

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Aplikace pro tvorbu časosběrných filmů



2021

Vedoucí práce: Mgr. Petr Krajča,  
Ph.D.

Tomáš Ďurďa

Studijní obor: Informatika pro vzdělá-  
vání, prezenční forma

## **Bibliografické údaje**

Autor: Tomáš Ďurda  
Název práce: Aplikace pro tvorbu časosběrných filmů  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2021  
Studijní obor: Informatika pro vzdělávání, prezenční forma  
Vedoucí práce: Mgr. Petr Krajča, Ph.D.  
Počet stran: 37  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Tomáš Ďurda  
Title: Application for creating timelapse movies  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2021  
Study field: Computer Science for Education, full-time form  
Supervisor: Mgr. Petr Krajča, Ph.D.  
Page count: 37  
Supplements: 1 CD/DVD  
Thesis language: Czech

## Anotace

*Tato práce se zabývá problematikou vytváření časosběrných filmů. Popisuje proces tvorby aplikace pro Linux, která umožní běžnému uživateli rychlou a snadnou tvorbu časosběrných filmů. Aplikace vytvoří videozáznam ze série obrázků. Mezi další funkce patří přidání obrazových filtrů, textu, zvukové stopy. Umožňuje také korekci posunutí snímků. Byla vytvořena v jazyce C++ s využitím knihoven Qt, FFmpeg a OpenCV.*

## Synopsis

*This thesis deals with the problematics of making timelapse movies. It describes a proces of making a Linux application which allows quick and easy creation of timelapse movies. Application creates a video file from a collection of images. Another features cover adding image filters, text or sound tracks. It allows correction of the shifted frames as well. The application was written in C++ language using Qt, FFmpeg and OpenCV libraries.*

**Klíčová slova:** linux, časosběrné filmy, stabilizace obrazu, obrazové filtry

**Keywords:** linux, timelapse movies, image stabilization, image filters

Děkuji Mgr. Petru Krajčovi, Ph.D. za cenné rady a připomínky při vedení této práce.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora



# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>   | <b>8</b>  |
| <b>2</b> | <b>Základní pojmy</b>   | <b>9</b>  |
| <b>3</b> | <b>Požadavky na vypracování</b>   | <b>11</b> |
| <b>4</b> | <b>Aplikace s podobným zaměřením</b>  | <b>12</b> |
| 4.1      | OpenShot . . . . .  | 12        |
| 4.2      | Kdenlive . . . . .  | 13        |
| 4.3      | Stopmotion . . . . .  | 13        |
| <b>5</b> | <b>Programátorská část</b>  | <b>14</b> |
| 5.1      | Použité technologie . . . . .   | 14        |
| 5.1.1    | Jazyk C++ . . . . .   | 14        |
| 5.1.2    | Qt FrameWork . . . . .  | 15        |
| 5.1.3    | FFmpeg . . . . .  | 15        |
| 5.1.4    | OpenCV . . . . .  | 16        |
| 5.1.5    | EasyEXIF . . . . .  | 16        |
| 5.2      | Popis aplikace . . . . .  | 17        |
| 5.2.1    | Třída Project . . . . .   | 17        |
| 5.2.2    | Třída TimelineElement . . . . .   | 19        |
| 5.2.3    | Třída VisualElement . . . . .   | 20        |
| 5.2.4    | Třída Image . . . . .   | 20        |
| 5.2.5    | Třída Text . . . . .  | 20        |
| 5.2.6    | Třída Filtr . . . . .   | 21        |
| 5.2.7    | Třída HSl . . . . .   | 21        |
| 5.2.8    | Třída SoundElement . . . . .  | 21        |
| 5.2.9    | Třída Sound . . . . .   | 21        |
| 5.2.10   | Třídy ImageInspector, TextInspector, FilterInspector a SoundInspector . . . . . | 22        |
| 5.2.11   | Třída Preview . . . . .   | 22        |
| 5.2.12   | Třída ExportDialog . . . . .  | 23        |
| 5.2.13   | Třída Muxing . . . . .  | 23        |
| 5.2.14   | Třída Settings . . . . .  | 25        |
| 5.2.15   | Třída MainWindow . . . . .  | 25        |
| 5.2.16   | Třída TimelineTicks . . . . .   | 26        |
| 5.2.17   | Třída Factory . . . . .   | 26        |
| <b>6</b> | <b>Uživatelská příručka</b>   | <b>27</b> |
| 6.1      | Návod ke spuštění . . . . .   | 27        |
| 6.2      | Uživatelské rozhraní . . . . .  | 27        |
| 6.2.1    | Horní panel . . . . .   | 27        |
| 6.2.2    | Inspektor . . . . .   | 29        |

|          |                                 |           |
|----------|---------------------------------|-----------|
| 6.2.3    | Náhled . . . . .                | 29        |
| 6.2.4    | Časová osa . . . . .            | 29        |
| 6.2.5    | Progress bar . . . . .          | 30        |
| <b>7</b> | <b>Analýza řešení</b>           | <b>31</b> |
|          | <b>Závěr</b>                    | <b>32</b> |
|          | <b>Conclusions</b>              | <b>33</b> |
| <b>A</b> | <b>Obsah přiloženého CD/DVD</b> | <b>34</b> |
|          | <b>Literatura</b>               | <b>35</b> |

## Seznam obrázků

|   |   |    |
|---|---|----|
| 1 | Hlavní okno aplikace OpenShot . . . . .   | 12 |
| 2 | Hlavní okno aplikace Kdenlive . . . . .   | 13 |
| 3 | Hlavní okno aplikace StopMotion . . . . . | 14 |
| 4 | UML diagram aplikace . . . . .            | 17 |
| 5 | Grafické rozhraní aplikace . . . . .      | 28 |

# 1 Úvod

Časoběrný film (též časoběrná fotografie, anglicky time-lapse) lze definovat jako sekvenci snímků, která je přehrávána s podstatně vyšší snímkovou frekvencí, než při jaké byla natočena. Jeho účelem je „rozběhnout“ pomalé pohyby tak, aby byly zřejmé pro lidské vnímání.[1] Jednotlivé snímky jsou snímány např. v rozmezí 1 minuty. Při době snímání 10 hodin získáme 600 snímků záznamu. Při převedení do standardní snímkové frekvence (tj. 25 snímků za sekundu) získáme 24 sekund dlouhé časoběrné video. Takto můžeme znázornit např. pohyb mraků na obloze, kvetoucí tulipány nebo stavbu rodinného domu. Nástup moderních technologií nám umožňuje výrazně zjednodušit proces tvorby takovýchto filmů. Tím se dostáváme k účelu této bakalářské práce.

## 2 Základní pojmy

V této kapitole si definujeme si definujeme pojmy, které jsou dále v textu používány.

- **Kodek** je specifikace, která popisuje způsob reprezentování informace. Kodek může být ztrátový nebo bezztrátový. Bezeztrátové kodeky umožňují přesně reprodukovat originální obsah, ztrátové kodeky pouze přibližně. [22]
- **Kódování** videa je proces, kdy je série snímků zkomprimována do souboru menší velikosti. Kódování probíhá pomocí operací specifikovaných v kodeku. [23]
- **Dekódování** videa je proces přeměny zkomprimovaných dat zpět do podoby, ve které mohou být zobrazena. [24]
- **Transkódování** videa je proces dekodování z jednoho kodeku a následné zakódování do kodeku jiného. [23]
- **Datový stream** neboli proud je sekvence digitálních signálů, které jsou přenášeny mezi zdrojem a cílem. [25]
- **Streamování** je proces přenášení dat od poskytovatele ke koncovému uživateli. [26]
- **Multimediální kontejner** je formát k uložení několika proudů multimediálních dat do jednoho souboru. [27]
- **Muxing** je proces používaný ke kombinování více signálů do jednoho, který je přenášen po sdíleném médiu. Můžeme vzít například audio a video proud a zkombinovat je do kontejneru MP4. [28]
- **Demuxing** je proces extrahování individuálních signálů z jednoho proudu. Jedná se o opak muxingu. [29]
- **Převzorkování** znamená změnu vzorkovací frekvence zvuku. [30]
- **EXIF** je standard, který definuje informace týkající se obrázků nebo jiných médií pořízených digitální kamerou. Může se jednat o datum, čas, nebo GPS lokaci. [32]
- **OOP** neboli objektově orientované programování je programovací paradigma založené na konceptu tříd a objektů. Třídy reprezentují kategorie, které sdílí určité atributy. Třídy obsahují speciální funkce zvané metody, které se vztahují pouze k objektům dané třídy. Objekty se nazývají instance tříd. Mezi základní vlastnosti OOP patří:
  - **Abstrakce** znamená použití jednoduchých nástrojů pro přístup ke složitému objektu. Např. řidič nepotřebuje znát všechny detaily fungování motoru k tomu, aby mohl řídit auto.

