# Dokumentacja

**Temat projektu: 5. Podstawowe operacje na obrazie.**

## Wstęp

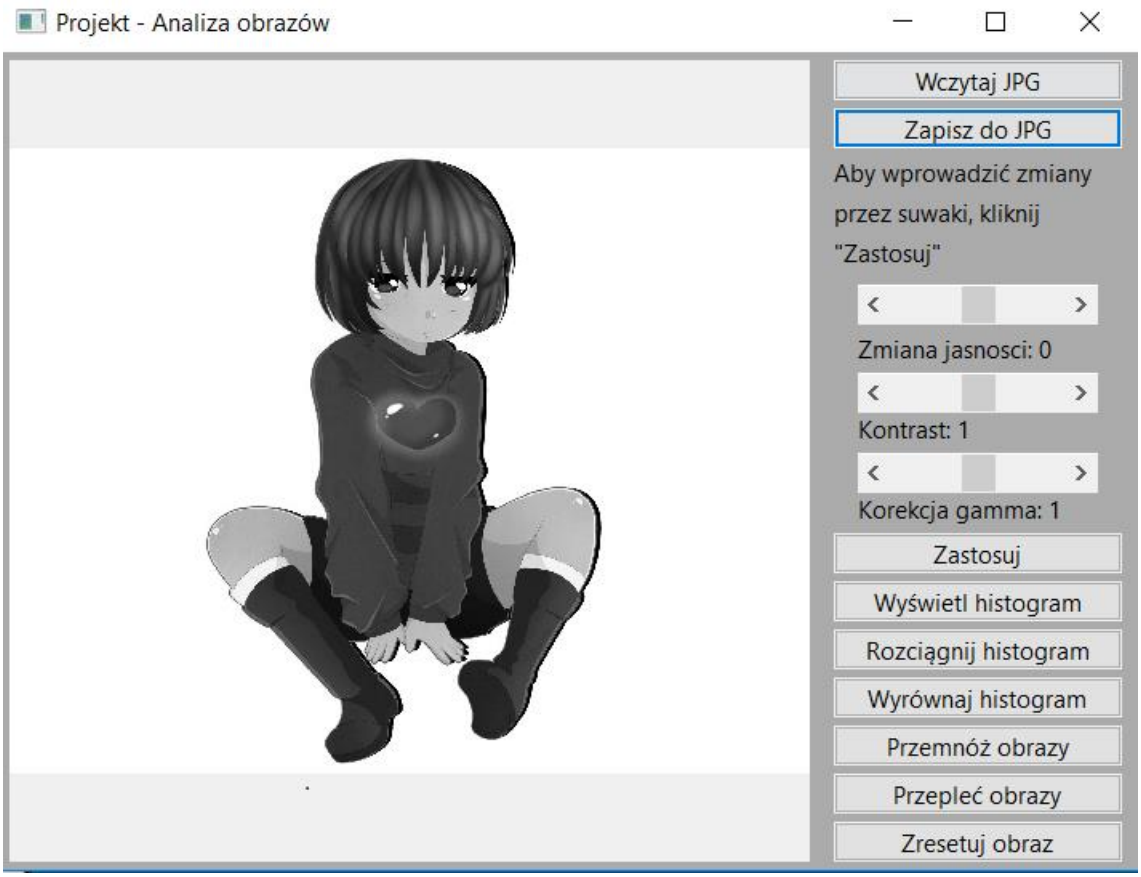
Program służy do przeprowadzania operacji na jednym lub dwóch obrazach, takich jak:
a) zmiana jasności
b) zmiana kontrastu
c) korekcja gamma
d) rozszerzenie histogramu
e) wyrównanie histogramu
f) mnożenie dwóch obrazów
g) przeplot dwóch obrazów.

## Wkład pracy

**Mateusz Sacha:** przygotowanie GUI i szkieletu programu, implementacja algorytmów.
**Maciej Domagalski:** przygotowanie i opracowanie algorytmów, implementacja algorytmów.

