

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce

## Fotogalerie s generátorem webstránek

**Petr Cipra**

Vedoucí: prof. Ing. Vlastimil Havran, Ph.D.

Obor: Otevřená informatika

Studijní program: Software

Srpen 2021



## Poděkování

Tímto bych rád poděkoval prof. Ing. Vlastimilu Havranovi, Ph.D. za užitečné a cenné rady a vstřícnost při vypracovávání této práce.

## Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu. Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, 13. srpna 2021

## Abstrakt

Tato práce se zabývá návrhem a implementací aplikace v jazyce C++ pro správu obrázků, která umožňuje jejich export do webové galerie. Dále také analýzou podobných již existujících programů a analýzou C++ knihoven pro tvoření grafických rozhraní. Výsledná aplikace umožňuje uživateli obrázky importovat i z adresářů a upravovat je přímo v aplikaci. Vygenerovaná galerie je přizpůsobitelná a kompatibilní s HTML 4.01 strict.

**Klíčová slova:** Webová galerie, Obrázky, Úprava obrázků, C++, HTML

**Vedoucí:** prof. Ing. Vlastimil Havran, Ph.D.  
Praha 2, Karlovo náměstí 13, E-420

## Abstract

This thesis deals with designing and implementation of an application written in C++ for image management, which allows the images to be exported to a web gallery. It also focuses on the analysis of similar existing programs and the analysis of C++ libraries for creating graphical interfaces. The resulting application allows the user to import images from directories and edit them directly in the application. The generated gallery is customizable and compatible with HTML 4.01 strict.

**Keywords:** Web gallery, Images, Image editing, C++, HTML

**Title translation:** Photogallery with webpages generator

## Obsah

<b>Zadání práce</b>	<b>1</b>	5.1.1 Obrázky programu . . . . .	19
<b>1 Úvod</b>	<b>3</b>	5.2 Easy Website Photo Gallery . . . . .	21
1.1 Cíl této práce . . . . .	3	5.2.1 Obrázky programu . . . . .	22
<b>2 Obrázkové formáty</b>	<b>5</b>	5.3 HTTPPhotos . . . . .	23
2.1 Formát JPEG/EXIF . . . . .	5	5.3.1 Obrázky programu . . . . .	24
2.2 Formát TIFF . . . . .	5	5.4 LightBox Video Web Gallery Creator . . . . .	25
2.3 Formát PNG . . . . .	6	5.4.1 Obrázky programu . . . . .	27
<b>3 EXIF metadata</b>	<b>7</b>	5.5 iWebAlbum . . . . .	28
3.1 Obrázky typu JPEG/EXIF . . . . .	9	5.5.1 Obrázky programu . . . . .	29
3.2 Obrázky typu TIFF . . . . .	11	5.6 Web Album Generator . . . . .	30
3.3 Obrázky typu PNG . . . . .	13	5.6.1 Obrázky programu . . . . .	32
3.4 Extrahovaná EXIF metadata . . . . .	14	5.7 Visual LightBox . . . . .	33
<b>4 Hypertext Markup Language (HTML)</b>	<b>15</b>	5.7.1 Obrázky programu . . . . .	35
<b>5 Analýza programů</b>	<b>17</b>	<b>6 Shrnutí analýzy programů</b>	<b>39</b>
5.1 jAlbum . . . . .	17	<b>7 Analýza C++ GUI knihoven</b>	<b>43</b>
		7.1 Dear ImGui . . . . .	43
		7.1.1 Obrázky . . . . .	44

7.2 NanoGUI .....	45	9.3.2 Implementace úprav obrázku a efektů .....	60
7.2.1 Obrázky .....	46	9.4 Přizpůsobení výsledné webové galerie .....	64
7.3 FLTK .....	47	9.4.1 Šablona .....	64
7.3.1 Obrázky .....	48	9.4.2 info.txt .....	64
7.4 wxWidgets .....	49	9.4.3 Formát textových vzorů pro dynamická data .....	65
7.4.1 Obrázky .....	50	9.4.4 Šablonové soubory .....	65
7.5 Agar .....	51	9.4.5 CSS galerie .....	66
7.5.1 Obrázky .....	52	9.4.6 Skripty (JS) galerie .....	66
<b>8 Shrnutí analýzy C++ GUI knihoven</b>	<b>53</b>	9.4.7 Uživatelské rozhraní .....	66
<b>9 Návrh implementace</b>	<b>55</b>	9.5 Vytvoření galerie .....	67
9.1 Import obrázků .....	55	9.5.1 Proces vytvoření .....	67
9.1.1 Možnosti importu .....	55	9.6 Okno pro správu stylů .....	68
9.1.2 Proces importu .....	56	9.7 Okno pro správu barev .....	68
9.1.3 Získávání informací ze struktury složek .....	57	9.8 Okno pro správu textů .....	68
9.2 Výpis obrázků .....	58	9.9 Vygenerování galerie .....	68
9.3 Editace obrázku .....	59	9.9.1 Uživatelské rozhraní .....	69
9.3.1 Uživatelské rozhraní .....	59		

<b>10 Grafický návrh programu</b>	<b>71</b>	11.7.4 info.txt .....	87
<b>11 Implementace programu</b>	<b>77</b>	11.7.5 settings.txt .....	88
11.1 Použité knihovny .....	77	11.7.6 Proces generování .....	88
11.2 Model .....	78	<b>12 Testování programu</b>	<b>89</b>
11.3 Import obrázků .....	78	12.1 Zpětná vazba .....	89
11.3.1 Výběr souborů .....	79	12.2 Navržené úpravy .....	90
11.3.2 Výber složky .....	79	12.3 Opětovné testování .....	91
11.4 Proces importování .....	79	<b>13 Závěr</b>	<b>93</b>
11.5 Výpis obrázků .....	81	13.1 Rozšíření práce .....	94
11.5.1 Hierarchie položek .....	82	<b>A Literatura</b>	<b>95</b>
11.5.2 Náhledový obrázek složky ..	83		
11.6 Editace obrázků .....	83		
11.6.1 Obdélníkové oříznutí .....	84		
11.6.2 Odstranění červených očí ..	84		
11.7 Generování webové galerie ....	84		
11.7.1 Textové vzory .....	85		
11.7.2 Šablony .....	85		
11.7.3 Téma .....	86		

## Obrázky

5.1 jAlbum, uživatelské rozhraní - výpis obrázků . . . . .	19	5.12 Web Album Generator, uživatelské rozhraní - výpis obrázků	32
5.2 jAlbum, uživatelské rozhraní - editace obrázku . . . . .	20	5.13 Web Album Generator, uživatelské rozhraní - generování galerie, sekce pro úpravu barev výsledné galerie . . . . .	33
5.3 jAlbum, uživatelské rozhraní - výsledná webová galerie . . . . .	20	5.14 Visual LightBox, uživatelské rozhraní - výpis obrázků . . . . .	35
5.4 Easy Website Photo Gallery, uživatelské rozhraní - výpis obrázků	22	5.15 Visual LightBox, uživatelské rozhraní - generování galerie, sekce pro úpravu vzhledu náhledových obrázků . . . . .	36
5.5 Easy Website Photo Gallery, uživatelské rozhraní - výsledná webová galerie . . . . .	23	5.16 Visual LightBox, uživatelské rozhraní - generování galerie, sekce pro výstup . . . . .	37
5.6 HTTPPhotos, uživatelské rozhraní - výpis obrázků . . . . .	24	7.1 Dear ImGui, demo . . . . .	44
5.7 HTTPPhotos, uživatelské rozhraní - editace obrázku . . . . .	25	7.2 Dear ImGui, prvek pro výběr barvy . . . . .	45
5.8 LightBox Video Web Gallery Creator, uživatelské rozhraní - hlavní okno programu . . . . .	27	7.3 NanoGUI, demo . . . . .	46
5.9 LightBox Video Web Gallery Creator, uživatelské rozhraní - výsledná webová galerie . . . . .	27	7.4 FLTK, demo . . . . .	48
5.10 iWebAlbum, uživatelské rozhraní - výpis obrázků . . . . .	29	7.5 FLTK, demo 2 . . . . .	48
5.11 iWebAlbum, uživatelské rozhraní - generování galerie . . . . .	30	7.6 wxWidgets, demo: Life . . . . .	50
		7.7 wxWidgets, demo: Bombs . . . . .	51
		7.8 Agar, demo: agartest . . . . .	52



10.1 Hlavní okno programu s výpisem obrázků .....	72
10.2 Import obrázků - dialogové okno pro výběr souborů .....	72
10.3 Import obrázků - dialogové okno pro výběr složky .....	73
10.4 Okno pro editaci obrázku .....	73
10.5 Okno pro editaci obrázku, s otevřeným menu .....	74
10.6 Editace obrázku - dialogové okno nastavení efektu .....	74
10.7 Okno přizpůsobení galerie .....	75
10.8 Okno vygenerování galerie .....	75

## Tabulky

6.1 Shrnutí analýzy programů, část 1.	40
6.2 Shrnutí analýzy programů, část 2.	41
8.1 Shrnutí analýzy C++ GUI knihoven .....	54



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Cipra** Jméno: **Petr** Osobní číslo: **483692**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Otevřená informatika**  
Specializace: **Software**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Fotogalerie s generátorem webstránek**

Název bakalářské práce anglicky:

**Photogallery with webpages generator**

Pokyny pro vypracování:

Analýzujte dostupné programy pro správu digitálních snímků se zaměřením na takové, které umožňují export do webových stránek ve formě fotoalba. Programy mezi sebou porovnejte z hlediska uživatelského a užitečných vlastností. Dále navrhnete a implementujete multiplatformní aplikaci v jazyku C++ pro správu fotografií s použitím knihovny QT. Musí podle pokynů zadavatele umožňovat import fotografií z adresářů, jednoduchou editaci (rotaci a výmaz, případně zvýšení jasu a kontrastu), anotaci textem a export do formátu HTML 4.01 strict. Případně volitelně do vyšší verze (XHTML). Aplikace bude umožňovat modifikaci vzhledu vytvářených stránek, velikost anotačního textu, velikost fotografie, data, pozadí atd. a vytvářet indexační stránku s orientačními obrázky atd. Aplikaci otestujte s použitím uživatelského testování alespoň tří osob, na základě zpětné vazby aplikaci vylepšete a znovu proveďte testování použitelnosti. Vytvořte ukázkové galerie a program umístěte do repozitáře sourceforge.

Seznam doporučené literatury:

- 1) Aplikace jAlbum, <https://jalbum.net/cs>
- 2) Aplikace Website Photo Gallery, <https://www.webgallerysoftware.com/>
- 3) Aplikace HTTPPhotos, <https://us.digicamsoft.com/softhttpphotos.html>
- 4) Lidwell, Holden, Butler, Universal Principles of Design, Revised and Updated: 115 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design ... Design Decisions, and Teach through Design , 2010, ISBN-10 : 1592535879.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**prof. Ing. Vlastimil Havran, Ph.D., Katedra počítačové grafiky a interakce**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **21.02.2021**

Termín odevzdání bakalářské práce: **13.08.2021**

Platnost zadání bakalářské práce: **19.02.2023**

prof. Ing. Vlastimil Havran, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



# Kapitola 1

## Úvod

Většina programů pro generování webových galerií nenabízí vysokou úroveň přizpůsobení výsledné webové galerie. Spousta z nich nabízí pouze několik šablon a nelze vše upravovat bez následné modifikace statických souborů po vygenerování galerie. Jen málokterý program také nabízí jednoduchou úpravu obrázků integrovanou přímo v programu.

Čtení, ukládání a úprava obrázkových dat s sebou přináší některé komplikace. Je potřeba prostudovat to, jak jednotlivé obrázkové formáty interně vypadají, aby se z nich dala extrahovat metadata. Je nutné postupně číst data ze souboru a logicky v souboru postupovat k požadovaným datům.

U ukládání dat do souborů HTML (Hypertext Markup Language) je nutné dát si pozor, aby výsledný obsah souboru představoval validní HTML 4.01 strict.



## 1.1 Cíl této práce

Cílem této práce je analyzovat již existující programy pro generování webových galerií, podobný program navrhnout a postupně projít veškeré záležitosti problematiky některých funkcí. Nakonec je nutné program i implementovat v programovacím jazyce C++.



## Kapitola 2

### Obrázkové formáty

Program musí umět pracovat s různými obrázkovými formáty, konkrétně alespoň s JPEG, TIFF a PNG. V této kapitole se nachází základní informace o daném obrázkovém formátu a rozdíly mezi nimi.

#### 2.1 Formát JPEG/EXIF[JPE]

JPEG/EXIF je obrázkový formát hojně využívaný pro kompresi obrázkových dat pomocí metody JPEG, především pro digitální fotografie. Podporuje možnost změnit úroveň komprese, za to ale nepodporuje průhlednost pixelů.

#### 2.2 Formát TIFF[TIF]

TIFF je flexibilní a přizpůsobitelný obrázkový formát pro zpracování obrázků a dat v rámci jednoho souboru. Podporuje bezztrátové ukládání obrázkových dat, lze ho použít i pro uchovávání JPEG obrázků a také umožňuje zahrnout tzv. \*hlavičkové značky\*.

## ■ 2.3 Formát PNG[PNGa]

PNG je rozšiřitelný obrázkový formát souboru pro bezztrátové, přenosné a komprimované úložiště rastrových obrázků. Podporuje například truecolor (až 48 bitů na pixel), plný alfa kanál (průhlednost) a detekci korupce souboru.



## Kapitola 3

### EXIF metadata

Program by měl umět získávat různé informace z obrázků během procesu importu. Konkrétně by měl program získávat metadata EXIF z obrázků alespoň těchto typů: *JPEG/JPG*, *TIFF* a *PNG*. Každý z těchto obrázkových formátů má odlišnou strukturu, nicméně rozebrání (tzv. parsování) EXIF metadat je u všech v zásadě stejné.<sup>[EXIa]</sup><sup>[SPEb]</sup><sup>[SPEa]</sup> Stačí tedy, aby program uměl pro každý z těchto obrázkových formátů extrahovat blok EXIF, jeho následné rozebrání se bude u všech pak provádět stejným způsobem.

**Poznámka:** V jednotlivých podsekcích této sekce se zabývám pouze extrakcí EXIF metadata a ne přímo obrázkových dat, ani jejich dekódování apod. Nejlépe by se problém čtení obrázkových dat řešil pomocí nějaké již existující knihovny, která tuto funkcionalitu podporuje. Tím by se dala přechíst jak obrázková data, tak metadata obrázku jedním průchodem datového proudu a zároveň by veškeré případné problémy a komplikace při čtení obrázkových dat byly vyřešeny.

**Příprava:** Mějme nějaký datový proud (stream) a následující funkce operující nad ním:

---

```
// Zkontroluje, zdali je v datovém proudu další v pořadí daná
// hodnota daného typu.
<Type> bool stream::hasNext(Type val);
// Přečte v datovém proudu následujících *length* bajtů a převede je
// do daného typu.
<Type> Type stream::readNext(uint32 length);
// Přeskočí v datovém proudu následujících *length* bajtů.
void stream::skip(uint32 length);
// Nastaví limit datového proudu na *length* bajtů. Jakmile je
// přečteno *length* bajtů, stream vyhodí EOF.
void stream::limit(uint32 length);
// Nastaví pořadí bajtů v datovém proudu. Hodnoty přečtené z
// datového proudu budou převáděny s pomocí tohoto pořadí. Např.:
// Datový proud má hodnoty 4D 2E ..., po zavolání
// readNext<uint16>(2), pokud je byteOrder == Byte_Order::LE, se
// vrátí hodnota 0x2E4D, pokud je byteOrder == Byte_Order::BE, se
// vrátí hodnota 0x4D2E.
void stream::setByteOrder(Byte_Order byteOrder);
// Posune se v datovém proudu na pozici *position*.
void stream::seek(uint32 position);
```

---

Mějme ještě definovány následující typy a funkci, která vezme nějaké argumenty a vytvoří z nich *EXIF\_Data*:

---

```
EXIF_Data_Entry {
    uint16 id, type;
    uint32 count, value;
}

EXIF_Data {
    vector<EXIF_Data_Entry> entries;
}

EXIF_Data create_EXIF_Data(/* argumenty */);
```

---

Tyto funkce a typy pak budou používány v pseudokódech pro extrakci EXIF metadat u jednotlivých typů obrázků.

## 3.1 Obrázky typu JPEG/EXIF

Podle specifikace[EXIa] je soubor strukturován pomocí tzv. markerů tvaru  $0xFFxx$ , kde  $xx$  je specifické číslo označující typ markeru. Za ním vždy (až na SOI a EOI markery, viz dále) následují 2 bajty pro délku obsahu (v bajtech) sekce označené daným markerem. Tato délka zahrnuje i tyto 2 bajty, takže je nutné přečíst pouze následujících  $N - 2$  bajtů, kde  $N$  je přečtená délka obsahu.

Soubor JPEG/EXIF začíná markerem **SOI** (Start Of Image) s hodnotou  $0xFFD8$  a končí markerem **EOI** (End Of Image) s hodnotou  $0xFFD9$ . Hledaná EXIF metadata jsou pak v sekci označené markerem **APP1** (Application marker 1) s hodnotou  $0xFFE1$ .

Pro extrakci EXIF metadat je tedy nutné přeskokovat sekce tak dlouho, dokud nepřečteme marker **APP1** a obsah sekce označené tímto markerem pak přečíst. Následně můžeme tyto data předat proceduře, která data rozebere. Konkrétně se v tomto případě jedná o stejnou proceduru, která čte soubor typu TIFF.

**Pseudokód:** Zde je ukázka, jak by funkce pro extrakci dat z obrázků typu JPEG/EXIF mohla vypadat (za pomoci předem definovaných funkcí, viz Příprava):

---

```
EXIF_Data parse_JPEG_EXIF(Stream stream) {
    // SOI (Start Of Image)
    if(!stream.hasNext<uint16>(0xFFD8)) return NULL;
    stream.skip(2); // Přeskočíme SOI
    bool found_app1 = false;
    while(!stream.isEOF())
        // Zdali je další v pořadí nějaký marker
        && stream.hasNext<uint8>(0xFF)) {
        uint8 marker = stream.readNext<uint8>(2);
        // Zdali jsme našli APP1 marker
        if(marker == 0xFFE1) {
            found_app1 = true;
            break; // Nemusíme dále přeskokovat markery
        }
        uint16 size = stream.readNext<uint16>(2);
        stream.skip(size - 2);
    }
    if(found_app1) {
        uint16 size = stream.readNext<uint16>(2);
        // Zdali se doopravdy jedná o EXIF blok
```

### 3. EXIF metadata

---

```
        if(!stream.hasNext<string>("Exif\0\0"))
            return NULL;
        // Přeskočíme EXIF header
        stream.skip("Exif\0\0".size());
        // Přečteme a rozebereme EXIF data
        return parse_TIFF_EXIF(stream, size);
    }
    // Jinak EXIF data nebyla nalezena
    return NULL;
}
```

---

## 3.2 Obrázky typu TIFF

Podle specifikace[SPEb] začíná soubor typu TIFF tzv. image file header o velikosti 8 bajtů. První 2 bajty říkají v jakém pořadí jsou bajty uspořádány, tedy zdali se jedná o little-endian (označeno jako "II-0x4949), nebo o big-endian (označeno jako "MM-0x4D4D). Dále budu uvažovat pouze variantu s pořadím bajtů v pořadí little-endian. Následují 2 bajty označující, že se jedná o TIFF soubor, konkrétně hodnota 42, v HEX 0x2A00. Další 4 bajty označují offset IDF (Image file directory), kde se nachází informace o obrázku. Offset je vzhledem k začátku souboru, tedy první bajt má offset 0.

**Image file directory:** Skládá se z tzv. IFD entries, tedy záznamů, kde každá obsahuje nějakou informaci o obrázku. První 2 bajty označují počet takovýchto záznamů, následuje jejich sekvence, která končí buď hodnotou offsetu na další IFD, nebo 4 bajty nul znamenající, že se jednalo o poslední IFD.

**IFD záznam (entry):** Skládá se z 12 bajtů - první 2 bajty je tag označující záznam, následují 2 bajty reprezentující typ záznamu, pak 2 bajty reprezentující počet hodnot a nakonec 4 bajty označující offset v souboru, kde se hodnota záznamu nachází, nebo rovnou hodnotu záznamu a to pouze tehdy, pokud se hodnota vejde do 4 bajtů. Pokud je hodnota menší než 2 bajty, pak je zarovnaná doleva (například hodnota 0x2A3C je uložena jako 0x2A3C0000).

**EXIF IFD záznamy:** Podle zdrojů[EXIa]·[SPEc]·[EXIb] se EXIF záznamy ukládají do tzv. Exif SubIFD, na kterou odkazuje TIFF Private tag s označením 0x6987. Právě tyto záznamy chceme z obrázku extrahovat. Formát dat v Exif SubIFD je stejný jako formát IFD.