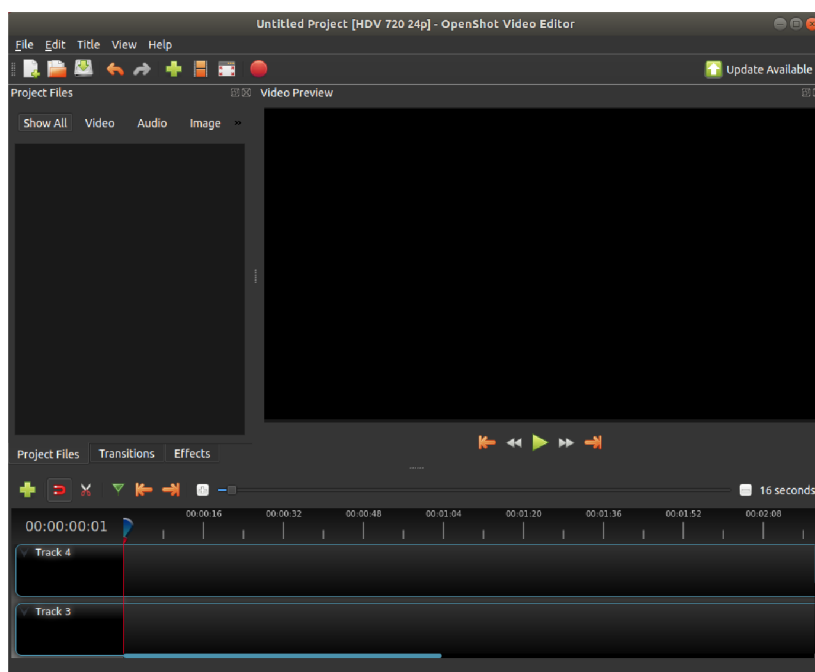
- **Dědičnost** znamená, že zděděné třídy přebírají atributy metody od nadřazených tříd.
  - **Zapouzdření** znamená, že hodnoty atributů objektu smí číst a měnit pouze metody daného objektu.
  - **Polymorfismus** znamená, že více metod může provádět stejný úkol. Zděděné objekty mohou nahradit chování rodičů vlastním, specifickým chováním. [31]
- **Návrhový vzor** je generální, vícenásobně použitelné řešení na běžně se vyskytující problém při vývoji software. [34]
  - **Factory** je návrhový vzor, který používáme např. pokud chceme vytvořit objekt pomocí parametrů, které nejsou známy před spuštěním. aplikace. [34]
  - **Framework** je platforma pro vyvíjení softwarových aplikací. [33]
  - **UML** je zkratka pro Unified Modeling Language. Jeho účelem je vytváření modelů pro objektově orientovaný software. Mezi třídami může existovat několik typů vztahů. Jedná se o:
    - **Dědičnost** - je znázorněna šipkou s trojúhelníkovým hrotem. Šipka směřuje směrem od potomka k rodičovské třídě.
    - **Agregace** - znamená, že třída, která je agregát, je určitým způsobem celek, zatímco jiná třída je součástí tohoto celku. Agregace je znázorněna diamantovou šipkou.
    - **Asociace** - je vztah, kdy třída obsahuje odkaz na jinou třídu. Na rozdíl od agregace zde však není vztah celku a součásti. Asociace je znázorněna obyčejnou šipkou.
    - **Závislost** - je případ, kdy je vztah mezi třídami velmi slabý. Tento vztah není implementován pomocí atributů, ale spíše pomocí argumentů v metodách. Závislost je znázorněna přerušovanou šipkou. [35]

### 3 Požadavky na vypracování

Primárním cílem této bakalářské práce je vypracovat grafickou aplikaci, která vytvoří časosběrný film ze série obrázků. Dalšími požadavky jsou aplikace obrazových filtrů, stříh, korekce posunutí, přidání textu a zvukové stopy. Uživatel od aplikace jistě očekává přehledné zobrazení všech součástí výsledného záznamu a možnost s nimi kdykoliv manipulovat. Předpokládá se tedy, že aplikace bude obsahovat časovou osu, pod kterou se budou zobrazovat jednotlivé překrývající se stopy (obrázky, text, zvuk a filtry). Každá stopa by měla mít možnost posunu, stříhu a odstranění, případně nastavení dalších specifických parametrů. Předpokládá se, že časová osa může obsahovat velmi dlouhé stopy, měla by mít tedy možnost přiblížení a oddálení.

Uživatel určitě očekává i náhled videa. Aplikace tedy bude mít funkci přehrávání a zastavení náhledu. Nad časovou osou bude posuvník signalizující, ve kterém místě se aktuální náhled nachází. Náhled by měl být pokud možno v reálném čase.

Aplikace by měla podporovat alespoň základní typy obrázků - .jpg, .png a .bmp. a zvukových souborů - .wav, .mp3 a .flac. Předpokládá se možnost ukládání výsledného videozáznamu do všeobecně podporovaných formátů, např. .mp4 nebo .avi. Mezi základní filtry, které by aplikace měla obsahovat, patří úprava jasu, kontrastu, sytosti a odstínu. Korekce posunutí by měla být automatická, tzn. že aplikace bude schopna nalézt snímky, kde došlo k posunu kamery a následně by měla provést adekvátní posunutí všech následujících snímků. Do textové stopy musí mít možné vkládat regulární výrazy, které umožní zobrazit informace o snímcích. Důležitým požadavkem je možnost nastavení snímkové frekvence podle délky videa, tedy aby záznam trval přesně stanovenou dobu.



Obrázek 1: Hlavní okno aplikace OpenShot

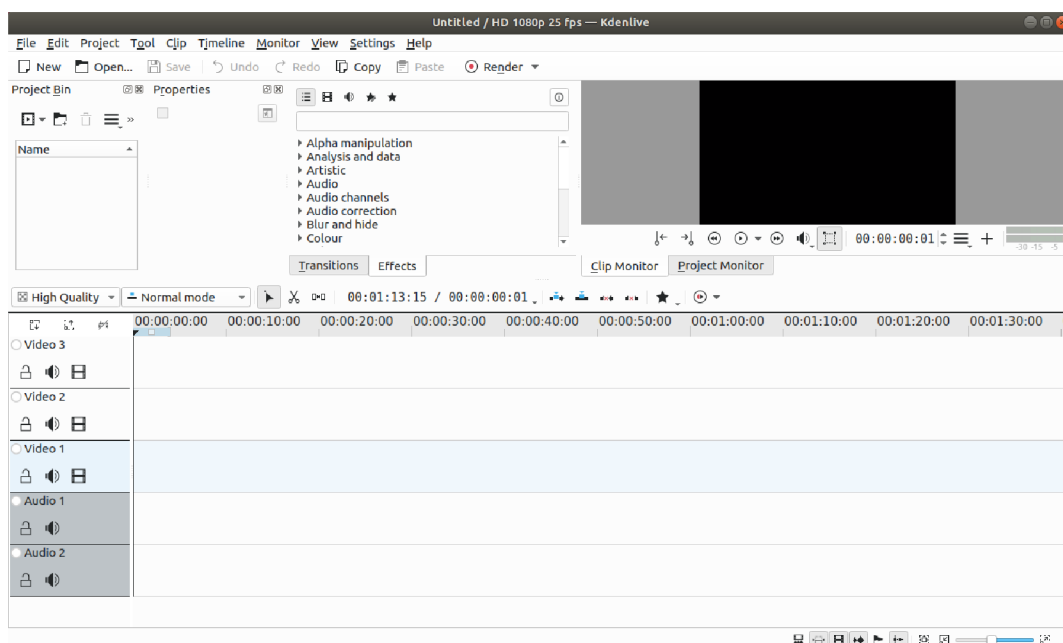
## 4 Aplikace s podobným zaměřením

V současné době na platformě Linux neexistuje žádná aplikace specializovaná přímo na tvorbu časosběrných videí. Uživatelé k tomuto účelu mohou použít klasické video editory. Mezi takovéto video editory patří např. Kdenlive nebo OpenShot. V této sekci stručně představíme tyto editory a porovnáme jejich výhody a nevýhody těchto aplikací při tvorbě časosběrných videí.

### 4.1 OpenShot

Na obrázku č. 1 vidíme hlavní okno aplikace OpenShot. OpenShot vytvořil vývojář Jonathan Thomas v roce 2008. Od roku 2012 projekt spadá pod společnost OpenShot Studios, LLC. Program je napsán v několika programovacích jazycích. Rozhraní je napsáno v Pythonu a PyQt5, zpracování videa je napsáno v C++ za pomoci knihovny FFmpeg. Interaktivní časová osa je sestavena pomocí jazyků HTML5, Javascript a AngularJS. Software je distribuován pod licencí GNU General Public Licence[2]. Na obrázku č. 1 vidíme hlavní okno aplikace. Výhodou je jednoduché ovládání. Aplikace je vhodná pro uživatele bez předchozích zkušeností s editací videa. Nevýhodou je časté padání aplikace z různých důvodů. Přidávání snímků do projektu je poněkud neintuitivní. Je nutné vybrat pouze jeden snímek a až poté se zobrazí dialog pro nahrání sekvence. Při vybrání více snímků najednou se tyto snímky nahrají jako samostatné obrázky.





