

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## NÁSTROJ PRO DABOVÁNÍ VIDEOA

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR KOŇÁREK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# NÁSTROJ PRO DABOVÁNÍ VIDEOA

VIDEO DUBBING TOOL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR KOŇÁREK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2015

## **Abstrakt**

Tato práce se zabývá návrhem a implementací multiplatformního nástroje pro dabing. Jde tedy převážně o zpracování videa a audia, respektive nahrávání hlasu dabérů a jejich následné editace. Naleznete zde technologie, které tuto problematiku řeší a přitom zajišťují multiplatformnost. Návrh nástroje je vytvořen, aby obsahoval všechny potřebné operace s audiem a přitom uživatelské rozhraní zůstalo lehce ovladatelné. Závěrem se dozvíte, jak byl nástroj otestován v soutěži v rychlodabingu.

## **Abstract**

This thesis deals with draft and implementation of a multi-platform dubbing tool. It is about video and audio processing, recording voice of separate voice actors and its subsequent editing. You can find a set of technologies that are used to solve these tasks and to ensure compatibility with various operating systems. The design is made to include all the necessary audio operations while preserving its simplicity. Finally, you will learn about the testing process of this tool in a fast-dubbing competition and the result of it.

## **Klíčová slova**

Dabing, Audio, Video, Multiplatformní, Qt, QtAV, QML, FFmpeg

## **Keywords**

Dubbing, Audio, Video, Multiplatform, Qt, QtAV, QML, FFmpeg

## **Citace**

Petr Koňárek: Nástroj pro dabování videa, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Nástroj pro dabování videa

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zbyňka Křivky, Ph.D.

.....

Petr Koňárek  
16. května 2015

## Poděkování

Děkuji mému vedoucímu Ing. Zbyňku Křivkovi, Ph.D. za rady a pomoc při řešení této bakalářské práce. Dále bych chtěl poděkovat vedení Animefestu, že mi umožnili nástroj prakticky vyzkoušet.

© Petr Koňárek, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

|                               |           |
|-------------------------------|-----------|
| <b>1 Úvod</b>                 | <b>2</b>  |
| <b>2 Analýza</b>              | <b>3</b>  |
| 2.1 Dabing                    | 3         |
| 2.2 Existující řešení         | 3         |
| 2.3 Použité technologie       | 4         |
| 2.3.1 Qt                      | 4         |
| 2.3.2 QtAV                    | 6         |
| 2.3.3 FFmpeg                  | 6         |
| <b>3 Návrh nástroje</b>       | <b>8</b>  |
| 3.1 Požadované funkce         | 8         |
| 3.2 Zvuk                      | 8         |
| 3.2.1 Nahrávání               | 9         |
| 3.2.2 Editace                 | 9         |
| 3.2.3 Ukládání                | 10        |
| 3.2.4 Export                  | 10        |
| 3.2.5 Odstranění hlasu        | 10        |
| 3.3 Video                     | 10        |
| <b>4 Uživatelské rozhraní</b> | <b>11</b> |
| 4.1 Hlavní okno               | 12        |
| 4.1.1 Mikrofóny               | 13        |
| 4.2 Vizualizační okno         | 13        |
| 4.2.1 Vizualizace             | 14        |
| 4.2.2 Nástrojová plocha       | 15        |
| 4.2.3 Časová osa              | 15        |
| 4.2.4 Editace                 | 15        |
| 4.3 Video okno                | 17        |
| <b>5 Implementace</b>         | <b>18</b> |
| 5.1 Video                     | 18        |
| 5.2 Zvuk                      | 19        |
| 5.2.1 Nahrávání               | 19        |
| 5.2.2 Přehrávání              | 20        |
| 5.2.3 Zobrazení               | 20        |
| 5.3 Metadata a ukládání       | 21        |
| 5.4 Editace                   | 21        |

|          |   |           |
|----------|---|-----------|
| 5.4.1    | Slučování a spojování . . . . .           | 22        |
| 5.4.2    | Rozdělování . . . . .                     | 22        |
| 5.4.3    | FFmpeg . . . . .                          | 22        |
| 5.4.4    | Undo a Redo . . . . .                     | 23        |
| 5.5      | Export . . . . .                          | 23        |
| <b>6</b> | <b>Testování</b>                          | <b>25</b> |
| 6.1      | Multiplatformnost . . . . .               | 25        |
| 6.2      | Soutěž . . . . .                          | 25        |
| 6.2.1    | Chyby a nedostatky . . . . .              | 26        |
| <b>7</b> | <b>Závěr</b>                              | <b>27</b> |
| 7.1      | Rozšíření a budoucnost projektu . . . . . | 27        |
| <b>A</b> | <b>Obsah CD</b>                           | <b>29</b> |
| <b>B</b> | <b>Manual</b>                             | <b>30</b> |

# Kapitola 1

## Úvod

Dabování je dnes ve světě filmového průmyslu běžná věc. Díky tomu jsou filmy dostupné v mnoha jazycích. K dobrému dabingu je ovšem nutné několik prvků. Asi tou nejdůležitější jsou dobří dabéři, kteří se svými hlasy kvalitou ovlivňují nejvíce. Samozřejmostí je i dobrý hardware čili mikrofony, filtry a různé mixážní pulty, protože kvalita samotného vstupu je důležitá, aby zvuk ve filmu nechrčel nebo nebyl zkreslený hlas. Další složkou, kterou se bude zabývat tato práce, je softwarová podpora.

Díky dobrému programu můžeme v postprodukcí spojovat zvukové stopy, které dabéři vytvořili, či je dokonce překrývat, pokud postavy mluví najednou. S touto podporou nemusí být přítomni najednou všichni dabéři, kteří jsou potřební pro danou scénu.

Vše co bylo řečeno platí pro profesionální dabovací společnosti, které mají přístup a peníze k tomu nejlepšímu co se na trhu nachází. Vytvořený nástroj bude určen spíše pro amatéry, kteří si chtějí předabovat kus svého oblíbeného filmu či seriálu nebo k přidání komentáře k rodinnému videu. K tomu jim bude stačit pouze mikrofón, který dnes již má většina lidí kvůli volání přes internet.

Nástroj tedy bude cílen na jednoduchou obsluhu a zaměření převážně na práce, které jsou potřeba pro dabing. To znamená extrahování zvukových stop z videa. Práce se zvukovou stopou či stopami, konkrétně půjde o posouvání, mixování, ořezávání atd. Program bude pracovat s těmi nejpoužívanějšími video a audio formáty. K tomu byl zvolen jazyk C++ s rámcem Qt. Ale pro práci se samotným videem a audiem je použit multimediální rámec QtAV a nástroj FFmpeg.

Tato práce vás tedy uvede do řešení základních problémů spojené s videem a audiem, které jsou nutné, aby dabovací nástroj fungoval. Jak tyto požadavky implementovat a vytvořit k tomu intuitivní, smysluplné a jednoduché prostředí, se dozvíte v následujících kapitolách.

## Kapitola 2

# Analýza

Pro dobré vytvoření návrhu a implementování jakéhokoliv softwaru, je důležité prozkoumat oblast, se kterou budete pracovat. Musíme tedy zjistit, co je všechno potřeba, aby nástroj zajistil vše nutné pro dabing viz sekce 2.1. Analýzou dosavadních řešení se inspirujeme, jak na problematiku práce se zvukem a videem nahlíží jiní vývojáři. Jelikož nemáme u většiny přístup ke zdrojovým kódům, tak se zaměříme na vizuální prvky a jak se pracuje s takovou aplikací, která edituje zvuk či video, z uživatelského hlediska. Co je všechno potřeba, aby nástroj uměl, jak rozvrhnout uživatelské prostředí pro lehkou orientaci a efektivnost. Všechny tyto vlastnosti se nejlépe zjistí při zkoušení ostatních, na tuto problematiku zaměřených, aplikací. Při tomto testování se zaměříme, co všechno je potřeba k chodu aplikace, jaké je prvotní seznámení atd. Dalším důležitým faktorem je výběr technologií, na kterých bude váš software postaven. Musíme si stanovit specifika jako multiplatformnost, závislost na jiném softwaru a doplňcích v systému. Nesmíme ovšem ani zanedbat, že nástroj bude primárně použit v soutěži rychlodabingu, a to se určitě do nástroje promítne.

