**Projekt nr 11**

**Korekty barwne**

Jan Baran

Jacek Rybak

Kacper Goraj

1 czerwca 2021

**1. Tytuł projektu i autorzy projektu**

Tytuł projektu to Korekty barwne.

Projekt został stworzony przez 3-osobowy zespół w składzie:

Jan Baran, Jacek Rybak, Kacper Goraj

**2. Opis projektu**

Celem projektu jest napisanie programu umożliwiającego wykonywanie selektywnych korekt barwnych wczytywanych plików graficznych. Zakresy zmian czyli proporcjonalność i mieszanie obrazu oryginalnego ze skorygowanym są regulowane suwakami. Dodatkowo istnieje możliwość dokonania korekt w kanałach jasności i nasycenia.

**3. Założenia wstępne przyjęte w realizacji projektu**

**Wymagania podstawowe**

* Program wczytuje i wyświetla plik graficzny na ekranie komputera
* Sześciokąt barw jest wyświetlany na ekranie z możliwością pobrania koloru
* Program dokonuje poprawnych korekt barwnych na pliku graficznym
* Suwaki proporcjonalności i mieszania obrazów odpowiednio zmieniają plik graficzny

**Wymagania rozszerzone**

* Korekty w kanale jasności
* Korekty w kanale nasycenia
* Wspomaganie korekt rysunkami odpowiednich histogramów, na których zaznacza się obszary korekty
* Wyświetlanie zmodyfikowanego ustawionymi korektami sześciokąta barw

**Dodatkowe**

* Możliwość zapisywania dokonanych zmian przez użytkownika

**4. Analiza projektu**

**4.1 Specyfikacja danych wejściowych**

Program umożliwia nam wczytanie pliku graficznego w formacie .jpg. Obraz zostaje wyświetlony w przeznaczonej do tego części na ekranie.

**4.2 Opis oczekiwanych danych wyjściowych**

Danymi wyjściowymi jest ten sam obraz co został wczytany tylko, że po dokonanych, przez użytkownika, korektach barwnych. Taką oto przetworzoną ilustracje można zapisać na swoim komputerze jako plik graficzny w formacie .jpg.

**4.3 Zdefiniowanie struktur danych**

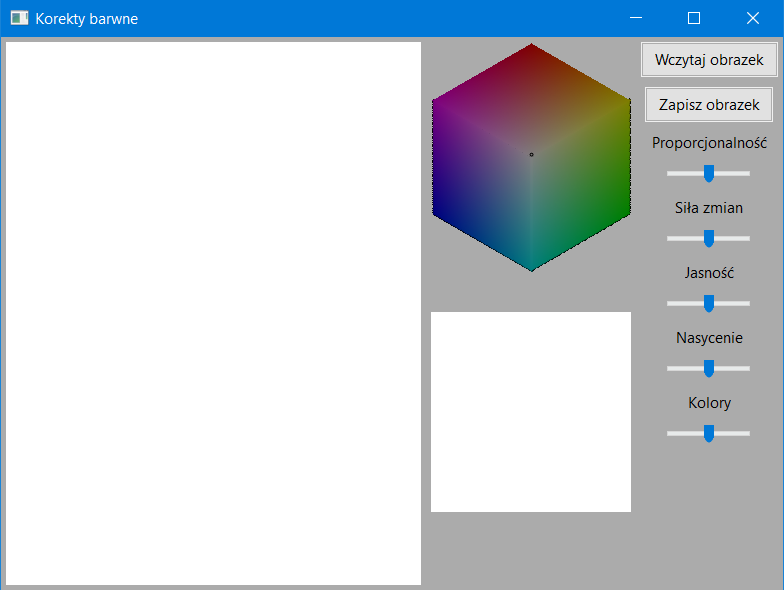
Obrazy są przetrzymywane przez wxImage natomiast zmiany ilustracji są przetrzymywane jako wxBitmap.

**4.4 Specyfikacja interfejsu użytkownika**

Interfejs użytkownika składa się z:

* Okna do przechowywania wczytanego pliku graficznego

Obraz dostosowuje się do rozmiarów okna. Z niego będą pobierane kolory z dowolnego miejsca oraz na nim będą widoczne korekty.



