



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## **APLIKACE PRO PODPORU VÝUKY BAREVNÝCH MODELŮ**

APPLICATIONS SUPPORTING LECTURES ON COLOUR MODELS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Kiryl Khaladzinski

**VEDOUCÍ PRÁCE**

SUPERVISOR

doc. Mgr. Pavel Rajmic, Ph. D

**BRNO 2018**

## **Abstrakt**

Cílem této bakalářské práce je tvorba webových aplikací, které na reálných příkladech ukážou možnosti práce s různými barevnými modely, také principy tvorby digitálních obrazů a možnosti zpracování dat digitálního obrazu. Teoretická část věnovaná základním pojmům v teorii barev a základem rastrových obrázků. Praktická část je návrh čtyř aplikací a jejich tvorby. První z apletů ukazuje princip míchání barev v různých barevných modelech, druhý na praktickém příkladu ukazuje základ indexové reprezentace obrazu, třetí ukazuje metody překladu barevného obrazu na šedotónový a poslední aplet se používá k překladu šedotónového obrazu na barevné.

## **Klíčová slova**

Applet, barevný model, digitální obraz, nepravé barvy, podpora výuky, rastrový obraz, webové aplikace, JavaScript.

## **Abstract**

The aim of this bachelor thesis is to create Web applications that show the possibilities of working with different color models in real examples shows principles of digital images creation and the possibility of processing digital image data. Theoretical part devoted to the basic concept in color theory and the basis of raster images. The practical part is the design of four applications and their creation. The first of the applets shows the principle of mixing colors in different color models, the second on the example shows the basis of the index representation of the image, the third shows the methods of translating the color image to grayscale and the last applet is used to translate grayscale image to color image.

## **Keywords**

Applet, color model, digital image, false colors, support of lectures, raster image, Web application, JavaScript.

## **Bibliografická citace:**

KHALADZINSKI, K. *Aplikace pro podporu výuky barevných modelů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 46 s. Vedoucí bakalářské práce doc. Mgr. Pavel Rajmic, Ph.D.

## Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „*Aplikace pro podporu výuky barevných modelů*“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(-a) autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následku porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **29. května 2018**

.....  
podpis autora



## **Poděkování**

Děkuji vedoucímu semestrální práce doc. Mgr. Pavel Rajmic, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne **29. května 2018**

.....  
podpis autora

Výzkum popsáný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

# Obsah

Úvod.....	11
1. Teorie barev a základní pojmy rastrových obrazu .....	13
1.1 Základní atributy barev .....	14
1.1.1 Barevný tón.....	14
1.1.2 Světlost.....	14
1.1.3 Sytost.....	14
1.2 Světlostní a sytostní proměny barevných tónů .....	15
1.2.1 Plošné uspořádání barev.....	15
1.2.2 Prostorové uspořádání barev.....	16
1.2.3 Barvy chromatické a achromatické .....	16
1.3 Principy míchání barev .....	17
1.3.1 Členění barev podle pořadí.....	17
1.3.2 Aditivní míchání barevných světel.....	17
1.3.3 Subtraktivní míchání barevných pigmentů .....	18
1.4 Měření a matematické vyjádření barev .....	18
1.4.1 Mezinárodní komise pro osvětlování (CIE) .....	18
1.4.2 Trichromatická měrná soustava.....	19
1.4.3 Chromatický diagram CIE 1931.....	20
1.5 Reprezentace barev v počítači.....	20
1.5.1 Reprezentace rastrového obrazu.....	21
1.6 Barevné prostory.....	22
1.6.1 Barevný prostor RGB.....	22
1.6.2 Barevný prostor CMY.....	23
1.6.3 Barevný prostor HLS.....	23
1.6.4 Barevný prostor Lab .....	23
1.6.5 Barevný prostor YCbCr.....	24
2. Návrh řešení.....	25
2.1 Použité technologie a jazyky.....	25
2.2 Důvody pro výběr jazyka JavaScript a knihovny React.....	25
2.3 Návrh appletu „Paletová reprezentace obrazu“ .....	26
2.3.1 Vybraní velikosti matic .....	27
2.3.2 Přidělení barvě čísla.....	28
2.3.3 Ovládací tlačítka .....	28
2.4 Návrh appletu „Míchání barev“ .....	29

2.5	Návrh appletu „Převod barevného obrazu na šedotónový“ .....	30
2.6	Návrh appletů „Barvené šedotónových obrazů (nepravé barvy)“ .....	32
3.	Popis řešení.....	34
3.1	Míchání barev v různých modelech.....	34
3.2	Paletova (indexová) reprezentace obrazu .....	36
3.3	Převod barevného obrazu na šedotónový.....	37
3.3.1	Průměrování R, G, B.....	38
3.3.2	Míchání R, G, B.....	38
3.3.3	Převod do Lab .....	39
3.4	Barvení šedotónových obrazů (nepravé barvy).....	41
3.4.1	Lineární gradient.....	41
3.4.2	HSL.....	42
3.4.3	Jet.....	42
3.4.4	COOL .....	42
3.4.5	HOT .....	42
	Závěr.....	43
	Literatura.....	44
	Seznam symbolů, veličin a zkratk.....	45
A	Obsah elektronické přílohy .....	46

## Seznam obrázků

Obr. 1.1	Vzájemná proměnlivost sytosti a světlosti.....	15
Obr. 1.2	Prostorové uspořádání barev.....	16
Obr. 1.3.	Chromatický diagram CIE 1931.....	20
Obr. 1.4.	Reprezentace obrazu v indexovém módu.....	21
Obr. 2.1.	Návrh appletu "Paletová reprezentace obrázu".....	26
Obr. 2.2.	Demonstrační část appletu "Paletová reprezentace obrázu".....	27
Obr. 2.3.	Vybraní velikostí matic v appletu "Paletová reprezentace obrázu".....	27
Obr. 2.4.	Přidělení barvě čísla v appletu "Paletová reprezentace obrázu".....	28
Obr. 2.5.	Ovládací tlačítka v appletu "Paletová reprezentace obrázu".....	28
Obr.2.6.	Návrh appletu "Míchání barev".....	29
Obr. 2.7	Blok barevných modelu appletu "Míchání barev" (fragment).....	30
Obr. 2.8.	Návrh appletu "Převod barevného obrazu na šedotónovy".....	30
Obr.2.9.	Demonstrační část appletu "Převod barevného obrazu na šedotónovy" ..	31
Obr.2.10.	Ovládací prvky appletu "Převod barevného obrazu na šedotónový" .....	31
Obr. 2.11	Navrh appletu "Barvení šedotónového obrazu" .....	32
Obr. 2.12	Ovladací prvky appletu „Barvení šedotónových obrazů“ .....	32
Obr. 2.13	Gamma korekce a doplňkové nastavení appletu "Barvení šedotónových obrazů" .....	33
Obr. 2.14	Demonstrační cast appletu "Barvení šedotónových obrazů" .....	33

## Seznam výpisů počítačových kódů

Výpis kódu 3.1: clrmodels.js – funkce převodu hodnot z modelu RGB do modelu HSL .....	35
Výpis kódu 3.2: clr2gray.js – funkce průměrování R, G, B.....	38
Výpis kódu 3.3: clr2gray.js – funkce míchaní R, G, B .....	39
Výpis kódu 3.4: clr2gray.js – funkce pro získávání hodnot kanálu L, barevného modelu Lab.....	40

# ÚVOD

Digitální obraz se stal nedílnou součástí našeho každodenního života. Každý den jsem otevíráme sociální sítě a vidíme jako jeden z našich příbuzných nebo přátel sdíleli fotku, nebo prostě rozhodl zlepšit vaši náladu legračním obrázkem, studenti se setkávají s grafy a diagramy v procesu studia různých disciplín, lékaři každý den si prohlížejí rentgenové snímky svých pacientů a takových příkladů možné uvést ještě velké množství.

Digitální obraz můžeme vidět nejen na obrazovce jakéhokoliv elektronického zařízení (telefon, laptop, tablet), a také na papíře, ať už tištěné fotografie nebo jen obraz v novinách. Nicméně, než se dostanete do papíru, obraz je opraven a zpracován na digitálním zařízení.

Všechno kolem nás se skládá z molekul a atomů, stejně digitální obraz se skládá z hodně malých prvků, které jsou umístěny v určitém pořadí tvoří obraz. Tyto drobné prvky mají v sobě hodnotu barvy, které v počítačových technologiích lze získat mícháním hlavních barev.

Aby bylo možné začít studovat a zpracovávat digitální obrázky, musíme pochopit základy barevné teorie.

Účelem této bakalářské práce je vytvořit několik aplikací, které směřovaný na pomoc při studiu teorie barevných modelů a zpracování obrazu.

V rámci této práce bude vytvořeno čtyři appletů:

- Míchání barev v různých modelech,
- Paletova (indexovaná) reprezentace obrazu,
- Převod barevného obrazu na šedotónový,
- Převod šedotónového obrazu na barevný (nepravé barvy).

Pro vytváření těchto aplikací je nutné pochopit teorii míchání barev, s čeho se skládá digitální obraz a principy jeho zpracování. Tyto aplikace mohou být užitečné pro zájemce o studium digitálních obrázků. Například aplikace, která míchá barvy v různých barevných modelech, může být použita jako vizuální příklad během přednášky nebo použita jako převodník pro barevné hodnoty mezi různými modely. Aplikace bude ilustrovat principy míchání barev, různé typy barevných modelů, a vztah mezi nimi.

Pro vytváření aplikace „Paletova reprezentace obrazu“ je nutné pochopit princip skládání takového obrázku.

Pro vytváření aplikace „Převod barevného obrazu na šedo tónový“ a „Převod šedotónového obrazu na barevný“ budou užitečné znalosti získané při vytváření prvních dvou aplikací. Tyto aplikace budou zobrazovat různé metody překladu z barevného obrazu do stupňů šedi, a naopak v reálném čase.

Všechny aplikace budou vytvořeny pomocí jazyka HTML, CSS a programovacího jazyka JavaScript.