**Pseudokód:** Zde je ukázka, jak by funkce pro extrakci dat z obrázků typu TIFF mohla vypadat (za pomoci předem definovaných funkcí, viz Příprava):

---

```

IFD_Entry {
    uint16 id, type;
    uint32 count, value;
}

IFD_Entry parse_IFD_Entry(Stream stream) {
    IFD_Entry entry = stream.readNext<IFD_Entry>(12);
    return entry;
}

vector<IFD_Entry> parse_IFD(Stream stream) {
    vector<IFD_Entry> entries;
    uint16 num_entries = stream.readNext<uint16>(2);
    // Přečteme všechny záznamy v IFD
    for(uint16 i = 0; i < num_entries; ++i) {
        IFD_Entry entry = parse_IFD_Entry(stream);
        entries.add(entry);
    }
    return entries;
}

EXIF_Data parse_TIFF_EXIF(Stream stream, uint32 limit) {
    if(limit > 0) stream.limit(limit);
    // Přečteme a nastavíme pořadí bajtů
    uint16 byte_order = stream.readNext<uint16>(2);
    if(byte_order == 0x4949) stream.setByteOrder(Byte_Order::LE);
    else stream.setByteOrder(Byte_Order::BE);
    // Offset první IFD
    uint32 offset = stream.readNext<uint32>(4);
    // Pro každou IFD (alespoň 1 musí existovat)
    do {
        stream.seek(offset);
        uint16 num_entries = stream.readNext<uint16>(2);
        // Přečteme všechny záznamy v IFD
        for(uint16 i = 0; i < num_entries; ++i) {
            IFD_Entry entry = parse_IFD_Entry(stream);
            if(entry.id == 0x6987) { // Jedná se o ExifOffset záznam
                stream.seek(entry.value); // Přejdeme na daný offset
                // Rozebereme EXIF záznamy
                vector<IFD_Entry> exif_entries = parse_IFD(stream);
                return create_EXIF_Data(exif_entries);
            }
        }
        // Posuneme se na další IFD
        offset = stream.readNext<uint32>(4);
    } while(offset != 0);
    // EXIF metadata nenalezena
    return NULL;
}

```

---

## 3.3 Obrázky typu PNG

Podle specifikace[PNGb] začíná soubor typu PNG tzv. PNG signature (8 bajtů) s hodnotou 0x89504E470D0A1A0A. Poté následuje sekvence tzv. PNG chunks. Každý chunk se skládá ze 4 částí: jeho velikost (4 bajty; jedná se pouze o velikost jeho dat, tedy bez hodnoty velikosti, typu a CRC), jeho typ (4 bajty), jeho data (o předešle uvedené velikosti) a kontrolní součet CRC (4 bajty).

**EXIF Chunk:** Jedním ze sekvence PNG chunks může být EXIF chunk[SPEa], který je označen typem 0x65584966 ("eXIf"). Tento chunk má stejný formát jako TIFF soubor, a tudíž pro jeho rozebrání lze použít stejnou proceduru jako pro rozebrání EXIF metadat u souboru TIFF.

**Pseudokód:** Zde je ukázka, jak by funkce pro extrakci dat z obrázků typu PNG mohla vypadat (za pomoci předem definovaných funkcí, viz Příprava):

---

```

EXIF_Data parse_PNG_EXIF(Stream stream) {
    // Zdali soubor obsahuje PNG signature
    if(!stream.hasNext<uint64>(0x89504E470D0A1A0A)) return NULL;
    // Pro každý PNG chunk (alespoň 1 musí existovat)
    do {
        uint32 chunk_length = stream.readNext<uint32>(4);
        uint32 chunk_type = stream.readNext<uint32>(4);
        if(chunk_type == 0x65584966) { // Jedná se o EXIF Chunk
            // Rozebereme EXIF chunk
            return parse_TIFF_EXIF(stream, chunk_length);
        }
        // Přeskočíme CRC (je možné ho případně využít pro kontrolu
        dat)
        stream.readNext<uint32>(4);
    } while(!stream.isEOF());
    // EXIF metadata nenalezena
    return NULL;
}

```

---

## 3.4 Extrahovaná EXIF metadata

Extrahovaná EXIF metadata (v pseudokódech jako *EXIF\_Data*) pak obsahují jednotlivé záznamy. U každého z nich je evidována identifikace záznamu, tzn. o jakou informaci se přesně jedná, a hodnota záznamu. U jednotlivých dat je hodnota reprezentována jinak, a tudíž, pokud by se tyto údaje měly nějak uživateli předávat, je nutné pro každou informaci specifikovat, v jakých jednotkách/jakém formátu by se měla zobrazovat. Například záznam *ExposureTime* je udáván v sekundách (číslo), ale záznam *DateTimeOriginal* je udáván jako text (ASCII). Řešením by mohlo být, že by se předem určená data (pokud v souboru existují) zobrazila správně formátovaná, a zbytek extrahovaných dat by se zobrazoval v tzv. RAW hodnotách, tedy v takových hodnotách, které byly nalezeny v souboru.





## Kapitola 4

# Hypertext Markup Language (HTML) [HTM]

HTML je standardní značkovací jazyk pro dokumenty určené k zobrazení ve webovém prohlížeči.

Základními bloky HTML dokumentu jsou HTML prvky. Každý dokument je strukturovaný pomocí těchto prvků, které jsou v kódu zapsány tzv. *\*tagy\**, jež jsou označeny pomocí špičatých závorek. Každý tag také může mít různé atributy, které dále specifikují vlastnosti daného prvku.

Častokrát je doprovázen technologiemi jako je CSS a JavaScript pro doručení lépe graficky vypadajícího a dynamičtějšího obsahu.



# Kapitola 5

## Analýza programů

V této sekci se nachází analýzy jednotlivých programů v příslušných podsekcích. Konkrétní body analýzy jsou (1) import obrázků do programu, (2) výpis obrázků v rozhraní programu, (3) editace obrázků, (4) přizpůsobení webové galerie, (5) výsledná webová galerie, případně další body specifické pro daný program.

### 5.1 jAlbum

**Verze:** 22.1

**Oficiální stránky:** <https://jalbum.net/cs/>

**Operační systém:** Windows, Mac, Linux (a další)

**Typ programu:** Placený (Free trial)

#### 1. Import obrázků

- Pomocí dialogového okna pro výběr souborů.
- Pomocí Drag and Drop.
- Lze specifikovat popis a tagy u obrázku.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.

#### 2. Výpis obrázků

- Změna pořadí pomocí Drag and Drop.
- Přejmenování pomocí dvojitého kliku.
- Možnost filtrace obrázků.
- Možnost vytváření hierarchie obrázků pomocí složek.

### 3. Editace obrázku

- Funkce: překlopení, otočení podle úhlu, ořez, auto-korekce, úprava jasů, kontrastu, sytosti a gamma.
- Efekty: červené oči, zaostření, rozmazání, stupně šedi, mozaika, inverze.
- Přidání textu do obrázku, jeho zarovnání, barva a font (včetně možnosti přidání stínu), náhled v reálném čase.
- Zobrazení histogramu při editaci.
- Jednoduché překlíkávání mezi obrázky v editačním módu pomocí šipek.

### 4. Přizpůsobení výsledné webové galerie

- Možnost vybrat si z již existujících šablon.
- Možnost změnit barvy pozadí stránky a navigace, odsazení, zaoblení rohů, fonty a barvy Lightboxu.
- Možnost přidat a odebrat sociální sítě u sdílení alba.
- Možnost vložit vlastní HTML do hlavičky a těla stránky, plus přidat i vlastní CSS a JS.
- Dodatečná správa SEO u obrázků a nadpisů ve webové galerii.

### 5. Výsledná webová galerie

- Zobrazení mřížky s obrázky a jejich popisky na indexové stránce.
- Rozklik obrázku pro vstup do slideshow.
- Možnost vyhledávat obrázky v galerii podle popisu či názvu.

### 6. Optimalizace obrázků

- Možnost zvolit kvalitu JPEG výsledného obrázku.

### 7. Vytvoření galerie

- Jedním kliknutím se album vytvoří.
- Spustí se i lokální server na portu 8080 pro zhlédnutí webové galerie.

## Hodnocení.

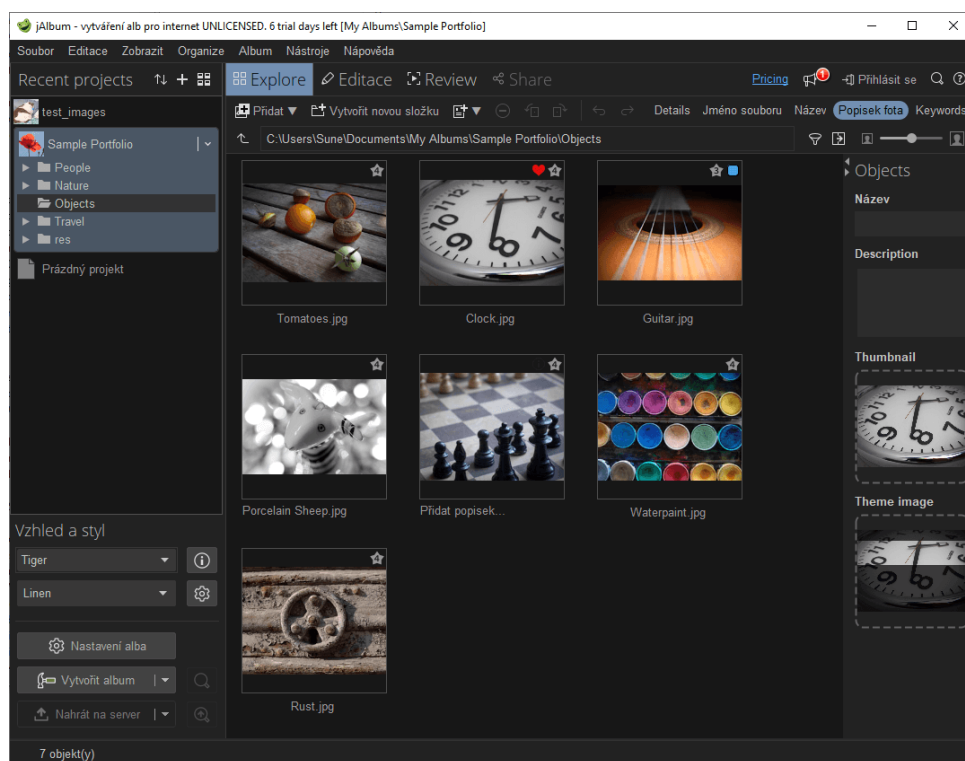
### 1. Co se mi líbí

- Intuitivnost rozhraní, i když program nabízí spoustu funkcí, není rozhraní příliš přehlcené.
- Jednoduchost změny barev a dalších nastavení vygenerované galerie.

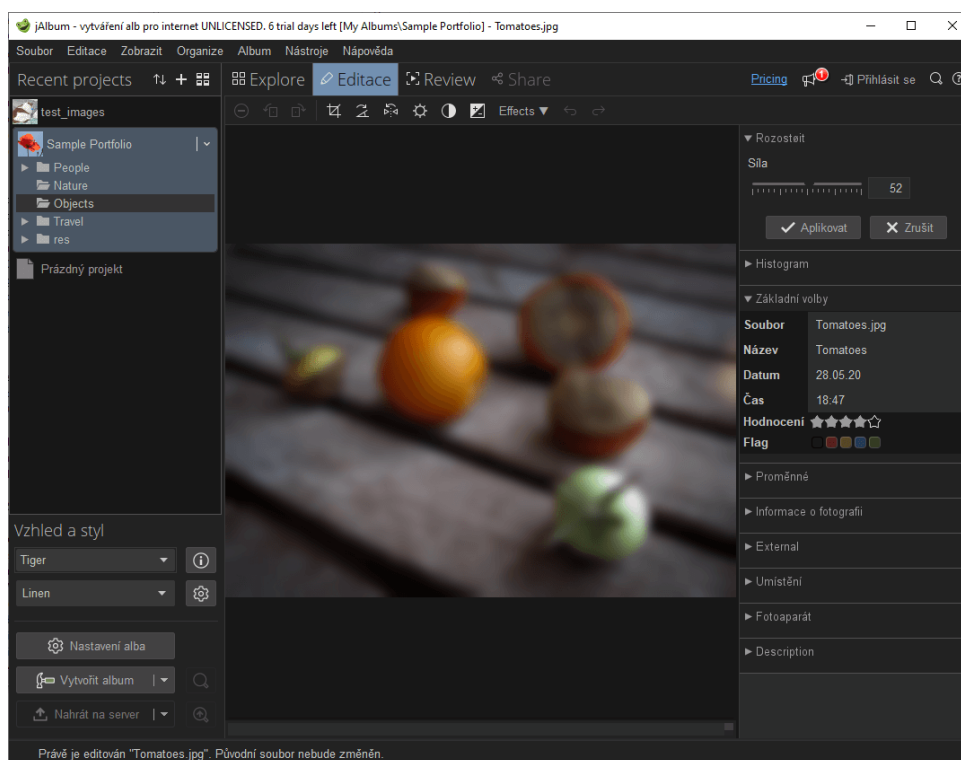
### 2. Co se mi nelíbí

- Složitě vypadající výsledná galerie, více efektů a přechodů než by bylo nutné (nenašel jsem možnost toto vypnout bez použití CSS).

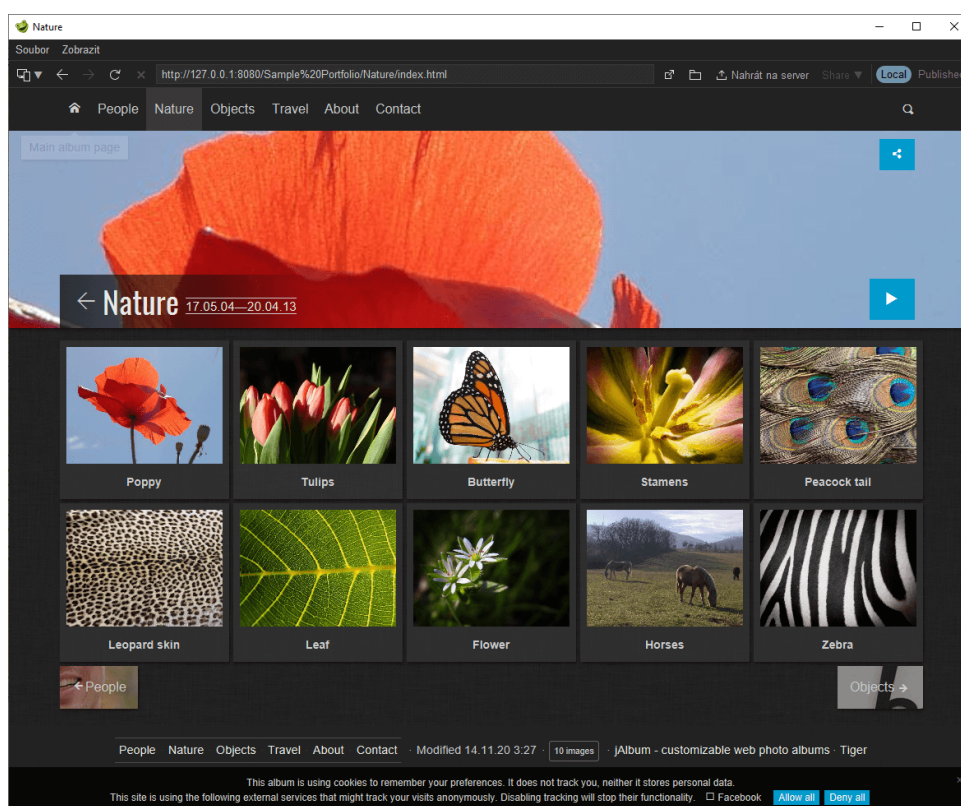
## 5.1.1 Obrázky programu



Obrázek 5.1: jAlbum, uživatelské rozhraní - výpis obrázků



Obrázek 5.2: jAlbum, uživatelské rozhraní - editace obrázku



Obrázek 5.3: jAlbum, uživatelské rozhraní - výsledná webová galerie

## 5.2 Easy Website Photo Gallery

**Verze:** 1.1

**Oficiální stránky:** <https://www.webgallerysoftware.com/>

**Operační systém:** Windows

**Typ programu:** Placený (Free trial)

### 1. Import obrázků

- Pomocí dialogového okna pro výběr souborů.
- Pomocí Drag and Drop.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.

### 2. Výpis obrázků

- Obrázky jsou zobrazeny v seznamu s jejich názvy.
- Změna pořadí pomocí tlačítek.
- Náhled obrázku v postranním panelu.

### 3. Editace obrázku

- Funkce: změna jasu, kontrastu, světlost, tmavost, saturace.
- Efekty: rozostření, šum, odstíny šedi, inverze.

### 4. Přizpůsobení výsledné webové galerie

- Možnost vybrat si z již existujících šablon.
- Úprava stylů a struktury vygenerované webové galerie pomocí HTML a CSS, texty jsou samostatně upravitelné v textových polích.
- Možnost změnit fonty na jednotlivých typech stránek (Index, Detail obrázku atd.)

### 5. Výsledná webová galerie

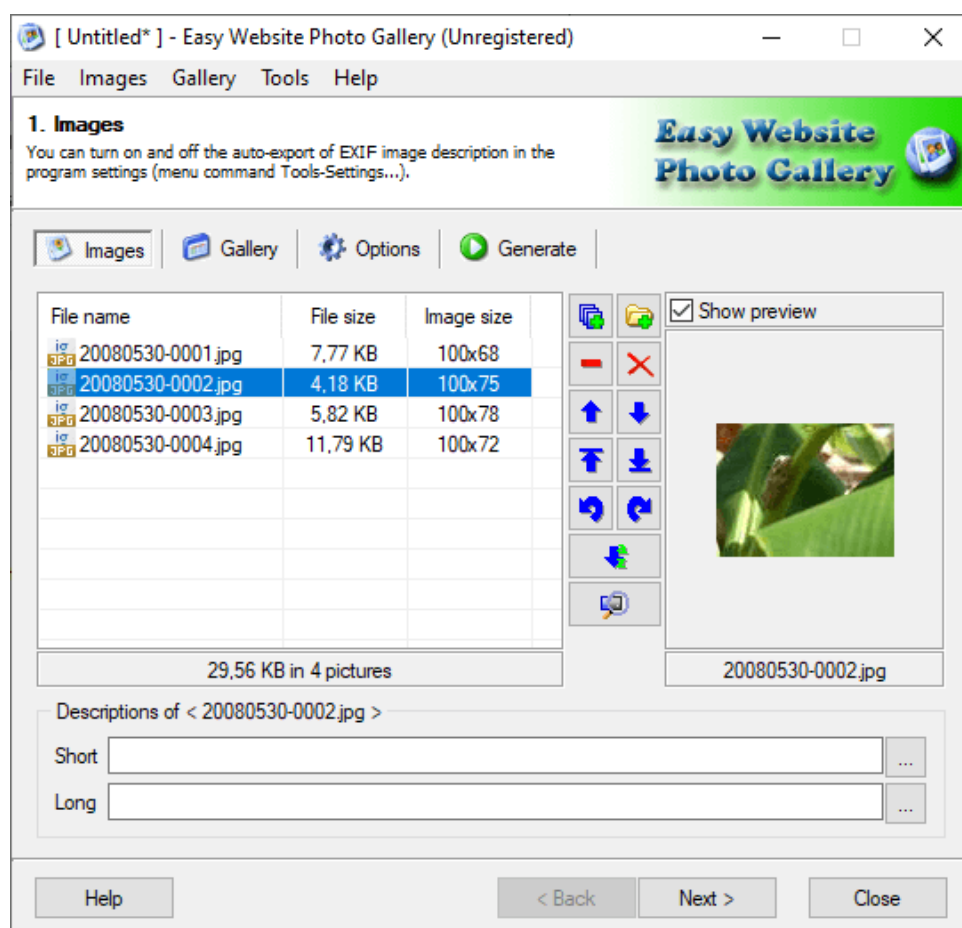
- Zobrazení mřížky s obrázky a jejich popisky na indexové stránce.
- Rozklik obrázku pro vstup na jeho detail.

### 6. Optimalizace obrázků

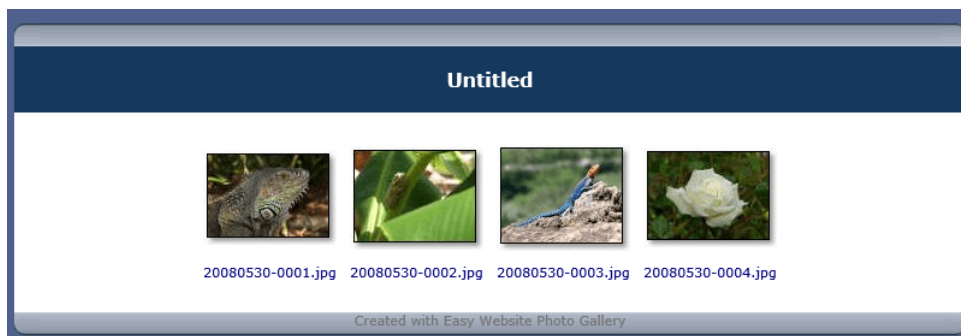
- Možnost zvolit kvalitu u změny rozměrů obrázku.