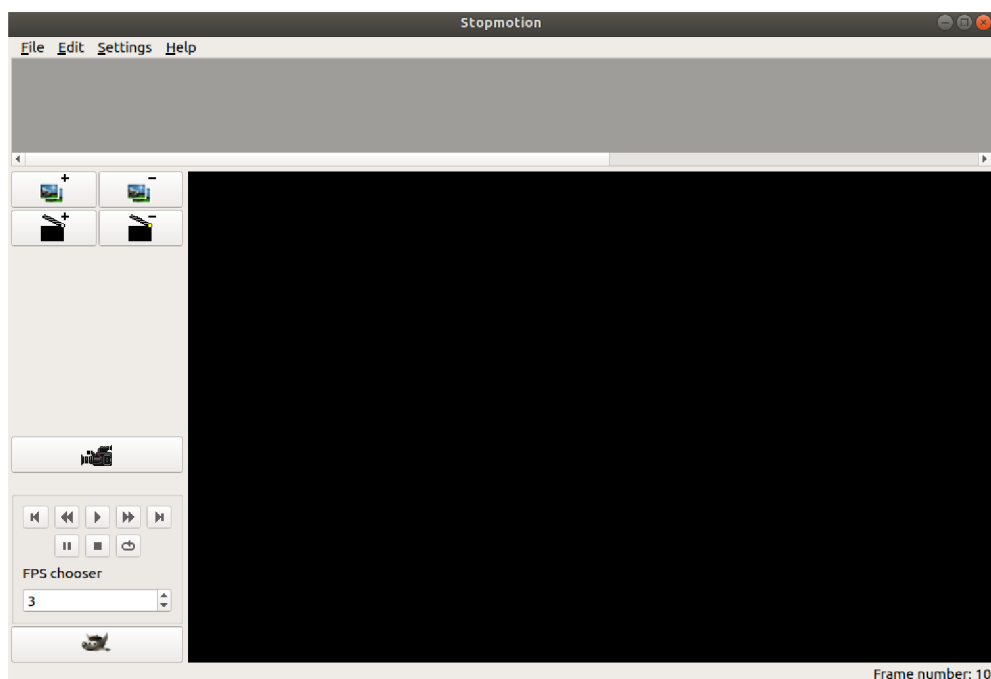
Obrázek 2: Hlavní okno aplikace Kdenlive

## 4.2 Kdenlive

Na obrázku č.2 vidíme hlavní okno aplikace Kdenlive. Kdenlive je akronym pro KDE non-linear video editor. Je distribuován pod licencí GNU General Public Licence. Projekt započal v roce 2003. Aplikace je naprogramována s pomocí knihovny Qt a frameworku KDE. Zpracování videa obstarává převážně MLT Framework. [4] Výhodou je početná komunita aktivních uživatelů a široké spektrum funkcí, které ocení především profesionální editoři. To může být naopak nevýhodou pro začátečníky, kteří se v obsáhlém GUI mohou velmi obtížně orientovat. Přidání sekvence snímků lze provést z horního menu Project -> Add SlideShow. Dialogové okno poté nabízí volbu složky se snímky. Je možné nastavit délku snímku v řádu desítek milisekund, avšak není možné nastavit délku celé sekvence.

## 4.3 Stopmotion

Na obrázku č.3 vidíme hlavní okno aplikace StopMotion. Stopmotion je jednoduchý nástroj pro tvorbu fázových animací. [5] Jedná se o původně studentský projekt vytvořený Hermanem Robakem. Dá se použít i na tvorbu časosběrných videí. Přidání sekvence je velmi jednoduché, provede se výběrem příslušných snímků v prohlížeči souborů. Aplikace sama o sobě nepodporuje přidávání filtrů, ale jednotlivé snímky je možné otevřít v grafickém editoru Gimp a tam je upravit. Aplikace podporuje pouze statickou hodnotu snímků za sekundu.



Obrázek 3: Hlavní okno aplikace StopMotion

## 5 Programátorská část

### 5.1 Použité technologie

Aplikace byla naprogramována v jazyce C++ s využitím frameworku Qt. Pro dekodování videí byla použita knihovna ffmpeg. Knihovna OpenCV byla použita pro detekci posunutí snímků. Knihovna easyEXIF byla použita pro extrakci metadat z obrazových souborů. Následující podkapitoly obsahují stručný popis těchto technologií.

#### 5.1.1 Jazyk C++

C++ je programovací jazyk, který vytvořil Bjarne Stroustrup a jeho tým v Bellových laboratořích. Jak jeho název napovídá, byl odvozen z jazyka C a obsahuje všechny jeho vlastnosti. Díky tomu je možné použít existující kód z jazyka C v C++ programech. Programování v C++ je efektivní a blízké ke strojovému programování. Umožňuje vytvářet přenosné programy pro různé platformy. Jazyk podporuje koncepty objektově orientované programování, mezi které patří abstrakce dat, zapouzdření, dědičnost a polymorfismus. Jazyk podporuje také reference, šablony a zacházení s výjimkami. Jedná se o kompilovaný jazyk. Zdrojové kódy obvykle končí příponou .cpp nebo .cc. Hlavičkové soubory obsahují informace používané ve zdrojových kódech, jako například definice proměnných a funkcí. Můžou mít příponu .h, a nebo mohou být bez přípony. [6]

### 5.1.2 Qt Framework

Qt je multiplatformní framework určený pro tvorbu softwaru. Je vyvíjen společnostmi Qt company a Qt Project Community. Mezi podporované platformy patří především Windows, Linux, macOS a Android. Podporované jazyky jsou C++, Python a Qt QML [7, 8]. Je dostupný v komerční nebo open-source variantě. Pokud si vývojář zvolí open-source variantu, může používat Qt pod licencí LGPL 3, GPL 2 nebo GPL 3. [9] Framework je možné nainstalovat pomocí online nebo offline instalátoru, nebo je možné sestavit zdrojové balíky. Instalátory obsahují Qt knihovny, dále vývojové prostředí Qt Creator, dokumentaci, ukázky, zdrojové kódy a doplňky specifické pro jednotlivé platformy. [11]

Základem API je modul Qt Core, jádro, které obsahuje negrafické třídy používané jinými moduly. Modul Qt GUI obsahuje základní třídy pro grafické komponenty. Framework nabízí více technologií pro vytváření uživatelského rozhraní. Prvním z nich je modul Qt Widgets, který nabízí tvorbu tradičních statických uživatelských rozhraní, která dokáží přizpůsobit svůj vzhled dle dané platformy. Modul Qt Quick poskytuje QML typy pro tvorbu více dynamických rozhraní s mnoha animacemi a efekty. Mezi další moduly patří např. Qt Multimedia, který umožňuje práci s audiem, videem, kamerou a rádiem, Qt Network, který usnadňuje audio video síťové programování, nebo Qt SQL, umožňující integraci databází. [10]

Ke komunikaci mezi objekty se používá mechanismus signálů a slotů. Signál je vyslán, když u objektu dojde k určité události. Slot je funkce, která je volána jako reakce na přijatý signál. Objekty v Qt mají mnoho předdefinovaných signálů a slotů, ale pokud vytvoříme vlastní zděděné třídy, můžeme implementovat vlastní signály a sloty. Je možné k jednomu slotu připojit libovolně mnoho signálů a jeden signál může být spojen s libovolným množstvím slotů. Mechanismus signálů a slotů je typově bezpečný, protože signatura signálu se musí shodovat se signaturou připojeného slotu. [13] Chceme-li využít tento mechanismus, musíme dědit ze třídy `QObject`.

Součástí frameworku je Meta-Object-Compiler. Tento nástroj čte hlavičky C++ souborů a vytváří zdrojové soubory obsahující meta-objektový kód pro tyto třídy. [12]

Nástroj QMake zjednodušuje proces sestavení na různých platformách. Automatizuje proces generování Makefile souborů, proto k vytvoření Makefile stačí pouze pár řádků informací. [14]

### 5.1.3 FFmpeg

FFmpeg je multimediální framework, schopný kódovat, dekodovat, transkodovat, muxovat, demuxovat, streamovat, filtrovat a přehrávat. Podporuje široké spektrum formátů, od nejzastaralejších až po nejmodernější. Jeho výhodou je vysoká přenosnost. Lze ho zkompilovat a spustit na systémech Linux, Mac OS, Microsoft Windows, Solaris nebo BSD. Vysokou prioritou frameworku je bezpečnost. [15] Je provozován pod licencí LGPL verze 2.1 nebo novější. Některé volitelné sou-

části jsou pokryty licencí GPL 2.[16] Framework obsahuje nástroj ffmpeg, který se používá v příkazovém řádku pro konverzi formátů multimediálních souborů. Nástroj ffmpeg je jednoduchý přehrávač médií založený na SDL. Nástroj ffprobe slouží k analýze multimediálních streamů. Framework nabízí také knihovny. Libavcodec obsahuje kodéry a dekodéry pro audio a video kodeky.

