

Zadanie projektowe:

Należy utworzyć specyfikację rastrowego pliku graficznego rejestrującego obraz kolorowy (z wykorzystaniem 256 narzuconych oraz 256 dedykowanych barw) oraz w skali szarości (z wykorzystaniem 256 stopniowej skali szarości). Alfabet wejściowy to wartość 8 bitowa. Dane o obrazie zbierane i gromadzone są w sposób omówiony na spotkaniach projektowych.

W ramach projektu należy także napisać aplikację, które zgodnie ze stworzoną specyfikacją dokonają filtracji danych wejściowych (przystosowanie obrazka do alfabetu wejściowego z ewentualnym zastosowaniem ditheringu) i konwersji z pliku BMP do nowego rodzaju pliku graficznego oraz z nowego rodzaju pliku do formatu BMP.

Użytkownik powinien mieć możliwość wyboru jednego z trzech trybów koloru obrazka (paleta narzucona, odcienie szarości, paleta dedykowana) oraz czy obrazek ma zostać poddany ditheringowi.

KONWERTER

Konwerter na rastrowy typ graficzny DGB dokonuje filtracji danych wejściowych (przystosowania obrazka do alfabetu wejściowego) i konwersji z pliku BMP do pliku DGB oraz konwersji odwrotnej, tj. z formatu DGB do formatu BMP. Użytkownik ma możliwość wyboru czy obrazek ma zostać poddany ditheringowi (do wyboru tablica Bayer'a lub algorytm Floyd-Steinberga). Program został napisany i przetestowany w środowisku „Code::Blocks 20.03”.

SPECYFIKACJA

W pliku DGB można wyróżnić 3 podstawowe części:

1. Nagłówek - przechowuje metadane o obrazku
2. Paletę kolorów - przechowuje wykorzystywane przez obrazek kolory (część ta pojawia się jedynie w przypadku palety dedykowanej)
3. Dane dotyczące obrazu (w postaci optymalnej).

NAGŁÓWEK

Nagłówek jest wykorzystywany do identyfikacji pliku oraz do odczytania metadanych dotyczących obrazka. Dodatkowo w nagłówku zamieszczone są dane pozwalające na weryfikację czy obrazek nie jest uszkodzony. Na nagłówek pliku w formacie D68 składają się:

Nazwa	Offset	Rozmiar	Wartości i przeznaczenie
Identyfikator	00	2 bajty	'D6' służy do identyfikacji pliku
Szerokość obrazka	02	2 bajty	0-65535 Określa szerokość obrazka w pikselach
Wysokość obrazka	04	2 bajty	0-65535 Określa wysokość obrazka w pikselach
Tryb	06	1 bajt	1-5 Określa tryb obrazu: 1 - obraz z paletą narzuconą 2 - obraz w skali szarości narzuconej 3 - obraz w skali szarości dedykowanej 4 - obraz z paletą wykrytą 5 - obraz z paletą dedykowaną (MedianCut) Tryby 3,4,5 wymuszają konieczność pojawienia się po nagłówku (offset 08) palety dedykowanej
Dithering	07	1 bajt	0-2 0 - brak ditheringu 1 - dithering (tablica Bayera 4x4) 2 - dithering (Floyd-Steinberg)
Rozmiar danych	08 lub 768+08	4 bajty	0-4294967296 Określa rozmiar danych obrazu

PALETA KOLORÓW

Każdy piksel opisywany jest przy pomocy 8 bitów, które w zależności od trybu obrazu określają:

1. Wartość z palety narzuconej (zdefiniowanej globalnie dla wszystkich obrazów)
2. Wartość skali szarości (zdefiniowanej globalnie dla wszystkich obrazów)
3. Wartość (indeks) z palety dedykowanej (zdefiniowanej w nagłówku pliku)