## Omówienie programu

Aby uruchomić aplikację, należy otworzyć plik *Projekt_Sacha_Domagalski.exe*. Po uruchomieniu ukaże się główne okno programu:



Najpierw należy wczytać jakiś obraz używając przycisku **Wczytaj JPG.** Zostanie on automatycznie przekonwertowany do skali szarości. Wtedy uaktywnią się pozostałe funkcje programu, a podgląd wprowadzonych zmian będzie można obserwować w oknie podglądu po lewej stronie.

Przycisk **Resetuj obraz** przywraca obraz oryginalny do stanu przed operacjami.

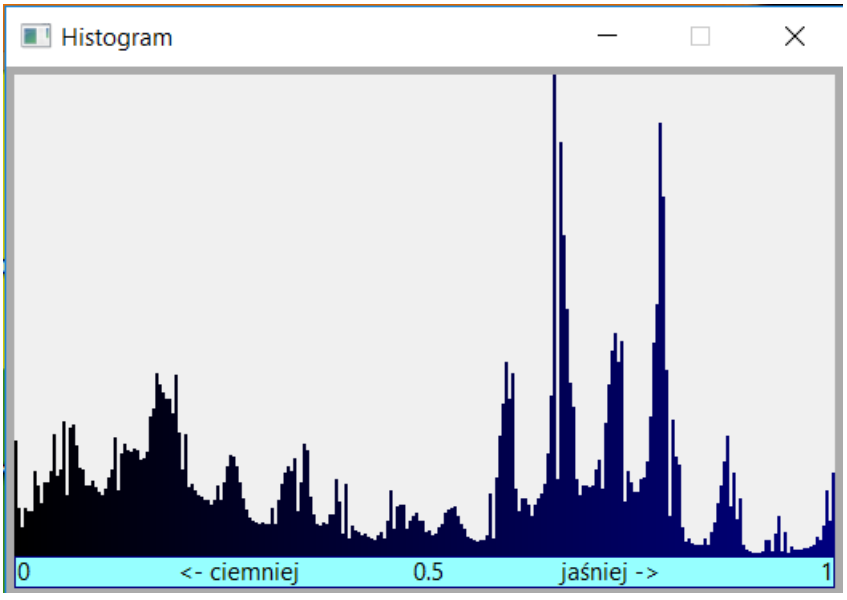
Po wykonaniu wszystkich potrzebnych zmian obraz można zapisać przyciskiem **Zapisz do JPG.**

Suwak **Zmiana jasności** służy do dobrania wartości $k$ (widać ją pod suwakiem) która zostanie dodana do obrazu, aby zmienić jego jasność.

Suwakiem **Kontrast** można ustawić wartość $k$ widoczną pod suwakiem, przez którą obraz zostanie przemnożony aby zmienić jego kontrast.
Suwak **Korekcja gamma** służy do ustawiania wartości gammy. Wartość pod suwakiem to $\frac{1}{\gamma}$. Jest ustalona w taki sposób, aby przesuwanie suwaka w prawo rozjaśniało obraz, a nie ściemniało.

Przycisk **Zastosuj** służy do potwierdzania zmian wprowadzonych przez suwaki. Efekt widać w oknie podglądu.