- **Libavutil** obsahuje funkce pro zjednodušení programování, např. generátory náhodných čísel, datové struktury apod.
- **Libavformat** obsahuje muxery a demuxery pro multimediální kontejnery.
- **Libavdevice** obsahuje vstupní a výstupní zařízení. Libavfilter obsahuje multimediální filtry.
- **Libswscale** je knihovna provádějící vysoce optimalizované škálování a konverze mezi pixelovými a barevnými formáty.
- **Libswresample** umožňuje optimalizované převzorkování audia.[15]

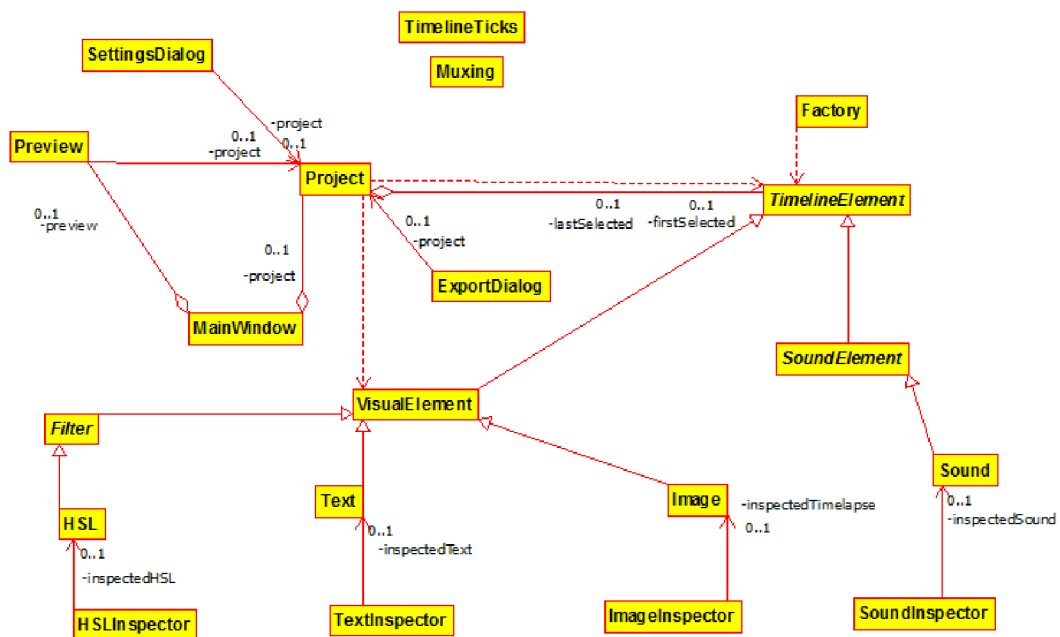
#### 5.1.4 OpenCV

OpenCV je knihovna pro počítačové vidění, napsaná v jazycích C a C++. Byla navržena s ohledem na výpočetní výkon a real-time aplikace. Jedním z hlavních cílů je poskytnout uživateli jednoduše použitelné rozhraní pro tvorbu sofistikovaných aplikací. Knihovna obsahuje funkce zaměřené např. na inspekci produktů, kalibraci kamery nebo robotiku. Knihovna je vydána pod open-source licencí. Tato licence umožňuje vytváření komerčních aplikací bez nutnosti zveřejňovat zdrojový kód. Struktura knihovny je rozdělena do pěti hlavních komponent:

- Komponenta **CV** obsahuje základní zpracování obrázků.
- **ML** je knihovna pro strojové učení.
- **HighGUI** obsahuje funkce pro ukládání a načítání obrázků a videí.
- **CXCore** obsahuje základní datové struktury.
- **CvAux** obsahuje experimentální algoritmy[17].

#### 5.1.5 EasyEXIF

TinyEXIF je jednoduchá C++ knihovna pro parsování metadat uvnitř JPEG obrázků. Její výhodou je její nezávislost na knihovnách třetích stran. Pro její použití stačí zkopírovat do cílového projektu pouze dva zdrojové soubory. Je distribuována pod licencí BSD. [18]



Obrázek 4: UML diagram aplikace

## 5.2 Popis aplikace

Na obrázku č. 4 vidíme UML diagram aplikace. Na počátku hierarchie stojí třída `MainVnindow`. Třídy `Preview` a `Project` jsou součástí `MainVnindow`, je zde vztah agregace. Třídy `Preview`, `SettingsDialog` a `ExportDialog` obsahují reference na třídu `Project`, jedná se o vztah asociace. Třída `Project` obsahuje metody s parametry, kterými jsou objekty tříd `TimelineElement` a `VisualElement`, je tedy na těchto třídách závislá. Navíc také agreguje třídu `TimelineElement`. `TimelineElement` je virtuální třída. Dědí třídy `VisualElement` a `SoundElement`, které jsou rovněž virtuální. Třída `VisualElement` dědí třídy `Filter`, `Text` a `Image`. `Filter` je virtuální třída, která dědí třídu `HSL`. Třídy `HSLInspector`, `TextInspector` a `ImageInspector` jsou asociovány s příslušnými třídami `HSL`, `Text` a `Image`. Třída `SoundElement` dědí třídu `Sound` a ta je asociována s třídou `SoundInspector`. Třída `Factory` je závislá na třídě `TimelineElement`. Třídy `TimelineTicks` a `Muxing` můžeme považovat za samostatné.

V následujícím kapitolách popíšeme vlastnosti a funkce těchto tříd.

### 5.2.1 Třída `Project`

Třída reprezentující projekt časoběrného filmu. Instance třídy jsou vytvářeny při startu aplikace, při vytvoření nového projektu, nebo otevření uloženého projektu.

Třída obsahuje následující atributy:

- **position** - aktuální pozice jezdce na časové ose v milisekundách.
- **elements** - seznam všech stop nacházejících se na časové ose.

- **resolution** - velikost video rámu v pixelech. Výchozí hodnota je 1280x720.
- **fps** - počet snímků za sekundu. Jedná se o desetinné číslo. Výchozí hodnota je 25.
- **motionCorrection** - pravdivostní hodnota určující povolení nebo zakázání automatické korekce posunutí.
- **maxMotionCorrection** - celé číslo určující maximální míru korekce posunutí.
- **samplerate** - vzorkovací frekvence zvuku. Výchozí hodnota je 44100.
- **firstSelected** a **lastSelected** - první a poslední vybraný prvek na časové ose při vícenásobném výběru.

Třída obsahuje tyto metody:

- **getVisualElementsAtFrame()** - jako argument bere číslo snímku a vrátí všechny vizuální komponenty, které na časové ose leží na stejné pozici jako tento snímek.
- **getSoundElementsAtSample()** - jako argument bere číslo zvukového vzorku a vrátí všechny audio stopy, které na časové ose leží na stejné pozici jako tento vzorek.
- **getLastTimelapsePosition()** - vrátí celkovou délku projektu jako desetinné číslo.
- **getLastTimelapseFrame()** - vrátí poslední snímek na časové ose.
- **getLastTimelapseFrame()** - vrátí poslední zvukový vzorek.
- **getFirstTimelapse()** - vrátí první snímek v projektu.
- **placeIsOccupied()** - argumentem jsou číslo snímku a element (kandidát na pozici). Funkce zjistí, zdali daný kandidát může být přemístěn na daný snímek bez kolize (tzn. zda pozice není obsazená).
- **fitInSize()** - vrátí takové rozlišení projektu, které vměstná do obdélníku zadaného argumentem při zachování poměru stran.
- **moveImages()** - argumentem je první snímek a celé číslo určující počet snímků. Funkce posune všechny obrazové snímky počínaje prvním o zvolený počet snímků.
- **firstSelected()** - vrátí první vybraný prvek na časové ose.
- **lastSelected()** - poslední zvolený prvek na časové ose. Vybrány jsou všechny prvky mezi tímto prvkem a prvním prvkem.

- **updateSelection()** - obstarává multiselekcí, tedy aby všechny prvky mezi prvním a posledním byly vybrány.
- **deleteSelection()** - odstraní všechny vybrané prvky.
- **read()** - nastaví atributy projektu podle dat ze vstupu.
- **write()** - zapíše hodnoty atributů na výstup.

### 5.2.2 Třída `TimelineElement`

Třída `TimelineElement` je abstraktní třídou poskytující základ pro všechny stopy projektu. Dědí ze třídy `QGraphicsItem`. Třída obsahuje tyto vlastnosti:

- **startPosMsecs** - pozice v milisekundách, kde stopa začíná na časové ose.
- **track** - číslo stopy, představuje vertikální pozici komponenty.
- **height** - výška stopy v pixelech.