### 2.1 Dabing

Jak bylo zmíněno v úvodu v dnešním filmářském průmyslu je dabing běžná věc. Zde se však používají profesionální nástroje, ať už jde o software nebo hardware. Pro ty, co by pojem dabing neznali, zde výtazek z diplomové práce o typu filmového překladu [4].

*Jako nejlepší definice dabingu nám může posloužit ta od O. Walló: Dabing je „...výrob[a] jinojazyčné verze k danému originálu.“ Jedná se tedy o to, že k hotovému filmu „pouze“ vytvoříme hlasovou kulisu. Znamená to, že potřebujeme převést slova z výchozího jazyka do cílového, a to většinou s významem změněným pro diváka cílového jazyka srozumitelným. Dabing totiž ruší jazykovou bariéru a umožňuje divákovi vnímat dílo v řeči, které plně rozumí: v trojici složek dabingu obraz - zvuk - slovo právě rovina slova jako jediná podléhá překladu a úpravě, proto je důležité ji vidět na pozadí ostatních rovin filmu.*

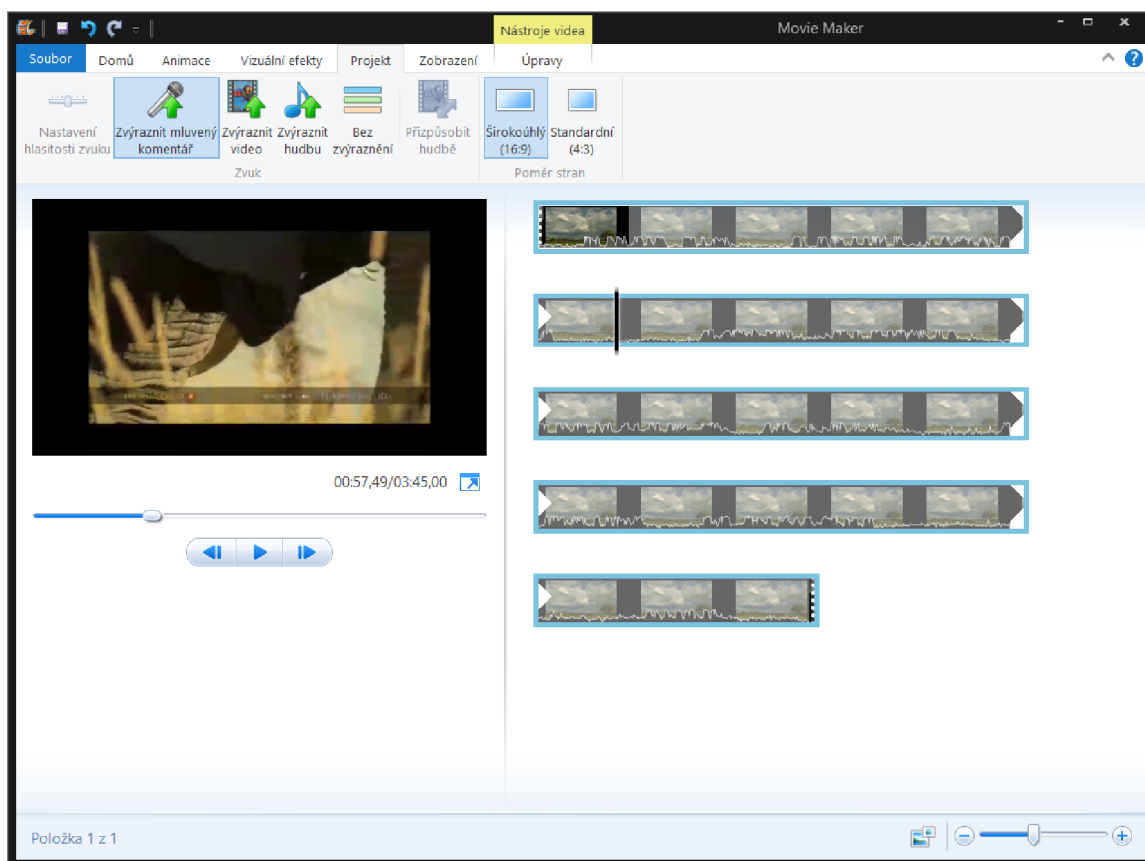
Ještě existuje tzv. voiceovering [1]. Tato technika pouze utlumuje původní zvukovou stopu a na takto upravenou zvukovou stopu nasazuje novou. Pro tento projekt je ovšem nejdůležitější, co k dabingu potřebujeme a jaké tedy funkce musí výsledný nástroj nabízet. U dabingu tedy určitě potřebujeme nahrávání zvukového vstupu čili jazykové stopy a její následnou editaci.



## 2.2 Existující řešení

Dnes na internetu lze nalézt mnoho programů, které umí pracovat se zvukem a obrazem. Jedním z nejpožívanějších editorů pro práci se zvukem je AudaCity. Jde o multiplatformní program umožňující rozsáhlou úpravu zvuku. Zvládne např. konvertování zvukových formátů, různý stříh stop, ořezávání frekvencí, vkládání různých efektů, nahrávání z mikrofonů atd. Bohužel pro dabing není příliš vhodný a to kvůli absenci přehrávání videa a tím pádem i nemožnosti nahraný dabing do videa vložit. Ovšem pro inspiraci, jak pracovat se zvukem na uživatelské úrovni, je vhodný.

Programy vhodnější pro dabing je Windows Movie Maker <sup>1</sup>. Windows Movie Maker (obr. 2.1) je jednoduchý video editor, kde můžete do zvukové stopy vkládat svoje komentáře nebo jiný zvukový podklad, který chcete do svého videa vložit. Zvládá i jiné efekty pro zpracování videa. Bohužel nelze odstranit původní stopu ale pouze ji ztlumit v místech uživatelem vloženého zvuku. Není tedy určen přímo pro dabování videí, ale pouze pro úpravu např. rodinných videí nebo pokud chcete vytvořit vaše oblíbené filmy vámi namluvenými komentáři.



Obrázek 2.1: Windows Movie Maker

Na internetu nalezneme i spousty profesionálních programů pro práci s multimédií, bohužel jsou většinou placené, ale zase nabízí mnoho funkcí, které k profesionální práci potřebujete. Jelikož jde o mocné nástroje s mnoha funkcemi, jsou složité a z toho vyplývá

<sup>1</sup><http://windows.microsoft.com/cs-cz/windows7/products/features/movie-maker>

pozvolná učící křivka. To může některé lidi od tvorby odradit, když ne to, že jde o placené programy. Jedná se např. o MAGIX Movie Edit, Adobe Audition CC, Corel VideoStudio.

## 2.3 Použité technologie

Technologie byly vybrány na základě schopnosti pracovat s videem a audiem a možností použití na co nejvíce platformách. Konkrétně byli vybrány rámce Qt, QtAV a nástroj v příkazové řádce FFmpeg. Začátek vývoje probíhal na linuxové distribuci Arch Linux a po úspěšném zkompileování QtAV pro operační systém Windows, práce pokračovali v systému Windows 8.1. V obou případech jsem použil vývojové prostředí Qt Creator, které, jak napovídá název, je určeno přímo pro vývoj Qt aplikací. Poskytuje tedy vše potřebné pro vývoj, včetně zobrazení grafického návrhu uživatelského rozhraní.

### 2.3.1 Qt

Qt<sup>2</sup> je známý multiplatformní rámec, který umožňuje jednoduchý vývoj grafických aplikací pro více operačních systémů. V projektu je použit ve spojení s C++, ale dá se používat i s mnoha jinými programovacími jazyky např. Python, Ruby, Java a jiné. Jedním z přínosů je, že jako programátor napíšete jenom jeden kód a s malými úpravami jste schopni aplikaci spustit na všech hlavních platformách čili Windows, Apple Mac OS a různé linuxové distribuce, ale třeba i na různých mobilních platformách. Qt také umožňuje vývoj grafických uživatelských rozhraní, a to pomocí tzv. widgetů nebo pomocí Qt Quick, který je použit i v tomto projektu [2].