V první kapitole budou zkoumány základy teorie barev a principy skládání rastrových obrazů. Druhá kapitola navrhne možnosti řešení problému. Třetí kapitola bude popisovat konečné řešení.



# 1. TEORIE BAREV A ZÁKLÁDNÍ POJMY RASTOVÝCH OBRAZU

Pojem počítačové grafiky je velmi široký a aby vysvětlil, co je v něm obsaženo potřebné stovky hodin. Jednou z takových oblastí je zpracování grafické informace a její vizualizace. [1]

Je dva základní způsoby ukládání a zpracování obrazové informace:

- Rastrová grafika (bitmapová grafika),
- Vektorová grafika. [3]

Obraz rastrové grafiky se poskládá z velmi malých bodů, které se nazývají pixely, co je zkratkou od anglických slov Picture Element. Obvykle v digitálním obraze počet pixelů přesahuje stovky tisíc a ve většině případů lidské oko není schopno jich rozlišit. Pixely uspořádané do rastru, co si možno představit jako velkou matici nebo mřížku barevných bodů. Většina výstupní zařízení (např. monitory, tiskárny), které jsme používáme pro vizualizace obrazů, používá princip poskládání rastrového obrazu.

Každý pixel má svou vlastní polohu v rámci rastru a také může být charakterizován konkrétní barvou, která je jednou z důležitých vlastností pixelu.

Na to abych člověk mohl vnímat barvy musí být dostatek světla. Na světlo reagují naše oko, které vysílá impulsy do mozku a teprve vzniká tam barevný vjem. Na to, jak lidí vnímají barvy má vliv celá řada faktorů, protože různé vědní disciplíny nahlízejí na tento proces zcela odlišným způsobem.

Barva ve fyzice je měřitelnou veličinou a závisí na dopadu a odrazu světla od povrchu předmětu. Fyzikální a fyziologické působení barev nejlépe vystihuje výrok Isaaca Newtona: "Ve fyzice žádné barvy neexistují, ve fyzice existují pouze spektrum. Barvy existují pouze v našich očích a v našem mozku."

V psychologii není barva vlastností světla, ale vjem zrakovým orgánem paprsku světla určité délky. Z pohledu této disciplíny, jak člověk vnímá barvy má vliv jeho myšlení, získané zkušenosti, charakteristiky a vlastnosti lidského oka jako zrakového orgánu.

V umění a výtvarné výchově má barva subjektivní charakter. Barvy jsou nedílnou součástí fotografií nebo obrazů, pomocí barev autory se snažejí předat náladu nebo dát emocionální barvu konkrétního detailu nebo celého obrazu. [1]

## **1.1 Základní atributy barev**

Barvy možné charakterizovat vlastnostmi, které jsou závislé na charakteru světelného zdroje, vlastnosti povrchu, od kterého odrazí světlo, a také na charakteristikách organismu a schopnostech člověka vnímat barvy. Hermann von Helmholtz byl jedním z prvních autorů, který pojmenoval atributy barev. Ten použil takové termíny jako barevný ton, světelnost a sytost. [1]

### **1.1.1 Barevný tón**

Základní vlastnost je barevný tón. Pomocí této vlastnosti můžeme odlišit jednu barvu od druhé. Spektrální barvy určité vlnové délky jsou označeny barevný tóny, příkladem je parsekvlnové délky 700 nm, který vyvolává vjem červené barvy. Každý barevný tón je charakterizován svou vlnovou délkou.

Všichni barevné tóny do sebe ve spektru přecházejí. Červený a fialový tóny možné spojit pomocí purpurového tónu, čímž vznikne uzavřený kruh barevných tónu.

Základní barevné tóny mají svou vlastní identifikaci a pojmenovaný podle známých názvů (např. červená, modrá, zelená atd.). [1]

### **1.1.2 Světlost**

Barevný tón je charakteristika barvy, která je ovlivněna délkou světelného paprsku, světlost je charakteristika, která je ovlivněna velikosti rozkmitu světelných vln. Světlost je hlavním atributem pro vyjádření kontrastu a ukazuje množství bílé nebo černé barvy, která byla přidána do základního barevného tónu.

Nejnižší stupeň světlosti má černá barva, naopak nejvyšší stupeň světlosti má bílá barva. Mezi těmito stupni se nachází škála šedých barev. Odstíny šedých barev vznikají snižováním světlosti bílé barvy.

Světlost lze také měnit pro všechny barevné tónu. Mezi bílou a černou barvou už bude konkrétní barevný tón. Přidáváním bílé barvy budeme měnit světlostní stupeň a blížit se k bílé barvě, ten proces zesvětlování. Naopak když budeme přidávat černou barvu, světlostní stupeň bude klesat a barevný ton bude se blížit k černé barvě, ten proces se nazývá ztemňování. Přidáváním černé nebo bílé barvy nedochází k vytváření nového barevného tónu, jenom k vytváření odstínu. [1]

### **1.1.3 Sytost**

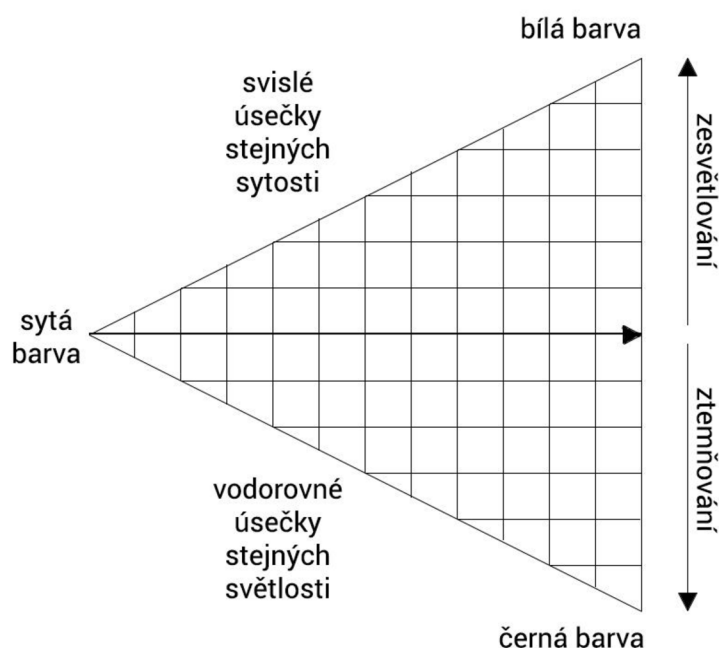
Sytost je hodnota, která ukazuje sílu určitého barevného tónu, stupeň intenzity barvy. Ukazuje relativní čistotu barvy, ovlivňuje její barevnost (pestrost). Sytost nese informace o tom, jak moc se barva odlišuje od spektrální barvy stejného barevného tónu co do čistoty tónu.

Nejvyšší hodnota sytosti je u čistých barev spektra. Míchání barevného tónu s jinými barvami snižuje jeho intenzitu. Sytost je mírou pestrosti každé barvy. [1]

## 1.2 Světlostní a sytostní proměny barevných tónů

### 1.2.1 Plošné uspořádání barev

Vztah mezi sytostí barevného tónu a jeho světlostí, můžeme zakreslit do trojúhelníku (Obr. 2.1). Na obrázku je vidět že přidáváním bílé barvy bude vzrůstat světlost barevného tónu a klesat jeho sytost. Přidáváním černé barvy bude klesat sytost a světlost barevného tónu. Ke změnám dochází nejenom na hranicích trojúhelníku ale také uvnitř. Přidáváním bíle nebo černé barvy, a při změně sytosti vznikají odstíny, ale barevný tón zůstává stejný.



Obr. 1.1 Vzájemná proměnlivost sytosti a světlosti.

Vodorovné úsečky uvnitř trojúhelníka geometricky reprezentují:

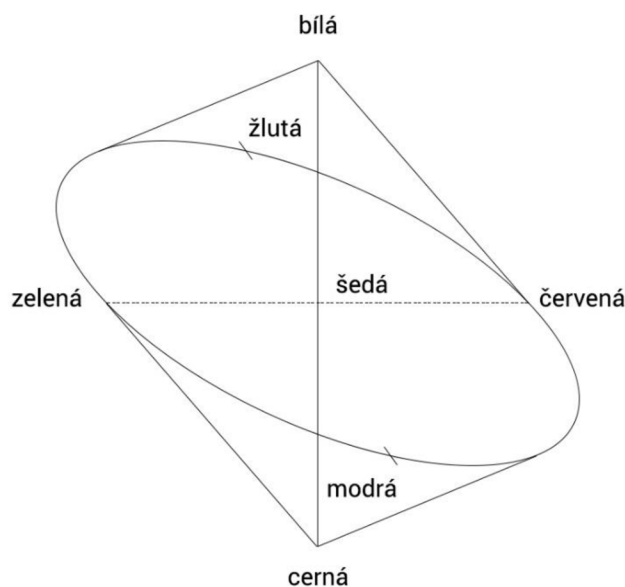
- odstíny stejného barevného tónu a světlosti,
- změnu sytosti při nezměněné světlosti.

Svislé úsečky uvnitř trojúhelníka geometricky reprezentují:

- odstíny stejného barevného tónu a sytosti,
- změnu světlosti při nezměněné sytosti. [1]

## 1.2.2 Prostorové uspořádání barev

Ke každému barevnému tónu můžeme sestrojít trojúhelník sytostí a světlostí. Získané trojúhelníky budou mít společnou základnu (hranice trojúhelníka která ukazuje změnu světlosti od černé barvy k bílé) a když spojíme dohromady všechny trojúhelníky dostaneme prostorový dvojkůžel. Na obvodu jsou rozmístěny syté barevné tony jako v barevném kruhu.



Obr. 1.2 Prostorové uspořádání barev

Tento trojrozměrný model ilustruje proměnlivost světlosti a sytosti celého barevného kruhu. [1]

## 1.2.3 Barvy chromatické a achromatické

Chromaticnost je pojem, který vyjadřuje barevnosti světla. Všechny čisté barevné tony můžeme označit za chromatické.