Třída obsahuje tyto metody:

- **isAtPos()** - virtuální metoda, jako argument bere celé číslo určující pozici. Může jít o pozici audio vozku nebo video snímku. Vrací pravdivostní hodnotu, pokud stopa leží na této pozici.
- **durationMsecs()** - virtuální metoda, vrací trvání komponenty v milisekundách jako desetinné číslo.
- **updateInspector()** - virtuální metoda. Nastaví, jaký typ inspektoru se bude zobrazovat u daného potomka.
- **clone()** - vytvoří duplikát objektu.
- Metody **read()** a **write()** fungují analogicky jako u třídy `Project`.

Třída přepisuje metody:

- **mousePressEvent()** - pokud uživatel klikne na komponentu levým tlačítkem, dojde k aktualizaci inspektoru. Objekt je vybrán.
- **mouseReleaseEvent()** - pokud uživatel klikne na komponentu pravým tlačítkem, zobrazí se kontextové menu.
- **paint()** - stará se o vykreslení základních tvarů objektu.
- **boundingRect()** - vrátí obdélník o stejném rozměru, jaký má tato stopa.

Třída obsahuje tyto sloty:

- **deleteTriggered()** - příslušná stopa se odstraní z projektu.
- **duplicateTriggered()** - vytvoří duplikát daného objektu.
- **fpsChanged()** - reaguje na změnu snímkové frekvence a zajišťuje překreslení všech snímků.

### 5.2.3 Třída VisualElement

Třída poskytuje základ pro všechny komponenty, které mají vizuální charakter. Třída přepisuje metodu `itemChange()`. Zajišťuje, aby se objekty daly posouvat pouze v jedné vertikální rovině (v rámci jedné stopy), a aby byl posuv možný pouze po snímcích (objekty se „přichytávají“ vždy k nejbližšímu snímku). Také zabraňuje vkládání objektů na pozici, která je obsazená. Důležitou vlastností je `startFrame`, která určuje pozici snímku, kde komponenta leží na časové ose. Třída definuje virtuální metodu `composeImage()`. Argumenty jsou snímek, jeho časová pozice a seznam metadat. Metoda modifikuje daný snímek a případně vloží nová metadata do seznamu.

### 5.2.4 Třída Image

Třída představující jeden snímek. Má atribut `image`, který obsahuje textový řetězec. V řetězci je uložena cesta k souboru s obrázkem. Atribut `mini` obsahuje zmenšeninu obrázku, která se používá mj. při přehrávání náhledu.

Třída přepisuje metody:

- **paint()** - při větším přiblížení vykreslí na stopu malý náhled obrázku a jeho jméno.
- **composeImage()** - vloží data z obrazového souboru do snímku, který dostává jako parametr. Stará se o čtení EXIF metadat ze souboru.

Třída obsahuje slot **decode()**. Slouží k načtení obrázku ze souboru a jeho zmenšení do miniatury používané v náhledu.

### 5.2.5 Třída Text

Třída je potomkem třídy `VisualElement`, představující textové pole. Má tyto atributy:

- **text** - textový řetězec, který se má zobrazit.
- **coords** - souřadnice v pixelech.
- **font** - instance objektu třídy `QFont`, specifikuje typ písma.
- **color** - instance objektu třídy `QColor`, specifikuje barvu písma.
- **frameCount** - doba trvání textového popisku ve snímcích.

Třída přepisuje metody:

- **paint()** - při větším přiblížení vykreslí na stopu textový řetězec.
- **composeImage()** - vloží text do snímku předaném v parametru. Při vkládání textu do aktuálního snímku má instance objektu k dispozici slovník metadat, které se vztahují k tomuto snímku. Obsahuje-li textový řetězec nějaký klíč ze slovníku, je tento klíč v řetězci nahrazen příslušnou hodnotou.



### 5.2.6 Třída `Filtr`

Třída vytvářející abstraktní základ pro komponenty, které filtrují obrazové snímky. Třída definuje atribut `frameCount`, což je doba trvání vyjádřená počtem snímků.

### 5.2.7 Třída `HSI`

Třída představující jednoduchý filtr. Obsahuje atributy `hue` (odstín), `saturation` (sytnost), `lightness(jas)`, `contrast` (kontrast) a `gamma`.

Aplikování filtru probíhá následovně: Z aktuálního snímku se extrahují barvy jednotlivých pixelů ve formátu RGB. Nejprve se pomocí daných hodnot kontrastu  $C$  vypočítá koeficient  $F$ .

$$F = 259 * (C + 255) / 255 * (259 - C)$$

Následně se vypočte nová hodnota každé barevné složky, např. hodnota červené se vypočte podle vzorce:

$$R' = F * (R - 128) + 128$$

Gamma korekce se provádí dle vzorce:

$$R' = 255 * (R / 255)^\gamma$$

Analogicky u zelené a modré složky [19, 20]. Při aplikaci odstínu je hodnota jasu jednoduše přičtena k jednotlivým složkám [21]. Při aplikaci odstínu a sytnosti jsou tyto atributy jednoduše přičteny k hodnotám barev převedených do modelu HSL. Hodnoty aplikovaných filtrů se vkládají do metadat příslušného snímku.

### 5.2.8 Třída `SoundElement`

Třída poskytuje základ pro všechny komponenty, které mají zvukový charakter. Důležitým argumentem je `startSample`, tedy číslo vzorku, kde komponenta začíná na časové ose. Třída definuje virtuální metodu `composeSound()`. Argumenty jsou audio vzorek a číslo vzorku. Metoda upraví hodnotu vzorku podle atributů dané komponenty.

Třída přepisuje metodu `itemChange()`. Zajišťuje, aby se objekty daly posouvat pouze v jedné vertikální rovině (v rámci jedné stopy). Také zabraňuje vkládání objektů na pozici, která je obsazená.

### 5.2.9 Třída `Sound`

Třída představující zvukovou nahrávku. Má tyto atributy:

- **audioData** - pole obsahující dekodované vzorky z audio souboru.
- **sampleCount** - celkový počet vzorků v dekodovaném souboru.
- **path** - cesta ke zvukovému souboru.

- **volume** - hlasitost.
- **crop** - pozice ořezání stopy v milisekundách.
- **duration** - délka zvukové stopy v milisekundách .
- **drop** - pozice začátku zvukové stopy v milisekundách.

Třída obsahuje slot `splitTriggered()` - pokud uživatel klikne na volbu `split` v kontextovém menu, dojde k rozdělení příslušné komponenty na dvě samostatné a `joinTriggered()`, který spojí dvě nebo více zvukových stop v jednu.

Zděděná metoda `composeSound()` vloží vzorek z proměnné `audioData` do vzorku předaného parametrem.

### 5.2.10 Třídy `ImageInspector`, `TextInspector`, `FilterInspector` a `SoundInspector`

Tyto třídy umožňují uživateli nastavit parametry jednotlivých komponent projektu. Instance třídy je vytvořena v okamžiku, kdy uživatel vybere objekt na časové ose. Tento objekt se stane inspektovaným objektem. Každá třída obsahuje ukazatel na inspektovaný objekt. Třídy obsahují grafické rozhraní pro úpravu parametrů, které je vloženo do hlavní aplikace. Obsahují také metody, které reagují na podněty uživatele a podle nich provádějí příslušné změny inspektovaných objektů, např. zvýšení nebo snížení hlasitosti apod.

### 5.2.11 Třída `Preview`

Třída `Preview` slouží k zobrazování snímků a přehrávání zvuku. Obsahuje tyto atributy:

- **playThread** - atribut typu `QThread`. Je to vlákno, ve kterém běží časovač zajišťující periodickou aktualizaci snímků.
- **size** - velikost náhledu v pixelech. Nastaveno na 640x320.
- **previewLabel** - plátno, na které se vykreslují snímky. Jde o instanci třídy `QLabel`.
- **device** - ukazatel na objekt třídy `QIODevice`. Jedná se o zařízení, na které se zapisují zvukové samplly.
- **audioOutput** - ukazatel na objekt třídy `QAudioOutput`. Tento objekt slouží k posílání zvukových dat do atributu `device`.
- **audioFormat** - objekt typu `QAudioFormat`. Jde o formát pro přehrávání audia. Obsahuje údaje o vzorkovací frekvenci, počtu kanálů, počtu bitů ve vzorku, pořadí bitů a typu vzorku. Tyto údaje jsou nastaveny v konstruktoru třídy `Preview`.

- **samplesPlayed** - celé číslo, vyjadřuje počet přehraných zvukových vzorků od spuštění přehrávače.