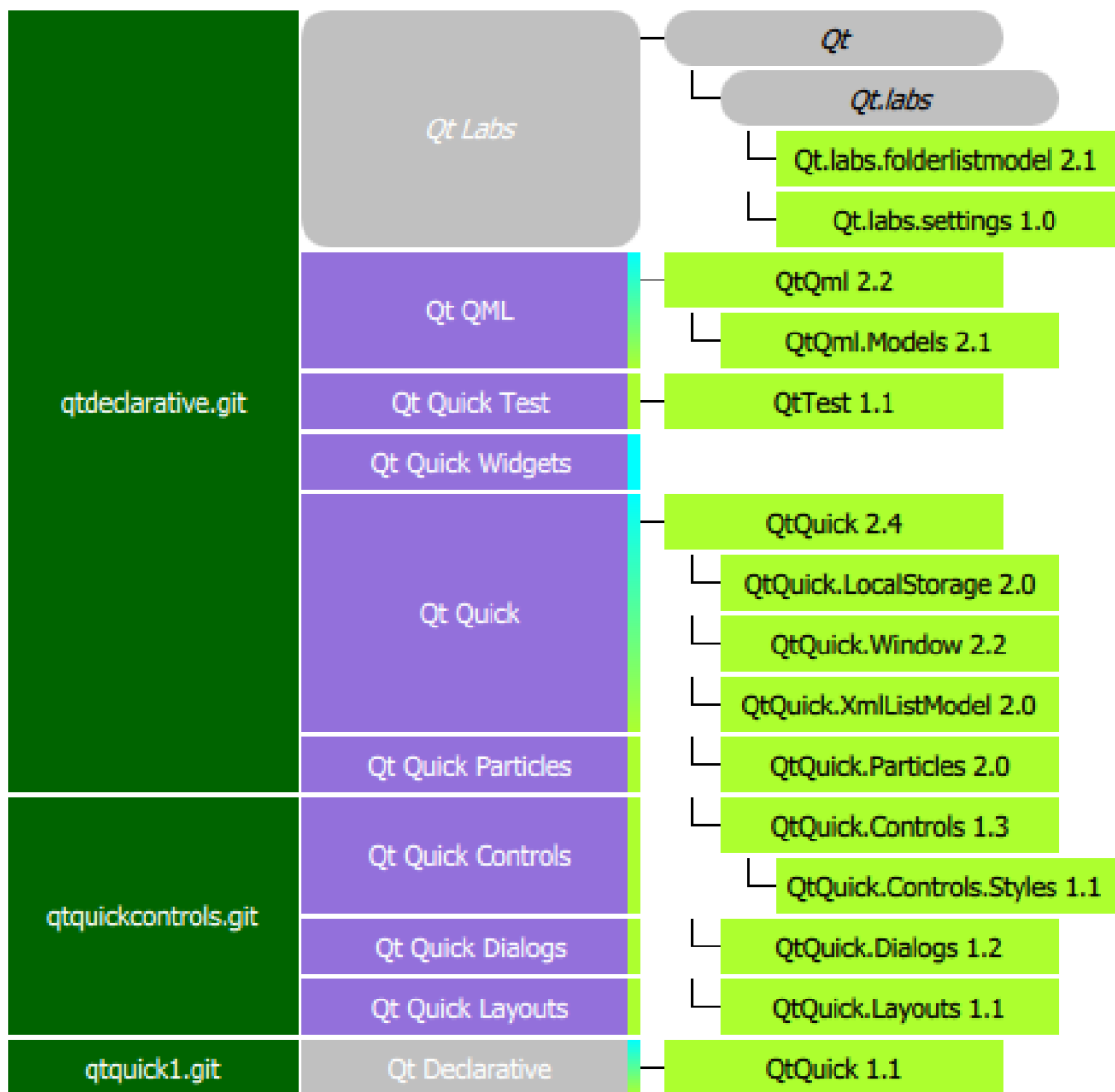
Qt Quick (Qt User Interface Creation Kit)<sup>3</sup> je moderní intuitivní kolekce technologií pro navrhování dynamického uživatelského prostředí v jazyku QML. Programátorovy dává k dispozici základní grafické typy jako tlačítko, výběrové boxy atd. Na tyto typy může aplikovat grafický styl, který si může různě upravovat dle své představy. Nebo má možnost za použití základní komponenty, ze kterého ostatní prvky dědí, vytvořit další, podle potřeby vyvíjené aplikace.

QML je vysokoúrovňový jazyk a podobný skriptovacím jazykům a kaskádovým stylům. Lze v něm také psát funkce v JavaScriptu [5] nebo volat funkce objektu C++, což je hlavně vhodné pokud chcete provádět nějaké výpočty či využívat různorodé datové struktury, které Qt nebo JavaScript používá. Na obrázku 2.2 je zobrazena struktura Qt Quick. V kapitole 5 je řečeno, které moduly byli v projektu použity a co nabízí [3].

---

<sup>2</sup><http://www.qt.io/>

<sup>3</sup><http://doc.qt.io/qt-5/qtquick-index.html>



Obrázek 2.2: Struktura QML

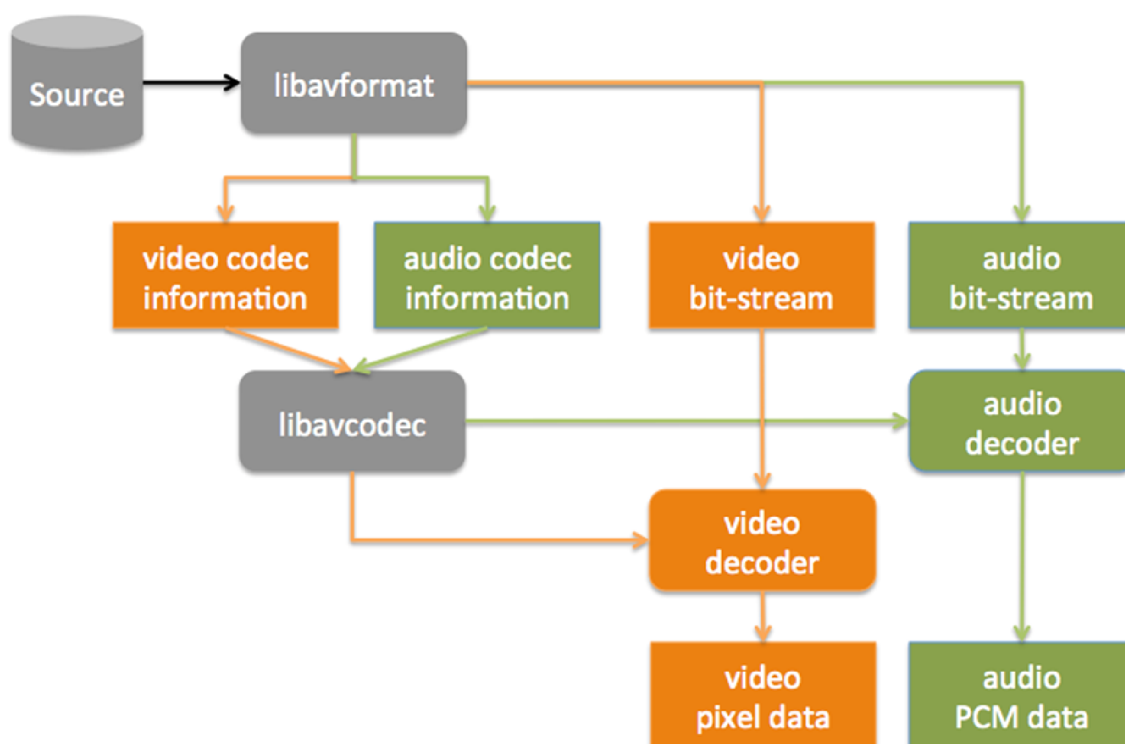
### 2.3.2 QtAV

Jde o rámec<sup>4</sup>, rozšiřují výše zmíněné Qt o lepší podporu přehrávání multimédií, od čínského vývojáře Wang Bin ze Shanghaiské university. Jelikož používá knihovny FFmpeg, viz v sekci 2.3.3, přehrává všechny nejrozšířenější formáty audia a videa a to nezávisle na tom jaký systém používáte a jaké kodeky jsou nainstalovány v systému. Je plně multiplatformní. Lze také využít jeho schopnosti hardwarové akcelerace při přehrávání a umí i vykreslování titulků. Lze použít jak u klasických Qt Widgetů tak i u novějšího Qt Quick. Vytvoření jednoduchého přehrávače je díky němu otázka několika řádků kódu.