**Hodnocení.**

1. Co se mi líbí
  - Jednoduchost programu a výsledné galerie.
2. Co se mi nelíbí
  - Úprava barev a stylu pouze přes CSS.

**5.2.1 Obrázky programu****Obrázek 5.4:** Easy Website Photo Gallery, uživatelské rozhraní - výpis obrázků





**Obrázek 5.5:** Easy Website Photo Gallery, uživatelské rozhraní - výsledná webová galerie

## 5.3 HTTPPhotos

**Verze:** 4.10

**Oficiální stránky:** <https://us.digicamsoft.com/softhttpphotos.html>

**Operační systém:** Windows

**Typ programu:** Free

1. Import obrázků
  - Pomocí dialogového okna pro výběr souborů.
  - Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.
2. Výpis obrázků
  - Změna pořadí pomocí Drag and Drop.
3. Editace obrázku
  - Funkce: změna jasu, kontrastu.
  - Efekty: rozostření, zostření, stupně šedi, inverze.
4. Přizpůsobení výsledné webové galerie
  - Možnost změnit rozložení indexové stránky.
  - Možnost změnit font, barvy pozadí a textu.
5. Výsledná webová galerie
  - Zobrazení mřížky s náhledy obrázků na indexové stránce.
  - Rozklik obrázku pro vstup na jeho detail v LightBoxu.

## Hodnocení.

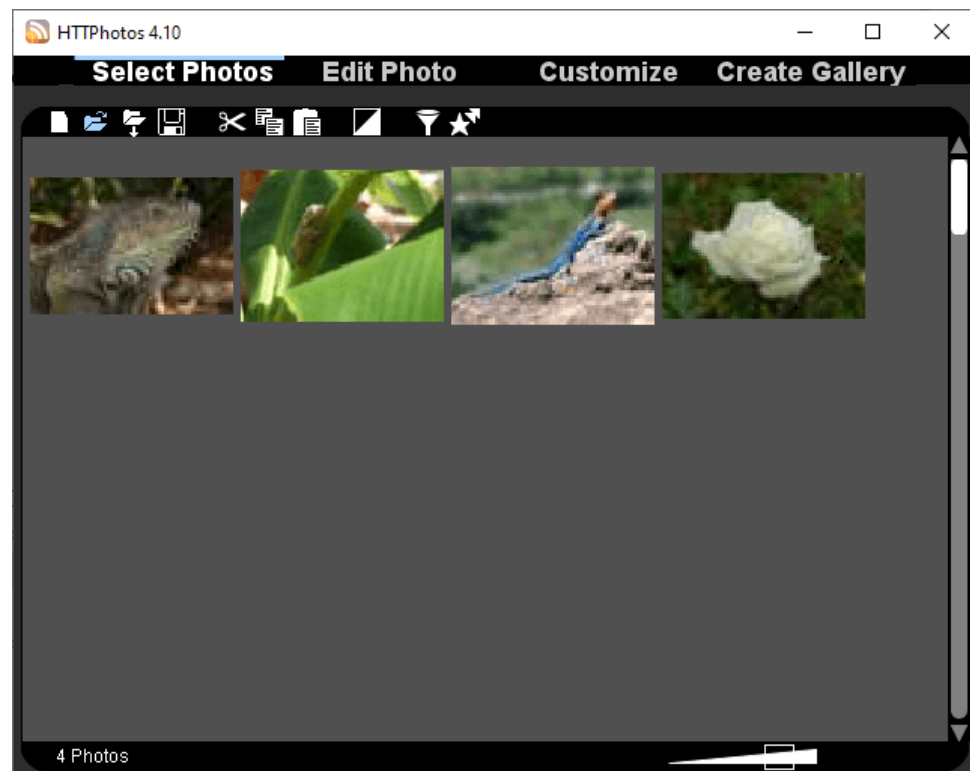
### 1. Co se mi líbí

- Jednoduchost programu a výsledné galerie.
- Program není nutné instalovat.

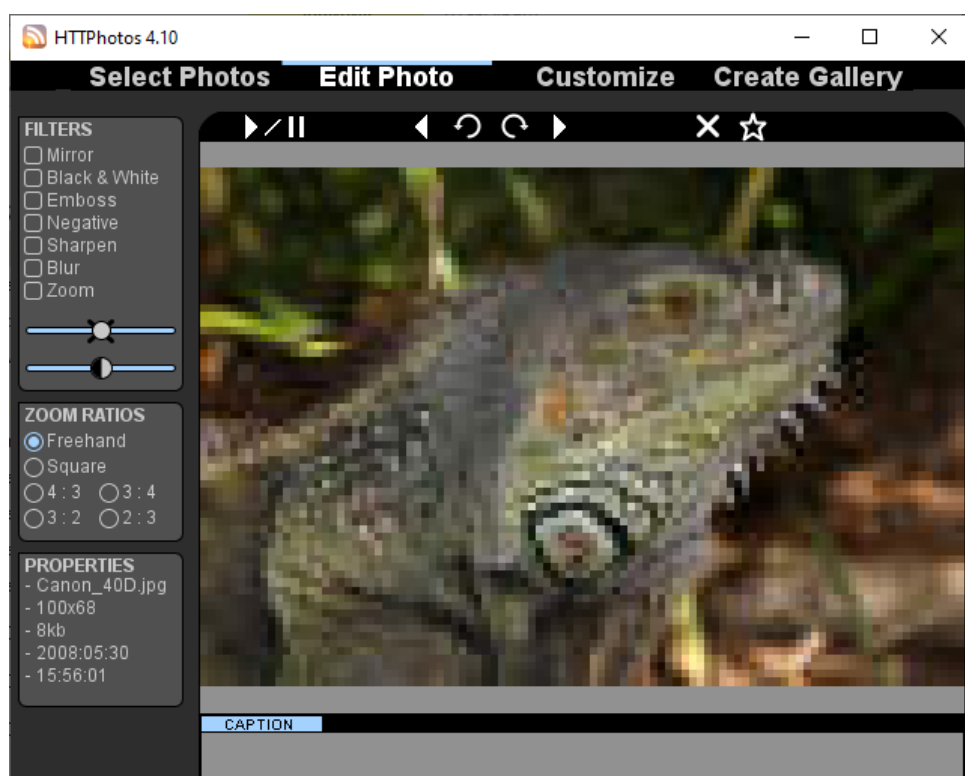
### 2. Co se mi nelíbí

- Ikona pro editaci obrázku je Lupa (místo například ikony tužky apod.)
- Nelze zvolit sílu efektů u editace obrázku (až na jas a kontrast).
- Překlopení obrázku je ve filtrech, zatímco jeho otočení je ve výpisu obrázků.
- Rozhraní aplikace je méně přehledné.

## 5.3.1 Obrázky programu



**Obrázek 5.6:** HTTPhotos, uživatelské rozhraní - výpis obrázků



Obrázek 5.7: HTTPotos, uživatelské rozhraní - editace obrázku

## 5.4 LightBox Video Web Gallery Creator

**Verze:** 2.1.7

**Oficiální stránky:** <http://pranas.net/webgallerycreator/>

**Operační systém:** Windows

**Typ programu:** Free

1. Import obrázků
  - Pomocí dialogového okna pro výběr souborů (pouze výběr složky).
  - Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.
2. Výpis obrázků - **Není**
3. Editace obrázku - **Není**
4. Přizpůsobení výsledné webové galerie
  - Možnost změnit velikost náhledů.

- Možnost změnit počet řádků a sloupců mřížky na stránce indexu.
- Možnost upravit kvalitu u náhledu obrázku.
- Možnost upravit CSS výsledných stránek.

5. Výsledná webová galerie

- Zobrazení mřížky s náhledy obrázků a jejich popisky na indexové stránce.
- Rozklik obrázku pro vstup na jeho detail v LightBoxu.

### Hodnocení.

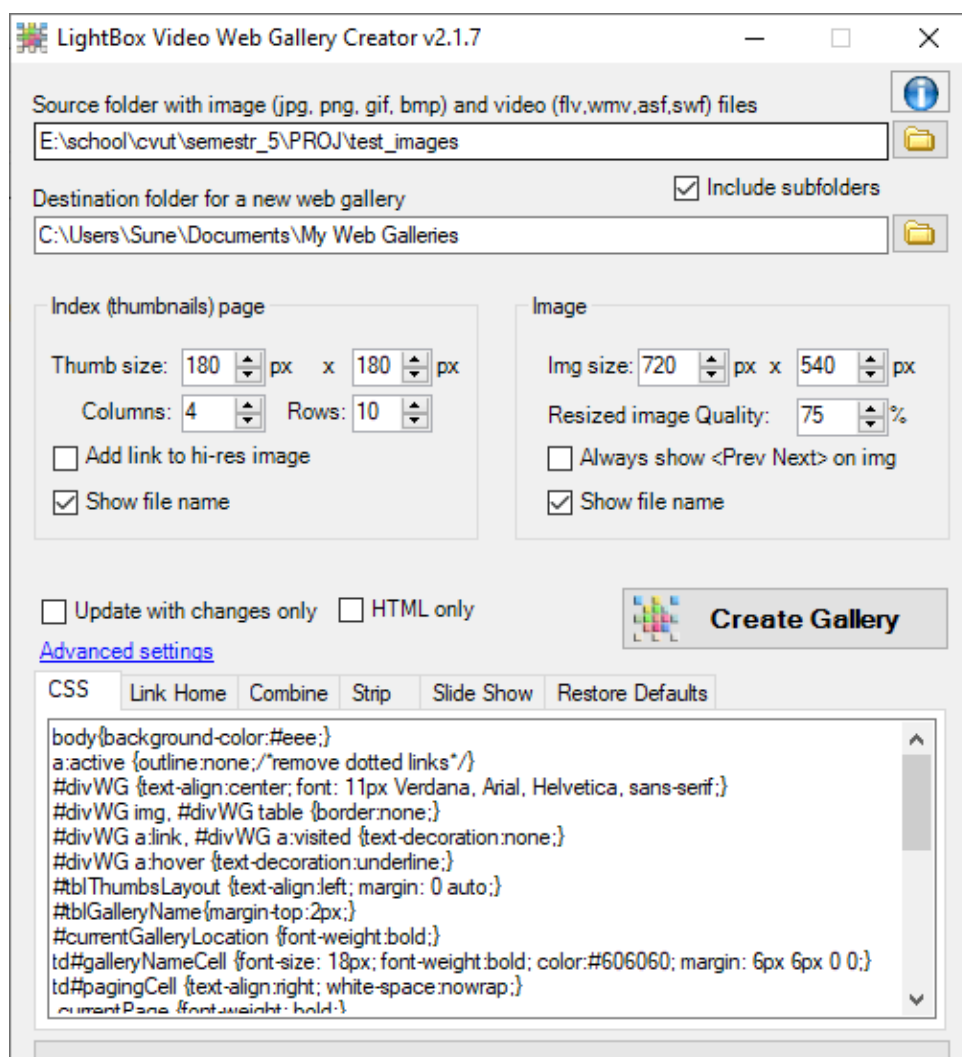
1. Co se mi líbí

- Jednoduchost programu a výsledné galerie.

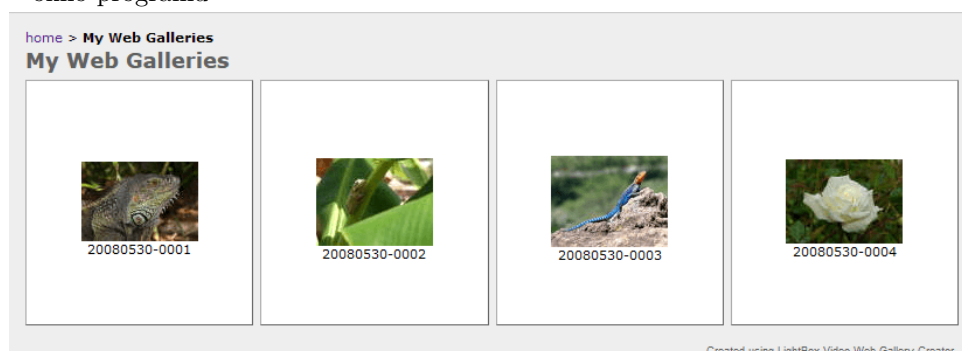
2. Co se mi nelíbí

- Program je nutné instalovat, i přesto, že se pouze extrahuje jeden EXE soubor.
- Nelze změnit barvy, fonty apod. bez znalosti CSS.
- Není možná úprava obrázků před vygenerováním galerie.

### 5.4.1 Obrázky programu



**Obrázek 5.8:** LightBox Video Web Gallery Creator, uživatelské rozhraní - hlavní okno programu



**Obrázek 5.9:** LightBox Video Web Gallery Creator, uživatelské rozhraní - výsledná webová galerie

## 5.5 iWebAlbum

**Verze:** 2.02

**Oficiální stránky:** <http://eunq.com/>

**Operační systém:** Windows

**Typ programu:** Opensource

### 1. Import obrázků

- Pomocí dialogového okna pro výběr souborů či složky.
- Pomocí Drag and Drop.
- Možnost změnit název a přidat komentář u jednotlivých obrázků.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.

### 2. Výpis obrázků

- Obrázky jsou zobrazeny v seznamu s jejich názvy a informacemi.
- Změna pořadí pomocí tlačítek.
- Náhled obrázku v postranním panelu.

### 3. Editace obrázku

- Funkce: Otočení a překlopení.

### 4. Přizpůsobení výsledné webové galerie

- Možnost vybrat si z již existujících šablon a stylů.
- Možnost změnit rozměry obrázků.
- Možnost změnit počet řádků a sloupců v mřížce.
- Možnost změnit název alba a komentáře k albu.

### 5. Vytvoření galerie

- Dodatečné informace (jako například datum pořízení) u obrázku nejsou zachovány.

## Hodnocení.

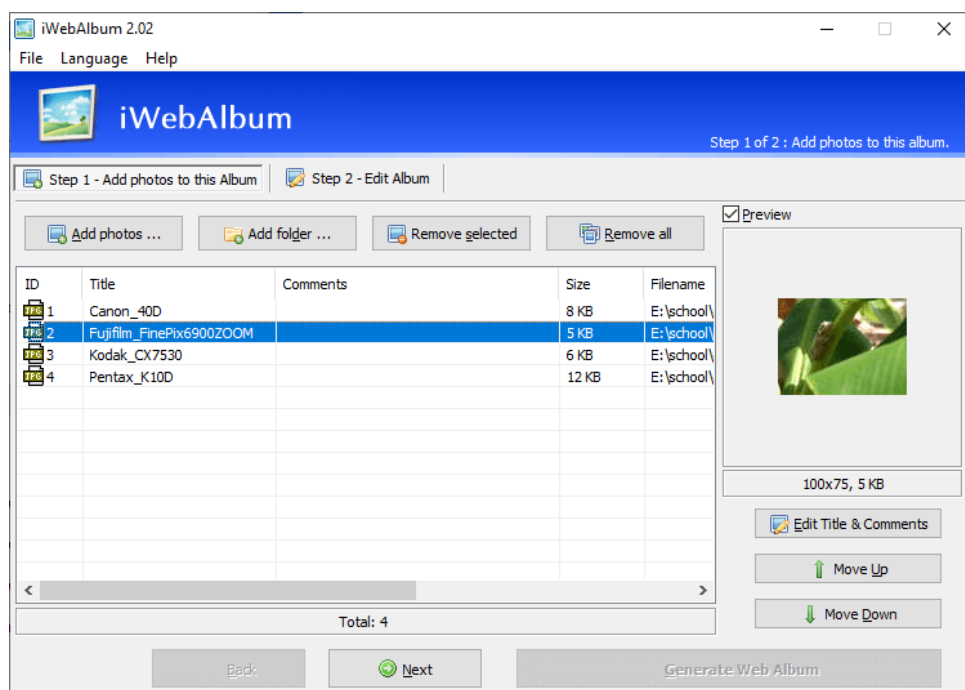
### 1. Co se mi líbí

- Jednoduchost programu a výsledné galerie.

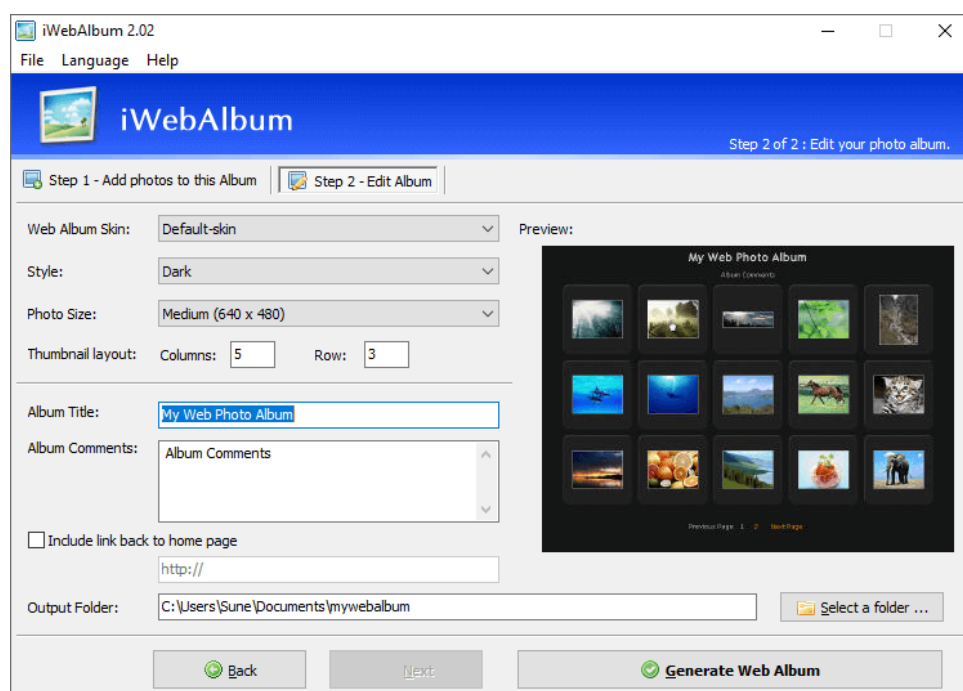
## 2. Co se mi nelíbí

- Lightbox u výsledné galerie funguje pouze v Internetu Exploreru a starší verzi prohlížeče Edge. Galerie je tedy nepoužitelná v jiných prohlížečích. Otestováno v IE 11.572.19041.0, Edge 44.19041.423.0, Firefox 82.0.2, Chrome 86.0.4240.111 a Opera 72.0.3815.186.
- Obrázky nelze v programu ořezávat či jinak upravovat, vyjma otáčení.

### 5.5.1 Obrázky programu



Obrázek 5.10: iWebAlbum, uživatelské rozhraní - výpis obrázků



Obrázek 5.11: iWebAlbum, uživatelské rozhraní - generování galerie

## 5.6 Web Album Generator

**Verze:** 1.8.2

**Oficiální stránky:**

<https://www.ornj.net/software/web-album-generator>

**Operační systém:** Windows

**Typ programu:** Free

### 1. Import obrázků

- Pomocí dialogového okna pro výběr souborů.
- Možnost změnit název a přidat komentář u jednotlivých obrázků.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.

### 2. Výpis obrázků

- Obrázky jsou zobrazeny v seznamu s jejich názvy.
- Změna pořadí pomocí tlačítek.
- Náhled obrázku v postranním panelu.



### 3. Editace obrázku

- Funkce: Otočení.

### 4. Přizpůsobení výsledné webové galerie

- Možnost vybrat si z již existujících šablon.
- Možnost změnit název alba a komentáře k albu.
- Možnost změnit barvy textu a pozadí.
- Možnost změnit počet řádků a sloupců v mřížce.
- Možnost změnit rozměry obrázků.
- Možnost změnit pozici navigace na stránce.

### 5. Vytvoření galerie

- Dodatečné informace (jako například datum pořízení) u obrázku nejsou zachovány.

