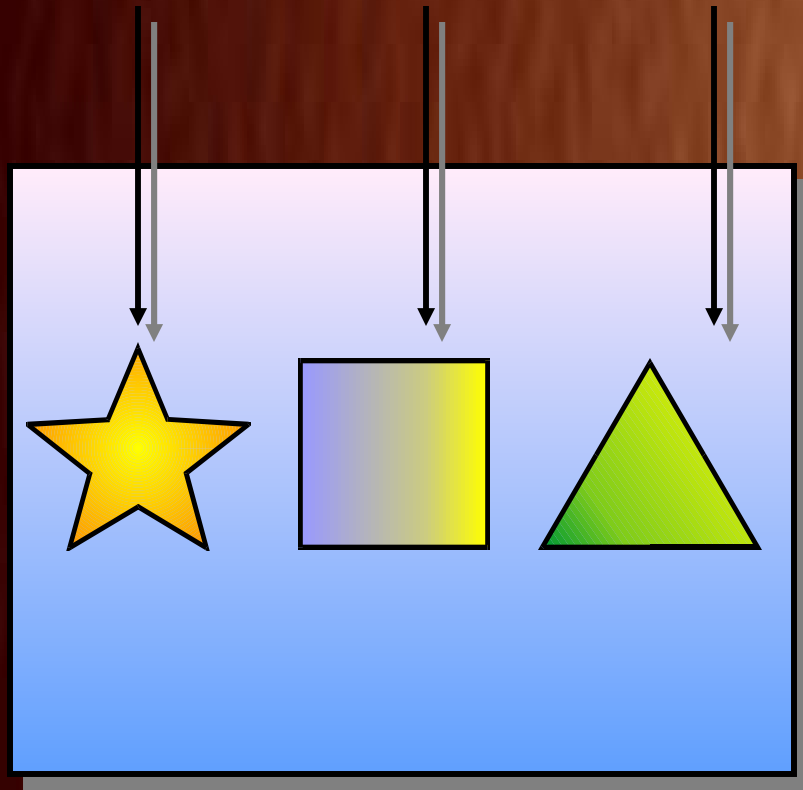
*Przykładowe fazy animacji teksturowanej pileczki.*



Zasadę działania techniki podwójnego buforowania ilustruje schemat:



Bufory w pamięci operacyjnej przechowujące kształty obiektów, które będą podlegały animacji.



**Bufor podwójny (Double Buffer)**

**Pamięć obrazu**

Dziękuję za uwagę.

---

Prelekcję przygotował  
Kamil Stokfiszewski