<sup>4</sup><http://www.qtav.org/>

### 2.3.3 FFmpeg

Jsou knihovny a stejnojmenný nástroj<sup>5</sup> pro práci s multimédií umožňující dekodování, kódování, spojování, oddělování jak videa tak i audia a má ještě mnoho dalších funkcí. Tyto knihovny navíc podporují snad všechny existující formáty videí a zvuku. Vývojáři tohoto rámce se snaží o co nejlepší technickou stránku, čili co nejmenší zatížení prostředků nebo lepší využitelnost pro ostatní programátory. V projektu bude použit pouze nástroj, který funguje v příkazové řádce, kvůli složitosti knihoven. FFmpeg bude použito pro finální exportování výsledků dabingu a dílčí mezi kroky, rozdělování a spojování nahrávek atd. Nejznámějším programem používající tyto knihovny je VLC media player<sup>6</sup>. V rámci FFmpeg jsou ještě k dispozici další nástroje. Jde o FFplay, jednoduchý přehrávač. FFserver, slouží jako multimediální streaming server pro živé vysílání a FFprobe, jenž se používá pro analýzu toku dat v multimédiích.



Obrázek 2.3: Dekódování pomocí FFmpeg

<sup>5</sup><https://www.ffmpeg.org/>

<sup>6</sup><http://www.videolan.org/vlc/>

## Kapitola 3

# Návrh nástroje

Cílem je vyvinout multiplatformní aplikaci, která zprostředkuje vše, co je potřeba k amatérskému, ale přesto kvalitnímu dabingu se zaměřením na použití v soutěži v rychlodabingu. Umožnit uživateli jednoduché ovládání a rychlou orientaci v uživatelském prostředí. Tím dosáhneme, že uživatel bude moci po krátkém seznámení začít ihned pracovat.

### 3.1 Požadované funkce

Nástroj musí umět nahrávat z několika mikrofónů zároveň a následně editovat zvuk, a to tak, jak je popsáno v sekci 3.2. Přehrávat videa, která budeme dabovat, i s mezivýsledky naší dosavadní práce. Podporovat co nejvíce formátu videa, bez nutnosti doinstalovávat kodeky a jiný software. Hlavně rodiny MPEG, jelikož jde o nejčastěji vyskytující se formát a audia (např. AAC, MP3, FLAC). A v poslední řadě ukládat mezivýsledky nebo rovnou exportovat finální verzi, což znamená spojení námi nahraného dabingu a původní zvukové stopy ve vybraném videu. Formáty, které jsou podporovány jako vstup pro přehrávání, by měli být podporovány i jako výstup. Veškeré tyto vlastnosti budou řešeny s ohledem na použití v soutěži.

Uživatelské prostředí musí být přehledně uspořádáno a vytvořeno na míru k práci se zvukem tak, aby uživatel nemusel číst manuály a za menší časovou dobu používání nástroje, byl schopný program alespoň v základu používat. Je nutno nějakým přijatelným způsobem zobrazit zvuk, aby se s ním dalo pracovat. Z toho vyplývají jednoduché akce několika kliknutí myši vedoucí k výsledkům, které uživatel zamýšlí.

### 3.2 Zvuk

U dabingu je nejdůležitější práce se zvukem, konkrétně s hlasem. Z toho lze vyvodit, že hlavní část této práce se bude věnovat hlavně zvuku, konkrétně tomu, co potřebujete u dabingu, tedy nahrávání v dostatečném frekvenčním rozsahu a kvalitě, pro nahrání lidského hlasu a následná editace. Zvuk musí být zobrazen, aby jeho grafická pozice odrážela jeho reálnou pozici v dabovaném videu. Proto bude zobrazen v posuvném horizontálním poli, kde budou jednotlivé grafické objekty reprezentovat nahrané části. Po vybrání některého objektu s ním můžeme manipulovat tak, jak je popsáno dále v této kapitole. Zde u zobrazených nahrávek bude přítomna i časová osa zobrazující pozici ve videu s možností tuto pozici i měnit.

Při načtení videa určeného k dabování se v pozadí načtou zvukové stopy a vyplní se uživatelské prostředí daty získané ze zvoleného souboru. Pokud budeme načítat meziprodukt předchozí práce, tak načítání je stejné, kromě načtení již vytvořeného dabingu. Ten se načte pouze jako metadata a zobrazí se grafické objekty podle těchto načtených dat. Dabing, jenž je uložený na disku, nebude nijak načítán ani jinak ovlivněn. Tyto metadata se budou načítat ze speciálního souboru s metadaty (sekce 3.2.3).

### 3.2.1 Nahrávání

Při dabingu je samozřejmě potřeba nahrávat hlas dabéra či rovnou několika dabérů zároveň. K tomu slouží mikrofon, který má dnes snad každý v notebooku. Uživatel ovšem pro zlepšení kvality může využívat i externí, a to mu musíme umožnit. Pokud bude mít více připojených mikrofonů, tak si bude moci vybrat, který konkrétně chce používat, či jestli bude nahrávat zvuk z více najednou. Při práci s více vstupy je nutné normalizovat sílu snímání zvuku, čili výslednou hlasitost. Zvuk z více mikrofonů bude nahráván každý samostatně a tak i ukládán. Pokud se první pokus nebude líbit, může uživatel nahrát několik verzí a pak porovnávat, která je nejlepší. Ostatní potom jednoduše smaže. Každá nahraná část se bude na pevný disk ukládat samostatně. Tak se s těmito kusy bude později lépe pracovat a lze je využít pro operace s historií čili operace Undo a Redo.

Nahrané sekvence dabingu je nutné nějak třídit. K tomu se využije grafické rozřazení nahrávek do jednotlivých řádků a přiřazení barvy grafickým objektům. Barva či řádek díky tomu mohou reprezentovat nějakou postavu či skupinu postav ve videu. Toto rozdělení ulehčí následnou editaci např., když některé postavy už jsou nadabované a doděláváme poslední charakter, zabráníme tím nepřehlednosti při větším množství nahrávek.

Po nahrání sekvence bude možno s kusem nahraného dabingu provádět různé editace typu posun, střih, sloučení atd.

### 3.2.2 Editace

Nahrané sekvence se budou muset nějakým způsobem upravovat. Na to v aplikaci budou funkce rozdělávání, přesouvání, spojování atd. Tyto prostředky musí být implementovány, jinak by nebyl nástroj příliš použitelný. Díky těmto funkcím bude možné přesnější překrývání našeho hlasu s artikulací postav ve videu bez nutnosti znovu nahrávat. Všechny postupné editace budou ihned reflektovány jak metadaty, tak i reálnými soubory na úložišti. Což znamená že například při rozdělávání nahrávky se z původní stanou dvě. Tyto nové části budou vytvořeny a uloženy na pevném disku. Původní soubor nahrávky zde zůstane také, aby mohlo dojít k navrácení stavu.

### Posun

Při dabování se musíme trefit do chvíle, kdy postava zrovna mluví. To se nám prvním nahráváním nemusí přesně povést, a proto je nutné sekvenci posunout do požadované pozice. Posouvání bude možno tažením s objektem nahrávky či pomocí tlačítek pro posun o jeden pixel, jakožto minimální zobrazovací jednotky, kde pixel bude reprezentovat určitou dobu např. 1 pixel se rovná 100ms. Tato reprezentace bude záležet na přiblížení zobrazení nahrávek. Veškeré grafické prvky pro zobrazování časů nahrávky musí toto posouvání okamžitě reflektovat, aby byl uživatel přesně informován. Metadata, která aplikace bude později používat pro export a editaci, budou aktualizována až po dokončení posunu. Aby se zabránilo

nekonzistenci dat posunem nahrávek při přehrávání dabingu, bude během přehrávání možnost přesunu vypnuta.