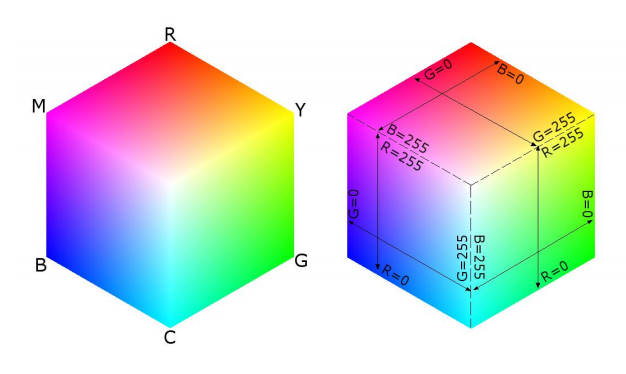
Rysunek 1. Okno użytkownika tuż po otwarciu programu.



Rysunek 2. Okno użytkownika programu po wczytaniu obrazka i zaznaczeniu kolory.

* Sześciokąt barw

Paleta kolorów z której użytkownik będzie wybierał na jaki kolor chce zmienić kolor na danym obrazku. Jest on w modelu RGB. Sześciokąt automatycznie dostosowuje swoją jasność aby pokazywać najbardziej odpowiedni kolor, który wybrał użytkownik na obrazku.



Rysunek 3. Sześciokąt barw

* Histogramów
* Okna które pokazuje jaki dokładnie kolor został wybrany przez użytkownika
* Suwaka proporcjonalności
* Suwaka mieszania obrazu oryginalnego ze skorygowanym
* Suwaka jasności
* Suwaka nasycenia
* Suwaka zamieniającego kolory na sześciokącie barw
* Przycisków wczytaj i zapisz obraz
* Okienka do zaznaczenia pod nazwą „Pokaż punkty”
* Suwaka regulującego zakres kolorów

**4.5 Wyodrębnienie i zdefiniowanie zadań**

Moduły projektu:

* Implementacja prostego GUI
* Stworzenie sześciokąta barw
* Wymyślenie i napisanie algorytmów potrzebnych do poprawnej korekcji barw
* Połączenie algorytmu z suwakami proporcjonalności i siły zmian
* Implementacja suwaków jasności i nasycenia

**4.6 Decyzja o wyborze narzędzi programistycznych**

Projekt został wykonany w języku C++ wraz z wykorzystaniem biblioteki wxWidgets. Biblioteka ta dała nam możliwość w szybki sposób stworzenia GUI za pomocą wxFormBuildier co ułatwiło nam pracę. Dodatkowo z tą biblioteką dobrze zapoznaliśmy się podczas wykładów i laboratoriów. Każda osoba z zespołu korzystała ze środowiska Microsoft Visual Studio 2019.

**5. Podział pracy i analiza czasowa**

Na początku uzgodniliśmy środowisko i dodaliśmy konieczne biblioteki. Następnie podzieliliśmy się równą pracą miedzy siebie. Oczywiście podczas wszelkich trudności konsultowaliśmy się ze sobą i wspólnie staraliśmy się rozwiązać problemy.

Zadania zrealizowane przez poszczególne osoby przedstawiają się następująco:

Jan Baran – GUI, dokumentacja – własne algorytmy, wczytywanie pliku graficznego,..

Jacek Rybak – GUI, dokumentacja – własne algorytmy,…

Kacper Goraj – GUI, dokumentacja projektu,…

**6. Opracowanie i opis niezbędnych algorytmów**

**Algorytm zmiany koloru**

**Algorytm do stworzenia sześciokąta barw**

**Algorytm do regulacji jasności**

Dodajemy (lub odejmujemy) tą samą wartość od kanału jasności: Inew = Iold + brigthess(jasność). Jeżeli wartości graniczne(czyli < 0 lub > 255) zostaną przekroczone to zmieniamy je na wartości graniczne.

**Algorytm do regulacji nasycenia**

Zwiększanie nasycenia kolorów można rozumieć jako oddalanie koloru danego piksela od poziomu szarości (czyli sytuacji, gdy składowe R,G i B są równe). Dla danego piksela poziom szarości obliczyliśmy jako średnią R,G, B, czyli (R+G+B)/3. Dodatkowo współczynnik modyfikacji nasycenia przedstawiliśmy w skali procentowej wartości położenia suwaka.