Barvy černá, bílá a všechny odstíny šedé jsou barvy achromatické. Všechny achromatické barvy jsou neutrální, protože se výborně kombinují se všemi chromatickými barvami.

Achromatické barvy odlišují se jedná od druhé pouze světlostí a mohou být použité ke změně světlosti a sytosti chromatických barev. [1]

## 1.3 Principy míchání barev

### 1.3.1 Členění barev podle pořadí

Většina barevných systémů je založená na principu míchání barev z několika barevných tónů. Podle určitého principu odvozuji další tóny. Podle pořadí, v jakém se barvy míchají, lze postupně rozlišit:

- Barvy primární (hlavní) jsou základní barvy, které nelze získat z jiných barev. V případě míchání barevných světél jsou primárními barvami červená, zelená a modrá.
- Barvy sekundární (podvojně) jsou barvy, které vznikají při míchání primárních barev, to buď optickým spojením dvou světelných paprsků (vzniknou barvy žlutá, azurová, purpurová).
- Barvy terciální (potrojně) jsou barvy, které vznikají smícháním jedné primární a jedné sekundární barvy. [1]

### 1.3.2 Aditivní míchání barevných světél

Zdrojem přirozeného světla jsou sluneční paprsky, které vnímáme jako bílé světlo (tzv. achromatické světlo). Každý sluneční paprsek tvořen celým spektrem barev, které lze pozorovat, když rozložíme světelný paprsek pomocí skleněného hranolu (např. spektroskop). Nejznámější příklad takového rozkladu je duha, kterou můžeme za deště vidět na obloze. Příčinou tohoto jevu je rozklad světelných paprsků jasného zdroje světla dešťovými kapkami. Když jsme díváme na duhu, vidíme jenom několik základních barev, ale ve skutečnosti je spektrum spojitě a jeho barvy hladce procházejí mezi sebou prostřednictvím mnoha odstínů. Spektrální barvy se proto někdy označují jako barvy duhové.

Aditivní míchání barev (nebo též součtové či přidávací) má takový princip, že přidává světelné paprsky různých vlnových délek k černé barvě. Tato metoda míchání barev vychází z toho, že černá barva reprezentuje stav, kdy nedochází k žádnému odrazení světla a postupným přičítáním barevných světél dostáváme barvu bílou.

Ostatní barvy můžeme dostat pomocí míchání červeného, zeleného a modrého paprsku světla. Když tyto základní barvy mají nulovou hodnotu, dostáváme barvu černou a postupným zvýšením hodnot budeme dostávat ostatní barvy spektra. Hlavní princip je, čím více hodnota každé složky, tím se přiblížíme k barvě bílé.

Všechny barvy v aditivním barevném prostředí jsou generovány světlem. Z toho vyplývá, že tento systém nepotřebuje žádný zdroj světla. Příkladem aditivního principu míchání barev je jakýkoliv barevný displej nebo monitor. [1]

### **1.3.3 Subtraktivní míchání barevných pigmentů**

Subtraktivní (odečtové) míchání barev má opačný princip jako aditivní míchání barev.

Nejznámějším příkladem je tiskárna a obrázek vytištěný na bílém papíře. Subtraktivní metoda vychází z toho, že místo světelných barev se používají krycí pigmentové barvy. V subtraktivním systému míchání barev se používají také tři hlavní barvy, ale na rozdíl od aditivního systému není to červená, zelená a modrá barvy, to je doplňkové barvy aditivního systému: žlutá, azurová a purpurová.

Smícháním dvou základních barev subtraktivního systému dostaneme sekundární barvu, která v aditivním systému je primární (např. smícháním žluté a purpurové barvy dostaneme zelenou barvu, což je primární barva aditivního systému míchání barev). Smícháme-li tři základní barvy subtraktivního systému, obdržíme černou barvu.

Subtraktivní systém míchání barev je založen na odrazu světla. Každá barva pohlcuje určitou část dopadajícího světla. To znamená že systém potřebuje vnější zdroj světla. A čím více barev přidáme (odečteme světla), tím více budeme blížit k černé barvě. [1]

## **1.4 Měření a matematické vyjádření barev**

Fyziologické charakteristiky lidského oka dovolují rozeznat 150 barevných tónů a kolem dvou tisíce odstínů. V praxi jsou pojmenovány kolem tisícovky různých barev, zbytek není pojmenován. Proto bylo řešeno vymyslet systém přesného měření barev, abych určit každý barevný tón.

Měří se světelný paprsky světelného zdroje a také paprsky odražené od povrchu objektu, pak se zkoumá reakce lidského oka na tento světelný podnět.

Princip měření barev navrhl univerzitní profesor a lékař Hermana von Helmholtze, který se zabýval teorií míchání světla. Pro přesné označení barvy zavedl tři hlavní atributy: barevný tón určený vlnovou délkou světelného paprsku, sytost vyjadřující čistotu dané barvy a světlost. [1]

### **1.4.1 Mezinárodní komise pro osvětlování (CIE)**

Princip měření barev od Hermana von Helmholtze Mezinárodní převzala osvětlovací komise (CIE – Commission internationale de l'éclairage). Tato technická mezinárodní organizace, která vyvíjí technické normy v oblasti světla, osvětlení, barev a barevných prostorů a zabývá se problematikou měření barevného světla či kalibraci různých technických přístrojů a zařízení, také definovala matematický model pro přesný popis barev. [1]

## 1.4.2 Trichromatická měrná soustava

Na základě mnoha experimentů vznikla trichromatická měrná soustava, která vychází z toho, že aditivním mícháním tří barevných světél lze dostat jakékoliv barvy. Základem se stala barevná světla (v závorce je uvedena jejich příslušná vlnová délka):

- červené (700,0 nm),
- zelené (546,1 nm),
- modré (435,8 nm).

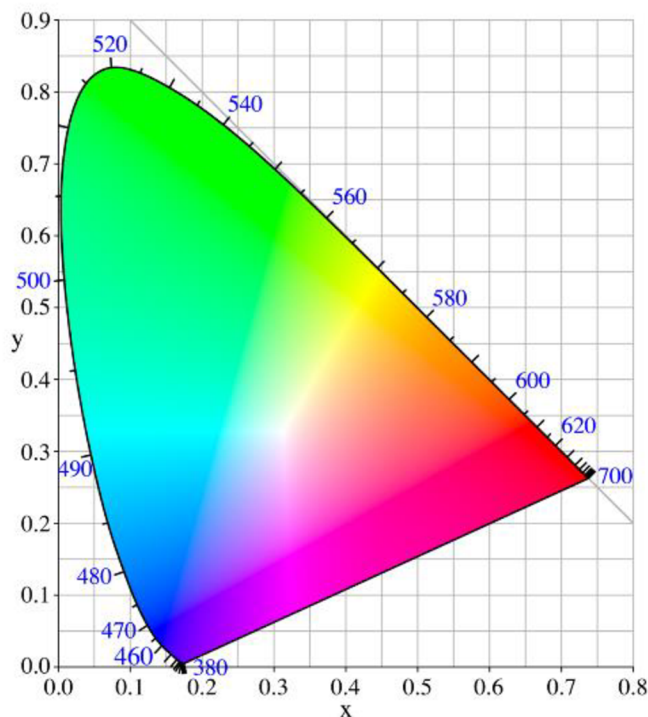
Každou barvu lze v trichromatické měrné soustavě popsat:

- množstvím tří barevných podnětů (tzv. trichromatickými složkami) - označují se velkými písmeny X, Y, Z a vypočítávají se integrací spektrálního barevného podnětu a trichromatických členitelů v celém rozsahu spektra.
- Poměrem tří barevných podnětů (tzv. trichromatickými souřadnicemi) vyjádřeným pomocí normových podílů  $x, y, z$ , kde:
  - $x = X/(X+Y+Z)$  je normový podíl červené barvy,
  - $y = Y/(X+Y+Z)$  je normový podíl zelené barvy,
  - $z = Z/(X+Y+Z)$  je normový podíl modré barvy.

Trichromatické složky tvoří kolorimetrický prostor CIE XYZ. V praxi se však prostorové zobrazení barev v souřadnicích X, Y, Z jevílo jako nepraktické. [1]

### 1.4.3 Chromatický diagram CIE 1931

Chromatický diagram CIE byl vytvořen v roce 1931 a je mezinárodním standardem barev.



Obr. 1.3. Chromatický diagram CIE 1931

Chromatický diagram ještě nazývají barevnou podkovou, v něm je všechny barvy, které lidské může rozlišit. [1]

### 1.5 Reprezentace barev v počítači

Práce každého počítače a zpracování jakékoliv informací v něm probíhá pomocí binárního kódu. To znamená, že informace v počítači je posloupnost jedniček a nul. Minimální velikost informací o barvách v počítači je 1 bit, což znamená, že v nejjednodušším případě na obrazovce, bude pixel "zapnutý" nebo "vypnutý", což představuje bílou nebo černou barvu. Není výjimkou a označení barev. Ve většině případů se používají tři hlavní složky: červená, zelená a modrá.

Každá složka má 256 úrovní (hodnoty složek jsou od 0 do 255, kde 0 je minimální úroveň a 255 je maximální). To znamená, že mícháním různých úrovní základních barev můžeme dostat velké množství barev a jejich odstínů. [1]

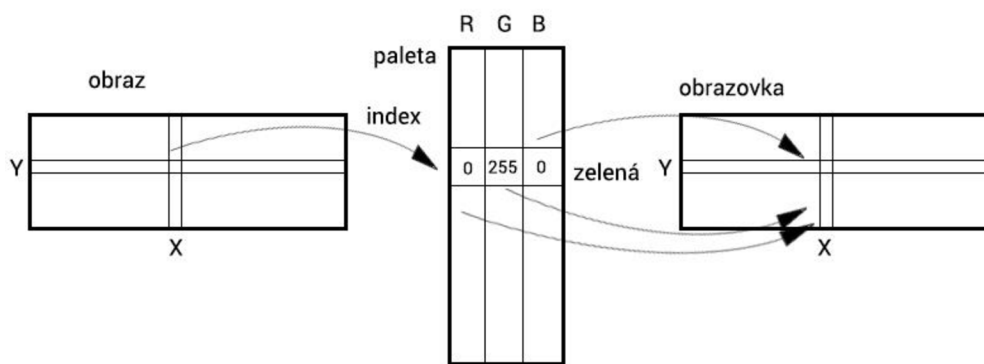


## 1.5.1 Repräsentace rastrového obrazu

Nejjednodušším modelem repräsentaci obrazu je dvourozměrná matice pixelů, kde každý pixel nabývá hodnotu podle typu obrazu.