### **Spojování a slučování**

Pokud například nahrajeme různé stopy z více mikrofonů a budeme s nimi spokojeni, můžeme tyto sekvence spojit do jedné, kvůli lepší manipulaci či přehlednosti. Stačí jenom tyto části vybrat a zmáčknout příslušné tlačítko. Pokud by byly sekvence od sebe odděleny částí, kde žádný námi nahraný dabing není, tak se zde vloží ticho. Při překrývání budou sloučeny, čili oba hlasy z nahrávek budou přítomny ve výsledném sloučení.

### **Střih**

Jde o rozdělování nahrané části na menší úseky. S těmi pak můžeme nakládat stejně jako s jednotlivými úseky. Výběr místa střihu bude možný pomocí dvou speciálních posuvníků na ose času. Akci lze využít např. pokud se vám polovina dabingu líbit bude, ale s druhou polovinou nejste spokojeni. V určeném místě nahrávku rozdělíte a špatnou půlku smažete.

### **3.2.3 Ukládání**

Všechno uživatelem nahraný dabing se bude rovnou ukládat na pevný disk do při vytváření projektu zvolené složky, kde bude přítomen i soubor pro metadata. Informace o tom, kde kus nahrané sekvence ve videu začíná, se budou ukládat do zvláštního textového souboru ve speciálním formátu XML. V tomto souboru budou i ostatní informace o zvukových souborech. Co bude konkrétně uloženo je popsáno v sekci 5.3. Po dokončení nahrávání a editaci je nutné ještě výsledek uložit. Uživatel si může postupně projekt ukládat sám nebo při vypnutí aplikace bude dotázán jestli chce práci uložit či nikoliv. Tento typ ukládání bude ukládat pouze metadata o nahrávkách. Jde tedy o typ téměř okamžitého uložení, jelikož nahrávky jsou už na pevném disku přítomny, a to i ty, které byly vytvořeny nějakou editací. Při vypnutí aplikace nahrávky, které nejsou přítomny v projektu, ale jsou na pevném disku, budou smazány, aby nedocházelo k zbytečnému zaplňování úložiště.

### **3.2.4 Export**

Provede se nad načteným projektem a nad aktuálním stavem, ve kterém se nachází, tak, jak zobrazuje grafické rozhraní. Při exportu dojde ke sloučení všech nahrávek a poté sloučení se zvukovou stopou ve videu. Tento výsledek bude ve zvláštním souboru, aby se neovlivnilo původní video a ani nahrávky dabingu. Díky tomu můžeme s projektem dále pracovat i po exportu.

### **3.2.5 Odstranění hlasu**

Jelikož půjde převážně o videa, která nemají oddělenou hlasovou složku od ostatních zvuků, bylo by vhodné hlas ze zvuku odstranit, aby váš dabing nebyl v těchto místech rušen hlasy původní zvukové stopy. Bohužel odstranění hlasu je samo o sobě velice obtížné, jelikož na stejné frekvenci, jako hlas, lze nalézt i jiné zvuky a nástroj tuto funkci nebude implementovat. K případnému odstranění hlasu bych doporučil aplikaci AudaCity, která tuto funkci podporuje. Aplikace ovšem umožní tzv. voiceovering viz sekce 2.1. Při exportu si můžete zvolit, na kolik procent bude původní zvuková složka ztlumena.

### 3.3 Video

Obrazová stopa hraje u dabingu menší roli. Umožňuje dabérovi se lépe trefit do emocí hlasu postavy, kterou dabuje. I zhodnocení celkového výsledku je lepší, když to zároveň vidíte a slyšíte. V aplikaci si budete moci vybírat místo, které dabujete, pomocí posunu ve videu. Je to mnohem lepší než hledání pouze pomocí audia, pokud nemáte text s časem, jelikož podle videa ihned poznáte, ve které části se nacházíte. Proto bude video mít klasické prvky přehrávání jako ostatní dostupné přehrávače. Což znamená pozastavování přehrávání, změna pozice odrážená posuvníkem, s kterým můžeme pozici také měnit atd.



## Kapitola 4

# Uživatelské rozhraní

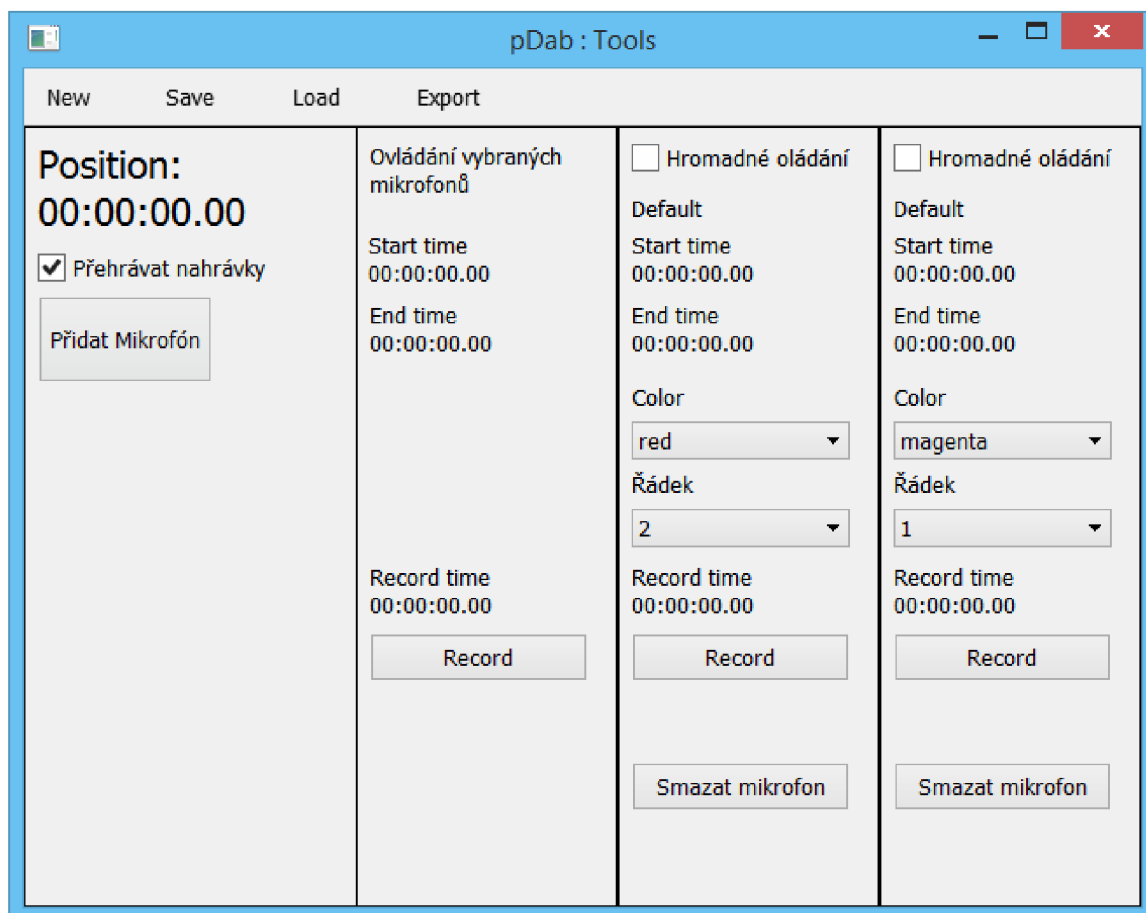
Vytvoření uživatelského prostředí pro editaci zvuku není jednoduchá věc, kvůli tomu, že zvuk musíme nějak zobrazit, aby se s ním dalo pracovat. K tomu bude využita vizualizace viz sekce 4.2. Jelikož s nástrojem budou nejspíše pracovat amatéři nebo poloprofesionální nadšenci, tak bude velký důraz kladen na jednoduchost uživatelského rozhraní. Ve finální verzi by uživatel neměl bloudit desítky minut v menu a na pracovní ploše a hledat operace nebo nastavení, které zrovna potřebuje. Bohužel práce se zvukem není zrovna nejjednodušší, což se na rozhraní jistě odrazí. Rozhraní je zaměřeno na co nejmenší potřebu přesného uživatelského vstupu, neboli různé zadávání textových hodnot. V našem případě například, kde se má nahrávka nacházet, v které místě ji rozdělíme atd. Vše se bude dít pomocí manipulace nějakých grafických objektů, třeba pomocí posuvníku po časové ose.