## Hodnocení.

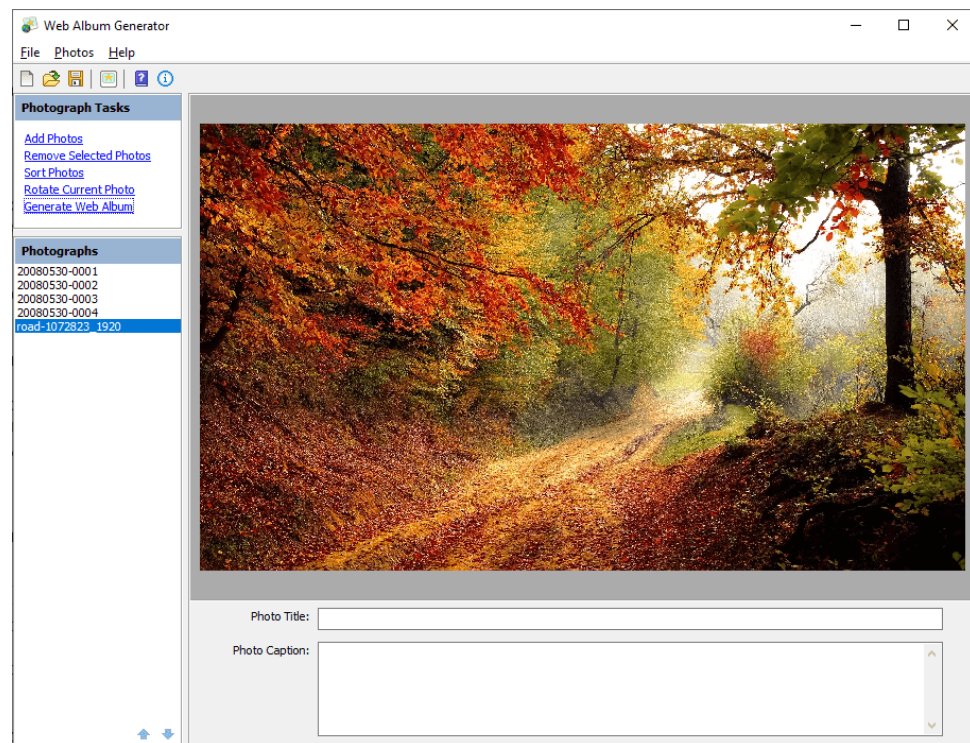
### 1. Co se mi líbí

- Jednoduchost programu a výsledné galerie.

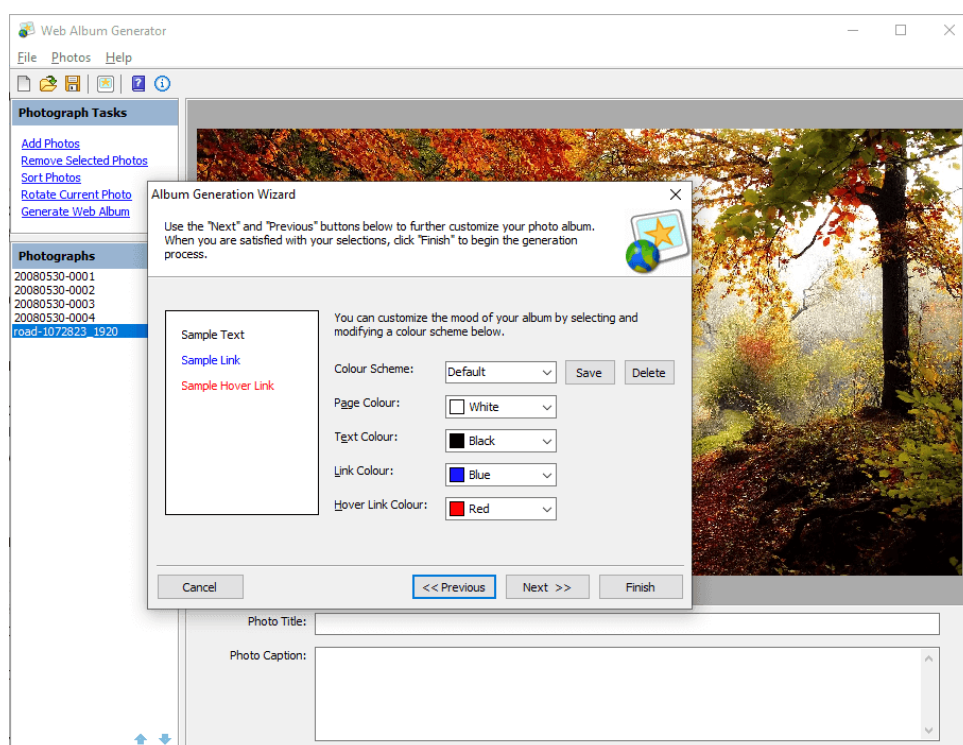
### 2. Co se mi nelíbí

- Výsledná galerie se nepřizpůsobuje velikosti obrazovky. U velkých obrázků se musí použít posuvníky či oddálit zobrazení stránky.
- Obrázky nelze v programu ořezávat či jinak upravovat, vyjma otáčení.

### 5.6.1 Obrázky programu



Obrázek 5.12: Web Album Generator, uživatelské rozhraní - výpis obrázků



**Obrázek 5.13:** Web Album Generator, uživatelské rozhraní - generování galerie, sekce pro úpravu barev výsledné galerie

## 5.7 Visual LightBox

**Verze:** 6.1

**Oficiální stránky:** <http://visuallightbox.com/>

**Operační systém:** Windows, Mac

**Typ programu:** Free (pro nekomerční použití)

### 1. Import obrázků

- Pomocí dialogového okna pro výběr souborů.
- Pomocí Drag and Drop.
- Pomocí importu ze služeb jako Flickr či Picasa.
- Možnost změnit název a přidat komentář u jednotlivých obrázků.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou importovány.

### 2. Výpis obrázků

- Obrázky jsou zobrazeny v mřížce s jejich názvy a náhledy.
- Změna pořadí pomocí Drag and Drop.

3. Editace obrázku

- Funkce: Otočení.

4. Přizpůsobení výsledné webové galerie

- Možnost vybrat si z již existujících šablon a stylů, jak mřížky, tak lightboxu jednotlivě.
- Možnost vybrat kvalitu výsledných náhledů obrázků.
- Možnost změnit nastavení slideshow.

5. Vytvoření galerie

- Možnost publikování galerie do CMS jako WordPress a Joomla.
- Dodatečné informace (jako například datum pořízení) u obrázku nejsou zachovány.

## Hodnocení.

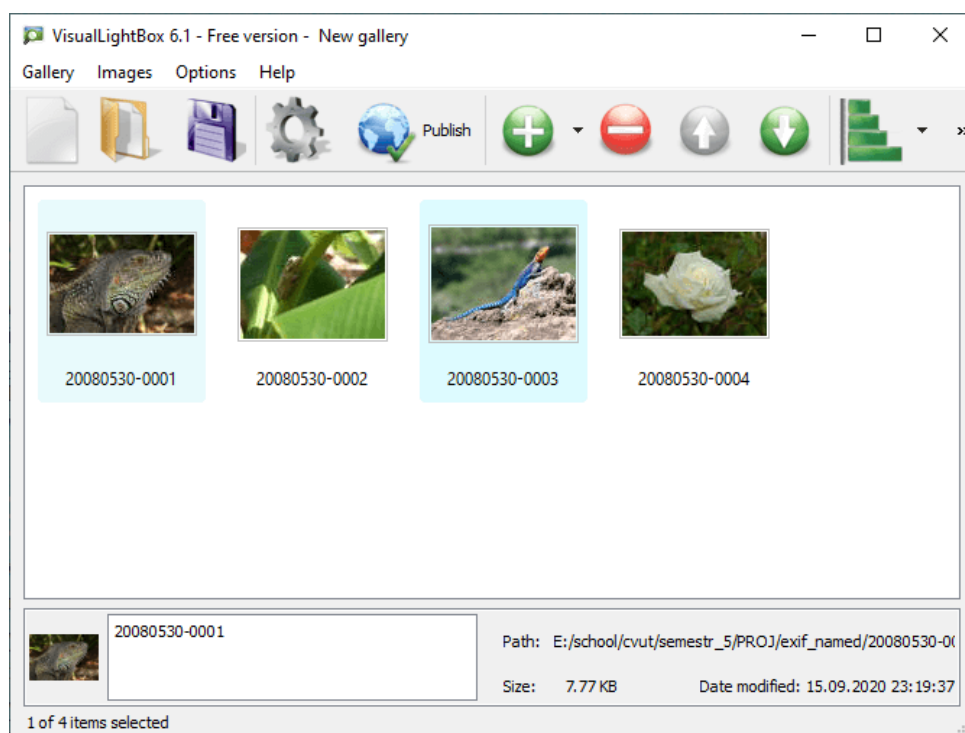
1. Co se mi líbí

- Jednoduchost programu a výsledné galerie.
- Oddělené styly pro Lightbox a mřížku galerie.
- Publikování do různých CMS, jako je WordPress či Joomla.

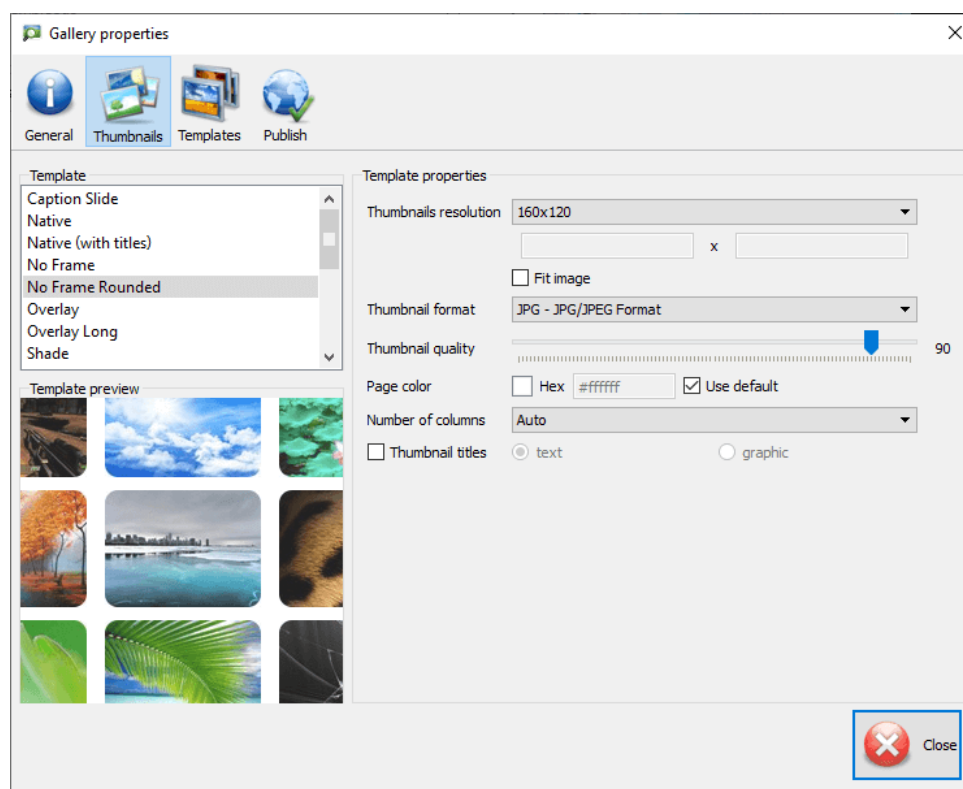
2. Co se mi nelíbí

- Obrázky nelze v programu ořezávat či jinak upravovat, vyjma otáčení.

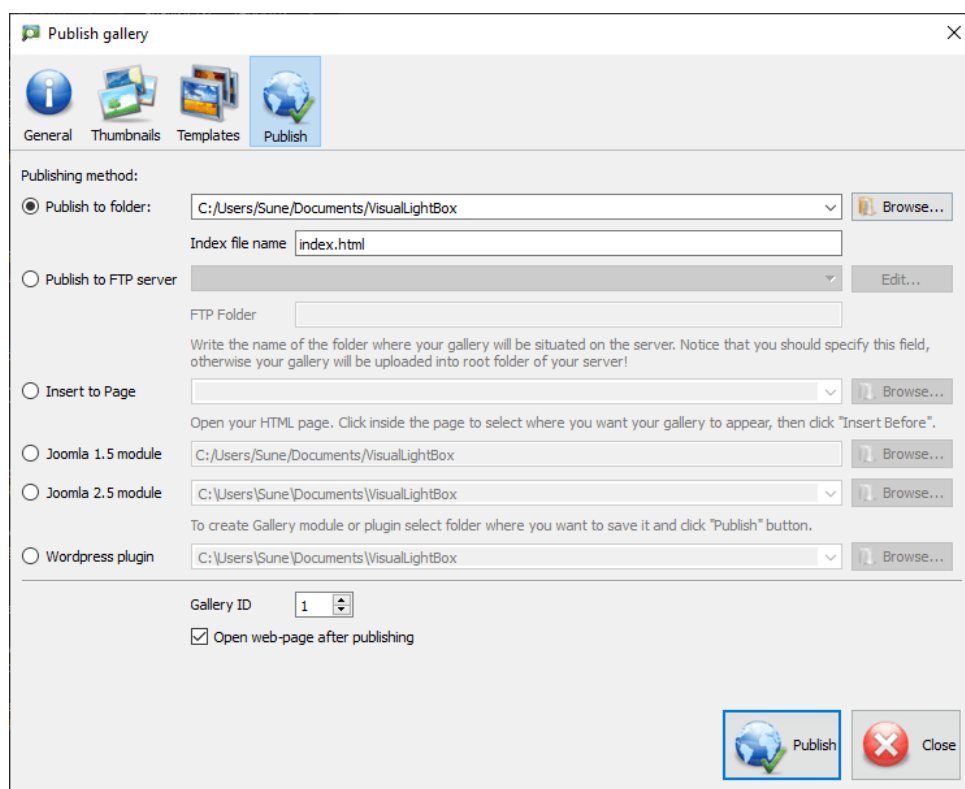
### 5.7.1 Obrázky programu



Obrázek 5.14: Visual LightBox, uživatelské rozhraní - výpis obrázků



**Obrázek 5.15:** Visual LightBox, uživatelské rozhraní - generování galerie, sekce pro úpravu vzhledu náhledových obrázků



**Obrázek 5.16:** Visual LightBox, uživatelské rozhraní - generování galerie, sekce pro výstup







## **Kapitola 6**

### **Shrnutí analýzy programů**

	<b>jAlbum</b>	<b>Easy Website Photo Gallery</b>	<b>HTTPPhotos</b>	<b>LightBox Video Web Gallery Creator</b>
<b>Import obrázků</b>	Pomocí dialogového okna (soubory) a drag and drop. Možnost popisů a tagů u obrázků.	Pomocí dialogového okna (soubory) a drag and drop.	Pomocí dialogového okna (soubory).	Pomocí dialogového okna (složky).
<b>Výpis obrázků</b>	Výpis s náhledy. Změna pořadí pomocí drag and drop. Možnost vytváření hierarchie pomocí složek.	Výpis v seznamu s názvy obrázků. Náhled obrázku v postranním panelu. Změna pořadí pomocí tlačítek.	Výpis s náhledy. Změna pořadí pomocí Drag and Drop.	Není.
<b>Editace obrázků</b>	Funkce: překlopení, otočení, ořez, úprava jasu, kontrastu a další. Efekty: červené oči, zaostření, rozmazání a další.	Funkce: změna jasu, kontrastu, světlost, tmavost, saturace. Efekty: rozostření, šum, odstíny šedi, inverze.	Funkce: změna jasu, kontrastu. Efekty: rozostření, zotření, stupně šedi, inverze.	Není.
<b>Přizpůsobení výsledné galerie</b>	Výběr stylu galerie z existujících šablon. Možnost změnit barvy, odsazení a další prvky Lightboxu. Vlastní HTML, CSS a JS. Správa SEO.	Výběr stylu galerie z existujících šablon. Možnost upravit styl a strukturu galerie pomocí HTML a CSS, texty samostatně upravitelné. Možná změna fontů.	Možnost změnit rozložení indexové stránky. Možnost změnit font, barvy pozadí a textu.	Možnost změnit velikost náhledů. Možnost změnit počet řádků a sloupců mřížky galerie. Možnost upravit kvalitu u náhledu obrázku. Možná úprava CSS.

Tabulka 6.1: Shrnutí analýzy programů, část 1.

	<b>iWebAlbum</b>	<b>Web Album Generator</b>	<b>Visual LightBox</b>
<b>Import obrázků</b>	Pomocí dialogového okna (soubory) a drag and drop. Možnost přidat komentář u obrázků.	Pomocí dialogového okna (soubory). Možnost přidat komentář u obrázků.	Pomocí dialogového okna (soubory), drag and drop a služeb jako Flickr či Picasa. Možnost přidat komentář u obrázků.
<b>Výpis obrázků</b>	Výpis v seznamu s názvy obrázků. Náhled obrázku v postranním panelu. Změna pořadí pomocí tlačítek.	Výpis v seznamu s názvy obrázků. Náhled obrázku v postranním panelu. Změna pořadí pomocí tlačítek.	Výpis s náhledy. Změna pořadí pomocí drag and drop.
<b>Editace obrázků</b>	Funkce: Otočení a překlopení.	Funkce: Otočení.	Funkce: Otočení.
<b>Přizpůsobení výsledné galerie</b>	Výběr stylu galerie z existujících šablon. Možná změna rozměrů obrázků. Možnost změnit počet řádků a sloupců v mřížce galerie.	Výběr stylu galerie z existujících šablon. Možnost změnit barvy textu a pozadí. Možnost změnit počet řádků a sloupců v mřížce galerie. Možná změna rozměrů obrázků.	Výběr stylu zvlášť mřížky galerie a lightboxu z existujících šablon. Možnost vybrat kvalitu výsledných náhledů obrázků. Možná změna nastavení slideshow.

Tabulka 6.2: Shrnutí analýzy programů, část 2.



# Kapitola 7

## Analýza C++ GUI knihoven

V této sekci analyzuji některé nalezené C++ GUI knihovny. Každá podsekce analyzuje jednu knihovnu.

### 7.1 Dear ImGui

**Verze:** 1.79

**Oficiální stránky:** <https://github.com/ocornut/imgui>

**Operační systém:** Cross-platform (Windows, Mac, Linux)

**Typ knihovny:** Open source

#### 1. Funkce

- Obsahuje základní prvky pro grafická rozhraní: Okna, tlačítka, menu, posuvníky (slider), textová pole atd.
- Lze vytvářet vlastní widgety (prvky).

#### 2. Použití

- Relativně jednoduché na použití, srozumitelné API.

#### 3. API

- Používá kontext pro vytváření a vkládání prvků.

## Hodnocení.

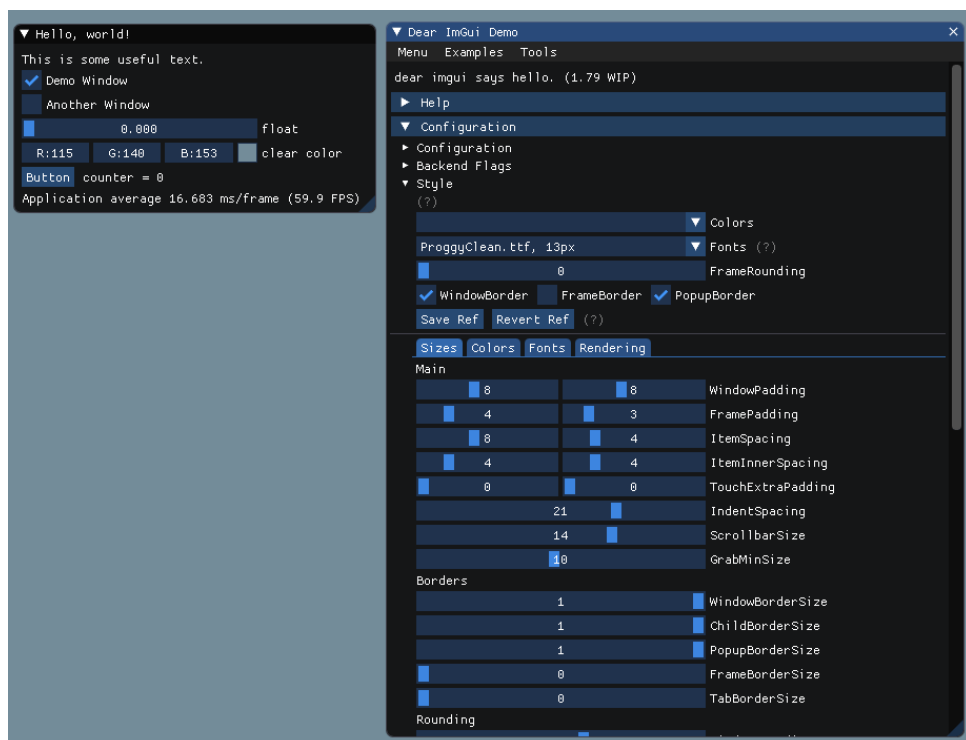
### 1. Co se mi líbí

- Vzhled jednotlivých prvků grafického rozhraní.
- Možnost rychle měnit tmavý a světlý styl.
- Již obsahuje prvek pro výběr barev, který vypadá mnohem lépe jak výchozí prvek na operačním systému Windows, není tedy nutné ho vytvářet.