PALETA NARZUCONA

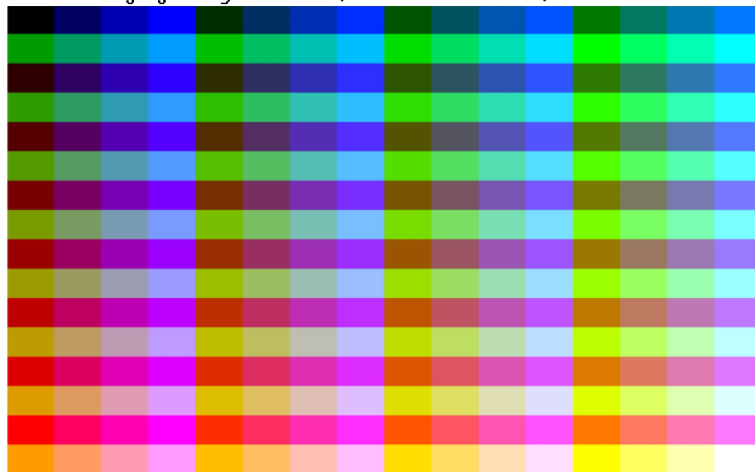
Paleta narzucona pozwala na uzyskanie 256 kolorów na które składają się:

8 możliwych stanów składowej czerwonej,

8 możliwych stanów składowej zielonej,

4 możliwe stany składowej niebieskiej.

Cała wartość (3 bity R, 3 bity G i 2 bity B) zapisywane są w jednym bajcie, gdzie składowa czerwona zajmuje bity od 5 do 7, zielona od 2 do 4, a niebieska od 0 do 1: RRRGGGBB



W przypadku składowej czerwonej i zielonej poszczególne wartości należy interpretować w następujący sposób:

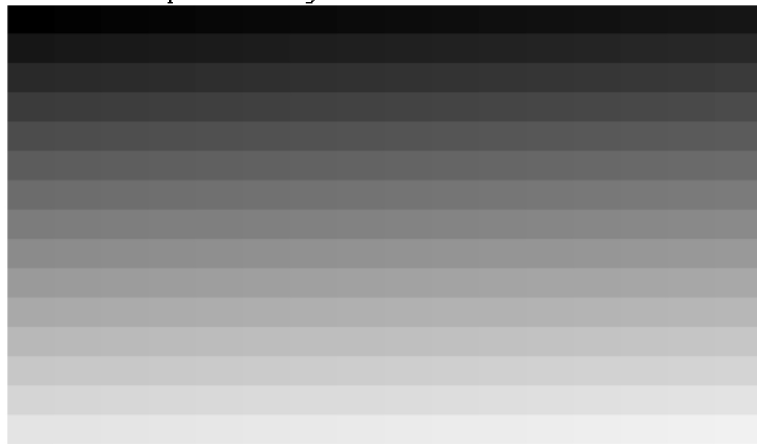
- 0 -> 0
- 1 -> 36
- 2 -> 73
- 3 -> 109
- 4 -> 146
- 5 -> 182
- 6 -> 218
- 7 -> 255

Wartość składowej niebieskiej należy interpretować w następujący sposób:

- 0 -> 0
- 1 -> 85
- 2 -> 170
- 3 -> 255

SKALA SZAROŚCI

Skala szarości pozwala na uzyskanie 256 odcieni szarości.



Wartość należy interpretować wprost jako intensywność odcienia szarości, gdzie 0 oznacza kolor czarny, a 255 kolor biały.

PALETA DEDYKOWANA

Paleta dedykowana to zestaw kolorów stworzony specjalnie dla konkretnego obrazka tak, aby zaproponowane barwy w jak największym stopniu pokrywały się z dominującymi kolorami danego obrazu. Szczególnym przypadkiem palety dedykowanej jest sytuacja, w której obrazek wykorzystuje nie więcej niż 256 kolorów, wtedy jako paletę dedykowaną możemy wykorzystać kolory wykorzystywane przez dany obrazek.