Nástroj se skládá ze tří oken, a to okno s videem, s vizualizovanými nahrávkami a hlavní okno s ovládáním mikrofónů. Hlavní důvod tohoto rozdělení je přehlednost, možnost volitelného uspořádání, a aby se mohlo každé okno dát na jiný monitor nebo projektor. To se týká především okna s videem, které musí lidé při soutěži vidět, aby se mohli podívat, co za scénku dabují, a vžít se do svých postav.

## 4.1 Hlavní okno

V tomto okně (viz obr. 4.1) se nachází lišta s ovládáním projektu. Konkrétně jde o tlačítka na vytvoření nového projektu, uložení, načtení a exportování výsledku.

Při vytváření nového projektu se otevře dialogové okno, kde si nastavíte název projektu, složku, kam se budou veškeré další soubory projektu ukládat a video, které hodláte dabovat. Vybrané video máte možnost si i okopírovat do složky projektu.



Obrázek 4.1: Hlavní okno

Hlavním obsahem hlavního okna je ovšem ovládání mikrofonů. Mikrofony jsou rozděleny do jednotlivých sloupečků. Při zapnutí aplikace může uživatel vidět pouze sloupeček pro hromadné ovládání a mikrofón, který je v systému nastavený jako výchozí. Pro přidání dalších mikrofonů je nutné použít tlačítko pro tuto akci. Po stisknutí se zobrazí dialogové okno, kde si vyberete mikrofón, jenž musí být připojen, podle systémového názvu a zadáte si vlastní pojmenování pro pozdější lepší orientaci, např jméno postavy. Každý mikrofón lze i odstranit, aby se ulehčila pozdější orientace.

Sloupec s hromadným ovládáním mikrofonů zapíná ty mikrofony, které mají zaškrtnuto povolení hromadného ovládání. Takto označené mikrofony již nejdou samostatně ovládat.

Ještě zde uživatel může vidět čas, kde se nachází ve videu a má zde také možnost vypnutí přehrávání nahraného dabingu spolu s videem pomocí zaškrťovacího políčka.

### 4.1.1 Mikrofóny

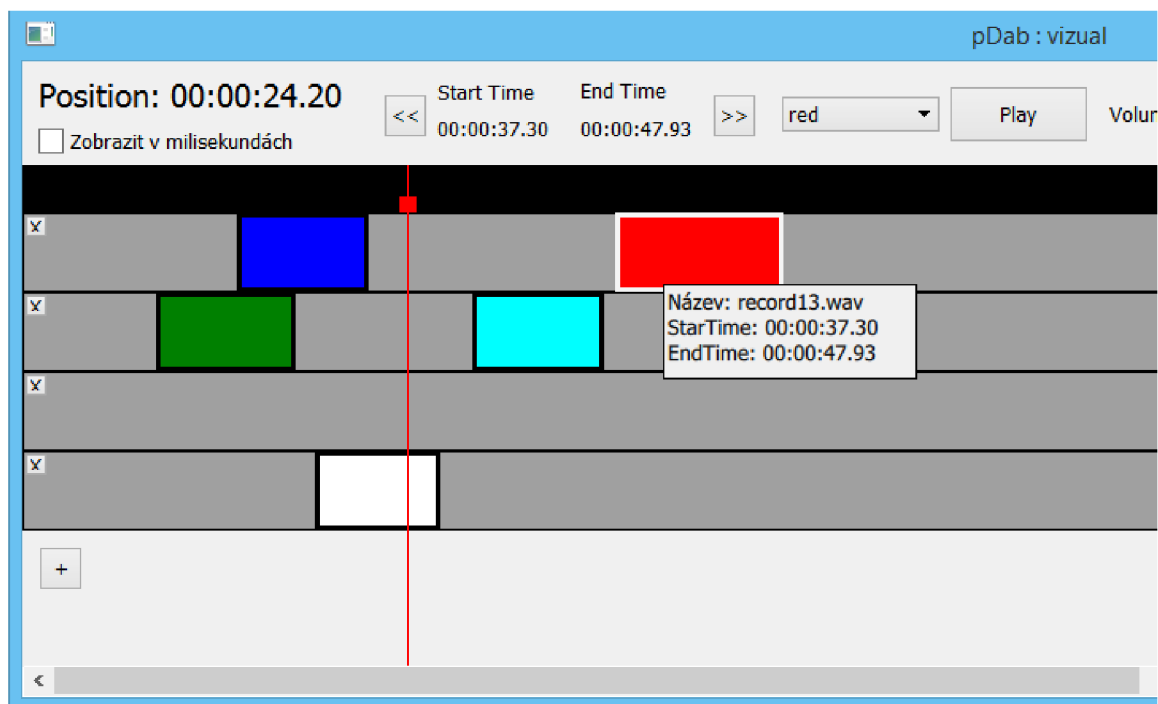
Každý sloupec pro ovládání mikrofónu je totožný. Popis z vrchu dolů je následující. Zaškrtnutá políčka pro to, jestli se bude mikrofón ovládat hromadně. Dále název který mu zadal uživatel, kde se po najetí myši zobrazí tooltip se systémovým názvem. Poté časy posledního nahraného dabingu. Konkrétně jde o začátek a konec nahrávky.

Pod těmito ukazateli času se nachází dva výsuvné seznamy. Jeden je pro výběr barvy, kterou si můžeme nahrávky dělit do různých kategorií či ji přiřazovat postavě. Druhý je pro výběr řádku, na kterém se nahrávka po dokončení nahrávání zobrazí. Zároveň, nelze nahrávat na jeden řádek z více mikrofónu. Toto omezení vzniklo z důvodů, aby se poté, co se zobrazí nahrávky v okně s vizualizací, nepřekrývali, a tím by došlo k zakrytí jedné či více z nich.

Následuje ukazatel délky aktuálně nahrávaného dabingu a tlačítka na zapnutí/stopnutí nahrávání a smazání mikrofónu.

## 4.2 Vizualizační okno

Zde nalezneme zobrazené naše nahrávky a také nástroje pro jejich editaci. Časová osa a samotné nahrávky jsou přítomny v posuvném pohledu, který mění svou velikost podle přiblížení a délky dabovaného videa. Ukázka části okna zobrazena na obrázku 4.2



Obrázek 4.2: Výřez okna s vizualizací

### 4.2.1 Vizualizace

Zvuk je tzv. vizualizovaný, což znamená, že jde vidět. Konkrétně jde o zobrazení pomocí barevných obdélníků, jejichž pozice a velikost odráží, kde se v časové ose nacházejí a jakou mají nahrávky délku. Po najetí myši na nahrávku se zobrazí, pomocí plovoucího okénka, informace o této nahrávce. Konkrétně jde o časy, kde se nachází její začátek a konec, a potom její název na pevném disku.

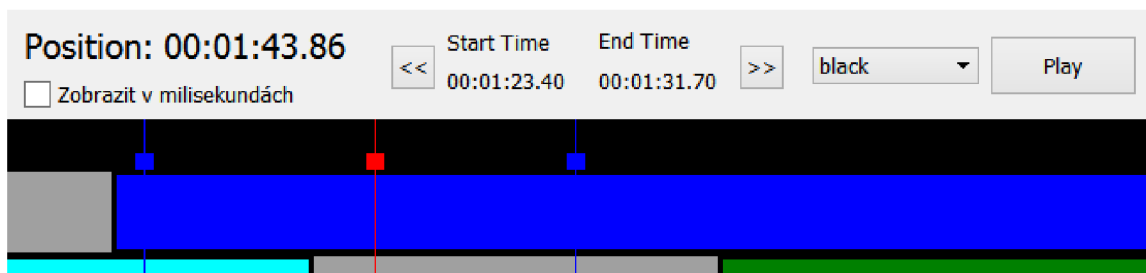
Tyto obdélníky jsou situovány v jednotlivých řádcích podle toho, kam si je uživatel vloží podle nastavení u mikrofónu. Řádky lze měnit i později, stačí pouze přetáhnout obdélník. Počet řádků není ničím omezen a lze je volně přidávat podle situace. Přidání řádku zajišťuje tlačítko nacházející se v levém dolním rohu tohoto okna. Nepotřebný řádek můžeme i odstranit tlačítkem, jenž je na začátku řádku. Nelze je ovšem odstranit, pokud se v řádku nacházejí nějaké nahrávky dabingu.