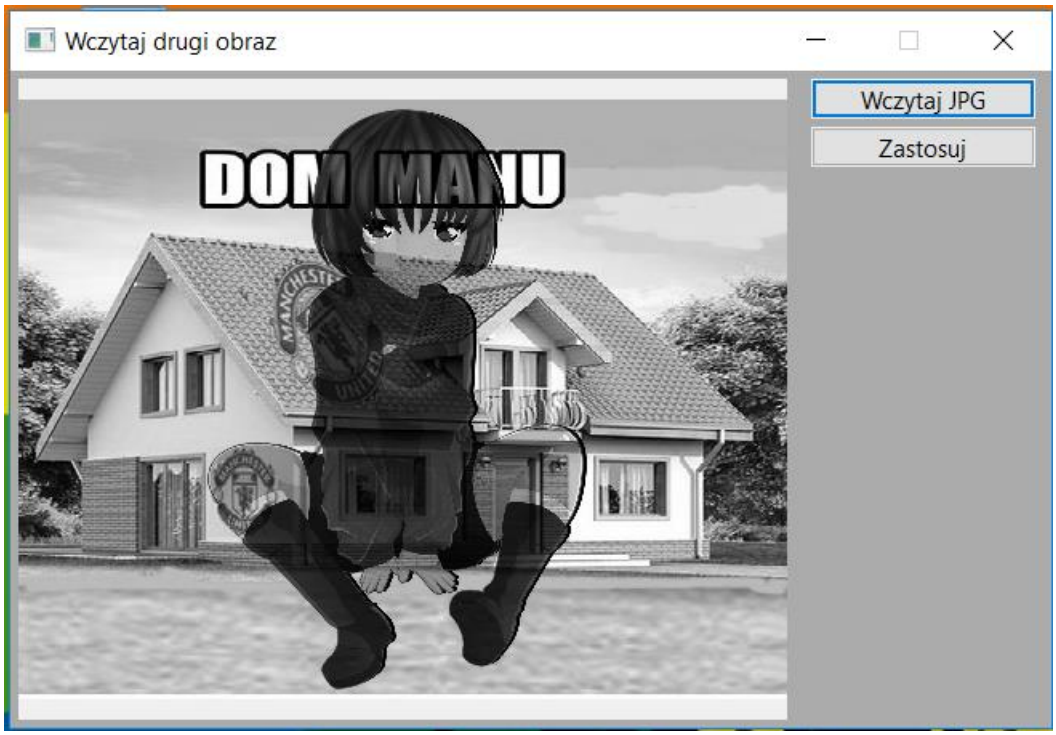
Przycisk **Wyświetl histogram** wyświetla histogram obrazu, który jest aktualizowany w locie po każdej operacji.




Przycisk **Rozciągnij histogram** normalizuje obraz tak, aby histogram zajmował całą szerokość od 0 do 1.

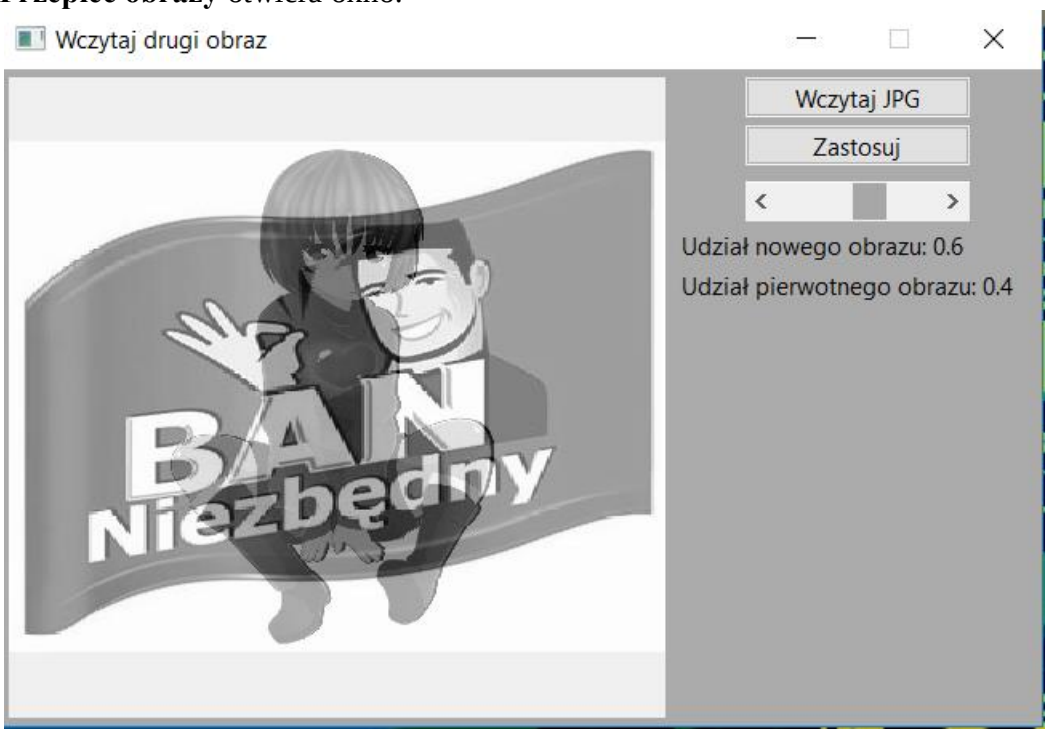
Przycisk **Wyrównaj histogram** modyfikuje obraz tak, aby jego histogram był możliwie równy.