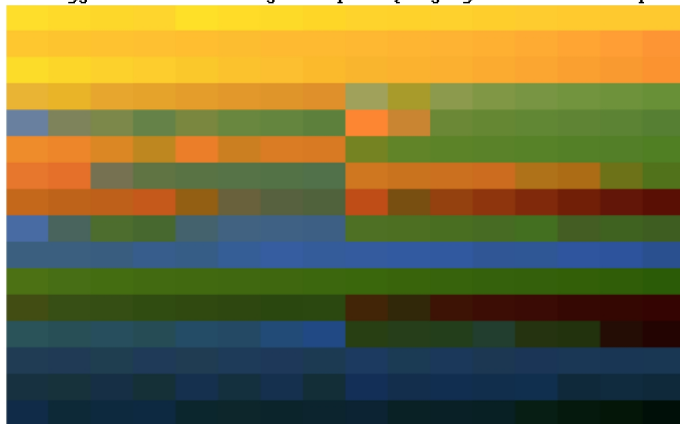
Wartość należy interpretować jako indeks do załączonej do pliku palety.

Paleta ta pojawia się zaraz za nagłówkiem (offset 12) i składa się z 256 zestawów RGB zajmujących $256 \times 3 = 768$ bajtów (pojedynczy kolor wykorzystuje 3 bajty, 1 bajt zawiera wartość składowej R, drugi składowej G, a trzeci składowej B).

Przykładowy obrazek



oraz wygenerowana dla niego (za pomocą algorytmu Median Cut) paleta 256 dedykowanych kolorów:



DANE OBRAZU

Po nagłówku od offsetu 12 (lub 780 w przypadku pliku z paletą dedykowaną) znajdują się dane obrazu. Dane te są zapisane w postaci nieskompresowanej.

Poszczególne wartości należy interpretować wprost według reguł określonych w części dotyczącej PALETY KOLORÓW.

Przykład:

Wartości heksadecymalne w pliku (od offsetu 12, paleta narzucona)

00 10 02 04 22 23

Należy interpretować jako:

00h → 00000000b → RGB(0, 0, 0)

10h → 00010000b → RGB(0, 146, 0)

02h → 00000010b → RGB(0, 0, 170)

04h → 00000100b → RGB(0, 36, 0)

22h → 00100010b → RGB(36, 0, 170)

23h → 00100011b → RGB(36, 0, 255)

Wartości heksadecymalne w pliku (od offsetu 12, skala szarości)

10 10 10 10 02 04 22 23

Należy interpretować jako:

10h → RGB(16, 16, 16)

10h → RGB(16, 16, 16)

10h → RGB(16, 16, 16)

10h → RGB(16, 16, 16)

02h → RGB(2, 2, 2)

04h → RGB(4, 4, 4)

22h → RGB(34, 34, 34)

23h → RGB(35, 35, 35)

KOLEJNOŚĆ ZBIERANIA DANYCH

Dane obrazu zbierane są określonej kolejności w blokach po 8 pikseli. Każdy blok 8 pikseli zajmuje 8 bajtów, gdzie pierwszy bajt to wartość pierwszego piksela, drugi – drugiego, itd.

Kolejność zbierania danych jest następująca:

Pierwszy blok danych zawiera 8 pikseli zbieranych od góry do dołu rozpoczynając od współrzędnej (0, 0), są to kolejno piksele o współrzędnych: (0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (0, 7). Kolejny blok danych zawiera piksele z drugiej kolumny, czyli rozpoczynając od współrzędnej (1, 0), są to kolejno piksele o współrzędnych: (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7). Kolejny blok danych zawiera piksele z trzeciej kolumny, itd. W momencie dotarcia do prawej krawędzi obrazu następny blok danych wraca do pierwszej kolumny obrazu i zawiera piksele rozpoczynające się od współrzędnej (0, 8), są to kolejno piksele o współrzędnych: (0, 8), (0, 9), (0, 10), (0, 11), (0, 12), (0, 13), (0, 14), (0, 15). Kolejny blok danych zawiera piksele z drugiej kolumny, czyli rozpoczynając od współrzędnej (1, 8), są to kolejno piksele o współrzędnych: (1, 8), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15). Kolejny blok danych zawiera piksele z trzeciej kolumny, itd. Po dotarciu do prawej krawędzi obrazu kolejny blok danych wraca do pierwszej kolumny obrazu i zawiera piksele z kolejnego 8-pikselowego wiersza, tj. zawiera piksele rozpoczynające się od współrzędnej (0, 16), są to kolejne piksele o współrzędnych: (0, 16), (0, 17), (0, 18), (0, 19), (0, 20), (0, 21), (0, 22), (0, 23). itd.