Táhnutím s nahrávkou se mění pozice, kde se nahrávka bude ve výsledném exportu nacházet. Pokus ovšem nahráváte dabing nebo pouze přehráváte video, nelze polohu táhnutím měnit, aby se zachoval kontext, v jakém čase se začnou přehrávat nahrávky. Měnit pozici lze i jiným způsobem, a to pomocí tlačítek označené šipkami. Tyto tlačítka slouží k přesnějšímu nastavení pozice. Po kliknutí se graficky posune o jeden pixel. Kde jeden pixel reprezentuje určitý počet milisekund podle toho, jaké máme zrovna přiblížení.

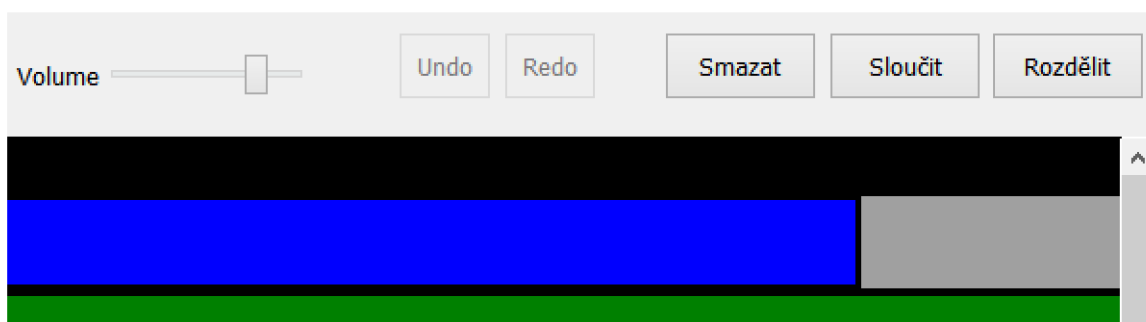
Výchozí měřítko je  $1 \text{ pixel} = 100 \text{ milisekund}$ . Pomocí tlačítka ctrl a kolečka myši lze přibližovat a oddalovat s tím, že existují 4 úrovně. Tyto úrovně přiblížení jsou odstupňované od  $1\text{px}/1\text{ms}$  až po  $1\text{px}/1\text{s}$ . Každé přiblížení měřítko 10x zmenšuje nebo zvětšuje a tato akce funguje, že přibližuje tzv. za myši. Což znamená, že po přiblížení máte v okénku kus, ve které se při otočení kolečkem nachází vaše myš. Díky tomu nemusíme opětovně hledat nahrávku.

## 4.2.2 Nástrojová plocha

Je zobrazena nad částí, kde se zobrazují nahrávky (viz obrázky 4.3 a 4.4). Na této ploše nalezneme vypsanou pozici na které se zrovna nacházíme. Pod tímto zobrazením času, je zaškrtnuté políčko pro změnu zobrazení času v milisekundách, jinak se čas zobrazuje ve formátu HH:MM:SS.mm. Vpravo vidíme detaily času vybrané nahrávky s tím, že pomocí tlačítek ji můžeme i posouvat. Následuje možnost měnit barvu již nahraného dabingu pomocí vysouvacího menu, kde je zobrazena barva v tu dobu vybrané nahrávky. Další je tlačítko pro přehrání vybrané nahrávky a nastavení hlasitosti přehrávané nahrávky. Jako další jsou tlačítka na operace Undo, Redo, mazání, rozdělování, spojování/slučování.



Obrázek 4.3: Detail nástrojové plochy s osami 1. část



Obrázek 4.4: Detail nástrojové plochy s osami 2. část

## 4.2.3 Časová osa

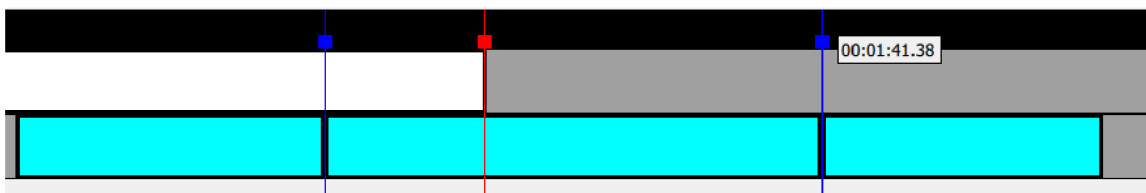
Je stěžejní pro orientaci v projektu. Zobrazuje se spolu s posuvníkem u videa, kde se časově nacházíte ve videu. Jde o malý červený čtvereček s čárou, která protíná všechny řádky, aby bylo snadno identifikovatelné, které nahrávky se v tomto čase vyskytují. Posuvník lze posouvat buď kliknutím na časovou osu, nebo táhnutím posuvníku.

## 4.2.4 Editace

S nahrávkou lze provádět základní editační operace. Na tyto operace jsou přítomny tlačítka v pravém horním rohu okna. Konkrétně můžeme nahrávku smazat, rozdělit nebo spojit dvě různé nahrávky. Veškeré tyto editace se ukládají a lze je zvrátit pomocí operace Undo nebo je obnovit pomocí Redo. Na tyto operace jsou v nástrojové liště speciální tlačítka.

## Rozdělování

Pokud chceme nahrávku rozdělovat musíme dát vědět kde. Pro tuto akci slouží dva ukazatelé, které se chovají stejně jako ukazatel času. Při spuštění aplikace nejsou viditelní a musí se zobrazit pomocí klávesy shift a kliknutím myši na časovou osu. Pokud kliknu na čtveřec a budu stále držet shift, ukazatel zmizí. Na těchto místech bude vybraná nahrávka rozdělena a neovlivní to ostatní nahrávky. Jelikož jsou přítomny dva tyto ukazatele lze nahrávku rozdělit na tři díly. Pokud bude v časovém rozmezí vybrané nahrávky pouze jeden ukazatel, tak se rozdělí jenom na dva díly. Po rozdělení se vytvoří nové grafické objekty na příslušném místě a barva se dědí z nahrávky, kterou jsme rozdělovali. Ukázka rozdělení na obrázku 4.5.