Třída obsahuje metody:

- **getOutput()** - aktualizuje snímek podle současné pozice na časové ose. Metoda nejdřív zjistí seznam všech vizuálních stop, které se nacházejí na současném snímku a vytvoří z nich nový obrázek, který se vykreslí na `previewLabel`.
- **getAudioOutput()** - jako parametry bere audio buffer, počet vzorků a číslo prvního vzorku. Vygeneruje daný počet `sample`ů a zapíše je do bufferu. Metoda nejdřív zjistí seznam všech zvukových stop, které se nacházejí na současném vzorku a vytvoří z nich nový vzorek, který zapíše na buffer. Pak navýší číslo aktuálního vzorku a opakuje, dokud není zapsán požadovaný počet vzorků.
- **play()** - reaguje na stisknutí tlačítka Play levým tlačítkem myši. Spustí zařízení pro přehrávání audia a vytvoří nové vlákno pro časovač. Ve chvíli, kdy se vlákno spustí, se zavolá `slot played()`.
- **stop()** - reaguje na stisknutí tlačítka Stop levým tlačítkem myši. Zastaví zařízení pro přehrávání audia a ukončí vlákno pro časovač.

Třída obsahuje slot `playing()`. Tento slot spustí časovač a kontroluje, zda-li už nastal čas pro aktualizaci snímku. Pokud ano, zavolá metodu `getOutput()`. Zároveň zjistí, kolik zvukových vzorků je potřeba přehrát a poté zavolá metodu `getAudioOutput()`.

### 5.2.12 Třída `ExportDialog`

Třída sloužící k nastavení parametrů vytvářeného video souboru. Je potomkem třídy `QDialog`.

### 5.2.13 Třída `Muxing`

Třída obsahující funkce pro kódování video souborů. Obsahuje modifikaci programu `muxing.c`, který je součástí projektu `FFmpeg`.

[Zde](#) je odkaz na původní dokumentaci.

Třída obsahuje následující metody:

- **makeVideoFile** - jako parametr bere textový řetězec a vytvoří video soubor v zadané destinaci. Metoda vytvoří video a audio proud, nastaví parametry audio a video kodeků, vypočte celkovou délku projektu a pak postupně kóduje jednotlivé bloky dat z proudů do souboru. Pokud je celková délka projektu vyšší, zobrazí ukazatel průběhu.
- **addStream()** - parametry jsou výstupní stream, výstupní formát, multimedialní kodek, a id kodeku. Metoda vytvoří audio nebo video stream. Nastaví parametry audio a video kodeků pro tento stream.
- **openVideo()** - má jako parametry formát výstupního souboru, multimedialní kodek, výstupní proud a slovník argumentů. Otevře kodek video proudu, alokuje video snímek a zkopíruje parametry proudu do muxeru.
- **allocPicture()** - parametry jsou pixelový formát snímku, jeho výška a šířka. Nastaví formát alokovaného snímku a alokuje datové buffery.
- **writeVideoFrame()** - má jako parametry formát výstupního souboru, výstupní proud a dobu trvání projektu. Vytvoří video snímek a zakóduje jej. Výsledkem je blok dat, který je zapsán do souboru.
- **getVideoFrame()** - jako parametry přijímá výstupní proud a dobu trvání projektu. Nejprve zkontroluje, zda-li je potřeba vykreslit další snímky (tzn. zdali už nebyla překročena doba trvání projektu). Pokud ano, volá metodu `fillYUVImage()`. Výsledný snímek se zapíše do streamu.
- **fillYUVImage()** - jako parametr bere video snímek, jeho číslo, výšku, šířku a proud. Metoda nejdřív zjistí seznam všech vizuálních stop, které se nacházejí na současném snímku a vytvoří z nich nový snímek. Dojde k převodu z barevného modelu RGB na YUV.
- **openAudio()** - má parametry formát výstupního souboru, multimedialní kodek, výstupní proud a slovník argumentů. Otevře kodek audio streamu, inicializuje zvukový generátor a zkopíruje parametry proudu do muxeru.
- **writeAudioFrame()** - má jako parametry formát výstupního souboru, výstupní proud a dobu trvání projektu. Vygeneruje audio vzorky a zakóduje je. Výsledkem je paket, který je zapsán do souboru.
- **getAudioFrame()** - jako parametry bere audio proud a počet vzorků. Metoda vygeneruje daný počet audio vzorků. Nejdřív zjistí seznam všech zvukových stop, které se nacházejí na současném vzorku a vytvoří z nich nový vzorek. Pak navýší číslo aktuálního vzorku a opakuje, dokud není zapsán požadovaný počet vzorků.

- **writeFrame()** - bere jako parametry formát výstupního souboru, časový základ (tj. počet snímků za sekundu na mínus prvou), dále výstupní proud a paket, který má být zapsán. Zapiše do souboru blok audio nebo video dat.
- **closeStream()** - parametrem je výstupní proud. Metoda uvolní alokovanou paměť.

Třída obsahuje sloty:

- **onButtonBoxAccepted()** - pokud uživatel v dialogu klikne na tlačítko OK, zobrazí se nový dialog pro volbu destinace exportovaného souboru. Při zadání validní destinace dojde k zavolání metody `makeVideoFile()` se zvolenou destinací jako parametrem.
- **onButtonBoxRejected()** - pokud uživatel v dialogu klikne na tlačítko Cancel, dojde ke smazání objektu.

#### 5.2.14 Třída Settings

Třída je potomkem třídy `QDialog`. Obsahuje dialog pro nastavení vlastností projektu. Instance se vytvoří při kliknutí na ikonu Options na horním panelu aplikace. Atributem je instance projektu.

#### 5.2.15 Třída MainWindow

Tato třída představuje hlavní okno aplikace. Slouží primárně k přijímání instrukcí od uživatele v hlavním okně aplikace. Je potomkem třídy `QMainWindow`. Obsahuje tyto atributy:

- **project** - ukazatel na objekt aktuálního projektu.
- **preview** - ukazatel na objekt náhledu.

Zde je seznam slotů třídy `MainWindow`:

- **onActionSettingsTriggered()** - otevře dialog s vlastnostmi projektu.
- **onActionOpenTriggered()** - otevře dialog se systémem souborů. Poté nastaví atributy projektu podle informací ze zvoleného souboru.
- **onActionSaveTriggered()** - otevře dialog se systémem souborů. Poté uloží informace o projektu do zvoleného souboru.
- **onActionExportTriggered()** - otevře dialog pro Export projektu.
- **onActionImagesTriggered()** - otevře dialog pro výběr obrázků. Zvolené soubory jsou načteny do projektu.
- **onActionFiltersTriggered()** - vloží nový filtr do časové osy.

- **onActionTextTriggred()** - vloží nový textový objekt do časové osy.
- **onActionSoundTriggred()** - otevře dialog pro import zvukového souboru.
- **onPlayButtonClicked()** - začne přehrávat náhled.
- **onStopButtonClicked()** - ukončí přehrávání náhledu.
- **zoomInClicked()** - přiblíží aktuální zobrazení časové osy o dvojnásobek.
- **zoomOutClicked()** - oddálí aktuální zobrazení časové osy o dvojnásobek.

### 5.2.16 Třída `TimelineTicks`

Třída je potomkem třídy `QSlider`. Jejím úkolem je vykreslovat časovou osu a jezdec, který se po ní pohybuje. Atribut `tickInt` určuje, v jak velké vzdálenosti se vykreslují časové značky. Metoda `getHandleWidth()` vrátí šířku jezdce v pixelech. Vykreslování časových značek provádí přepsaná metoda `paintEvent()`. Třída obsahuje sloty:

- **onActionTiggered()** - reaguje na posun jezdce a v závislosti na jeho pozici upravuje pozici časové osy.
- **updateTicks()** - aktualizuje časové popisky podle současné pozice a míry přiblížení.

### 5.2.17 Třída `Factory`

Třída obsahuje jednoduchou implementaci návrhového vzoru `Factory`. Třída má atribut `map`. Jedná se o asociativní pole, jehož klíči jsou řetězce obsahující názvy tříd a hodnotami jsou instance příslušných tříd. Třída se využívá při načítání projektů. Důvodem je potřeba schopnosti vytvářet instance podle názvů tříd, které jsou uloženy v datových souborech. Tuto schopnost, obecně označovanou jako reflexe, jazyk `C++` nepodporuje.