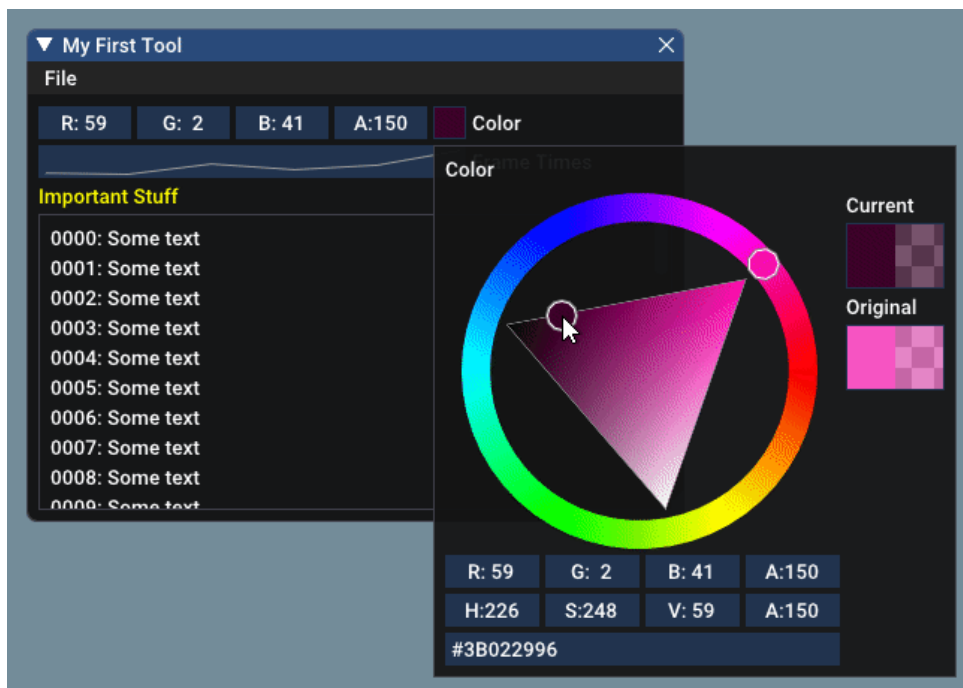
### 2. Co se mi nelíbí

- Občasná krkolomnost při vytváření binárních spustitelných souborů.
- Ne úplně kompletní tutoriály/příklady na oficiálních stránkách.

## 7.1.1 Obrázky



Obrázek 7.1: Dear ImGui, demo



Obrázek 7.2: Dear ImGui, prvek pro výběr barvy

## 7.2 NanoGUI

Verze: 22.09.2019

Oficiální stránky: <https://github.com/wjakob/nanogui>

Operační systém: Cross-platform (Windows, Mac, Linux)

Typ knihovny: Open source

### 1. Funkce

- Obsahuje základní prvky pro grafická rozhraní: Okna, tlačítka, menu, posuvníky (slider), textová pole atd.

### 2. Použití

- Relativně jednoduché na použití.

### 3. API

- Používá objekty pro vytváření a vkládání prvků.

**Hodnocení.**

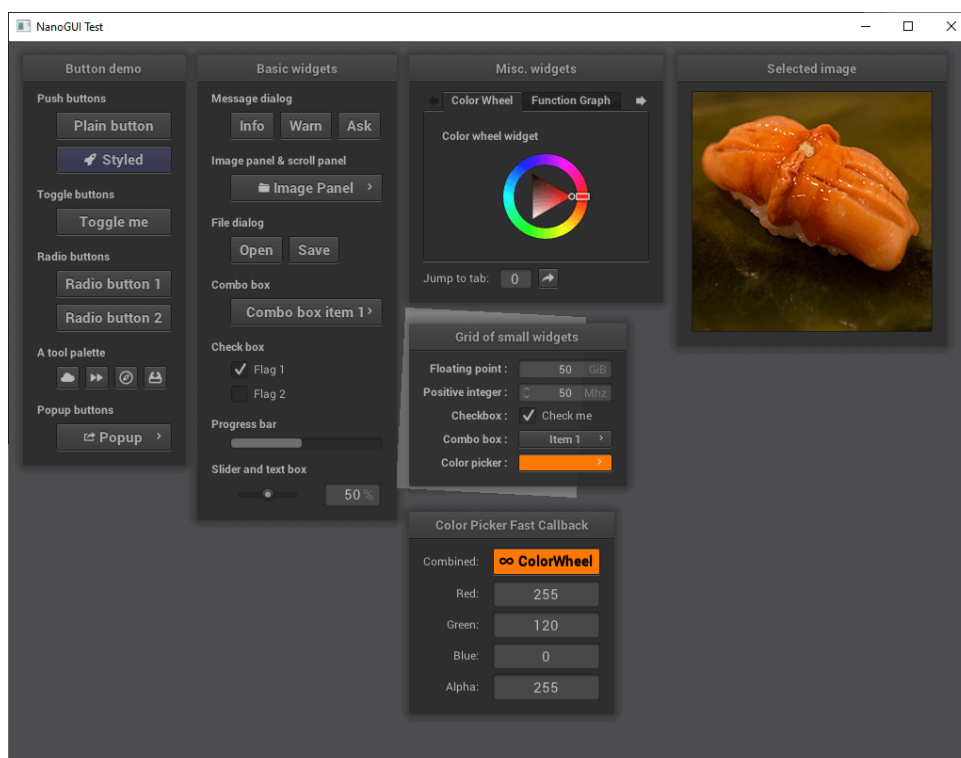
1. Co se mi líbí

- Vzhled jednotlivých prvků grafického rozhraní.
- Již obsahuje prvek pro výběr barev.
- Hezky zpracované demo aplikace.

2. Co se mi nelíbí

- Knihovna již není plně vyvíjena.

## 7.2.1 Obrázky



Obrázek 7.3: NanoGUI, demo



## 7.3 FLTK

**Verze:** 1.3.5

**Oficiální stránky:** <https://www.fltk.org/>

**Operační systém:** Cross-platform (Windows, Mac, Linux)

**Typ knihovny:** Open source

### 1. Funkce

- Obsahuje základní prvky pro grafická rozhraní: Okna, tlačítka, menu, posuvníky (slider), textová pole atd.

### 2. Použití

- Relativně jednoduché na použití.

### 3. API

- Používá objekty pro vytváření a vkládání prvků.

## Hodnocení.

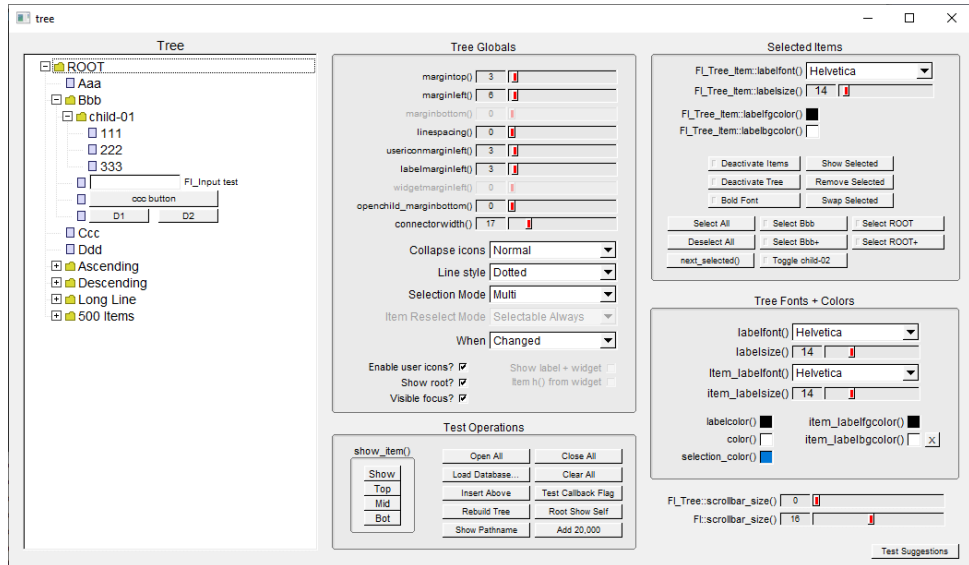
### 1. Co se mi líbí

- Již obsahuje prvek pro výběr barev.
- Velmi rozsáhlá dokumentace.

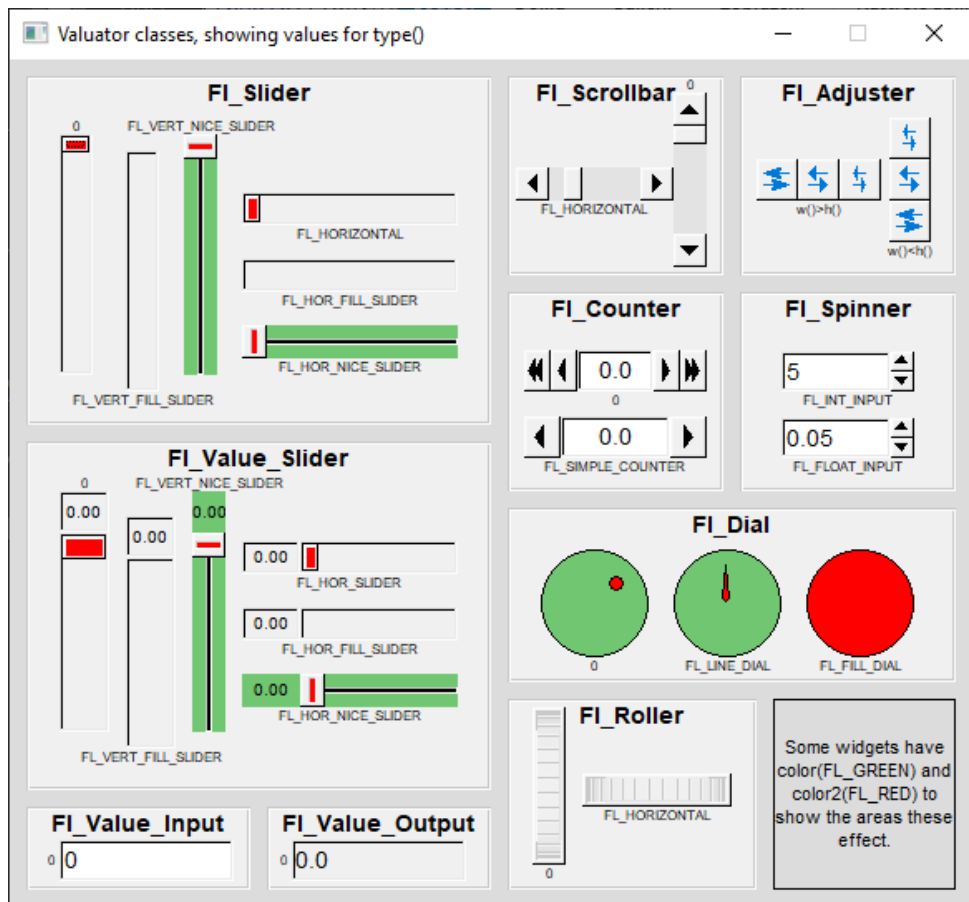
### 2. Co se mi nelíbí

- Vzhled jednotlivých prvků grafického rozhraní.

### 7.3.1 Obrázky



Obrázek 7.4: FLTK, demo



Obrázek 7.5: FLTK, demo 2

## 7.4 wxWidgets

**Verze:** 3.1.4

**Oficiální stránky:** <https://www.wxwidgets.org>

**Operační systém:** Cross-platform (Windows, Mac, Linux)

**Typ knihovny:** Open source

### 1. Funkce

- Obsahuje základní prvky pro grafická rozhraní: Okna, tlačítka, menu, posuvníky (slider), textová pole atd.

### 2. Použití

- Občas zmatečné při použití.

### 3. API

- Používá objekty pro vytváření a vkládání prvků.

## Hodnocení.

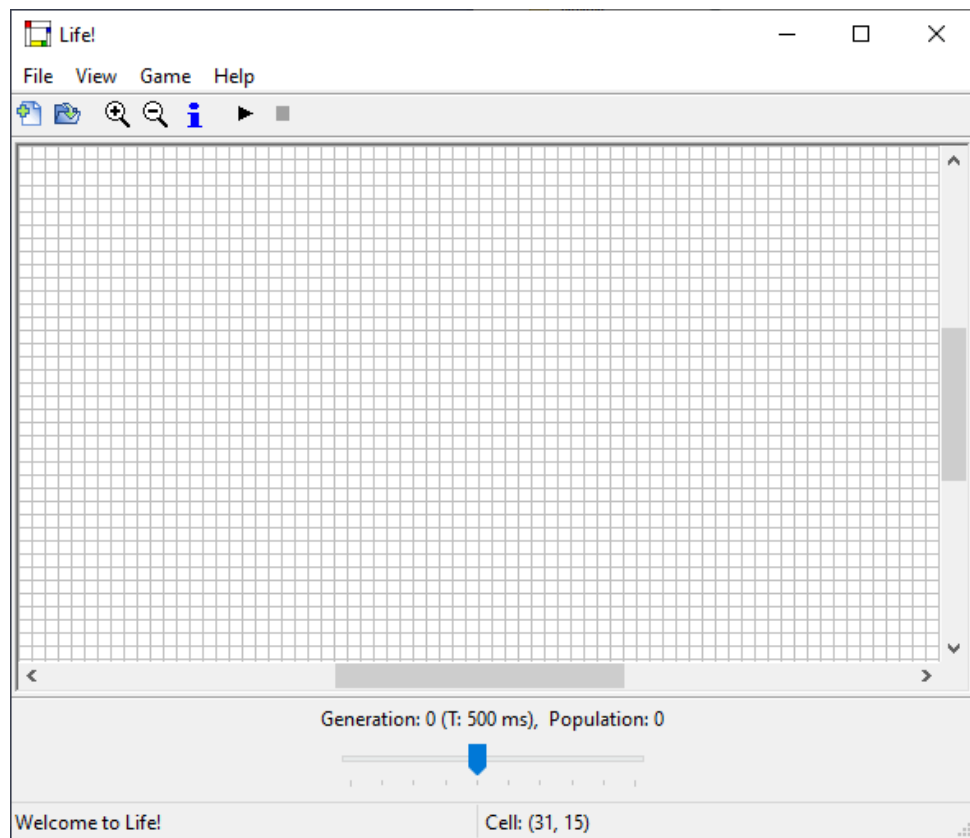
### 1. Co se mi líbí

- Již obsahuje prvek pro výběr barev.
- Obsáhlá dokumentace.

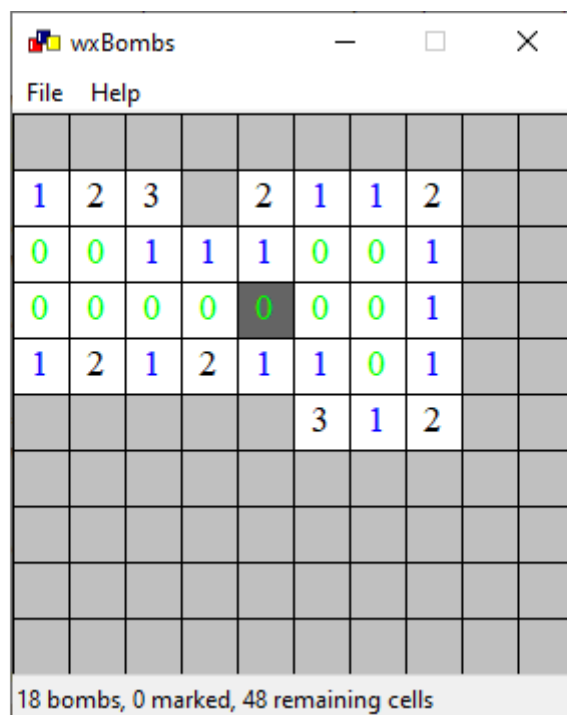
### 2. Co se mi nelíbí

- Vzhled jednotlivých prvků grafického rozhraní.

### 7.4.1 Obrázky



Obrázek 7.6: wxWidgets, demo: Life



Obrázek 7.7: wxWidgets, demo: Bombs

## 7.5 Agar

Verze: 1.6.0

Oficiální stránky: <http://libagar.org>

Operační systém: Cross-platform (Windows, Mac, Linux)

Typ knihovny: Open source

### 1. Funkce

- Obsahuje základní prvky pro grafická rozhraní: Okna, tlačítka, menu, posuvníky (slider), textová pole atd.

### 2. Použití

- Občas zmatečné při použití.
- Psané v C.

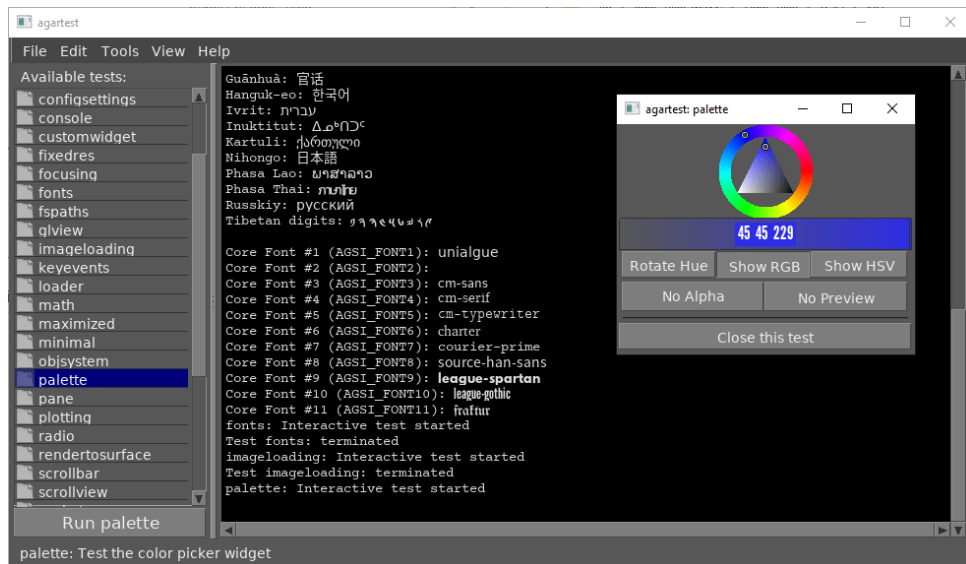
### 3. API

- Používá objekty pro vytváření a vkládání prvků.

## Hodnocení.

1. Co se mi líbí
  - Vzhled jednotlivých prvků grafického rozhraní.
  - Již obsahuje prvek pro výběr barev.
  - Obsáhlá dokumentace.
2. Co se mi nelíbí
  - Není psané v C++.

### 7.5.1 Obrázky



Obrázek 7.8: Agar, demo: agartest



## Kapitola 8

### Shrnutí analýzy C++ GUI knihoven





# Kapitola 9

## Návrh implementace

Tato sekce se věnuje návrhu implementace funkcí v budoucím programu tohoto projektu. Každá podsekce se zabývá jednou požadovanou funkcí, kterou by měl program umět.

### 9.1 Import obrázků

Tato podsekce se zabývá možnostmi importu obrázků do programu a následným procesem importu.

#### 9.1.1 Možnosti importu

Import obrázků do programu by měl být pro uživatele co nejvíce intuitivní a jednoduchý, nicméně každý uživatel může mít jiné preference, proto by se mělo implementovat více možností importu. Nejvíce vhodnými jsou následující:

##### 1. Dialog pro výběr souborů

Uživatel zvolí jeden či více souborů. Soubory pak následně projdou procesem importu. Většina operačních systémů umožňuje do dialogu přidat různé filtry, jako například *Všechny obrázky*, *JPEG/JPG soubory*, *PNG soubory* apod., které by měly být pro uživatele také dostupné.



samotné datum bez času. Pokud je možnost zvolení názvu složky jako data pořízení obrázků zvolena, ale název složky není validní datum či nelze rozpoznat formát data, pak se objeví chybová hláška a uživatel musí vybrat, zdali import přerušit, nebo chybu akceptuje a název složky se bude ignorovat.

### ■ 9.1.3 Získávání informací ze struktury složek

Pokud bude programu při importu pomocí složky předána určitá struktura složek, lze z ní získat informace o datu pořízení jednotlivých obrázků bez toho, aniž by se musela číst jejich metadata. Pokud se v obrázku nenachází informace o datu pořízení, nejde správnost tohoto získaného data nijak ověřit. Uživateli ale lze umožnit využít, pokud data v obrázku neexistují, právě data extrahovaná ze struktury složek.

Struktura složek nicméně může být různá:

#### ■ Plochá struktura

Jedná se o strukturu, kde je programu předána složka, ve které se nachází podsložky, jejichž název obsahuje celé datum v různých formátech, tedy například: *2020-10-14*, *14. 10. 2020* atd. Všem obrázkům z takovýchto podsložek bude přiřazeno právě toto celé datum.

#### ■ Hierarchická struktura

Jedná se o strukturu, kde je programu předána složka, ve které se nachází hierarchie složek. A to taková, že nejdříve jsou složky s názvy roků, v nich složky jednotlivých měsíců a v nich pak složky jednotlivých dní. Je samozřejmě možné uživateli umožnit tuto hierarchii změnit třeba tak, že nejdříve je den, pak měsíc a pak rok.

Řazení obrázků v jednotlivých složkách je nicméně na uživateli. Jelikož program bude pravděpodobně soubory procházet abecedně, k jejich nahrání ve správném pořadí stačí například je očíslovat nebo u nich specifikovat čas pořízení (ve tvaru *hh-mm-ss* apod.)



