

Korekty barwne

Projekt nr 11

Mateusz Cichostępski

Jakub Wójcik

Mikhail Lemiasheuski

Czerwiec 2022

Spis treści

1. Tytuł i autorzy projektu	3
2. Opis projektu	3
3. Założenia wstępne przyjęte w realizacji projektu	3
4. Analiza projektu	4
4.1 Specyfikacja danych wejściowych	4
4.2 Opis oczekiwanych danych wyjściowych	4
4.3 Zdefiniowanie struktur danych	4
4.4 Specyfikacja interfejsu użytkownika	4
5. Podział pracy i analiza czasowa	4
6. Opracowanie i opis niezbędnych algorytmów	4
7. Kodowanie	5
8. Testowanie	7
9. Wdrożenie, raport i poprawki	8

1. Tytuł i autorzy projektu

Tytuł projektu to “Korekty barwne”, jest to projekt nr 11. Autorami są:

- Mateusz Cichostępski
- Jakub Wójcik
- Mikhail Lemiasheuski

2. Opis projektu

Program posiada funkcje wczytywania oraz wyświetlania na ekranie pliku graficznego. Użytkownik ma możliwość pobrania koloru poprzez kliknięcie w dowolnym miejscu obrazka. Pobrany kolor zostaje umieszczony na pasie barw. Po wskazaniu na “color-picker” nowego koloru użytkownik ma możliwość określić współczynnik proporcjonalności oraz siłę korekcji. Na podstawie tych wartości jest wyznaczany zakres edytowanych pikseli oraz proporcję mieszania obrazu. Użytkownik również może pobrać zmodyfikowany obraz.

3. Założenia wstępne przyjęte w realizacji projektu

Ze względu na to, że wszyscy członkowie zespołu posiadają doświadczenie programowania w bibliotece React.js zdecydowaliśmy zrezygnować z użycia wykorzystywanej w podczas zajęć laboratoryjnych biblioteki wxWidgets.

W wyniku powstała prosta aplikacja internetowa, realizująca podstawowe funkcje opisane w wymaganiach do projektu. Do stworzenia aplikacji, oprócz React.js, użyta została również biblioteka p5.js, dostarczająca narzędzia dla wygodniejszej pracy z obiektami graficznymi.

Program posiada funkcjonalności:

- Zapis oraz odczyt pliku
 - Format pliku: jpg, jpeg, bmp, png
- “Color picker” pozwalający na wybranie nowego koloru do modyfikacji
- “Color slider” na którym zaznaczony jest kolor modyfikowanego piksela

- Paski do regulacji:
 - “Proportionality factor” - współczynnik proporcjonalności modyfikowanej barwy
 - “Power of change” - regulacja sił zmian poprzez mieszanie obrazu oryginalnego ze skorygowanym

4. Analiza projektu

4.1 Specyfikacja danych wejściowych

Danymi wejściowymi są uploadowane przez użytkownika pliki w formacie .jpg, .jpeg, .bmp lub .png

4.2 Opis oczekiwanych danych wyjściowych

Na wyjściu programu użytkownik ma możliwość pobrania zmodyfikowanego obrazu w formatach .jpg, .jpeg, .bmp, .png.

4.3 Zdefiniowanie struktur danych

Podczas działania programu funkcjonalność programu przechowywana jest w komponencie *Canvas*. W nim znajdują się wszystkie funkcje i obiekty potrzebne do prawidłowego działania programu.

4.4 Specyfikacja interfejsu użytkownika

Interfejs użytkownika składa się z jednego okna w którym po prawej stronie wyświetlany jest modyfikowany obraz oraz po lewej wszystkie narzędzia pracy (suwaki, color-picker...)

5. Podział pracy i analiza czasowa

W celu optymalizacji pracy nad projektem zdecydowaliśmy się na przydzielenie każdemu z członków zespołu obowiązków, najbardziej odpowiadających jego umiejętnościom. Poza wspólnymi dyskusjami nad implementacją głównych funkcji (przekształcających obrazek) podział odpowiedzialności wyglądał następująco:

Jakub Wójcik	Opracowanie architektury projektu, podstawowy interfejs użytkownika, dokumentacja
Mateusz Cichostępski	Implementacja szablonu oraz głównych funkcji projektu
Mikhail Lemiasheuski	Opracowanie algorytmów, dokumentacja

6. Opracowanie i opis niezbędnych algorytmów

Po wybraniu przez użytkownika koloru z obrazka (poprzez kliknięcie w dowolnym miejscu) przetwarzany jest cały obrazek w celu zaznaczenia pikseli które:

- mają kolor taki sam jak wybrany piksel (składowe (R,G,B) są równe)
- są “odległe” od wybranego koloru nie więcej, niż o określony przez pierwszy suwak moduł,

*innymi słowy, jeżeli *max_diff* jest wartością określoną przez suwak a (R,G,B) - wartość wybranego koloru to edytowane są piksele, dla których spełnia się warunek:

$$\begin{aligned} |R - \text{piksel}.R| &\leq \text{max_diff} \\ \&\& |G - \text{piksel}.G| &\leq \text{max_diff} \\ \&\& |B - \text{piksel}.B| &\leq \text{max_diff} \end{aligned}$$

Algorytm łączący kolory w odpowiedniej proporcji (na podstawie przejrzystości nowego obrazu określonej przez drugi suwak) dla poszczególnego piksela wygląda następująco:

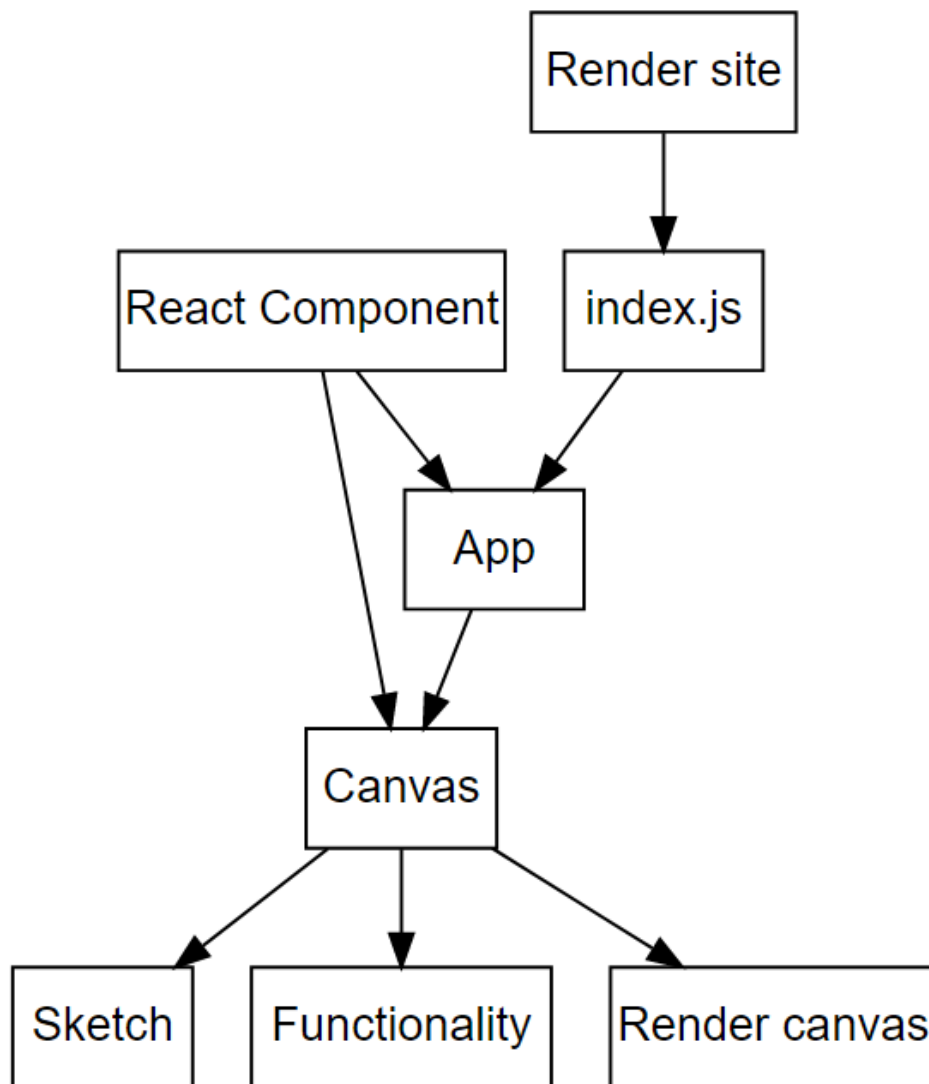
niech początkowe wartości składowe – (Xr,Xg,Xb) zaś zmodyfikowane- (Yr,Yg,Yb),
wtedy:

$$\begin{aligned} Xr &= Xr * (100 - \text{factor}) + Yr * \text{factor} \\ Xg &= Xg * (100 - \text{factor}) + Yg * \text{factor} \\ Xb &= Xb * (100 - \text{factor}) + Yb * \text{factor} \end{aligned}$$

gdzie: factor – parametr określający siłę korekcji

7. Kodowanie

Aplikacja została stworzoną na szablonie dostarczonym przez React.js. Głównym komponentem kodu jest Canvas(), który zwraca strukturę HTML, zawierającą wszystkie elementy aplikacji. Komponent ten zawiera w sobie implementacje pozostałych funkcji oraz wszystkie struktury danych.