Když každý pixel bude popsán jediným bitem, říkáme že obraz je monochromatický nebo černobílý. Binární informace nese v sebe jedničky a nuly, ale to neznamená že to je právě černá a bílá barva, můžeme požit libovolně dvě barvy.

Indexový mód pro repräsentace obrazu požívá mapu barev (colormap). Každý pixel v takovém obrazu nenabývá hodnotu barvy, ale je ukazatelem do tabulky – mapa barev (barevná paleta).



Obr. 1.4. Repräsentace obrazu v indexovém módu.

Barevná paleta je tabulka, která obsahuje hodnoty barev každého pixelu. Paleta se skládá ze tří barevných kanálů, každý repräsentován 8 bity a index repräsentován jedním bytem (paleta má 256 řádků maximálně). Z toho vyplývá že každý pixel může nabývat hodnotu některé z 256 barev, které se vybírají z celkového počtu barev (16 777 216 barev). Tento způsob přiřazení barev také má název pseudo color. Barevná paleta může být buď částí obrázku, nebo může být vytvořena při zobrazení obrázku. [3]

Výhodou bitmapové grafiky je možnost zobrazovat obrázky a editovat jich na různých typech zařízení s různým typem softwaru. Také jednou z výhod rastrových obrázků je, že na rozdíl od obrázků vektorové grafiky, podporují velký počet formátů souborů a pro uložení potřebují menší množství paměti.

Hlavní nevýhody bitmapových obrázků je závislost velikosti souboru na velikosti samotného obrazu a také omezení ve změně velikosti obrazu. [8]

## 1.6 Barevné prostory

Moderní počítačové technologie pro zobrazování barev používá různé barevné modely. Každý barevný model je založen na aditivním nebo subtraktivním systému míchání barev. V počítačové technice jsou hodnoty barev specifikovány libovolnými jednotkami měření, buď matematickými nebo fyzikálními.

Různá technická zařízení fungují v různých barevných modelů, které mají různý počet barev k dispozici. Některé barvy získané v jednom barevném prostoru nemusí být v jiném prostoru, takže tato barva by měla být nahrazena nejbližší verzí, v blízkosti originálu.

Důvod, pro který existuje několik barevných modelů je v tom, že všechna technická zařízení mají různé technické charakteristiky, které ovlivňují na zobrazení barev. [1]

### 1.6.1 Barevný prostor RGB

Jeden z hlavních barevných modelů, který slouží k získání různých barev je barevný model RGB. Skládá se ze tří hlavních kanálů (barev), jejichž kombinace dostaneme jiné barvy. Hlavní složky jsou červená (Red), zelená (Green) a modrá (Blue). Tento model používá aditivní systém míchání barev.

Každý kanál se skládá z 256 možných úrovní zobrazení. To znamená, že při skládání tří kanálů, je možné získat 16 777 216 různých kombinací barev a jejich odstínů.

Tento barevný model se používá hlavně v zařízeních, která mohou vyzařovat světlo, například monitor, scanner nebo digitální fotoaparát. [1]

Barevný prostor RGB je nejbližší barevný model pro lidské vnímání. Existuje však několik typů modelů RGB: sRGB, Adobe RGB, ProPhoto RGB.

Prostor sRGB je světově známý standard vyvinutý společně firmami HP a Microsoft v roce 1995. Specifikace sRGB umožňuje správně zobrazit barvy na monitorech a televizorech, co stalo hlavním faktorem pro jeho přijetí jako standardu. [5]

Prostor Adobe RGB byl vyvinut v roce 1998 firmou Adobe Systémem. Hlavním účelem vytvoření tohoto prostoru bylo vytvoření standardu s rozšířeným barevným gamutem, více než u sRGB. [6]

ProPhoto RGB prostor, také známý jako ROMM RGB, byl vyvinut společností Kodak a nabízí ještě více barev než Adobe RGB. Tento prostor je navržen pro práci s fotomateriály a pokrývá 90% možných barev v barevném prostoru CIE L\*a\*b. [7]

### 1.6.2 Barevný prostor CMY

Barevný model CMY je typickým představitelem subtraktivního systému míchání barev. Podobně jako barevný model RGB, se skládá ze 3 hlavních barev, které jsou sekundární barvy v modelu RGB, je žlutá (Yellow), purpurová (Magenta) a azurová (Cyan).

Princip míchání barev a počet úrovní jednotlivých kanálů je přesně stejný jako u modelu RGB.

Tento barevný model se používá hlavně v tiskařské technice. Příkladem může být barevná tiskárna. [1]

### 1.6.3 Barevný prostor HLS

Hlavní nevýhodou barevných modelů RGB a CMY je, že jsou zaměřeny na určité charakteristiky technických zařízení. Takže vznikali jiné barevné prostory, ale na rozdíl od prostoru RGB a CMY, oni se neskládají ze tří hlavních barev, ale ze tří hodnot, jako je odstín, sytost a jas.

Typickým příkladem je prostor HLS (v některých zdrojích může mít název HSB). Tento název se získá z písmen rozlohy, což znamená:

- H (Hue) – barevný tón – nabývá hodnoty od 0° do 360°, jako jdou barvy v barevném kruhu,
- S (Saturation) – sytost – nabývá hodnoty od 0 do 1, nebo 0 do 100,
- L (Lightness), B (Brightness) – jasová hodnota – která jako sytost také nabývá hodnoty od 0 do 1, nebo od 0 do 100. [1]

### 1.6.4 Barevný prostor Lab

Barevný prostor Lab má oficiální název CIE 1976 L\*a\*b, byl vyvinutý pro vědecké účely. Byl navržena tak, aby byl nezávislá na vlastnostech zobrazovacích zařízení.

Barevný prostor Lab je složen ze tří kanálů:

- Jasový kanál L (Lightness) - světlost bodu – nabývá hodnoty v rozsahu 0 až 100.
- Barevný kanál **a** reprezentuje přechod mezi doplňkovými barvami červenou (**+a**) a zelenou (**-a**), nabývá hodnoty v rozmezí -128 až 127.
- Barevný kanál **b** reprezentuje přechod mezi doplňkovými barvami žlutou (**+b**) a modrou (**-b**), jako kanál **a** nabývá hodnoty v rozmezí -128 až 127.

V barevném prostoru CIE 1976 L\*a\*b je jasová složka obrazu oddělena od složky barevné. Barevný prostor Lab moc často se používá v počítačové grafice. [1]

### **1.6.5 Barevný prostor YCbCr**

YCbCr nebo Y'CbCr používá se u videa nebo u digitální fotografie. Y' je řazen do laminace (jasu) komponentu a Cb s Cr jsou modrý a červený chrominanční komponent.

Y'CbCr není absolutní barevný model, ale existuje jako způsob kódování RGB informací. Zobrazení barev hodně závisí na hodnotách RGB barev v signálu.

Y' nabývá hodnoty v rozmezí 0 až 1, kanály Cb a Cr nabývají hodnoty od -0.5 do +0.5, ale v některých zdrojích můžeme setkat se s tím, že všechny tři kanály nabývají hodnoty od 0 do 255, což odpovídá úrovni RGB modelu. [4]

## 2. NÁVRH REŠENÍ

### 2.1 Použité technologie a jazyky

Všechny aplikace budou fungovat v režimu online proto základ bude napsán jazyce HTML (od angl. Hypertextmarkuplanguage). Pro napsání funkcí každé aplikace bude použit objektově orientovaný jazyk JavaScript, nebo spíše knihovna React (někdy v literatuře je nalezen jako React.js nebo ReactJS).

React byla vytvořena společností Facebook a Instagram, první verze byla vydána v březnu 2013. React je ideálním nástrojem pro vytváření webových aplikací.

### 2.2 Důvody pro výběr jazyka JavaScript a knihovny React

Myslím, že jazyk JavaScript je lepší varianta pro vytváření aplikací souvisejících s zpracováním obrazu. Tento objektově orientovaný jazyk je používán mnoha velkými internetovými zdroji a sociálními sítěmi, protože umožňuje zpracování velkého množství dat během několika sekund. Tento jazyk se vyvíjí ve velkém tempu, což umožňuje urychlit aplikace, a také má obrovskou komunitu, která je připravena pomoc v případě problému. Důležitým faktorem je, že tento jazyk je podporován všemi hlavními internetovými prohlížeči, které vám umožní používat aplikace na různých platformách a zařízeních.

Charakteristickým rysem vybrané knihovny React je virtuální DOM (Document Object Model). Celou strukturu webové stránky lze znázornit pomocí modelu DOM. Ve skutečnosti je to organizace prvků HTML, které můžeme upravovat, odstraňovat nebo přidávat nové prvky. Jazyk JavaScript se používá k interakci s DOM. Při manipulaci s prvky HTML pomocí jazyka JavaScript, může dojít ke snížení výkonu, zejména při velkém množství prvků a operace na prvky může trvat delší dobu. Koncept virtuální DOM byl zaveden k vyřešení problému s výkonem.

Virtuální DOM představuje lehkou kopii normální DOM. A charakteristickým rysem React je, že tato knihovna pracuje s virtuálním DOM spíše než obvyklým.