Przycisk **Przemnóż obrazy** otwiera następujące okno:




Najpierw trzeba wczytać drugi obraz używając przycisku **Wczytaj JPG.** To wyświetli wynik mnożenia go z obrazem pierwotnym. Drugi obraz jest automatycznie przeskalowany, aby pasował do pierwszego.
Jeśli wynik odpowiada oczekiwaniom, należy kliknąć **Zastosuj** i wrócić do okna głównego. W przeciwnym przypadku przycisk **X** spowoduje powrót bez wprowadzania zmian.

Przycisk **Przepleć obrazy** otwiera okno:



Przycisk **Wczytaj JPG s**łuży do wybrania obrazu, który zostanie najpierw przeskalowany, a następnie przepleciony z oryginalnym. Domyślna wartość *k* wynosi 0,5.
**Suwak** służy do ustalenia wartości *k*, przez którą zostanie pomnożony obraz pierwotny. Automatycznie dobierana jest wartość *1-k*, przez który przemnożony zostanie nowy obraz przed dodaniem go do przygotowanego obrazu pierwotnego.
Przycisk **Zastosuj** zapisuje zmiany i wraca do okna głównego. Przycisk **X** spowoduje powrót bez zmian.

## Opis kodu źródłowego

Do zrealizowania projektu wykorzystano bibliotekę wxWidgets, gdyż pozwala ona na łatwe zaprogramowanie interakcji użytkownika z programem przy pomocy licznych kontrolek i przycisków, jak również dzięki dostarczonej przez niej obsługi obrazów i plików graficznych.

Pliki gui.cpp i gui.h zawierają klasę AOFrame. Dziedziczy ona po wxFrame (z biblioteki wxWidgets), co w praktyce oznacza, że reprezentuje ona pewne okno (w tym przypadku główne okno programu). Znajdują się w niej wszystkie kontrolki, takie jak przyciski, suwaki, wyświetlane teksty oraz panel, na którym wyświetlany jest obraz, jak również funkcje – handlery zdarzeń związanych z tymi kontrolkami.
W klasie znajdują się również dwa obiekty wxImage (z biblioteki wxWidgets) przechowujące dane o obrazach – jedna przechowuje oryginał (by móc cofnąć modyfikacje), druga kopię, która jest dowolnie modyfikowana. Wszelkie operacje na obrazach odbywają się dzięki możliwości bezpośredniego dostępu do danych pikseli tych obrazów (metoda GetData() klasy wxImage zwraca wskaźnik do tablicy wszystkich pikseli).
Klasa AOFrame zajmuje się również wyświetlaniem obrazu i dostosowaniem jego rozmiaru w zależności od rozmiaru okna, jak również zarządzaniem pozostałymi oknami (tworzeniem i usuwaniem ich).
Wszelkie operacje, które modyfikują zawartość obrazu znajdują się w plikach operations.cpp i operations.h. Znajdujące się tam funkcje przyjmują jako argument wspomniany wcześniej wskaźnik na tablicę pikseli, by móc dokonywać na niej zmian.