## 9.3 Editace obrázku

Uživatel může provést editaci obrázku přímo v programu. Jelikož se nejedná o plnohodnotný obrázkový editor, program bude nabízet pouze nějaké ze základních funkcí pro editaci. Konkrétně minimálně tyto funkce: *překlopení obrázku, rotace obrázku (o 90 stupňů, jako volitelná funkcionalita i libovolný úhel), oříznutí obrázku a úprava jasu a kontrastu obrázku (jako volitelná funkcionalita i úprava saturace)*; a dále konkrétně minimálně tyto efekty: *stupně šedi, inverze, rozostření, zostření a odstranění červených očí*.

### 9.3.1 Uživatelské rozhraní

Uživatelské rozhraní by mělo být pro uživatele intuitivní a nemělo by být přehlcené. Pro nejjednodušší variantu editace obrázku je potřeba:

- plátno, kde se obrázek v současném stavu zobrazí,
- výběr úpravy obrázku a efektu,
- nastavení jednotlivých úprav obrázku a efektů,
- tlačítka pro potvrzení a zrušení editace.

**Hlavní okno:** V okně editace se nachází plátno pro zobrazení obrázku v současném stavu, které zabírá značnou část prostoru okna. Umístění výběru pro úpravu obrázku a efektu může být buď (a) lišta menu nahoře v okně s kategoriemi *Úpravy* a *Efekty*, nebo (b) panel vlevo či vpravo vedle plátna s tlačítky jednotlivých úprav a efektů. V poslední řadě se v okně nachází v dolní části dvě tlačítka, jedno pro potvrzení editace a druhé pro zrušení editace.

**Okno/panel nastavení úpravy či efektu:** V okně/panelu nastavení, pokud to úprava či efekt umožňuje, je nastavení síly dané úpravy či efektu (příkladem může být třeba síla rozmazání). U některých úprav či efektů (například inverze) lze umožnit změnu síly, nicméně v předešle analyzovaných programech takováto funkce není, proto je brána jako volitelná funkcionalita. Dále je nutné mít tlačítko pro potvrzení a zrušení úpravy či efektu. Veškeré změny v rozhraní (jako je změna síly) se projeví v obrázku zobrazeném na plátně. Takto uživatel

může vidět úpravy či efekty ihned a není potřeba nic potvrzovat a následně se pak zbytečně vracet o krok zpět před tím, než byla úprava či efekt na obrázek aplikován.

### ■ 9.3.2 Implementace úprav obrázku a efektů

Pro úpravy obrázku a efekty lze použít (a) vlastní implementaci, nebo (b) použít již existující knihovnu. Pokud by se zvolila první varianta, bylo by vhodné i přesto použít nějakou již existující knihovnu pro čtení a ukládání různých obrázkových formátů. Obrázkové formáty jsou komplexní, využívají různé algoritmy a mají třeba i určitá rozšíření (například PNG extensions). Vlastní implementací čtení a ukládání obrázků by se mohly objevovat chyby, některé obrázky by ani nemusely být správně přečteny a hlavně by se pouze na této části programu strávila většina času. Většina knihoven je řádně otestovaná, a tudíž je odzkoušeno, že doopravdy funguje.

Po přečtení obrázku je ale stále nutné stanovit, jak se obrázky budou interně v programu reprezentovat, aby se na ně mohly aplikovat algoritmy úprav a efektů. Knihovny většinou mají více možností, jak obrázky reprezentovat - například `BYTE_3_RGB` (3 byte pro každou složku RGB), `INT_ARGB` (1 integer pro všechny složky ARGB) a podobně. Jelikož se nejedná o plnohodnotný obrázkový editor, stačí zvolit pouze jednu interní reprezentaci. Proto je nutné přečtené obrázky převést do jednotného formátu. Tím, že program umožňuje pracovat s obrázky PNG a ty podporují průhlednost, je nutné umožnit pracovat s průhledností každého pixelu. Proto bych osobně volil `INT_ARGB`, kde je pro každý pixel použito jedno celé číslo (integer), možností je ale třeba také `BYTE_4_ARGB`, kde pro každý pixel jsou použity 4 byte. Dále by jednotlivé pixely byly v paměti reprezentovány po řádcích.

Po stanovení jednotného formátu v předchozím odstavci, v tomto případě konkrétně `INT_ARGB`, by se jednotlivé úpravy a efekty daly (naivně) implementovat následovně:

#### ■ Překlopení obrázku horizontálně

Pro každý řádek obrázku uděláme následující:

Pro  $0 \leq i < (w - 1)/2$  vezmeme dva pixely, jeden na pozici  $i$  a druhý na pozici  $w - i - 1$ , kde  $w$  je šířka obrázku, a tyto pixely prohodíme.

#### ■ Překlopení obrázku vertikálně

Pro každý sloupec obrázku uděláme následující:

Pro  $0 \leq i < (h - 1)/2$  vezmeme dva pixely, jeden na pozici  $i$  a druhý na pozici  $h - i - 1$ , kde  $h$  je výška obrázku, a tyto pixely prohodíme.

#### ■ Otočení obrázku (o 90 stupňů)

Vytvoříme si nové pole o velikosti  $w * h$ , kde  $w$  je šířka obrázku a  $h$  je výška obrázku. Pro každý pixel na pozici  $[x, y]$  (indexově  $y * w + x$ ) aplikujeme rotaci o 90 stupňů, tedy  $R_{\theta=90}([x, y]) = [-y, x]$ , a uložíme ho na výslednou pozici (indexově  $x * h - y$ ). Nové pole pak reprezentuje otočený obrázek s šířkou  $h$  a výškou  $w$ .

#### ■ Oříznutí obrázku (obdélníkové oříznutí)

Vytvoříme si nové pole o velikosti  $w_c * h_c$ , kde  $0 < w_c \leq w$  je šířka ořezu,  $0 < h_c \leq h$  je výška ořezu,  $w$  je původní šířka obrázku a  $h$  je původní výška obrázku. Pak každý pixel  $[x, y]$  v oblasti  $x_c \leq x < x_c + w_c$ ,  $y_c \leq y < y_c + h_c$ , kde bod  $[x_c, y_c]$ ,  $0 \leq x_c < w - 1$ ,  $0 \leq y_c < h - 1$ , je počáteční bod ořezu, uložíme do nového pole na pozici  $[x - x_c, y - y_c]$  (indexově  $(y - y_c) * w + (x - x_c)$ ).

#### ■ Úprava jasu[BRI]

Mějme hodnotu  $v \in [-255, 255]$  reprezentující sílu úpravy jasu. Pro každý pixel obrázku pak uděláme následující:

```
r := clamp(((p >> 16) & 0xff) + v, 0, 255),
g := clamp(((p >> 8) & 0xff) + v, 0, 255),
b := clamp((p & 0xff) + v, 0, 255),
p' := (p & 0xff000000) | (r << 16) | (g << 8) | b,
```

kde  $p$  je hodnota pixelu,  $r$  je červená složka pixelu,  $g$  je zelená složka pixelu,  $b$  je modrá složka pixelu, `clamp` je pomocná funkce pro omezení hodnoty (mezi hodnotu minima a hodnotu maxima), `>>` je binární posun doprava, `<<` je binární posun doleva, `&` je binární AND, `|` je binární OR, hodnoty začínající `0x` jsou celá čísla reprezentovaná v hexadecimální soustavě a  $p'$  je výsledná hodnota pixelu.

#### ■ Úprava kontrastu[CON]

Mějme hodnotu  $c \in [-255, 255]$ , pak  $v = 259 * (c + 255) / (255 * (259 - c))$  reprezentující sílu úpravy kontrastu. Pro každý pixel obrázku pak uděláme následující:

```
r := clamp(v * (((p >> 16) & 0xff) - 128) + 128, 0, 255),
g := clamp(v * (((p >> 8) & 0xff) - 128) + 128, 0, 255),
b := clamp(v * ((p & 0xff) - 128) + 128, 0, 255),
p' := (p & 0xff000000) | (r << 16) | (g << 8) | b,
```

kde  $p$  je hodnota pixelu,  $r$  je červená složka pixelu,  $g$  je zelená složka pixelu,  $b$  je modrá složka pixelu, `clamp` je pomocná funkce pro omezení hodnoty, `>>` je binární posun doprava, `<<` je binární posun doleva, `&` je binární AND, `|` je binární OR, hodnoty začínající `0x` jsou celá čísla reprezentovaná v hexadecimální soustavě a  $p'$  je výsledná hodnota pixelu.

#### ■ Stupně šedi[GRA]

Pro každý pixel obrázku pak uděláme následující:

```

r := (p >> 16) & 0xff),
g := (p >> 8) & 0xff),
b := (p & 0xff),
t := clamp(0.299 * r + 0.587 * g + 0.114 * b, 0, 255),
p' := (p & 0xff000000) | (t << 16) | (t << 8) | t,

```

kde  $p$  je hodnota pixelu,  $r$  je červená složka pixelu,  $g$  je zelená složka pixelu,  $b$  je modrá složka pixelu, `clamp` je pomocná funkce pro omezení hodnoty, `>>` je binární posun doprava, `<<` je binární posun doleva, `&` je binární AND, `|` je binární OR, hodnoty začínající `0x` jsou celá čísla reprezentovaná v hexadecimální soustavě a  $p'$  je výsledná hodnota pixelu.

#### ■ Inverze[INV]

Pro každý pixel obrázku pak uděláme následující:

```

r := 255 - (p >> 16) & 0xff),
g := 255 - (p >> 8) & 0xff),
b := 255 - (p & 0xff),
p' := (p & 0xff000000) | (r << 16) | (g << 8) | b,

```

kde  $p$  je hodnota pixelu,  $r$  je červená složka pixelu,  $g$  je zelená složka pixelu,  $b$  je modrá složka pixelu, `>>` je binární posun doprava, `<<` je binární posun doleva, `&` je binární AND, `|` je binární OR, hodnoty začínající `0x` jsou celá čísla reprezentovaná v hexadecimální soustavě a  $p'$  je výsledná hodnota pixelu.

#### ■ Rozostření (Box Blur)[BLU]

*Poznámka: V této naivní implementaci nejsou ošetřeny okraje obrázku.* Mějme hodnotu  $r \in \mathbb{N}$  reprezentující poloměr rozostření. Vytvoříme si nové pole o velikost  $w * h$ , kde  $w$  je šířka obrázku a  $h$  je výška obrázku. Pak pro každý pixel uděláme následující:

Podíváme se okolo současného pixelu do čtvercové oblasti o poloměru  $r$  s současným pixelem ve středu, tedy čtvercová oblast je pomyslná mřížka se stranou o velikosti  $2 * r + 1$ . Pak mějme proměnné  $acc_a = 0$ ,  $acc_r = 0$ ,  $acc_g = 0$ ,  $acc_b = 0$  a  $ctr = 0$ , pro každý pixel v této čtvercové oblasti provedeme toto:

```

acc_a := acc_a + (p >> 24) & 0xff),
acc_r := acc_r + (p >> 16) & 0xff),
acc_g := acc_g + (p >> 8) & 0xff),
acc_b := acc_b + (p & 0xff),
ctr := ctr + 1.

```

Pixel má pak tuto výslednou hodnotu:

```

p' := ((acc_a / ctr) << 24)
      | ((acc_r / ctr) << 16)
      | ((acc_g / ctr) << 8)
      | (acc_b / ctr),

```

kde  $p$  je hodnota pixelu, `>>` je binární posun doprava, `<<` je binární posun doleva, `&` je binární AND, `|` je binární OR a hodnoty začínající `0x` jsou celá čísla reprezentovaná v hexadecimální soustavě.



### ■ Zostření[SHA]

*Poznámka: V této naivní implementaci nejsou ošetřeny okraje obrázku.* Mějme hodnotu  $v \in \mathbb{N}$  reprezentující sílu zostření. Vytvoříme si nové pole o velikost  $w * h$ , kde  $w$  je šířka obrázku a  $h$  je výška obrázku. Pak pro každý pixel provedeme stejné rozostření jako v předchozím bodě akorát s tím rozdílem, že výsledná hodnota pixelu počítána takto:

```

r := (p >> 16) & 0xff),
g := (p >> 8) & 0xff),
b := (p & 0xff),
p' := (p & 0xff000000)
      | ((r + (r - (acc_r / ctr)) * 0.5 * v) << 16)
      | ((g + (g - (acc_g / ctr)) * 0.5 * v) << 8)
      | (b + (b - (acc_b / ctr)) * 0.5 * v)),

```

kde  $p$  je hodnota pixelu,  $r$  je červená složka pixelu,  $g$  je zelená složka pixelu,  $b$  je modrá složka pixelu,  $>>$  je binární posun doprava,  $<<$  je binární posun doleva,  $&$  je binární AND,  $|$  je binární OR, hodnoty začínající  $0x$  jsou celá čísla reprezentovaná v hexadecimální soustavě,  $p'$  je výsledná hodnota pixelu a proměnné  $acc_r$ ,  $acc_g$ ,  $acc_b$ ,  $ctr$  jsou stejné proměnné jako u rozmazání.

### ■ Odstranění červených očí

U této funkce je nutné nějakým způsobem nejdříve získat oblast, na kterou se algoritmus pro odstranění červených očí má aplikovat. Jelikož program nemá být plnohodnotný obrázkový editor a nemá být v tomto směru komplexní, zvolíme jednodušší metodu, tj. necháme uživatele provést výběr oblasti, kde se červené oči nacházejí. Uživatel tedy nejdříve vybere v obrázku obdélníkovou oblast  $([x, y], w, h)$ , kde  $[x, y]$  je počáteční bod oblasti,  $w$  je šířka oblasti a  $h$  je výška oblasti. Jelikož je oční zornice spíše kulatého tvaru, následně se pro takovou oblast zvolí střed

$C = [x + w/2, y + h/2]$  a poloměr  $r = (max(w, h))/2$ , kde  $max$  je pomocná funkce, která ze všech svých vstupních hodnot zvolí tu největší. Pro každý pixel v oblasti  $(C, r)$  pak vypočítáme jeho výslednou hodnotu takto[RED]:

```

m := ((p >> 8) & 0xff) + (p & 0xff)) / 2,
p' := (p & 0xff000000) | (m << 16) | (m << 8) | m,

```

kde  $p$  je hodnota pixelu,  $>>$  je binární posun doprava,  $<<$  je binární posun doleva,  $&$  je binární AND,  $|$  je binární OR, hodnoty začínající  $0x$  jsou celá čísla reprezentovaná v hexadecimální soustavě a  $p'$  je výsledná hodnota pixelu.

## 9.4 Přizpůsobení výsledné webové galerie

Uživatel by měl mít možnost upravit všechny barvy a texty ve výsledné webové galerii. Pro pokročilejší uživatele by měl program nabízet i možnost upravit CSS a JS galerie. Samotná struktura galerie - počet řádků a sloupců mřížky, pozice navigace a podobně - by také měla jít upravit, alespoň v nějaké omezené míře. Stejně jako v předešle analyzovaných programech by měly být dostupné nějaké předpřipravené styly, které uživatel může využít nebo z nich při dalších následujících úpravách může vycházet. Všechny tyto požadavky řeší program pomocí šablon.

### 9.4.1 Šablona

Šablona je složka obsahující veškeré soubory potřebné pro generování galerie. Pro ukládání informací o samotné šabloně, jako je název, verze, autor a podobně, je v šabloně soubor `info.txt`. Dále zde existují soubory, které obsahují strukturu stránek galerie: `index.html`, `list.html`, `single.html` atd. V těchto souborech jsou pak určité speciální textové vzory, které do těchto souborů umožňují dynamicky vkládat data, jako je například seznam obrázků a podobně.

### 9.4.2 `info.txt`

Tento soubor obsahuje informace (metadata) o samotné šabloně. Program je umí z tohoto souboru přečíst a ty pak zobrazit uživateli (třeba název šablony, který se objeví ve výběru šablony v programu). Na každém řádku je uvedena jedna informace, kde název je první, pak následuje dvojtečka a pak je hodnota. Takový soubor pak může vypadat třeba takto:

```
Name: Název šablony
```

```
Version: 1.0
```

```
Author: Jan Novák
```

```
Description: Jednoduchá šablona, podporuje Lightbox.
```

### ■ 9.4.3 Formát textových vzorů pro dynamická data

V šablonových souborech (template files) lze použít textové vzory, se kterými je možné do takovýchto souborů vkládat dynamicky data. Textový vzor má tento formát: `{{ data_name }}`, kde `data_name` je název informace, která se má na to dané místo vložit. Příkladem pak třeba může být:

```
<p>Tato galerie používá šablonu {{ template.name }}</p>
```

### ■ 9.4.4 Šablonové soubory

V šabloně mohou existovat určité šablonové soubory, které program využije, pokud jsou dostupné. Pokud dostupné nejsou, využijí se výchozí. Takové soubory jsou:

- **index.html**  
Šablona pro index galerie, tedy taková domovská stránka galerie. Na této stránce se většinou vypisují informace o galerii a seznam složek a obrázků v první úrovni hierarchie.
- **directory.html**  
Šablona pro složku v galerii. Tento soubor se použije, když uživatel přejde do složky, například kliknutím na odkaz z indexu galerie.
- **single.html**  
Šablona pro stránku se samotným obrázkem. Tento soubor se použije, když uživatel klikne na odkaz nějakého obrázku ve výpisu obrázků.
- **header.html**  
Šablona pro záhlaví stránek. Zobrazuje se na každé stránce, kde se nachází textový vzor `{{ header }}`.
- **footer.html**  
Šablona pro záhlaví stránek. Zobrazuje se na každé stránce, kde se nachází textový vzor `{{ footer }}`.
- **navigation.html**  
Šablona pro navigaci galerie. Zobrazuje se na každé stránce, kde se nachází textový vzor `{{ navigation }}`.
- **list.html**  
Šablona pro výpis obrázků na dané stránce. Zobrazuje se na každé stránce, kde se nachází textový vzor `{{ list }}`. Pro jednotlivé položky v seznamu využívá šablonový soubor `list-item.html`.

- `list-item.html`  
Šablona pro položku ve výpisu obrázků/složek.
- `pagination.html`  
Šablona pro stránkování galerie. Zobrazuje se na každé stránce, kde se nachází textový vzor `{{ pagination }}`.

#### ■ 9.4.5 CSS galerie

Hlavním souborem CSS galerie je ve výchozím nastavení soubor `style.css`. Toto lze upravit změnou souboru `header.html`. Nachází se zde všechny styly, barvy apod.

#### ■ 9.4.6 Skripty (JS) galerie

Hlavním souborem JS galerie je ve výchozím nastavení soubor `script.js`, pokud existuje. Toto lze upravit změnou souboru `header.html`. Nachází se zde případná JS funkcionality, například pro Lightbox.

#### ■ 9.4.7 Uživatelské rozhraní

Vydeme-li z předešle navržených funkcí, v okně přizpůsobení galerie se nachází:

- pole pro vyplnění názvu galerie,
- výběr stylu galerie a případná správa stylů (jelikož ve správě stylů mají jít přidávat a odebírat styly a nejedná se přímo o přizpůsobení webové galerie, volil bych umístění této správy do samostatného okna),
- pole pro zadání počtu řádků a sloupců mřížky galerie,
- výběr pozice navigace,
- správa barev (jelikož bude většinou obsahovat vícero polí, volil bych umístění do samostatného okna)

- správa textů (jelikož bude většinou obsahovat vícero polí, volil bych umístění do samostatného okna)
- pole pro vlastní CSS,
- pole pro vlastní JavaScript.

Některá pole v tomto okně nemusí nutně u některých stylů být použita, a proto je tedy na dalším zvážení, zdali by se vůbec měla takováto pole zobrazovat v tomto okně, nebo zdali raději v nějakém jiném - například v okně pro nastavení stylu.

## 9.5 Vytvoření galerie

Program vezme uživatelem vybranou šablonu a veškeré informace o galerii, jako je seznam obrázků atd. a vygeneruje webovou galerii pomocí procesu vytvoření.

### 9.5.1 Proces vytvoření

Program při vytváření prochází následujícím procesem:

- Vytvoří se soubor `index.html` a vloží se do něj dynamická data, tedy šablonové soubory jako `header.html`, `footer.html` apod. Tím se vygeneruje statický HTML kód pro index galerie.
- Po jednom se pak procházejí obrázky (a složky) ze seznamu, který uživatel importoval do programu, a pro každý z nich se vygeneruje statický HTML soubor pomocí šablonových souborů. Každý obrázek se také zkopíruje do připravené podsložky ve složce `images` odpovídající jeho hierarchii v programu a vygeneruje se i náhledový obrázek, který se pak zobrazuje ve výpisu obrázků.
- Z dynamických dat se vloží barvy do CSS šablony, přidá se uživatelský CSS kód, pokud byl vyplněn, a vygeneruje se soubor `style.css`.
- Přidá se uživatelský JS kód, pokud byl vyplněn, a vygeneruje se soubor `script.js`.

Po tomto procesu je galerie připravená k použití. Jelikož se jedná o statické HTML soubory, lze ji přenášet, archivovat atd.

## 9.6 Okno pro správu stylů

V okně se nachází výpis všech existujících a programem úspěšně přečtených stylů (případně i oddělený seznam chybných či neúspěšně přečtených stylů s dodatečnou chybovou hláškou), tlačítko pro odebrání stylu a pro přidání stylu, u kterého se po kliknutí na něj zobrazí dialogové okno pro výběr složky (nebo například i pro výběr ZIP souboru) se soubory stylu, který uživatel chce přidat.