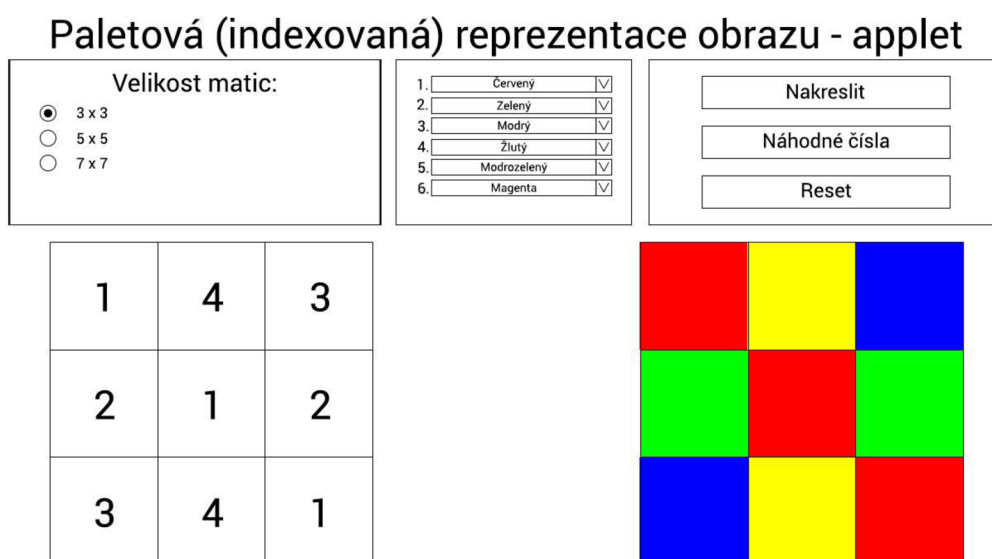
Pokud aplikace potřebuje znát stav prvků, tak se obrátit na virtuální DOM. Pokud chcete změnit prvky webové stránky, změny jsou nejprve budou provedeny ve virtuálním DOM a pak nový stav virtuálního DOM se srovnává s aktuálním stavem. A pokud tyto státy jsou různé, React najde minimální počet manipulací, které jsou požadovány před aktualizací na nový stav a vyrábí je. Tato interakce s

prvky webové stránky funguje mnohem rychleji a efektivněji, než kdybyste pracovali z JavaScriptu přímo s DOM.

Také jedním z důvodů je, že jsem mám praktické dovednosti práci s tímto programovacím jazykem a knihovnou React, což mi umožňuje plně vrhnout do vytváření aplikací a ne ve studiu základu jazyka JavaScript.

## 2.3 Návrh appletu „Paletová reprezentace obrazu“

Applet „Indexová reprezentace obrazu“ bude sloužit pro názorný příklad principu poskládání rastrového obrazu a také pro podporu ovládnutí nastudovanou teorii v rámci tématu barevných modelů. Návrh appletu je zobrazen na obrázku 6.

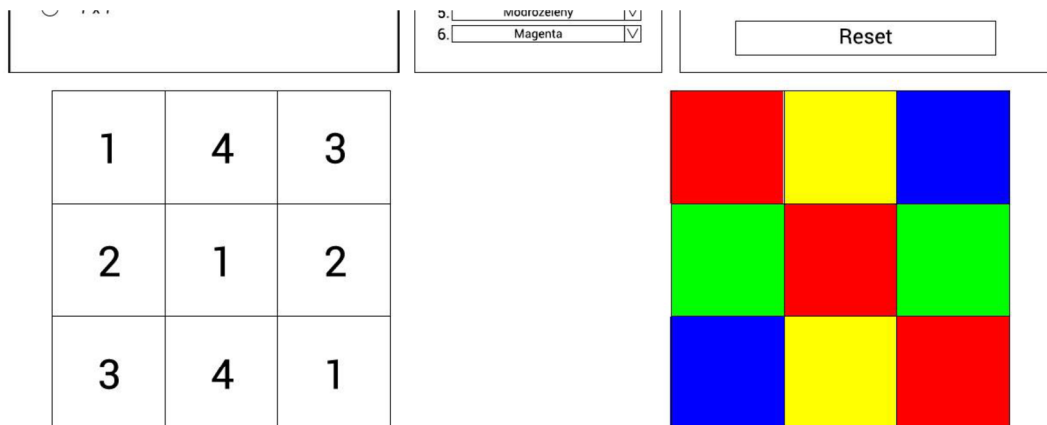


Obr. 2.1. Návrh appletu „Paletová reprezentace obrázku“

Applet možné rozdělit na dvě hlavní části:

- nastavení (nastavení a změna parametrů),
- demonstrační.

Demonstrační část bude se skládat ze dvou stejné velikosti matic. První matice (sleva na obrázku) slouží pro vyplnění jí čísla z dostupného rozsahu. V jednu buňku matice je možné zapsat jenom jedno číslo. Druhá matice (zprava na obrázku) bude sloužit pro demonstraci principu indexová móda. Tato matice bude zabarvená v souladu s vyplněnou první matice.



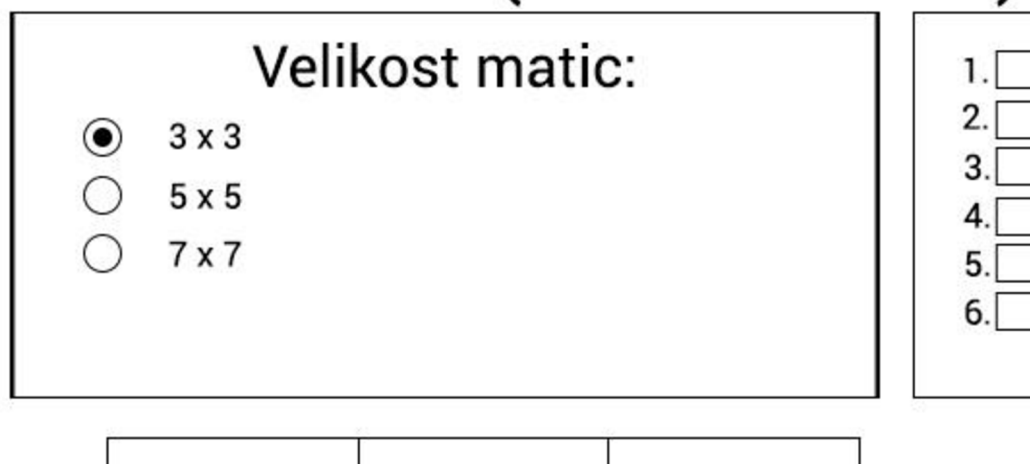
Obr. 2.2. Demonstrační část appletu "Paletová reprezentace obrázu"

Část appletu z nastavení bude se skládat ze tří bloku:

1. Vybraní velikostí matic.
2. Přidělení barvě čísla.
3. Ovládací tlačítka.

### 2.3.1 Vybraní velikosti matic

Blok bude se skládat z několika rádií (z angl. "radio button"), které budou dovolovat vybírat jedenu z velikostí matic. Výběr velikosti může být proveden ze tří možností: 3 x 3, 5 x 5 a 7 x 7. Vyber se bude provádět pomocí stisknutí na jednu z rádií.



Obr. 2.3. Vybraní velikostí matic v appletu "Paletová reprezentace obrázu"

### 2.3.2 Přidělení barvě čísla

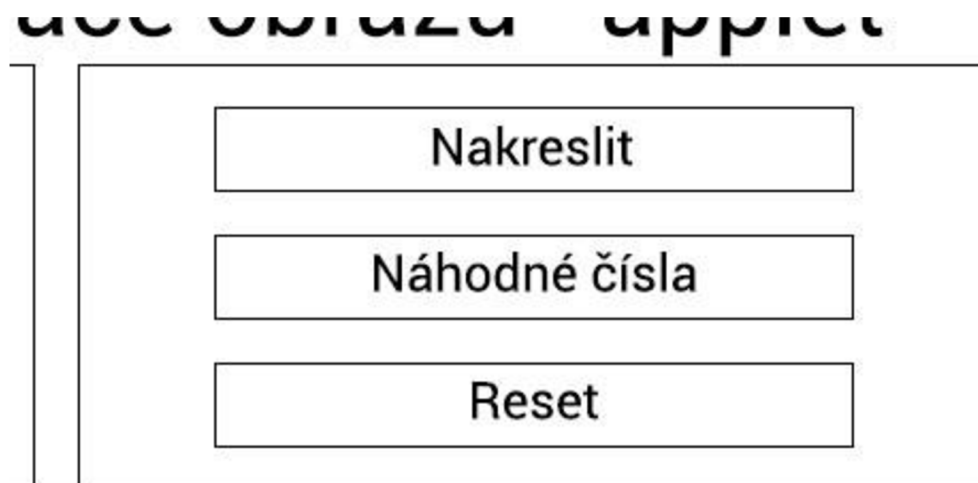
V tomto bloku v levé části budou se nacházet čísla, které uživatel může požit pro vyplnění první matice. Zprava od každého čísla bude se nacházet rozbalovací nabídka (angl. “select”), v které uživatel může zvolit jednu barvu ze seznamu a tato barva bude přiřazena k číslu zleva.