 //poziom szarości dla danego piksela

*int szary = (R + G + B) / 3;*

*// tmp - współczynnik modyfikacji nasycenia w skali procentowej*

*R = szary + (R - szary)\*(1.0 + tmp);*

*G = szary + (G - szary)\*(1.0 + tmp);*

B = szary + (B - szary)\*(1.0 + tmp);

**7. Kodowanie**

Klasy w projekcie wraz z ich metodami i atrybutami:

MyFrame1 - klasa

GUIMyFrame1 - klasa pochodna klasy MyFrame1

* void m\_panel3OnLeft( wxMouseEvent& event );
* void m\_panel3OnUpdateUI( wxUpdateUIEvent& event );
* void m\_panel4OnClick( wxMouseEvent& event );
* void m\_panel4OnUpdateUI( wxUpdateUIEvent& event );
* void m\_panel5OnUpdateUI( wxUpdateUIEvent& event );
* void m\_button1OnButtonClick( wxCommandEvent& event );
* void m\_button2OnButtonClick( wxCommandEvent& event );
* void m\_slider1OnScroll( wxScrollEvent& event );
* void m\_slider1OnScrollChanged( wxScrollEvent& event );
* void m\_slider2OnScroll( wxScrollEvent& event );
* void m\_slider2OnScrollChanged( wxScrollEvent& event );
* void m\_slider3OnScroll( wxScrollEvent& event );
* void m\_slider3OnScrollChanged( wxScrollEvent& event );
* void m\_slider4OnScroll( wxScrollEvent& event );
* void m\_slider4OnScrollChanged( wxScrollEvent& event );
* void m\_slider5OnScroll( wxScrollEvent& event );
* GUIMyFrame1( wxWindow\* parent );
* wxImage ImageOrg; - orginalny obraz który wcztuje użytkownik
* wxImage MyImage; - kopia obrazy orginalego
* wxImage ImageCpy; - druga kopia obrazy orginalego
* wxBitmap MyBitmap; -bitmapa przechowująca tymczasowo obraz
* wxColour ChosenColour; – kolor wybrany przez użytkownika, zapisany w modelu RGB
* Hexagon\* hexagon; - wskaźnik na klasę Hexagon
* int bright = 0; - zmienna pokazująca aktualną wartość jasności na suwaku
* bool bright\_flag = false; - flaga jasności
* bool sat\_flag = false; - flaga nasycenia
* bool zmiana\_flag = false; - flaga siły zmian
* double sat = 0; zmienna pokazująca aktualną wartość nasycenia na suwaku
* bool CC = 0; -…
* double zmiana = 0; zmienna pokazująca aktualną wartość siły zmian na suwaku
* void Brightness(int value); - funkcja zmieniajaca jasnosc obrazka o podany jako parametr współczynnik
* void Saturation(double value); - funkcja zmieniajaca nasycenie obrazka o podany jako parametr współczynnik
* void SilaZmian(double value); - funkcja zmieniajaca siłę zmian obrazka oryginalnego ze zmienionym o podany jako parametr współczynnik
* void DrawPicture(int bright,double sat); - funkcja rysująca oktualny obrazek
* void DrawColour();
* void ChangeColour(wxImage \*Image);- funkcja zmieniająca kolor obrazu

**8. Testowanie**

Etapy testowania przebiegały równorzędnie z implementacjami poszczególnych etapów. Pozwoliło to nam na szybką reakcję i niwelowanie pojawiających się błędów. W trakcie prac oraz na końcu po zbudowaniu projektu dokonywaliśmy testów z przykładowymi obrazkami ściągniętymi z Internetu z różnorodnymi kolorami. Obrazki testowe zostały załączone do projektu.

**9. Wdrożenie, raport i wnioski**

**Co się udało:**

W projekcie zostały spełnione wszystkie wymagania podstawowe. Użytkownik może swobodnie wczytać plik graficzny i edytować jego kolory. Wygodę w celu uzyskania zamierzonego efektu poprawiają suwaki regulujące proporcjonalność zmian oraz ich siłę. Dodatkowo stworzony histogram też spełnia swoje oczekiwania. Jeśli chodzi o wymagania rozszerzone to udało nam się zaimplementować suwaki umożliwiające dokonania korekt w kanałach jasności i nasycenia.

**Co się nie udało:**

Z wymagań rozszerzonych nie udało nam się stworzyć rysunków odpowiednich histogramów, na których zaznacza się obszary korekt. Wyświetlany sześciokąt barw nie modyfikuje się według ustawionych korekt.

**Co można poprawić w przyszłości:**

W przyszłości można by dodać możliwość wycinania pewnych części obrazu i wklejaniu innych albo też dodanie animacji lub ruszania się elementów na obrazie. Narzuca się również aby dodać możliwości regulowani kontrastu barw.