## 6 Uživatelská příručka

### 6.1 Návod ke spuštění

Pro správnou funkci aplikace je nutné nejprve nainstalovat výše zmíněné knihovny. Otevřete terminál a vložte následující příkazy:

```
sudo apt-get install build-essential qt5-qmake qt5-default  
libavcodec-dev libavformat-dev libavfilter-dev libavutil-dev  
libswscale-dev libswresample-dev libqt5svg5* libopencv-dev  
qtmultimedia5-dev qtgstreamer-plugins-qt5 libqt5gstreamer-dev  
libqt5multimedia5-plugins
```

Nyní je nutné vygenerovat Makefile pomocí příkazu `qmake` s následující syntaxí:

```
qmake -makefile -o makefile "CONFIG+=test" projectfile
```

Nyní můžete aplikaci sestavit pomocí příkazu `make`:

```
make -f makefile
```

Následně spustíte soubor s názvem `Bakalarskaprace`.

```
./Bakalarskaprace
```

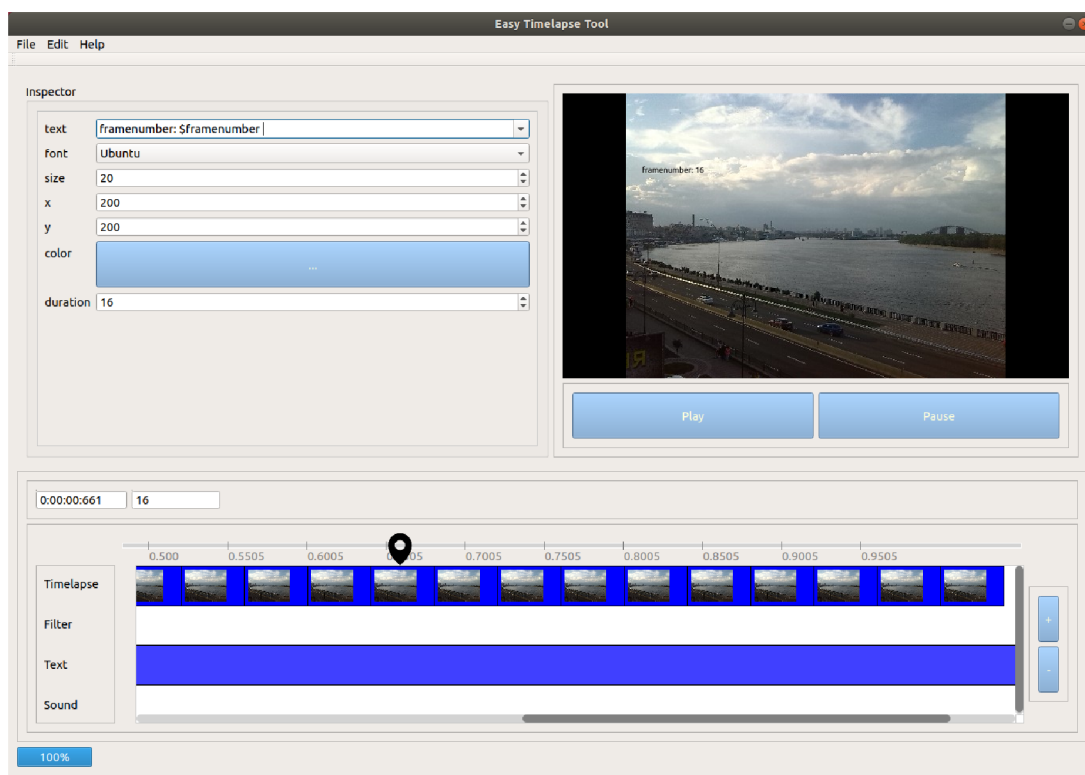
Na obrázku 5 vidíme okno aplikace bezprostředně po spuštění. Grafické rozhraní se dělí na čtyři základní části: horní panel, pod ním na levé straně inspektor, na pravé straně náhled videa a dole časová osa. V následující kapitole detailně popíšeme jednotlivé součásti uživatelského rozhraní a jejich funkci.

### 6.2 Uživatelské rozhraní

#### 6.2.1 Horní panel

Horní panel obsahuje základní nástroje pro práci s aplikací. Následuje jejich výčet (zleva doprava):

- **New** - Vytvoření nového projektu. Pokud uživatel současně pracuje na jiném projektu, zobrazí se dialog dávající na výběr uložení nebo zahození aktuálního projektu. Následně se zobrazí dialog umožňující nastavení základních parametrů projektu, jako je například rozlišení nebo snímková frekvence. Tyto parametry je později možné změnit (viz. odrážka `Options`).
- **Open** - Otevření projektu ze souboru. Mohou být otevřeny pouze soubory s příponou `.dat`.
- **Save** - Uložení projektu do souboru s příponou `.dat`.



Obrázek 5: Grafické rozhraní aplikace

- Export - uložení výsledného videosouboru. Export nabízí možnost nastavení kontejneru, audio a video kodeku a přenosové rychlosti audia videa. Každý datový kontejner podporuje jiné kombinace kodeků. Kodeky, které nejsou podporovány u aktuálně zvoleného kontejneru, jsou zašedlé.
- Exit - opuštění aplikace.
- Add Images - vložení obrázků do časové osy. Zobrazí okno se systémem souborů. Pomocí klávesy Shift je možné vybrat několik obrázků současně. Podporovanými formáty jsou .png, .bmp a .jpg. Obrázky se do časové osy vloží v pořadí, ve kterém byly vybrány.
- Add Audio - vložení zvukové stopy do časové osy. Mezi podporované formáty patří .mp3, .flac a .wav.
- Add Text - vložení textu do časové osy.
- Add Filter - vložení obrazového filtru.
- Settings - zobrazí dialogové okno umožňující nastavit základní parametry projektu (rozlišení a snímková frekvence). U snímkové frekvence jsou dvě možnosti nastavení. První možnost je klasické nastavení snímkové frekvence. Druhá možnost nabízí uživateli stanovení doby trvání časosběrného



filmu. Podle této doby bude odvozena nová snímková frekvence. Rovněž je možné povolit nebo zakázat funkci automatické korekce posunutí.

- About - zobrazí informace o aplikaci.

### 6.2.2 Inspektor

Každá položka na časové ose má své specifické nastavitelné parametry, které se zobrazují v inspektoru. Inspektor se nachází v levé horní části grafického rozhraní. Parametry v inspektoru se zobrazí při kliknutí levým tlačítkem myši.

- U Obrazové stopy se zobrazují EXIF informace (jsou-li k dispozici).
- U Zvukové stopy je možné nastavit hlasitost v procentech, délku trvání a počáteční pozici.
- Textová stopa má šest parametrů - text, font, velikost písma, barvu a dobu trvání. Za pozici se považuje levý horní okraj textového pole. Do textu lze vkládat regulární výrazy obsahující názvy proměnných. Proměnná je uvedena symbolem \$ a ukončena mezerou. Textové pole obsahuje dropdown menu, které obsahuje seznam všech podporovaných proměnných. Po výběru proměnné se název proměnné přidá do aktuálního textového pole. Pokud má proměnná na aktuální pozici nějakou hodnotu, pak se její jméno nahradí její hodnotou. Např. \$cameramodel se přepíše na řetězec Canon.
- Filtry obsahují pět nastavitelných atributů - Brightness (jas), Contrast (nastavení kontrastu), Saturation (sytost), Hue (barevný odstín) jsou nastavitelné na škále od -127 do 127. Gamma korekce je nastavitelná na škále 0 až nekonečno.

### 6.2.3 Náhled

Sekce Náhled se nachází v pravé horní části grafického rozhraní. Zobrazuje miniaturu aktuálního snímku. Tlačítko Play slouží pro kontinuální přehrávání snímků, ukončené tlačítkem Stop. Uživatel může nastavit pozici nahlíženého snímku pomocí jezdcy na časové ose nebo kliknutím na požadovaný snímek. Náhled lze přehrávat pouze v případě, kdy jsou načteny všechny snímky.

### 6.2.4 Časová osa

Sekce Časová osa se nachází ve spodní části grafického rozhraní. Obsahuje vizuální reprezentaci všech stop vytvářeného projektu. Je možné ji přibližovat nebo oddalovat pomocí tlačítek umístěných v pravém dolním rohu. U standardního přiblížení je 1:1, tedy jedna milisekunda odpovídá jednomu pixelu na časové ose. Maximální přiblížení je v poměru 1:32, minimální přiblížení v poměru 1024:1. Nad časovou osou se nachází jezdec, který definuje pozici aktuálně nahlíženého snímku. Jezdec je interaktivní a jeho pozice může být změněna buď tažením myši

nebo kliknutím na požadovanou pozici na ose. Nad časovou osou se nachází počítadlo. Zobrazuje číslo aktuálního snímku při dané snímkové frekvenci a jeho pozici v milisekundách. Pod časovou osou se nachází jednotlivé stopy projektu. Každá stopa je graficky reprezentovaná jako obdélník s levým okrajem na časové pozici, kde začíná, a pravým okrajem na pozici, kde končí. Stopy jsou interaktivní. Při kliknutí na stopu je stopa označena a její parametry se zobrazí v inspektoru. Při tažení myši je možné stopu přesunout doleva nebo doprava. Grafické stopy se přesunují po snímcích, zvukové stopy lze přesunovat po vzorcích. Každou označenou položku lze vymazat kliknutím pravým tlačítkem a zvolením položky Delete. Obrázky lze navíc duplikovat, duplikát je vložen bezprostředně za originální obrázek. Je možné označit více stop najednou pomocí klávesy Shift.

### **6.2.5 Progress bar**

Jedná se o progress bar nacházející se v levém dolním rohu aplikace. Znázorňuje průběh načítání obrázků v procentech. Při dosažení 100% se aktivuje tlačítko přehrávání náhledu.