Rysunek 1: Schemat blokowy działania aplikacji.

Canvas() - główny komponent zawierająca całą funkcjonalność, zwracająca strukturę HTML ze wszystkimi elementami aplikacji

Sketch(p) - klasa przechowująca funkcjonalności biblioteki P5.js

isSimilarPixel({R,G,B}, pixel, max_difference) – funkcja zwraca *true* , gdy żadna ze składowych wybranego piksela nie różni się od wartości odniesienia więcej, niż o wartość **max_difference** ($|pixel.R - R| \leq max_difference$ i t.d.) i *false* w przeciwnym wypadku.

numToRGBString(R,G,B) – funkcja zwracająca łańcuch zawierający informacje o wartości składowych (R,G,B) w postaci “rgb({R},{G},{B})”

changeProportionFactor(event) - funkcja przetwarzająca zdarzenie przesunięcia suwaka, odpowiadającego za zakres edytowanych pikseli.

changePowerChange(event) - funkcja przetwarzająca zdarzenie przesunięcia suwaka, odpowiadającego za siłę przekształcenia

setProportionFactor(N) – funkcja aktualizująca zakres zmian

setPowerChange(N)- funkcja aktualizująca parametr siły korekcji

preload() - funkcja dostarczana przez p5.js, odpowiada za załadowanie zdjęcia w sposób asynchroniczny. Wywołanie jej tuż przed **setup()** sprawia, iż funkcja **setup()** czeka na załadowanie wszystkich elementów.

setup() - funkcja wywoływana jest przy uruchomieniu programu w celu inicjalizacji własności środowiskowych (rozmiar ekranu, background itp.)

draw() - funkcja działająca w nieskończonej pętli do rysowania obrazu

8. Testowanie

Aplikacja została przetestowana manualnie. Każda funkcjonalność została sprawdzona po wdrożeniu. Testy zostały przeprowadzone na różnych obrazkach z sporą różnorodnością kolorów.

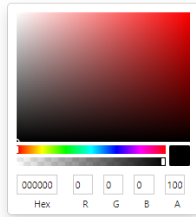
Podczas testowania sprawdzona została następująca funkcjonalność:

- kolor wybiera się poprzez kliknięcie w dowolnym miejscu po czym jest zaznaczony na color-picker'ze
- w celu wybrania nowego koloru trzeba skorzystać z tego samego color-picker'a (wybrany w poprzednim kroku kolor jest przechowywany w pamięci)
- po wybraniu nowego koloru obrazek zostaje zmodyfikowany na podstawie wartości domyślnych otoczenia oraz współczynnik do mieszania z kanałem alpha
- dla minimalnej wartości pierwszego suwaka zmieniają się wyłącznie piksele o takim samym kolorze, natomiast dla wartości maksymalnej edytowane są piksele o kolorach których żadna ze składowych (R,G,B) nie różni się od wybranego piksela więcej niż o 100.

1) Click on the image to choose pixel

Old color

2) Choose new color below



Proportionality factor

Power of change

Upload file

Save file

Reset image



Rysunek 2: Oryginalny obraz.

1) Click on the image to choose pixel

Old color

2) Choose new color below



Proportionality factor

Power of change

Upload file

Save file

Reset image



Rysunek 3: Obraz po przekształceniu z wskaźnikiem proporcji ustawionym na 100%.

9. Wdrożenie, raport i poprawki

Projekt został udostępniony na publicznej platformie Netlify, pod linkiem <https://korekty-barwne.netlify.app>.

Najbardziej wyraźną różnicą pomiędzy postacią naszego projektu a wymaganą jest brak sześciokąta barwnego. W zamian tego zaimplementowaliśmy “color picker” umożliwiający wybór nowego koloru oraz pasek barw, na którym się zaznacza wybrany z obrazka kolor “oryginalny”.

Poza tymi drobnymi niezgodnościami udało się nam, w naszej ocenie, w pełni zrealizować podstawową wersję projektu nr 11 “Korekty barwne”.