Obr. 2.4. Přidělení barvě čísla v appletu “Paletová reprezentace obrázu”

### 2.3.3 Ovládací tlačítka

Blok bude se skládat ze tří ovládací tlačítka. Tlačítko “Vykreslit” po jeho stisknutí zabarví každou buňku druhé matice v souladu z první matice nebo uživateli že je problém v nastavení nebo první matice vyplněna špatně. Tlačítko “Náhodné” vyplní



Obr. 2.5. Ovládací tlačítka v appletu “Paletová reprezentace obrázu”

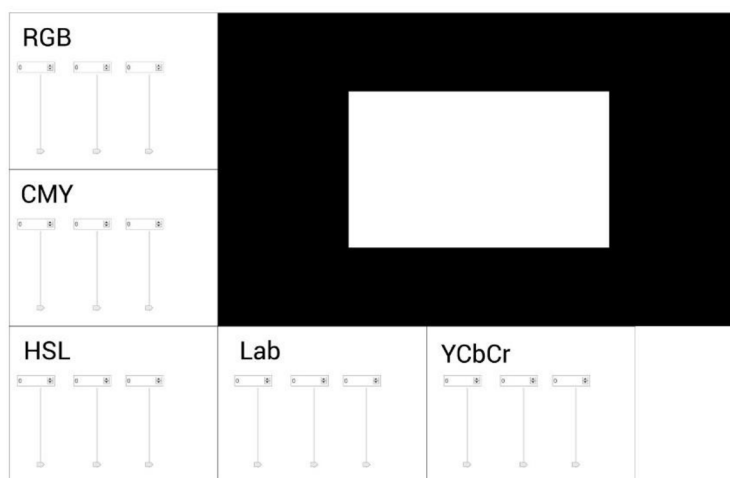


první matici náhodnými čísly a zbarví druhou matici v souladu s první. Tlačítko "Reset" bude sloužit pro smazání vyplněných buněk obojí matic.

## 2.4 Návrh appletu „Míchání barev“

Aplikace "Míchání barev" bude sloužit pro vizuální ukázkou překladu hodnot barvy mezi různými barevnými modely. Aplikace se bude skládat ze dvou hlavních bloků.

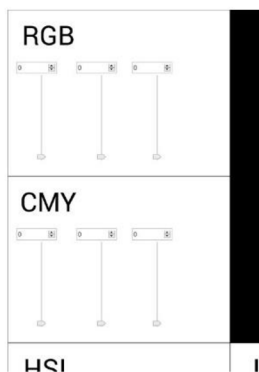
Převod barevného obrazu na šedetónový - applet



Obr.2.6. Návrh appletu "Míchání barev"

V pravé horní části bude blok, ve kterém budou dva obdélníky různé velikosti, jeden bude umístěn uvnitř druhého. Hlavní (na obrázku 3.6 je zobrazen černě) zobrazí barvu, která je získána z vybraných hodnot. Druhý obdélník (označen bílou barvou na obrázku 3.6) zobrazí doplňkovou barvu.

Kolem bloku s barvami budou bloky s různými barevnými modely (Obr. 3.7). V každém barevném modelu bloku, můžete změnit barvu a zobrazí se v demonstračním bloku.



**Obr. 2.7 Blok barevných modelu appletu "Míchání barev" (fragment)**

## 2.5 Návrh appletu „Převod barevného obrazu na šedotónový“

Applet "Převod barevného obrazu na šedotónový" bude sloužit pro podporu výuky barevných modelů. Applet bude provádět demonstraci obrazu z barevného modelů na šedotónový.

### Převod barevného obrazu na šedotónový - applet



Výber obrazku



Metody



**Obr. 2.8. Návrh appletu "Převod barevného obrazu na šedotónový"**

Applet možné rozdělit na dvě části. První část (obr. 2.9) bude se skládat ze dvou obrázků. První obrázek (zleva na obr. 2.9) bude zobrazovat originální obrázek, druhý (zprava na obr. 2.9) je výstupní, který bude zobrazovat vstupní obrázek po transformaci.





Výber obrazku



Metody

**Obr.2.9. Demonstrační část appletu "Převod barevného obrazu na šedotónovy "**

Druhá část bude se skládat ze ovládacích elementů appletu. V této části bude dvě rozbalovací nabídky. Jedna z nich bude sloužit pro výběr možných obrázků ze seznamů, a pomocí druhé nabídky uživatel bude vybírat metodu, pomocí které vstupní obrázek bude transformován do šedotónového modelu.

 Výber obrazku	 Metody
<input type="text" value="Název"/> <input type="button" value="v"/>	<input type="text" value="Název"/> <input type="button" value="v"/>
<input type="text" value="Další nastavení"/>	

**Obr.2.10. Ovládací prvky appletu "Převod barevného obrazu na šedotónový"**

V nejnižší části appletu bude se nacházet blok doplňkových nastavení vybrané metody. Tento blok bude skrytý. Když zvolená uživatelem metoda bude dovolat změnit nebo zvolit nějaké doplňkové nastavení, blok se zobrazovat.

## 2.6 Návrh appletů „Barvené šedotónových obrazů (nepravé barvy)“

Aplikace bude použita pro ilustraci barvení obrázku (nepravé barvy). Tato aplikace se skládá ze dvou základních částí. V horní části budou umístěny ovládací prvky, a v centru obrazovky se zobrazí původní obrázek a výsledek barvení.

### Barvení šedotónových obrazů (nepravě barvy) - applet

Výber obrázku

Název

Proved' převod

Výber barvici palety

Název



Nastavit gamma korekci

Další nastavení

Obr. 2.11 Navrh appletu "Barvení šedotónového obrazu"

V horní části obrazovky budou dva seznamy, jeden bude sloužit k výběru obrázku, druhý k výběru barevné palety. Také tam bude tlačítko, které bude použito po zmáčknutí kterého dojde k barvení vybraného obrazu s použitím vybrané barevné palety.

### Barvení šedotónových obrazů (nepravé barvy) - applet


Výber obrázku

Název

Proved' převod

Výber barvici palety

Název



Obr. 2.12 Ovládací prvky appletu „Barvení šedotónových obrazů“

V dolní části aplikace bude ovládací prvek, který umožňuje nastavit hodnotu gamma korekce, a také blok s další parametry, které uživatel může změnit. Tento blok bude k dispozici pouze v případě, že vybraná barevná paleta to dovoluje.



**Obr. 2.13 Gamma korekce a doplňkové nastavení appletu "Barvení šedotónových obrazů"**

Ve centru obrazovky budou umístěny dva obrázky. Obrázek na levé straně bude ilustrovat původní obrázek, a obraz na pravé straně se zobrazí konečný výsledek.



**Obr. 2.14 Demonstrační část appletu "Barvení šedotónových obrazů"**

## 3. POPIS ŘEŠENÍ

Vytváření aplikací bylo rozhodnuto začít s appletu „Míchání barev v různých barevných modelech“, protože funkcí převodu mezi barevnými modely v budoucnu by mohly být použité v jiných aplikacích.

### 3.1 Míchání barev v různých modelech

Aplikace se skládá z mnoha součástí, které mohou být rozděleny do hlavních částí. První je pět bloků, z nichž každý představuje barevný model (RGB, CMY, HSL, Lab, YCbCr), druhá část je obdélníková oblast, ve které se zobrazí výsledná barva.

Prvním krokem při budování aplikace bylo postavit kostru. Každá komponenta barevného modelu se skládá ze třech šoupátek, kterými uživatel může změnit hodnotu barevného modelu a tři pole, které se používají k zadání hodnoty barevného modelu.

Aplikace bylo rozhodnuto sestavit tak, aby byly vybrány tři globální proměnné (protože každý model se skládá ze třech kanálů), které pak budou přepočítány do hodnot nezbytných pro každý model. Hodnoty červeného, zeleného a modrého kanálů v modelu RGB byly vybrány jako základní hodnoty:

- `redValue`,
- `greenValue`,
- `blueValue`.

První komponenta je barevný model bloku RGB. Funkce tohoto bloku je sestavena tak, že když uživatel přesune některé z šoupátek nebo zadá libovolnou hodnotu do pole pro zadání dat, funkce `HandleChange` určuje, které šoupátko bylo přesunuto nebo do kterého pole bylo zadána nova hodnota a zapíše tuto hodnotu do globální proměnné. Tato funkce také kontroluje, aby hodnoty zadané do vstupního pole nepřekročily maximální povolené hodnoty. Funkčnost RGB barevného modelu bloku se mírně liší od ostatních bloků barevných modelů, protože obsahuje barevné pruhy, které ukazují, jak se mění celková barva při přesunutí šoupátka. Každý z pásů se mění v závislosti na skutečné hodnotě proměnných primárních barev. Tato funkce je implementována pomocí jazyku kaskádového stylu CSS.

Dalším blokem je barevný blok modelu CMY. Funkce tohoto bloku je zcela totožná s blokem RGB. Rozdíl je v tom, že při zadávání tohoto bloku jsou globální proměnné převedeny na hodnoty barevného modelu CMY pomocí funkce `convertRgbToCmy`.

Překlad hodnot mezi modely CMY a RGB se provádí odečtením odpovídající hodnoty od 255. Například přeložit hodnotu kanálu R z modelu RGB do kanálu C modelu CMY, musíme odečíst hodnotu kanálu R od 255.

Třetí blok je pro barevný model HSL. Převod hodnot z modelu RGB na model HSL se provádí pomocí funkce `rgb2hsl`. Na vstup funkce přichází tři proměnné a vrátí funkce pole hodnot, které obsahují hodnoty pro model HSL.

### Výpis kódu 3.1: `clrmodels.js` - funkce převodu hodnot z modelu RGB do modelu HSL

```
rgb2hsl(rV, gV, bV) {  
  // Deklarace proměnných  
  let H = 0;  
  let S = 0;  
  let L = 0;  
  let red = (rV / 255);  
  let green = (gV / 255);  
  let blue = (bV / 255);  
  
  // Výpočet maximální a minimální hodnoty, hodnoty delta  
  let minValue = Math.min(red, green, blue);  
  let maxValue = Math.max(red, green, blue);  
  let delta = maxValue - minValue;  
  
  // Vypočet jasové hodnoty L  
  L = (minValue + maxValue) / 2;  
  
  // Vypočet hodnot H a S pro model HSL  
  if(delta === 0) {  
    H = 0;  
    S = 0;  
  } else {  
    if ( L < 0.5 ) {  
      S = delta / (maxValue + minValue);  
    } else {  
      S = delta / (2 - maxValue - minValue);  
    }  
    let del_R = ((maxValue-red)/6) + (delta/2))/delta;  
    let del_G = ((maxValue-green)/6) + (delta/2))/delta;  
    let del_B = ((maxValue-blue)/6) + (delta/2))/delta;  
  
    if(red === maxValue) {  
      H = del_B - del_G;  
    }  
  }  
}
```

```

    } else if (green === maxValue) {
        H = ( 1 / 3 ) + del_R - del_B;
    } else if (blue === maxValue) {
        H = ( 2 / 3 ) + del_G - del_R;
    }

    if (H < 0) H = 1;
    if (H > 1) H = 1;
}

return [
    Math.round(360 * H).toString(),
    Math.round(S * 100).toString(),
    Math.round(L * 100).toString()
];
}

```

Zpětný překlad hodnot z modelů HSL do modelu RGB se provádí pomocí funkce `hsl2rgb` a `hue2rgb`.