Poniżej opisane są wszystkie istotne funkcje:
a) grayScale() - operacja, która zmienia obraz z kolorowego, na obraz o odcieniach szarości, wyliczając wartość odcienia jako średnia ważona kanałów RGB.
b) setBrightness – operacja, która dodaje do każdej pikseli obrazu ustaloną wartość (wybraną przez użytkownika), w rezultacie zmieniając jego jasność.
c) setContrast() – operacja, która przemnaża każdy piksel o ustaloną wartość (wybraną przez użytkownika), w rezultacie zmieniając poziom kontrastu.
d) setGamma() - operacja, która każdy piksel podnosi do potęgi o ustalonym wykładniku (wybranym przez użytkownika), w rezultacie dokonując korekcji gamma.
e) normalizeHistogram() - operacja, która rozciąga histogram tak, by najjaśniejszy piksel przyjął maksymalną wartość (255 w zapisie bajtowym), a piksel najciemniejszy – wartość 0.
f) equalHistogram() - operacja, która wyrównuje histogram, by wartości pikseli były bardziej równomiernie rozłożone.
Powyższe operacje (rozciąganie i wyrównywanie histogramu) odbywają sie z pomocą tablicy LUT, która dla każdej możliwej wartości koloru przypisuje nową, odpowiadającą mu wartość. Obie operacje najpierw wyznaczają taką tablicę, a potem modyfikują każdy piksel obrazu zgodnie z jej wartościami.
g) multiplicate() - operacja, która przemnaża przez siebie odpowiadające sobie wartości pikseli dwóch obrazów według wzoru: [piksel1 * piksel2 / maksPiksel], Gdzie maksPiksel wynosi 255, a piksel1 i piksel2 pochodzą odpowiednio z pierwszego i drugiego obrazu.

h) interlace() - operacja, która przeplata dwa obrazy, z uwzględnieniem procentowego udziału drugiego obrazu względem pierwszego. Innymi słowy, tworzy obraz o wartościach podanych według wzoru: [piksel1 * (100% - udział) + piksel2 * udział]

W plikach histogram.cpp i histogram.h znajduje się klasa HistFrame – dziedzicząc po wxFrame, reprezentuje okienko, w którym wyświetlany jest histogram. Dane o częstotliwości występowania poszczególnych kolorów przechowywane są w tablicy colorCount. Wyliczenie histogramu (a więc również wypełnienie tej tablicy) następuje w chwili wywołania metody drawHistogram(). Klasa przechowuje również informację o wartości pikseli o największej częstotliwości, by móc stosownie znormalizować wielkość wyświetlanego na panelu wykresu.

Ostatnią klasą jest MultFrame, również dziedziczącą po wxFrame i reprezentującą okienko, w którym wyświetlany jest podgląd obrazu podczas wykonywania operacji przemnażania lub przeplatania obrazu. Mimo, że poświęcone są na nią pliki multiplication.cpp i multiplication.h, to jednak do wykonania rzeczywistych operacji przemnażania i przeplatania, klasa sięga do odpowiednich funkcji z plików operations.h i operations.cpp. Klasa zajmuje się jedynie przygotowaniem GUI z podglądem i obsługą wczytania drugiego obrazu oraz ustawienia jego udziału (w przypadku przeplatania).

## Problemy

a) Po użyciu operacji wymagającej dwóch obrazów (przeplot lub mnożenie) suwaki w oknie głównym (jasność, kontrast i gamma) usuwają drugi obraz i wprowadzają zmiany tylko do oryginalnego.

b) Przy dużej różnicy wymiarów drugi obraz będzie wyraźnie zniekształcony.