## 7 Analýza řešení

Aplikace funguje uspokojivě, existuje však několik známých problémů. Mezi ně patří:

- Chybějící náhled zvukové stopy. Při pohledu na časovou osu není zřejmé, jaká část zvukové stopy se nachází na určitém místě. Řešením by bylo vykreslit na zvukové stopě křivku zobrazující průběh zvukové vlny.
- Při přehrávání náhledu občas dochází k přeskokování snímků. Náhled se nepřehrává plynule zejména při kombinaci aplikace filtrů a vyšší snímkové frekvence, zejména nad 30 fps.
- Posuvník na časové ose za určitých okolností nereaguje plynule. K tomu dochází, pokud zobrazení přiblíženo na úroveň, kdy se zobrazují malé náhledy na obrazových stopách.
- Možnost změny velikosti náhledu. Náhled má maximální velikost 640x360, což nemusí být ve všech případech dostačující. Menší náhled však snižuje nároky na výkon.
- Přidání klávesových zkratk pro některé úkony, např. vymazání aktuálního výběru.
- Chybí možnost změnit délku stopy zatažením myši za její okraj.
- Bylo by možné přidat více druhů filtrů.
- Hlavní menu neobsahuje funkce Zpět/Znovu, které by usnadnily práci s aplikací.
- Aplikace postrádá modernější grafické rozhraní. Současné grafické rozhraní neodpovídá současným trendům, řešením by bylo tedy přepracovat jej podle zásad tzv. flat designu. Bylo by možné vymazat horní menu a nahradit jej přehlednější nástrojovou lištou.

## Závěr

V rámci bakalářské práce byla vytvořena funkční aplikace, která splňuje všechny požadavky, které vyplývají ze zadání. K aplikaci byla vytvořena přehledná uživatelská i technická dokumentace. Bylo ukázáno, že aplikace je v určitých aspektech srovnatelná s již existujícími aplikacemi podobného zaměření. Je možné další ladění a rozšiřování aplikace.

## **Conclusions**

The result of this bachelor thesis is a functional application meeting all the requirements specified in the assignment. Application has a proper technical and user documentation. Application is partially comparable with similar applications. A further improving and extending is possible.

## A Obsah přiloženého CD/DVD

### **doc/**

Text práce ve formátu PDF, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu.

### **src/**

Kompletní zdrojové texty programu BAKALARSKAPRACE.

### **readme.txt**

Instrukce pro instalaci a spuštění programu BAKALARSKAPRACE.

Navíc CD/DVD obsahuje:

### **data/**

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

## Literatura

- [1] Časosběr. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2019 [cit. 2021-04-15]. Dostupné z: <https://cs.wikipedia.org/wiki/%C4%8Casosb%C4%9Br>
- [2] About OpenShot. *OpenShot Video Editor* [online]. Rockwall: OpenShot Studios, LLC., c2008-2021 [cit. 2021-04-09]. Dostupné z: <https://www.openshot.org/about/>
- [3] Features. *Kdenlive* [online]. [cit. 2021-04-09]. Dostupné z: <https://kdenlive.org/en/features/>
- [4] About. *Kdenlive* [online]. [cit. 2021-04-09]. Dostupné z: <https://kdenlive.org/en/about/>
- [5] Stopmotion. *Skolelinux* [online]. c2006 [cit. 2021-04-09]. Dostupné z: <http://developer.skolelinux.no/info/studentgrupper/2005-hig-stopmotion/>
- [6] KIRCH-PRINZ, Ulla a Peter PRINZ. *A Complete guide to C++ programming*. Sudbury, Massachusetts: Jones & Bartlett Publishers, 2002. ISBN 0-7637-1817-3.
- [7] Supported Platforms. *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://doc.qt.io/qt-5/supported-platforms.html>
- [8] Supported Platforms & languages *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://www.qt.io/platforms-languages>
- [9] Open Source Qt Use *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://www.qt.io/download-open-source>
- [10] Qt Features. *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://www.qt.io/product/features>
- [11] Getting Started with Qt. *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://doc.qt.io/qt-5/gettingstarted.html>
- [12] Using the Meta-Object Compiler (moc). *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://doc.qt.io/qt-5/moc.html>
- [13] Signals & Slots *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://doc.qt.io/qt-5/signalsandslots.html>
- [14] qmake Manual. *Qt* [online]. Espoo: The Qt Company, c2020 [cit. 2021-04-09]. Dostupné z: <https://doc.qt.io/qt-5/qmake-manual.html>
- [15] About FFmpeg. *FFmpeg* [online]. [cit. 2021-04-09]. Dostupné z: <https://www.ffmpeg.org/about.html>

- [16] FFmpeg License and Legal Considerations. *FFmpeg* [online]. [cit. 2021-04-09]. Dostupné z: <https://www.ffmpeg.org/legal.html>
- [17] BRADSKI, Gary R. a Adrian KAEHLER. *Learning OpenCV*. Sebastopol: O'Reilly, c2008. ISBN 978-0-596-51613-0.
- [18] LAHIRI, Mayank. Easyexif. Github [online]. San Francisco: Github, 2021 [cit. 2021-04-09]. Dostupné z: <https://github.com/mayanklahiri/easyexif>
- [19] LOCH, Francis. Image Processing Algorithms Part 5: Contrast Adjustment. *The Crypt Magazine* [online]. RIYAN Productions, 2009 [cit. 2021-04-14]. Dostupné z: [http://www.thecryptmag.com/Online/56/imgproc\\_5.html](http://www.thecryptmag.com/Online/56/imgproc_5.html)
- [20] LOCH, Francis. Image Processing Algorithms Part 6: Gamma Correction. *The Crypt Magazine* [online]. RIYAN Productions, 2009 [cit. 2021-04-14]. Dostupné z: [http://www.thecryptmag.com/Online/57/imgproc\\_6.html](http://www.thecryptmag.com/Online/57/imgproc_6.html)
- [21] LOCH, Francis. Image Processing Algorithms Part 4: Brightness Adjustment. *The Crypt Magazine* [online]. RIYAN Productions, 2009 [cit. 2021-04-14]. Dostupné z: [http://www.thecryptmag.com/Online/55/imgproc\\_4.html](http://www.thecryptmag.com/Online/55/imgproc_4.html)
- [22] Codec. *Mux* [online]. c2021 [cit. 30.07.2021]. Dostupné z: <https://mux.com/video-glossary/codec>
- [23] Encoding. *Mux* [online]. c2021 [cit. 30.07.2021]. Dostupné z: <https://mux.com/video-glossary/encoding>
- [24] Decoding. *Mux* [online]. c2021 [cit. 30.07.2021]. Dostupné z: <https://mux.com/video-glossary/decoding>
- [25] Datový proud. *IT slovník* [online]. IT slovník team, c2008-2021 [cit. 2021-7-30]. Dostupné z: <https://it-slovník.cz/pojem/datovy-proud>
- [26] Video Streaming Definition. *Haivision* [online]. c2021. [cit. 30.07.2021]. Dostupné z: <https://www.haivision.com/resources/streaming-video-definitions/streaming-media>
- [27] Definition of multimedia container. *PCMag* [online]. c1996 [cit. 30.07.2021]. Dostupné z: <https://www.pcmag.com/encyclopedia/term/multimedia-container>
- [28] Multiplexing / Muxing. *Mux* [online]. c2021 [cit. 30.07.2021]. Dostupné z: <https://mux.com/video-glossary/multiplexing-muxing>
- [29] Demultiplexing / Demuxing. *Mux* [online]. c2021 [cit. 30.07.2021]. Dostupné z: <https://mux.com/video-glossary/demultiplexing-demuxing>
- [30] Resampling. *DSP Guru* [online]. Iowegian International Corporation, c1999-2020 [cit. 30.07.2021]. Dostupné z: <https://dspguru.com/dsp/faqs/multirate/resampling>



- [31] What is Object Oriented Programming? OOP Explained in Depth. *Educative: Interactive Courses for Software Developers* [online]. c2021 [cit. 01.08.2021]. Dostupné z: <https://www.educative.io/blog/object-oriented-programming>
- [32] What is EXIF Data and How You Can Remove it From Your Photos. *Photography Life* [online]. Dostupné z: <https://photographylife.com/what-is-exif-data>
- [33] Framework definition. *The Tech Terms Computer Dictionary* [online]. Dostupné z: <https://techterms.com/definition/framework>
- [34] Learning Design Patterns [online]. ZZZ Projects, 2019 [cit. 2021-8-1]. Dostupné z: <https://riptutorial.com/Download/design-patterns.pdf>
- [35] UML Tutorial. *Nonmouth University* [online]. West Long Branch: Nonmouth University, 1997 [cit. 2021-8-2]. Dostupné z: <http://bluehawk.monmouth.edu/~jwang/umlClassDiagrams.pdf>