Další blok je pro barevný model Lab. překlad hodnot pro model Lab a zpět se neprochází přímo, ale prostřednictvím barevného modelu XYZ. Funkce `rgb2xyz` a `xyz2lab` se používají k překladu hodnoty z modelu RGB do modelu Lab a `lab2xyz` a `xyz2rgb` funkce se používají pro zpětný překlad.

Poslední blok barevných modelů je určen pro barevný model YCbCr. Převod hodnot z modelu RGB na hodnoty určené pro model YCbCr se vyskytují ve funkci `rgb2ycbcr`. Zpětný převod hodnot je implementován přímo ve funkci `handleChangeYCbCr`, která sleduje pohyb šoupátek a změny hodnot v polích pro zadávání dat.

## 3.2 Paletova (indexová) reprezentace obrazu

Tento applet slouží jako vizuální příklad reprezentace obrazu v indexovém modu. Obrazové body neobsahují přímo hodnotu barvy, ale obsahují odkaz na index v paletě barev. V appletu uživatel může vybrat barevné hodnoty z poskytnutého seznamu, pak je zapsat do dvourozměrné matice a získat konečný výsledek.

Applet lze rozdělit do několika bloků. První blok je volba velikosti matice. Uživatel má možnost vybrat ze 3 možností (3 x 3, 5 x 5 nebo 7 x 7). Aplikace pomocí funkce `handleChangeMatrixSize` kontroluje změny v tomto bloku. Tato funkce řídí změny v tomto bloku a pokud uživatel zvolí jinou velikost matice, funkce zapíše tuto hodnotu do globální proměnné `matrixSize`.



Druhý blok představuje paletu barev, ve kterém je konkrétní hodnotě přiřazena konkrétní barva. Funkce `handleChangeColorValue` kontroluje změny v tomto bloku. Kdy uživatel vybere barvu v poli, funkce upravuje globální proměnu `colorValues`. Tato proměnná je pole hodnot, ve kterých jsou zapsány hodnoty palety barev.

Dále je blok skládající se ze dvou tlačítek "Reset" a "Náhodné". Pomocí tlačítka Reset funkce vrátí všechny hodnoty matice a vybrané barvy v paletě barev na původní. Tlačítko "Náhodné" slouží k vyplnění matice náhodnými hodnotami z možného rozsahu i náhodně přiřadí barvy v bloku pro výběr barvy. Funkce tohoto tlačítka je funkce `randomValues`, která používá standardní metodu `Random` pro vyber náhodných hodnot.

Další blok je matice, do které uživatel zadá číselné hodnoty indexů barev. Funkce `handleChangeValue` řídí změny v matici a zavádí je do globální proměnné `matrixValues`. Tato funkce také kontroluje, že vstupní data uživatele byly platné.

Poslední blok je konečna matice, která na základě všech globálních proměnných ukazuje konečný výsledek. Cyklus porovná hodnoty zadané do numerické matice s barevnými hodnotami v paletě barev a vytvoří na ní výslednou matici.

### 3.3 Převod barevného obrazu na šedotónový

Tento applet slouží jako příklad překlada barevného obrazu do obrazu ve stupních šedi. Applet se skládá ze 3 hlavních částí.

První část je dva obrázky. Na levé straně je původní obraz v barvě, a na pravé straně konečný obraz v odstínech šedé. Níže jsou bloky výběru obrazu a volba transformační metody. Poslední blok je skrytý panel s dodatečným nastavením parametrů, tento blok se objeví pouze v případě, že je vybrána určitá metoda, v tomto případě metoda číslo dvě "Míchání R, G, B". Tento skrytý panel má tři šoupátka, které umožňují změnit hodnoty proměnných. Každá hodnota je zadána v procentech.

V appletu byly implementovány tři metody překlada obrazu do odstínu šedé:

- Průměrování R, G, B,
- Míchání R, G, B,
- Překlad do Lab.

Práce s daty obrazu se provádí pomocí tagu `<canvas>`, který umožňuje pracovat s obrazem pixel po pixelu. Tag `<canvas>` je elementem jazyka HTML, ale práce s daty se provádí pomocí jazyka JavaScript. Tento tag možno použít k získání všech barevných hodnot každého pixelu v obraze. Tato data budou reprezentována

jako pole hodnot, ve kterém jsou každé čtyři hodnoty jsou barvou obrazového bodu, kde první hodnotou je červený kanál, druhá hodnota je zelený kanál, třetí je modrý kanál a čtvrtá je hodnota průhlednosti obrazových bodů.

### 3.3.1 Průměrování R, G, B

Tato metoda, jak je vidět z názvu, je založena na průměrování jednotlivých kanálů barevného modelu RGB. To znamená, že každý kanál bude přetypován na stejnou váhu 33,3 procenta.

Metoda v aplikaci implementována pomocí funkce `averagingMethod`. Tato funkce načítá data všech obrazových bodů původního obrazu a pak pomocí cyklu každý z kanálů přiřadí průměrnou hodnotu všech třech RGB kanálů každého obrazového bodu, hodnota průhlednosti je nastavena na 255 (což odpovídá plné neprůhlednost pixelu). Funkce vrátí pole hodnot každého obrazového bodu, které se zobrazí na obrazovce.

Výpis kódu 3.2: `clr2gray.js` - funkce průměrování R, G, B

```
averagingMethod(data) {
  let imgData = data; // Deklarace proměnné

  // Cyklus, který prochází všemi hodnotami datového pole
  for(let i = 0; i < imgData.data.length; i = i + 4) {
    // Průměrování každého kanálu modelu RGB
    let averageValue = Math.round((imgData.data[i]
      + imgData.data[i + 1]
      + imgData.data[i + 2]) * 0.33);
    // Přiřazení nové hodnoty každému kanálu
    imgData.data[i]=imgData.data[i + 1]
      = imgData.data[i + 2]
      = averageValue;
    imgData.data[i + 3] = 255;
  }

  return imgData;
}
```

### 3.3.2 Míchání R, G, B

Tato metoda je založena na tom, že každý kanál barevného modelu RGB má svou intenzitu *i* může mít hodnotu od 0 do 100 procentu. Stejně jako v předchozí metodě obraz zpracován pixel po pixelu. Hodnota každého kanálu je vynásobena

přiřazenou jí intenzitou, pak všechny hodnoty se skládají a výsledek přiřazen ke každému kanálu. Výsledné hodnoty nahradí původní.

Metoda apletu je implementována pomocí funkce `mixingRGB`, která ze vstupu načítá data všech obrazových bodů jako pole, pomocí cyklu kanyly každého obrazového bodů vynásobený přiřazenou intenzitou, pak sečtený a přiřazený ke každému kanálu. Funkce vrací nové hodnoty kanálu každého obrazového bodu jako pole.

### Výpis kódu 3.3: `clr2gray.js` – funkce míchaní R, G, B

```
mixingRGB(data) {
    // Deklarace proměnné a přiřazení jí hodnoty intenzit
    kanálu
    let optionsValues = this.props.optionsValues;
    // Deklarace proměnné a přiřazení hodnot kanálu pro
    každý obrazový bod
    let imgData = data;

    // Cyklus zpracování každého pixelu
    for (let i = 0; i < imgData.data.length; i = i + 4) {
    // Násobení hodnot kanálů na přiřazenou intenzitu
        imgData.data[i]=imgData.data[i]
            *(Number(optionsValues[0]) / 100);
        imgData.data[i+1]=imgData.data[i+1]
            *(Number(optionsValues[1])/100);
        imgData.data[i+2]=imgData.data[i+2]
            *(Number(optionsValues[2])/100);
    // Součet všech hodnot kanálu
        let value = imgData.data[i]
            + imgData.data[i+1]
            + imgData.data[i+2];
    // Přiřazení nových hodnot kanálem
        imgData.data[i] = imgData.data[i+1]
            = imgData.data[i + 2] = value;
        imgData.data[3] = 255;
    }
    return imgData;
}
```

### 3.3.3 Převod do Lab

Tato metoda je založena na překladu hodnot kanálu barevného modelu RGB do barevného modelu Lab, po kterém se zobrazují pouze kanál jasu L. Jak je uvedeno

výše v kapitole 3,1, překlad hodnot mezi barevnými modely RGB a Lab prochází přes model XYZ.

Metoda implementována pomocí funkce `convertToLab`. V této funkci jsou hodnoty barev každého obrazového bodu přeloženy do barevného modelu Lab, kanálem `a` i `b` jsou přiřazeny hodnoty 0, výsledek je přeložen zpět do modelu RGB, aby se zobrazil výsledný obrázek. Překladové funkce mezi barevnými modely použité z appletu "Míchání barev v různých barevných modelech".