## 9.7 Okno pro správu barev

V okně se nachází výpis všech názvů barev a jejich příslušných hodnot, které lze měnit. V dolní části se pak ještě nachází tlačítka pro zrušení a potvrzení změn.

## 9.8 Okno pro správu textů

V okně se nachází výpis všech názvů textů a jejich příslušných hodnot, které lze měnit. V dolní části se pak ještě nachází tlačítka pro zrušení a potvrzení změn.

## 9.9 Vygenerování galerie

Funkcionalita samotného vytvoření galerie již byla navržena v minulé zprávě, v této sekci se tedy budu věnovat pouze návrhu okna, které se zobrazí uživateli, který chce galerii vygenerovat.

### ■ 9.9.1 Uživatelské rozhraní

Pro vygenerování galerie je nutná výstupní cesta, tedy cesta, kam se uloží soubory vygenerované galerie, vybraná šablona galerie a seznam obrázků. Styl galerie a seznam obrázků uživatel zvolil v okně přizpůsobení galerie a okně s výpisem obrázků, a tudíž zbývá zvolit akorát výstupní cestu. V okně vygenerování galerie by se tedy mělo nacházet pole pro výstupní cestu a pro uživatelskou přívětivost také i tlačítko pro otevření dialogu pro výběr složky, aby uživatel nemusel případně vypisovat celou cestu ručně. Dále také tlačítko pro zrušení a tlačítko pro vygenerování galerie, u kterého se po kliknutí na něj spustí proces vytvoření galerie.



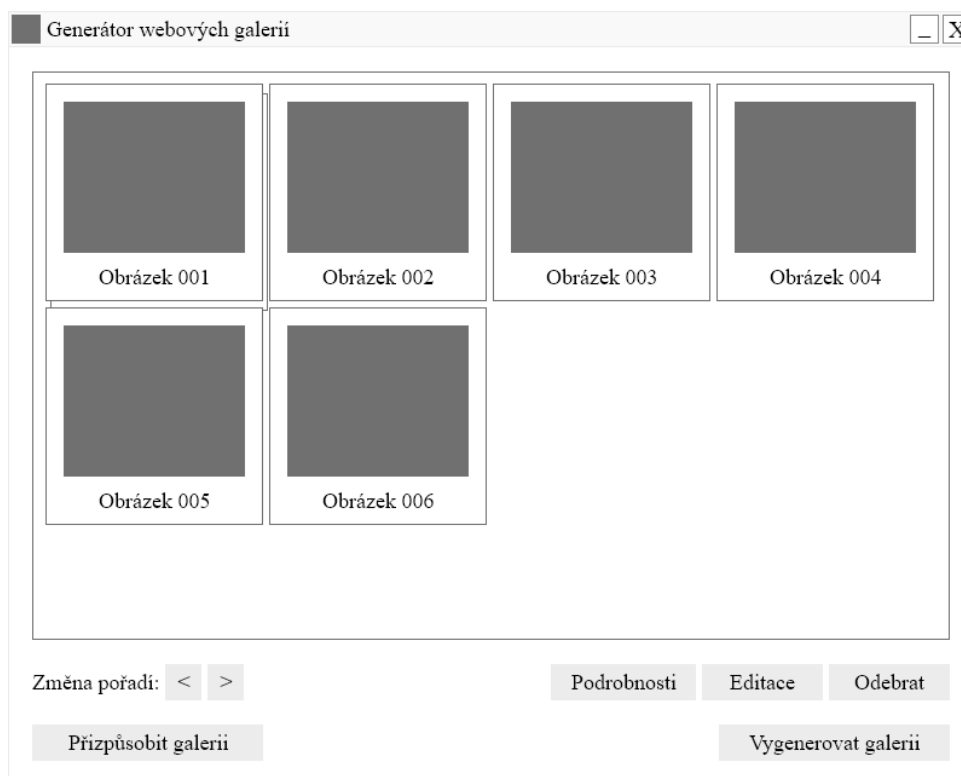




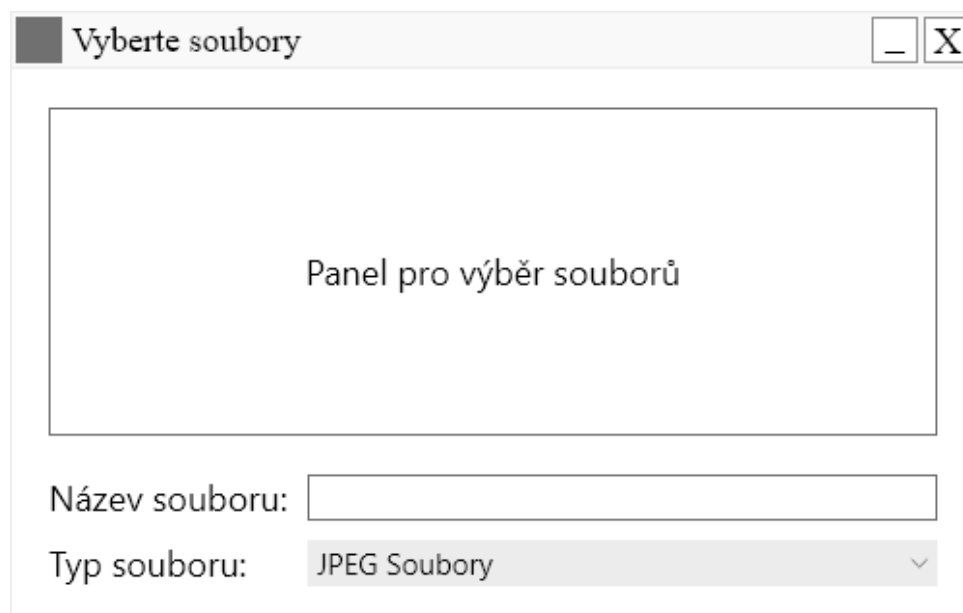
## Kapitola 10

### Grafický návrh programu

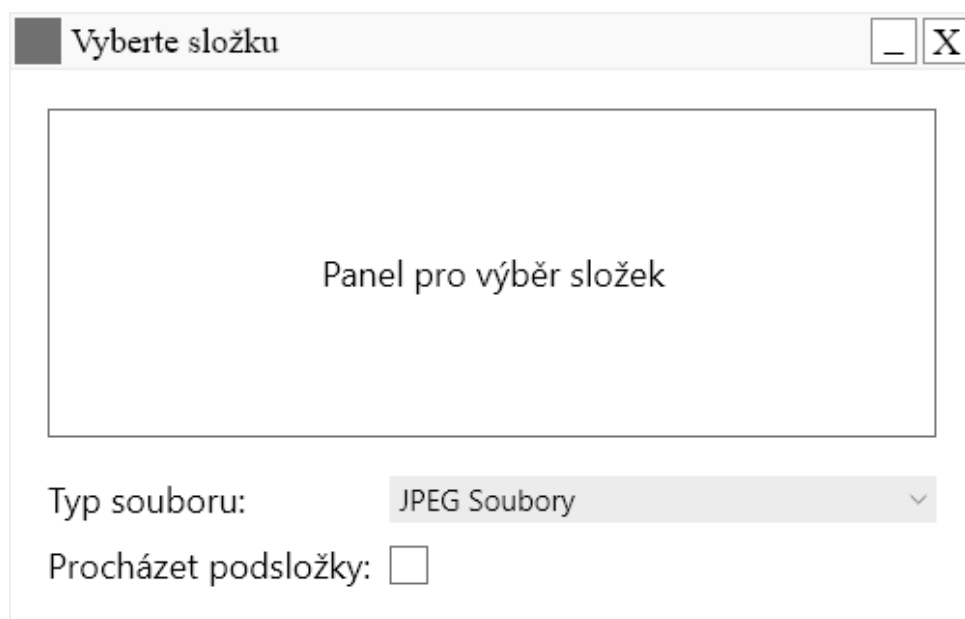
V této sekci se nacházejí grafické návrhy pro jednotlivá okna programu. Jedná se především o přetvoření textového popisu v předchozí kapitole do nějaké vizuální podoby. Každý z návrhů bere na vědomí jednoduchost, přehlednost a srozumitelnost grafického rozhraní a využívá často již existující prvky nacházející se v různých grafických knihovnách.



**Obrázek 10.1:** Hlavní okno programu s výpisem obrázků



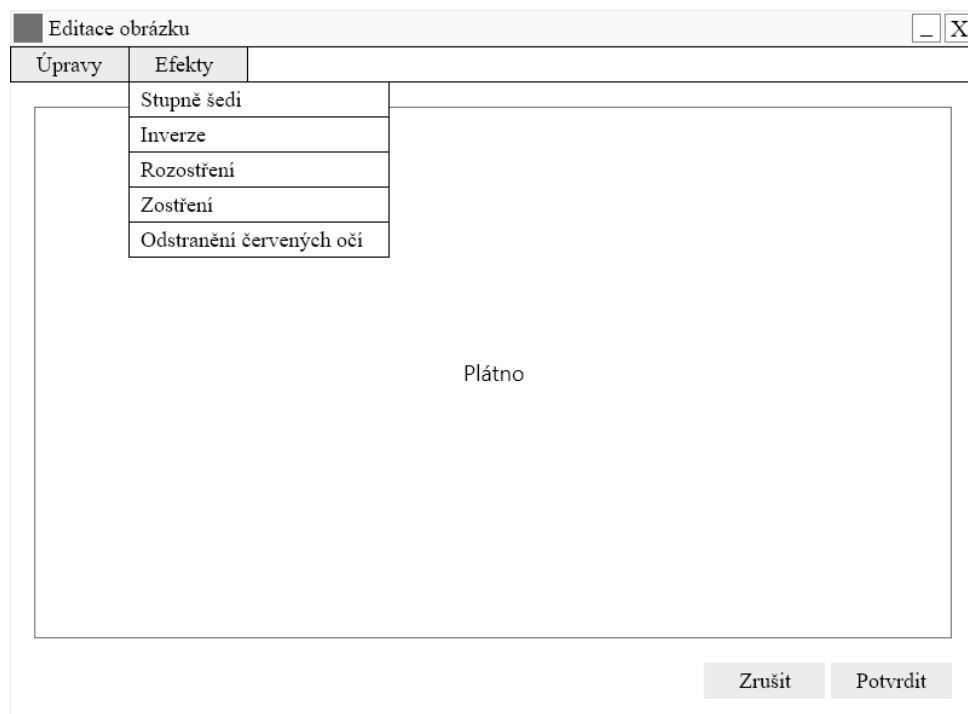
**Obrázek 10.2:** Import obrázků - dialogové okno pro výběr souborů



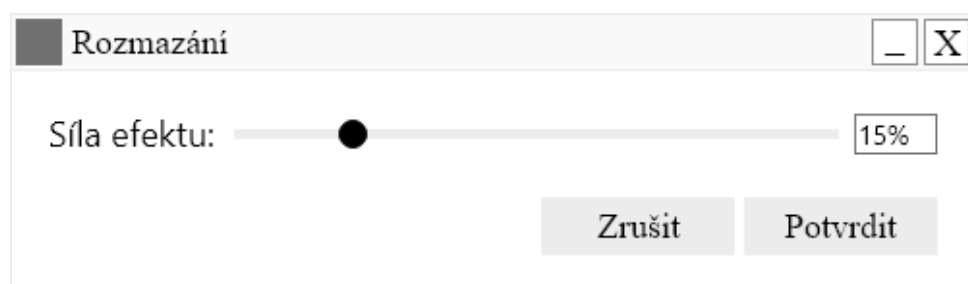
**Obrázek 10.3:** Import obrázků - dialogové okno pro výběr složky



**Obrázek 10.4:** Okno pro editaci obrázku



**Obrázek 10.5:** Okno pro editaci obrázku, s otevřeným menu



**Obrázek 10.6:** Editace obrázku - dialogové okno nastavení efektu

Prizpusobit galerii

Název galerie:

Styl galerie: Jednoduchý styl ▾

Spravovat styly

Mřížka galerie:  x

Pozice navigace: Nahoře ▾

Spravovat barvy Spravovat texty

Vlastní CSS:

Vlastní JavaScript:

Zrušit Potvrdit

**Obrázek 10.7:** Okno přizpůsobení galerie

Vygenerovat galerii

Výstupní cesta:  ...

Zrušit Vygenerovat

**Obrázek 10.8:** Okno vygenerování galerie



# Kapitola 11

## Implementace programu

Tato sekce se zabývá implementací samotného programu. Implementace vychází primárně ze zadání a z návrhu implementace v jedné z předchozích kapitol. Program je implementován v programovacím jazyce C++ a je využíván standard C++17.

### 11.1 Použité knihovny

Implementace programu využívá nejen standardní knihovnu jazyka C++, ale také knihovny třetích stran, jimiž jsou:

- Pro grafické rozhraní aplikace je použita knihovna *Dear ImGui*, která využívá dále platformu *GLFW* a knihovnu *OpenGL* pro vykreslování.
- Pro čtení a ukládání obrázků a práci s obrázkovými daty je v implementaci použita knihovna *CImg*.
- Pro dialog výběru souborů a složky je použita knihovna *ImFileDialog*.

## 11.2 Model

Jedním z hlavních tříd v implementaci, která se využívá napříč programem, je třída **Modelu**, jež umožňuje globálně v instanci programu ukládat data a reagovat na jejich změny. V Modelu jsou umístěny páry klíčů a hodnot, kde každý klíč má svůj vlastní název a má přiřazen povolený typ hodnoty. Pomocí daného klíče lze pak do Modelu ukládat pouze takové hodnoty, které typově odpovídají povolenému typu hodnoty klíče. Model pak umožňuje data vkládat, mazat, získávat, kontrolovat, zdali je hodnota přítomná, a navíc také, pokud je hodnota typu `std::vector`, přidávat položky do seznamu.

Model také využívá principu událostí a posluchačů (Events and Listeners), který umožňuje jednoduše upozorňovat všechny objekty, které se přihlásili k poslouchání, na změnu dat. Při přidání, odebrání a nastavení hodnoty u nějakého klíče, informuje Model všechny zaregistrované posluchače o změně této hodnoty, včetně informace o tom, jaká změna vlastně nastala.

## 11.3 Import obrázků

Import obrázků je v programu možný dvěma způsoby - buď lze importovat jednotlivé obrázky výběrem souborů z dialogového okna pro výběr souborů nebo lze importovat celou složku pomocí dialogového okna pro výběr složky. Oba tyto způsoby lze nalézt v podnabídce *Import* v hlavní nabídce, která je umístěna nahoře v okně programu.

Pro import jsou v implementaci definovány následující typy, které usnadňují práci s importovanými soubory a složkami, jejich daty a také jsou dále využívány při jiných operacích v programu:

- `image_tex_t` - uchovává ID OpenGL textury, její formát, šířku a výšku.
- `image_file_data_t` - uchovává doplňující informace o souboru, konkrétně jeho popis a datum vytvoření.
- `image_file_t` - uchovává hlavní informace o souboru, jako je jeho cesta, obrázková data, textura (pro OpenGL), stav (pro informaci, zdali nastala při importování chyba či nikoliv) a doplňující informace o souboru (`image_file_data_t`).



- `image_dir_t` - uchovává informace o složce, jako je její cesta, podsložky a soubory a její stav (pro informaci o výsledku importování, stejně jako u `image_file_t`).

Definicí těchto typů a implementací operací nad těmito typy lze v programu uchovávat všechna důležitá data o importované položce na jednom místě.

### ■ 11.3.1 Výběr souborů

Zvolením položky *File* v podnabídce *Import* se otevře dialogové okno pro výběr souborů. Uživatel pak po výběru jednoho či více souborů kliknutím na tlačítko *Open* importuje dané soubory. Pro každý vybraný soubor proběhne proces importování souboru.

### ■ 11.3.2 Výber složky

Zvolením položky *Directory* v podnabídce *Import* se otevře dialogové okno pro výběr složky. Uživatel pak po výběru složky kliknutím na tlačítko *Open* importuje danou složku. Pro vybranou složku proběhne proces importování složky.

## ■ 11.4 Proces importování

Po vybrání souborů či složky uživatelem se tyto dané položky předají třídě *Importer*, která se stará o import souborů a složek. Ta se tyto položky pokusí importovat a extrahovat z nich požadovaná data. Takto je odděleno vybírání souborů a složek a samotná logika importování.

### ■ Proces importování souboru

Importování souboru probíhá v několika fázích:

- V první fázi se program pokusí otevřít soubor. Pokud soubor neexistuje či nastala jiná chyba, program soubor přeskočí.
- V druhé fázi se program pokusí přečíst metadata souboru (toto je dále popsáno v sekci Extrakce metadat ze souboru). Pokud nelze metadata přečíst či extrahovat, pak program podporovaná metadata vyplní jako prázdná a název obrázku se nastaví na stejnou hodnotu jako je název souboru.
- Ve třetí fázi se program pokusí pomocí knihovny *CImg* přečíst samotná data obrázku. Pokud data nelze přečíst, obrázek je přeskočen.

Knihovna *CImg* umí načítat obrázky buď z textového řetězce cesty obrázku nebo z ukazatele na `std::FILE`. Způsob s ukazatelem ale knihovna nepodporuje pro čtení *TIFF* souborů, a tudíž je nutné v tomto případně otevřít soubor dvakrát - jednou při extrakci metadat a podruhé při čtení obrázkových dat.

Program používá ke čtení souborů novější způsob pomocí datových proudů, tzv. streams, konkrétně třídu `std::ifstream`. Instanci této třídy ale nelze žádným způsobem přeměnit na ukazatel na `std::FILE`, a tudíž se ve všech případech otevírá každý soubor dvakrát.

Po úspěšném přečtení obrázkových dat jsou jednotlivé pixely obrázku převedeny do formátu RGBA (red, green, blue, alpha). Všechny pixely pak mají tedy čtyři komponenty. Tímto se následně zjednoduší proces editace obrázku (viz sekce Editace obrázků).

- Ve čtvrté fázi se pouze uzavře daný soubor. Pokud nenastala žádná chyba v předchozích fázích, program úspěšně importoval daný obrázek.

Těmito fázemi se do programu naimportují veškerá potřebná data - metadata obrázku, tedy především datum pořízení obrázku, a samotná obrázková data.

## ■ Proces importování složky

Pro importování složky program nejprve rekurzivně prochází všechny položky složky, tedy nejdříve prochází soubory a složky, které jsou přímými potomky. Jedná-li se o soubor, je daný soubor importován podle procesu importování souboru. Jedná-li se však o složku, program pro tuto složku rekurzivně provede proces importování složky. Tímto způsobem se naimportuje celá struktura vybrané složky včetně všech souborů, které lze importovat.

## ■ Extrakce metadat ze souboru

Program podporuje extrakci metadat ze souboru obrázku typu *JPEG/EXIF* a *TIFF*. Konkrétně se jedná především o datum pořízení obrázku, které pak lze zobrazit i ve výsledné webové galerii. Extrakce pro soubory typu *JPEG/EXIF* a *TIFF* je avšak navzájem odlišná.

Pro *JPEG/EXIF* soubory program nejdříve přečte značku *Start of Image* (SOI) a určí pořadí bajtů (endianitu). Následně prochází sekce označené značkou `0xFFxx`, kde `xx` je specifické číslo označující typ značky, dokud nenarazí na sekci **APP1**, jež obsahuje hledaná EXIF metadata. Obsah této sekce je pak strukturován stejně jako u souboru *TIFF*, proto program následně využije stejné procedury jako pro čtení tohoto typu souborů.

Pro *TIFF* soubory program nejdříve přečte hlavičku souboru - pořadí bajtů (endianitu), značku, že se jedná o soubor typu *TIFF*, a pozici (tzv. offset) prvního adresáře (tzv. Image file directory), kde se nachází záznamy (tzv. entries) dat. Pak prochází záznamy těchto adresářů tak dlouho, dokud buď již žádné další adresáře nejsou, nebo narazil na adresář se záznamem s označením `0x8769`, který označuje záznam s pozicí EXIF metadat. Po přesunu na danou pozici, program zpracovává EXIF záznamy. Jakmile narazí na záznam se značkou `0x9003` a typem 2, tj. textový řetězec v kódování ASCII (American Standard Code for Information Interchange), tak tento záznam zpracuje, jelikož se jedná o záznam s datem pořízení obrázku. Jelikož program momentálně neextrahuje žádná další metadata, po extrakci hodnoty tohoto záznamu, čtení metadat zastaví a vrátí nashromážděná data.