**Výpis kódu 3.4: `clr2gray.js` – funkce pro získávání hodnot kanálu L, barevného modelu Lab**

```
convertToLab(data) {  
    // Deklarace proměnné a přiřazení jí všech hodnot  
    každého pixelu  
    let imgData = data;  
  
    for (let i = 0; i < imgData.data.length; i = i + 4)  
    {  
        // Převod hodnot z modelu RGB do modelu XYZ  
        let XYZ=this.rgb2xyz(  
            imgData.data[i],  
            imgData.data[i+1],  
            imgData.data[i + 2]  
        );  
        // Převod hodnot do modelu Lab  
        let Lab = this.xyz2lab(XYZ);  
        // Získání hodnot kanálu L  
        XYZ = this.lab2xyz(Lab[0], 0, 0);  
        // Převod hodnot do modelu RGB  
        let RGB = this.xyz2rgb(XYZ);  
  
        // Přiřazení nových hodnot  
        imgData.data[i] = RGB[0];  
        imgData.data[i + 1] = RGB[1];  
        imgData.data[i + 2] = RGB[2];  
        imgData.data[i + 3] = 255;  
    }  
  
    return imgData;  
}
```

### 3.4 Barvení šedotónových obrazů (nepravé barvy)

V některých oblastech jsou získaná data reprezentována jako černobíle obrázky. Jako příklad je možno uvést rentgenové nebo sonografické snímky v medicíně. Nicméně, tyto obrázky jsou reprezentované pouze v odstínech jedné barvy, v praxi se jedná o odstínech šedé barvy. Pro prozkoumání podrobnosti snímku a zvýšení pravděpodobnost správného dešifrování dat, používají metodu nepravých barev[4].

Tato aplikace pomůže na příkladech pochopit princip metody nepravých barev.

Aplikace se skládá ze dvou hlavních částí. První část obsahuje ovládací prvky. První rozbalovací nabídka se používá k výběru snímku nebo obrazu. Druhým se používá pro výběr metody barvení k vytvoření výsledného obrazu. Také zde je skrytý prvek, ke kterému dochází pouze při výběru konkrétní metody, v tomto případě to je metoda „Lineární gradient“. Druhá část se skládá ze dvou oken v každém z nich je obraz a paleta barev použitých na obrázku. V levé části je počáteční obrázek, v pravé – zpracovaná verze.

Princip aplikace je založen na nahrazení původních barevných hodnot snímku hodnotami barev vybrané metody. Stejně jako v kapitole 3.3 byl použit tag <canvas> umožňující pracovat s obrazem pixel po pixelu.

Každý bod obrazu v odstínech šedi, stejně jako barevný obraz je reprezentován jako čtyři hodnoty. První tři jsou hodnoty tří kanálů červeného (R), zeleného (G) a modrého (B). Hodnoty z prvních tří kanálů jsou stejné a mohou mít hodnoty v rozsahu od 0 do 255. Poslední, čtvrtá, je hodnota průhlednosti v obrazových bodech. Chcete-li nahradit původní hodnotu barevných kanálů, aplikace vypočítá 256 barevných hodnot podle zvolené barevné palety, pokud je nutné přeloží hodnoty do modelu RGB a nahradí základní hodnoty.

#### 3.4.1 Lineární gradient

Když uživatel vybere barevnou paletu „Lineární gradient“ v ovládací části aplikace, zobrazí se okno pro zadání dvou hodnoty. Toto je počáteční a koncové hodnoty lineárního gradientu. Aplikace zpočátku vypočítá lineární gradient se základních hodnot a pak sleduje změny ve vstupních polích. Když uživatel zadá nové hodnoty, program vypočítá lineární gradient, potom vypočte 256 hodnot tohoto gradientu, porovná pixel po pixelu hodnoty kanálu původního obrazu a nahradí je novými hodnotami. Funkce také kontroluje aby hodnoty zadané v polích byly v rozsahu platných čísel.

### **3.4.2 HSL**

Druhá barevná paleta představuje barevný model HSL. Pro získání hodnot barevného modelu, aplikace vypočítá požadovaný počet hodnot v barevném modelu HSL, pak převede tyto hodnoty do barevného modelu RGB. Při překladu hodnoty z jednoho barevného modelu do druhého, aplikace používá funkce z aplikace „Míchání barev v různých barevných modelech“.

### **3.4.3 Jet**

Barevná paleta Jet je hladký přechod z tmavě modré na modrou, dále na zelenou, žlutou a končí červenou barvou. Takový hladký přechod lze získat z barevného modelu HSL, ale použít necelé měřítko kanálu H. Pro paletu barev stačí použít hodnoty v rozsahu od 0 do 256 barevného modelu HSL. Stejně jako u předchozích barevných palet aplikace vypočítá požadovaný počet barevných hodnot, převede je do barevného modelu RGB a nahradí původní hodnoty výslednými.

### **3.4.4 COOL**

Barevná paleta COOL představuje odstíny modré a purpurové barvy, nebo lineární gradient který hladké se přechází z modré barvy do purpurové. Princip barvení původního obrazu je totožný s předchozími metodami.

### **3.4.5 HOT**

Tato barevná paleta je zcela totožná s barevnou paletou COOL, s výjimkou, že používá barvy procházející z černé na červenou, pak na žlutou a na konci bílou barvou.

# ZÁVĚR

Cílem této bakalářské práce bylo prostudovat problematiku související s aplikacemi pro podporu výuky, předložit návrhy a vytvářet tyto aplikací. Všechny aplikace byly vytvořeny s pomocí HTML, CSS a JavaScript jazyky, takže tyto applety mohou být použity jako samostatné aplikace nebo vložené do již připravených webových stránek. K napsání kódu byla použita knihovna React.

První kapitola popisuje základy teorie barev a tvorby rastrových obrázků, základní pojmy vztahující se k této oblasti. Byly prozkoumány takové pojmy jako barevný tón, světlost a sytost. Dále byly popsány chromatické a achromatické barvy, principy míchání barev. Bylo vysvětleno, jak barvy mohou být prezentovány v počítači. Byly prozkoumány různé barevné modely.

Druhá kapitola je věnována grafickému návrhu všech aplikací, které měly být vytvořeny v rámci této práce, jakož i volbě jazyka pro psaní appletů a popisování výhod těchto jazyků. Byly prezentovány návrhy pro applety „Míchání barev v různých barevných modelech“, „Paletova (indexová) reprezentace obrazu“, „Převod barevného obrazu na šedotónový“ a „Barvení šedotónového obrazu (nepravé barvy)“. Byly prezentovány stručně popis, struktura a funkčnost každého z appletů.

Třetí kapitola je věnována výhradně popisu hotových řešení. O každém z hotových aplikací bylo řečeno o hlavní bloky, principy funkčnosti každé aplikace a funkčnost jednotlivých částí. Byly popsány některé funkce a data se kterými pracovali a co se vracely. Také byly uvedeny oddělené fragmenty kódu, které ukazují princip práce některých funkcí.

# Literatura

- [1] DANNHOFEROVÁ, Jana. *Velká kniha barev: kompletní průvodce pro grafiky, fotografie a designéry*. Brno: Computer Press, 2012. ISBN 978-80-251-3785-7.
- [2] ŽÁRA, Jiří, Bedřich BENEŠ, Jiří SOCHOR a Petr FELKEL. *Moderní počítačová grafika*. 2. přepracované a rozšířené vydání. Brno: Computer Press, 2010. ISBN 80-251-0454-0.
- [3] BOLEČEK, Libor. Program pro zobrazení černobílých snímků v nepravých barvách (Pseudo-colour Paging of the Monochromatic Picture). *Elektrorevue* [online]. 2010, 20.12.2010, **2010**(125), 9 [cit. 2018-05-28]. Dostupné z: <http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/15/program-pro-zobrazeni-cernobilych-snimku-v-nepravych-barvach/>
- [4] Rastrová grafika. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-12-08]. Dostupné z: [https://cs.wikipedia.org/wiki/Rastrová\\_grafika](https://cs.wikipedia.org/wiki/Rastrová_grafika).
- [5] YCbCr. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-12-11]. Dostupné z: <https://cs.wikipedia.org/wiki/YCbCr>
- [6] SRGB. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-01-28]. Dostupné z: <https://cs.wikipedia.org/wiki/SRGB>
- [7] Adobe RGB color space. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-01-28]. Dostupné z: [https://en.wikipedia.org/wiki/Adobe\\_RGB\\_color\\_space](https://en.wikipedia.org/wiki/Adobe_RGB_color_space)
- [8] ProPhoto RGB color space. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-01-28]. Dostupné z: [https://en.wikipedia.org/wiki/ProPhoto\\_RGB\\_color\\_space](https://en.wikipedia.org/wiki/ProPhoto_RGB_color_space)
- [9] Bitmapová grafika. Wikisofia [online]. Praha [cit. 2018-01-30]. Dostupné z: [https://wikisofia.cz/wiki/Bitmapov%C3%A1\\_grafika](https://wikisofia.cz/wiki/Bitmapov%C3%A1_grafika)



## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RGB	<i>barevný model neboli Red, Green, Blue</i>
CMY	<i>barevný model neboli Cyan, Magenta, Yellow</i>
HSL	<i>barevný model neboli Hue, Saturation, Lightness</i>
Lab	<i>barevný model neboli Lightness, osa zelená – červená, osa modrá – žlutá</i>
YCbCr	<i>barevný model neboli Luminance, modrý a zelený chrominanční komponenty</i>
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>

# A OBSAH ELEKTRONICKÉ PŘILOHY

Na přiloženém CD se nachází elektronická verze této bakalářské práce ve formátu PDF, zdrojové kódy appletů. Struktura CD je následující:

- BakalarskaPrace.pdf
  - zdroj – složka se zdrojovými kódy
    - node\_modules – složka s moduly a knihovnou React
    - public – složka s hlavní stránkou appletů
      - favicon.ico – ikona
      - index.html – hlavní stránka appletů
      - manifest.json – JSON-soubor pro WebExtension
    - src – složka se zdrojovými kódy appletů
      - img – složka s obřízkami appletů
      - clr2gray.js – zdrojový kód apletu převod barevného obrazu
      - clrmodels.js – zdrojový kód apletu míchaní barev
      - gray2clr.js – zdrojový kód apletu nepravé barvy
      - index.css – kaskádové styly
      - index.js – zdrojový kód rozcestníku
      - matrix.js – zdrojový kód apletu paletová reprezentace obrazu
  - package-lock.json – seznam všech modulu a jejich verzí
  - package.json – informace o aplikaci
- Aplety jsou umístěny na internetu pod následujícím odkazem:
- <http://www.stud.feec.vutbr.cz/~xkhala00/>