## ■ 11.5 Výpis obrázků

Po importování souborů či složek uživatelem otevře program nové okno s výpisem importovaných položek. V tomto okně se nachází textový vstup pro název nové složky, informace o tom, v jaké složce se momentálně uživatel nachází (označení / znamená, že se uživatel nachází v kořenové složce), tlačítko pro odebrání vybrané položky ze seznamu, tlačítko pro vybrání náhledového obrázku vybrané složky a tlačítka pro přesun vybrané položky. Pod tímto panelem se pak nachází samotný výpis obrázků, jenž je řešen zobrazením mřížky náhledů souborů a složek. Tato mřížka se sama přizpůsobuje velikosti okna, kde se buď přidávají, či odebírají sloupce a položky se tak samy zarovnávají. Lze také měnit pořadí položek, buď pomocí táhnutí a puštění (Drag and drop) či pomocí tlačítek přesunu položky. Dále lze dvojklikem na položku souboru

zobrazit okno s doplňujícími informacemi o souboru, jež je možné i přímo měnit, a tlačítkem pro vstup do integrovaného obrázkového editoru, kde je možné obrázek upravit. Takto má uživatel k dispozici všechny funkce, které program nabízí, na jednom místě a to včetně přehledného výpisu položek.

Pro výpis položek jsou v implementaci definovány následující typy, které usnadňují práci s daty ve výpisu a následně pak i v generování samotné webové galerie:

- `gallery_dir_t` - uchovává informace o doplňujících datech potřebných k zobrazení složky ve výpisu obrázků, konkrétně obrázková data a texturu (pro OpenGL) náhledového obrázku složky. Dále také informaci o tom, zdali se má použít výchozí náhledový obrázek, který je umístěn ve složce programu a dále `resources/directory.png`.
- `gallery_item_t` - uchovává informace potřebné k zobrazení položky ve výpisu obrázků, konkrétně nadřazenou složku položky (v případě, že se jedná o kořenovou složku, pak je hodnota `nullptr`), seznam podřazených položek (pouze složky), ukazatel na dodatečné informace o položce (tj. v případě souboru hlavní i doplňující informace, v případě složky pak pouze doplňující informace), zdali je položka složkou či nikoliv a název položky.

Pomocí těchto definovaných typů lze v programu vytvořit stromovou strukturu hierarchie položek, která se pak následně také použije i pro vygenerování samotné webové galerie.

### 11.5.1 Hierarchie položek

Program umožňuje vytvářet složky ve složkách a nabízí tak uživateli možnost určit hierarchie souborů. Novou složku lze vytvořit zadáním názvu nové složky v okně s výpisem obrázků a kliknutím na tlačítko *Create*. Tímto se nová složka vytvoří v současné složce a lze do ní vstoupit. V té lze pak importovat soubory a složky stejným způsobem jako v kořenové složce.

### ■ 11.5.2 Náhledový obrázek složky

Pro každou složku, kromě kořenové složky, lze vybrat její náhledový obrázek, který se následně poté zobrazuje ve výsledné webové galerii. Takto je uživateli umožněna větší přizpůsobitelnost webové galerie.

## ■ 11.6 Editace obrázků

V programu se nachází i integrovaný obrázkový editor, kde je uživateli umožněno upravit importované obrázky. Nejedná se o plnohodnotný obrázkový editor, a tudíž jsou k dispozici pouze základní funkce pro úpravu obrázků. Obrázkový editor je přístupný přes okno informací o vybraném obrázku.

V okně editoru obrázků se nachází tlačítka operací, úprav a filtrů, jež lze u obrázku provést. Pod nimi je lišta přiblížení obrázku, kde se nachází informace o současném přiblížení, možnost obrázek přiblížit a oddálit a obnovit původní přiblížení. Pod touto lištou jsou tlačítka pro uložení a zrušení provedených změn. Zbytek prostoru pak vyplňuje plátno, kde se zobrazuje současný stav obrázku.

Všechny obrázky mají své pixely při importování převedeny do formátu *RGBA* (red, green, blue, alpha). Jejich pixel mají tedy vždy čtyři komponenty - červenou, zelenou, modrou a alfa kanál (průhlednost pixelu). Tímto se zjednoduší celý proces editace obrázku - všechny funkce se musí implementovat pouze pro variantu s pixely s formátem *RGBA*.

Všechny funkce, které editor podporuje, jsou na nižší úrovni (tj. pracující pouze s pixely obrázku) implementovány ve třídě `ImgOp` a to především podle návrhu implementace s občasnými menšími úpravami.

Funkce nad samotným obrázkem pak obstarává třída `ImageEditor`, jež uchovává současný stav obrázku (hodnoty pixelů, šířku, výšku) a jeden uložený předchozí stav obrázku, který se využívá při zrušení úprav, jako je úprava jasu atd. Dále také obstarává aktualizaci textury pro OpenGL, jež je vykreslována v okně editoru.

Jelikož textury OpenGL a třída `CImg` mají interně komponenty pixelů uložené v jiném pořadí, je nutné před každou aktualizací instance třídy `CImg`

data zpřeházet. K tomu slouží funkce `pixels_to_cimg_pixels`, opačně pak funkce `image_pixels_rgba`.

Editor nabízí následující funkce:

- **Operace** - horizontální a vertikální překlopení, otočení doleva či doprava a obdélníkové oříznutí.
- **Úpravy** - úprava jasu, kontrastu, převod na odstíny šedi, invertování barev a odstranění červených očí.
- **Filtry** - rozmazání a zostření.

### 11.6.1 Obdélníkové oříznutí

Obdélníkové oříznutí je jedna ze dvou funkcí, která vyžaduje větší vstup od uživatele než jen jednu hodnotu. V editoru je tato funkce řešena tak, že po kliknutí na tlačítko *Crop* se uživatel přepne do módu oříznutí a po kliknutí na obrázek a tažením myši se vytvoří obdélníková oblast, jež má být novým obrázkem. Tato oblast je v okně zobrazena jako červený obdélník.

### 11.6.2 Odstranění červených očí

Podobně jako je řešeno obdélníkové oříznutí, je řešeno i odstranění červených očí. Nicméně jelikož je tvar zornice oka spíše kruhový, místo obdélníkové oblasti se vezme místo prvního kliknutí myši jako střed kružnice a druhý bod jako poloměr dané kružnice. Tažením myši druhým bodem lze pak kružnici zvětšovat a zmenšovat. Kliknutím na jinou pozici lze obnovit původní střed kružnice a začít tak znovu. Oblast, která bude upravena, je pak zobrazena v okně jako modrá kružnice.

## 11.7 Generování webové galerie

Po importování obrázků a jejich případné úpravě má uživatel možnost vygenerovat si z nich webovou galerii. Měl by mít také možnost si galerii upravit,

tj. například změnit barvy a velikost textu, počet obrázků na stránku atd. V implementaci programu je toto řešeno přes témata, šablony a konfigurační soubory tématu. Oproti návrhu programu je ve finální implementaci použito pojmenování Téma namísto Šablony a šablonové soubory jsou pojmenovány jen jako šablony. Dále jsou také některé změny v pojmenování šablon.

Uživatel si veškeré nastavení výsledné webové galerie volí v okně Generování galerie, kde jsou vypsané všechny podporované vlastnosti, jako je název galerie, počet položek na stránku apod.

### ■ 11.7.1 Textové vzory

Textové vzory jsou textové řetězce formátu `{{ text }}`, kde `text` je obsah daného vzoru. Obsahem může být:

- Název nějaké proměnné (např. šířka právě vykreslovaného obrázku v galerii či třeba jeho název).
- Výraz pro vložení jiného souboru, tj. formát textu: `include("název-souboru")`.
- Podmínkový výraz tj. formát textu: `if (var)`, kde `var` je název proměnné pro kontrolu existence a prázdné hodnoty. Podmínku lze také negovat, tedy: `if(not var)`. Podmínkový blok pak musí být ukončen textovým výrazem `{{ endif() }}`. Pokud je podmínka splněna, vše mezi začátkem bloku (`if`) a koncem bloku (`endif`) zůstane ve výsledném souboru, pokud není splněna, vše mezi začátkem a koncem bloku nebude ve výsledném souboru.

### ■ 11.7.2 Šablony

Šablony jsou soubory typu `.html`, `.css` nebo `.js`, které mohou obsahovat textové vzory. Šablony jsou zpracovány programem tak, že vezmou existující proměnné (argumenty) z daného kontextu a pomocí nich nahradí textové vzory obsahem dané proměnné. Pokud proměnná neexistuje, pak je textový vzor nahrazen prázdným textovým řetězcem.

### 11.7.3 Téma

Hlavním prvkem při generování webové galerie je nutnost tématu. Téma si uživatel vybere před spuštěním generování galerie a následně je pak použito pro vytvoření HTML struktury a spojení všech dat z programu dohromady.

Každé téma obsahuje několik základních souborů:

- **info.txt** - Obsahuje informace o daném tématu, které se načítají do programu do okna Generování galerie.
- **\_\_base.html** - Hlavní soubor galerie obsahující základní strukturu HTML dokumentu. Měly by se v něm nacházet HTML prvky jako `<html>`, `<head>`, `<body>`, `<link>` pro kaskádové styly a `<script>` pro JavaScript soubory či obsah. Dalé by měl také obsahovat textové vzory `{{ header }}`, `{{ footer }}`, `{{ content }}` a `{{ title }}`, případně `{{ custom_css }}`, `{{ custom_js }}`
- **header.html** - Soubor používaný pro záhlaví galerie. Měl by se v něm nacházet název galerie (`{{ gallery.name }}`), cesta složky (`{{ directory_path }}`) a navigační menu (`{{ navigation }}`).
- **navigation.html** - Soubor používaný pro navigační menu, měl by obsahovat textový vzor `{{ navigation.items }}`.
- **navigation-item.html** - Soubor používaný pro položku navigačního menu, měl by obsahovat textový vzor `{{ navigation.item.is_current }}`, `{{ navigation.item.url }}` a `{{ navigation.item.title }}`.
- **directory-path-part.html** - Soubor používaný pro položku navigačního menu, lze v něm použít textové vzory `{{ directory.is_root }}`, `{{ directory.is_current }}`, `{{ directory.url }}` a `{{ directory.name }}`.
- **footer.html** - Soubor používaný pro zápatí galerie.
- **index.html** - Soubor používaný pro hlavní stránku galerie, tj. pro kořenovou složku galerie. Měl by obsahovat i textový vzor `{{ pagination }}`.
- **directory.html** - Soubor používaný pro ostatní složky galerie. Měl by obsahovat i textový vzor `{{ pagination }}`.
- **list.html** - Soubor používaný pro výpis položek v galerii. Měl by obsahovat textový vzor `{{ items }}`.



- **list-item.html** - Soubor používaný pro jednotlivé položky ve výpisu položek v galerii. Lze použít textové vzory `{{ item.url }}`, `{{ item.path }}`, `{{ item.width }}`, `{{ item.height }}`, `{{ item.name }}`.
- **pagination.html** - Soubor používaný pro stránkování. Měl by obsahovat textové vzory `{{ pagination.items }}`, případně i `{{ pagination.previous.url }}` a `{{ pagination.next.url }}`.
- **pagination-item.html** - Soubor používaný pro položku ve stránkování. Měl by obsahovat textové vzory `{{ pagination.item.is_current }}`, `{{ pagination.item.url }}` a `{{ pagination.item.title }}`.
- **pagination-separator.html** - Soubor používaný pro položku značící vícero stránek ve stránkování.
- **single.html** - Soubor používaný pro stránku jednotlivých obrázků. Měl by obsahovat textové vzory `{{ item.path }}`, `{{ item.name }}`, `{{ item.width }}`, `{{ item.height }}`, `{{ item.date }}`, `{{ item.description }}`, případně `{{ list.page }}`.
- **settings.txt** - Volitelný soubor. Obsahuje definice doplňujících vlastností, které jsou použité v šablonách tématu. Ty se pak zobrazí i v programu v okně Generování galerie.

Program nabízí jedno výchozí téma. Lze z něho vycházet při tvorbě vlastních témat.

#### ■ 11.7.4 info.txt

Tento soubor obsahuje hlavní informace o tématu, stejně jako bylo v návrhu programu. Program pak uživateli zobrazuje název tématu přímo v programu před spuštěním generování galerie. Na každém řádku je uvedena jedna informace, kde název je první, pak následuje dvojtečka a pak je hodnota. Tento soubor je povinný, jinak se téma nezobrazí v programu v seznamu témat.

Takový soubor pak může vypadat třeba takto:

```
name: A theme
version: 1.0
author: Author
description: A theme that works!
```

### 11.7.5 settings.txt

Volitelný soubor obsahující doplňující proměnné, které lze následně použít v šablonách pro uživatelem měnitelná data. Proměnné uvedené v tomto souboru se zobrazí i v programu v okně Generování galerie a uživatel jim bude moci zadat nějaké hodnoty. Každá proměnná má svůj název, pod kterým ji pak lze vložit do šablon, typ (string, int{min, max}, float{min, max}, select{item0:title0, ...}, color), nadpis (zobrazuje se v programu) a výchozí hodnotu. Takový soubor pak může vypadat třeba takto:

```
items_per_row|int{1,10}|Items per row|3  
content_width|string|Content width|1000px
```

### 11.7.6 Proces generování

Program nejdříve vezme nastavení zadané uživatelem a s ním pak pokračuje jako s proměnnými při generování galerie.

Program začíná generováním kořenové složky. Pro každou složku, ať je kořenová, nebo ne, je stejný postup. Nejprve se vytvoří složka fyzicky na disku, pak se vezmou všechny podřazené položky složky a rozdělí se do stránek (v úvahu se bere nastavení počtu položek na stránku). Tím se vytvoří stránkování a jednotlivé stránky složky, stránkování však nemusí existovat kvůli malému počtu položek a výsledkem tak bude jen jediný soubor. Pro každý soubor složky se nejdříve vygeneruje cesta složky a navigace (i když jsou stejná pro každou stránku složky, některé proměnné jsou jiné), poté záhlaví a zápatí a nakonec se vše pomocí základního souboru (`_base.html`) vše spojí dohromady. Soubor s obsahem stránky se uloží na disk. Pokud má složka náhledový obrázek, vygeneruje se z původního souboru a uloží se do složky náhledových obrázků. Takto se pokračuje rekurzivně pro všechny podřazené složky.

Pro každý soubor ve složce je pak obdobný postup, jen s tou výjimkou, že se používá jiná šablona, tj. namísto `index.html` nebo `directory.html` se použije `single.html`. Mimo náhledového obrázku se ještě zkopíruje původní obrázek do složky obrázků.

# Kapitola 12

## Testování programu

Testování programu probíhalo po celou dobu vývoje programu. Při implementaci se program průběžně testoval na vybraných operačních systémech, konkrétně na Ubuntu 20.04.2, Debian 10.9.0 a Windows 10. Byla nejenom testována kompilace zdrojových souborů a zdali lze program úspěšně sestavit, ale také použitelnost programu na daných operačních systémech. Program se vždy nejdříve muselo podařit sestavit a při úspěchu sestavení se následně zkusilo projít programem, tj. například importovat obrázky a upravit je apod.

Další testování probíhalo na konci vývoje programu a to pomocí uživatelského testování. Program byl předán třem lidem s různou zkušeností s prací na počítači a různou zkušeností s podobnými programy. Všichni z nich však používají primárně operační systém Windows. Uživatelé dostali za úkol si program projít a pokusit se vytvořit si webovou galerii z vlastně vybraných obrázků. Poté jsem od nich získal zpětnou vazbu, kterou jsem zpracoval a navrhl úpravy a ty pak následně opět implementoval do programu.

### 12.1 Zpětná vazba

Zpětná vazba od uživatelů byla jak pozitivní v některých ohledech, tak negativní v jiných.

- Rychlý report vybraných obrázků s přesnou kvalitou úpravy obrázku.

- Chybějící možnost vybírat stránky do navigace.
- Nefunkční přejmenování obrázku na název bez koncovky.
- Při otevřeném okně informací o obrázku se po výběru jiné položky neobnoví obsah okna.
- Dostatečné množství základních funkcí pro úpravy obrázků.
- V editoru obrázku zůstává pomocná grafika obdélníkového oříznutí a odstranění červených očí.

Uživatelé byli obecně s programem spokojeni a dokázali se v něm bez větších problémů orientovat.

## 12.2 Navržené úpravy

- *Chybějící možnost vybírat stránky do navigace.*  
Přidání nového textového pole pouze pro čtení do okna informací o obrázku. Toto pole obsahuje výslednou URL v galerii a uživatel ji tak může jednoduše zkopírovat a vložit do pole pro URL u nové položky v navigačním menu.
- *Nefunkční přejmenování obrázku na název bez koncovky.*  
Během generování galerie je nutné zkontrolovat, zdali je nový název souboru bez koncovky. Pokud ano, lze ji získat zpět z cesty daného souboru.
- *Při otevřeném okně informací o obrázku se po výběru jiné položky neobnoví obsah okna.*  
Pokud je okno informací o obrázku již otevřeno a uživatel překlikne na jiný obrázek, musí se nastavit i vybraná položka v tomto okně na nově zvolenou.
- *V editoru obrázku zůstává pomocná grafika obdélníkového oříznutí a odstranění červených očí.*  
Při zavření editoru obrázků je nutné obnovit jeho stav na výchozí.

## ■ 12.3 Opětovné testování

Opětovné testování probíhalo obdobně. Zpětná vazba byla nyní pozitivní, jelikož se vyřešili všechny zmíněné problémy.





## Kapitola 13

### Závěr

Tato práce měla za úkol analyzovat již existující programy pro správu obrázků s možností exportu do webové galerie, navrhnout takovýto program a poté ho i následně implementovat.

Postupně jsem analyzoval programy se stejnými či podobnými funkcemi, inspiroval jsem se jimi a po částech jsem utvářel vizi, jak by program mohl vypadat. Každý z analyzovaných programů implementoval některé funkce trochu jinak, a tudíž bylo potřeba vybírat ty více uživatelsky přívětivé. V některých případech jsou funkce řešeny velmi podobně jako u analyzovaných programů a v jiných případech jsou naopak navrženy úplně od začátku.

Dále jsem se také věnoval analýze C++ GUI knihoven, kde jsem si u každé z knihoven prošel krátce její dokumentaci, vyzkoušel ji a případně zkusil zkompilovat a spustit předpřipravené demo programy, pokud byly k dispozici.

U návrhu implementace jsem se věnoval i menším detailům v programu a pokryl jsem všechny požadované funkce. V samotné implementaci jsem pak většinu navržených funkcí implementoval, nicméně některé byly pozměněny či rozšířeny, podle toho, jak bylo potřeba. Program ve výsledku obsahuje všechny potřebné a požadované funkce.

## 13.1 Rozšíření práce

Lze dále pokračovat v analýze podobných programů a získávat více informací o tom, jak jsou některé další funkce řešené a zdali je třeba i implementovat do tohoto programu. Daly by se také dále analyzovat C++ GUI knihovny a porovnávat jejich výhody a nevýhody, kde by se ke konci mohla vybrat doopravdy ta nejvhodnější knihovna pro tento projekt. V této fázi by byl ale přechod na jinou GUI knihovnu časově náročnější.

Program by bylo vhodné přeložit i do jiných jazyků, aby ho mohlo využívat více lidí. Lze také řešit různé úpravy a uhlazování grafického rozhraní či také program rozšířit o automatickou kontrolu verzí a následnou automatickou aktualizaci. Tímto vším by se zvedla kvalita a použitelnost tohoto programu.



# Příloha A

## Literatura

- [BLU] *Rozmazání*, "[https://en.wikipedia.org/wiki/Box\\_blur](https://en.wikipedia.org/wiki/Box_blur)".
- [BRI] *Úprava jasů*, "<https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-4-brightness-adjustment/>".
- [CON] *Úprava kontrastu*, "<https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/>".
- [EXIa] *Exif formát*, "<https://www.media.mit.edu/pia/Research/deepview/exif.html>".
- [EXIb] *Informace o exif*, "<https://en.wikipedia.org/wiki/Exif#Technical>".
- [GRA] *Odstíny šedi*, "<https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-3-greyscale-conversion/>".
- [HTM] *Hypertext markup language (html)*, "<https://en.wikipedia.org/wiki/HTML>".
- [INV] *Inverzní barvy*, "<https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-7-colour-inversion-solarisation/>".
- [JPE] *Formát jpeg*, "<https://en.wikipedia.org/wiki/JPEG>".
- [PNGa] *Formát png*, "<http://www.libpng.org/pub/png/spec/1.2/PNG-Contents.html>".

- [PNGb] *Struktura png*, "<https://ftp-osl.osuosl.org/pub/libpng/documents/png-1.2-pdg.html#Structure>".
- [RED] *Zostření*, "<https://www.learnopencv.com/automatic-red-eye-remover-using-opencv-cpp-python/>".
- [SHA] *Zostření*, "<https://web.stanford.edu/class/cs448f/lectures/2.1/Sharpening.pdf>, strana 8".
- [SPEa] *Specifikace formátu png*, "<https://ftp-osl.osuosl.org/pub/libpng/documents/pngext-1.5.0.html#C.eXIf>".
- [SPEb] *Specifikace formátu tiff*, "<https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf>".
- [SPEc] *Specifikace tiff exif*, "<https://www.awaresystems.be/imaging/tiff/tifftags/exififd.html>".
- [TIF] *Formát tiff*, "<https://en.wikipedia.org/wiki/TIFF>".