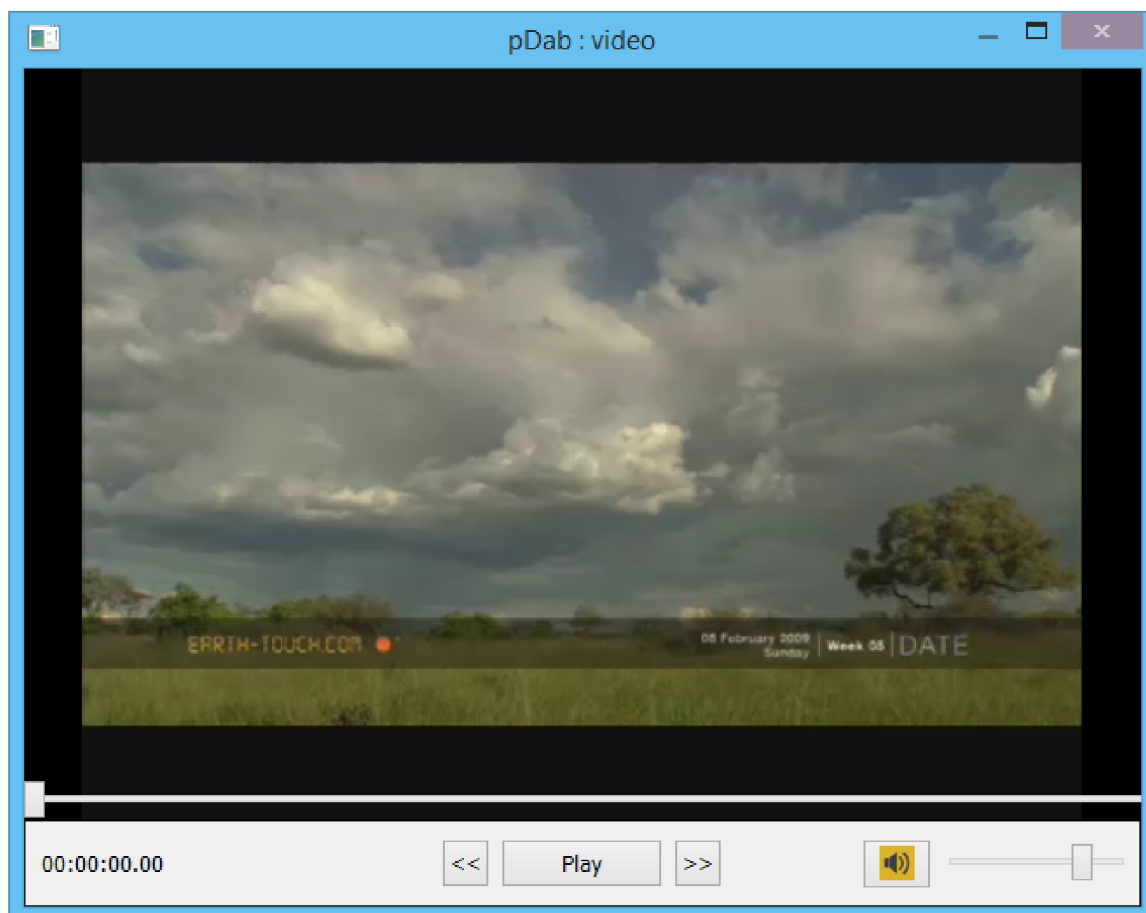
Obrázek 4.5: Nahrávky po rozdělení

## Spojování/slučování

Pokud bude uživatel chtít sloučit dvě nahrávky do jedné, stačí je vybrat. Výběr dvou nahrávek se provede pomocí kliknutí myši a během výběru druhé nahrávky musí být stisknutá klávesa shift. Vybrané nahrávky pro spojení musí být přítomny na jednom řádku. Po stisknutí tlačítka na spojení dojde ke dvěma scénářům. V prvním případě, pokud se nahrávky nepřekrývají, bude mezi ně vloženo ticho. V druhém, pokud se překrývají, budou sloučeny. Výsledné spojení se zobrazí tak, že se vytvoří nová vizualizace spojené nahrávky, která bude zobrazena tak, aby reprezentovala začátek první a konec druhé nahrávky. Tato vizualizace bude mít barvu jedné z vybraných nahrávek pro spojení.

### 4.3 Video okno

Zobrazení videa (viz obr. 4.6) je pro soutěžní či amatérské dabování stěžejní, abyste se mohli dobře časově synchronizovat s postavou a tzv. vžít se do ní. V tomto okně se tedy nachází samotné video a pár prvků pro jeho ovládání. Jde o pozastavení a obnovu přehrávání, ovládání hlasitosti. Můžete také krokovat video pomocí tlačítek kolem přehrávání. Velikost kroku je odvíjena od přiblížení v okně vizualizace.



Obrázek 4.6: Okno s videem

## Kapitola 5

# Implementace

Jako dnes téměř každá aplikace je i tato rozdělena na část backend a část frontend. První část je programována v klasickém C++ s rámcem Qt verze 5 a druhá využívá grafické standardní knihovny Qt Quick verze 2 pro psaní QML aplikací. V každé následující sekci budou proto popsány obě tyto části, jelikož se od sebe hodně liší, i když jsou částečně propojené.

Hlavní a jedinou částí Qt je třída `Dab`. V ní jsou všechny funkce potřebné pro fungování aplikace. Jde např. o funkce ukládání a načítání, střih zvuku, finální export atd. Hlavní důvod vytvoření jenom jedné třídy je, že pokud chceme volat funkce z Qt, tak musíme vytvořit objekt požadované třídy a vložit ji jako kontext do objektu QML, kde chceme funkce objektu volat.

QML, které implementuje uživatelské rozhraní, je rozděleno do více souboru. Každé okno má svůj vlastní soubor grafického rozhraní kromě dialogových oken. Většina JavaScriptových funkcí a volání C++ kódu, který se má provádět při nějaké akci uživatel, např. když klikne na tlačítko, se nachází v souboru `main.qml` reprezentující hlavní okno.

Na obrázku 2.2 je zobrazena struktura QML prvků. V projektu je použit modul Qt Quick 2.x jakožto základní modul, který musí být přítomen. Dále Window pro podporu zobrazených oken. V Controls se nacházejí prvky typu tlačítko, posuvník, zaškrťovací políčko, posuvný pohled atd. Pokud bychom tento modul nepoužili, veškeré prvky by bylo nutno programovat znovu, což je zbytečné, když již existují. Posledním modulem je Dialog. Tento modul obsahuje různá dialogová okna typu uložit soubor, načíst soubor nebo pro různé odpovědi typu ano/ne.

Hlavním komunikačním prostředkem v Qt jsou signály a sloty, neboli funkce, která se vyková při spuštění signálu. Veškeré signály z grafického prostředí zpracovává JavaScript a případně volá funkce z třídy `Dab`. Výhodné je, že můžete na jeden signál napojit více funkcí.

Pokus chceme volat z QML nějaké členské funkce, které máme napsané v C++, musíme vytvořený objekt reprezentující naši třídu vložit do kontextu objektu QML. Členské funkce v této třídě ještě navíc musí být označené jako `Q_INVOKABLE`. V případě tohoto projektu jde o objekt `dab` třídy `Dab`. Je tu i další možnost, a to napojit signály z QML na sloty v objektu C++. Toto propojení je použito pouze při volání exportu, aby mohlo grafické rozhraní reflektovat postup exportu. Bohužel i přesto grafické rozhraní tzv. zamrze přes všechnu mou snahu.



## 5.1 Video

Ať už se zakládá nový projekt nebo se načítá nějaký uložený, musí se provést načtení videa. Tato akce musí být provedena jako první, jelikož se od času videa odvíjí délka plochy, kde se zobrazují nahrávky, čili i velikost zobrazení těchto nahrávek. Po načtení videa je stopnuté, pouze se video posune na nultou milisekundu, aby se zobrazil první snímek obrazu. Video se ovládá pouze v QML a jde o prvek rámcu QtAV jménem `Video`, které je vloženo jako potomek prvku `Rectangle` a v něm se video vykresluje. Celá definice okna s videem je v souboru `SVideoForm.ui.qml`, kde "ui" naznačuje, že jde pouze o uživatelské rozhraní. V těchto typech souborů nejsou dovoleny žádné JavaScriptové funkce. Video je stěžejním synchronizačním prvkem aplikace. Od jeho pozice se odvíjí, kde nahrávání začíná a kde končí. Podle toho má poté nahrávka svoji počáteční pozici a velikost. K této synchronizaci se používá signál `PositionChanged`. Při obsluze toho signálu, který se aktivuje přibližně každou půl sekundu, se synchronizují ukazatelé času a ukazatelé na časových osách u videa a ve vizualizačním okně.

## 5.2 Zvuk

Hlavní část aplikace je správa a editace zvuku respektive nahraného dabingu. K editování zvuku je využíván nástroj FFmpeg, který se spouští jako externí program s danými parametry, co má provést. O jaké parametry jde je, pro jednotlivé akce popsáno v sekci 5.4.

Mikrofóny se načítají do svého seznamu, aby se s nimi dalo dobře manipulovat. Při spuštění je načtený pouze výchozí mikrofón v systému. K ovládání mikrofónů se používá třída `QAudioRecorder`. Z ní se vytvoří instance pro každý mikrofón zvlášť. Tomuto objektu musíme jednotlivé mikrofóny přiřadit. To se děje pomocí jejich systémového jména, které vytáhneme z funkce `QAudioDeviceInfo::availableDevices(QAudio::AudioInput)`, která vrací seznam s objekty informací o připojených vstupních zvukových zařízeních. Jedna vlastnost tohoto objektu je i systémové jméno mikrofónu. Jelikož chceme jenom vstupní zařízení, musíme jako parametr použít `QAudio::AudioInput`. Objektu mikrofónů lze nastavovat i různé parametry jako například počet kanálů čili stereo, mono.

Další nastavitelnou vlastností je vzorkovací kmitočet, který je v aplikaci nastaven na 44100 Hz, jakožto nejpoužívanější a nejrozšířenější. Většina vestavěných mikrofónů v noteboocích tento kmitočet podporuje. S tímto kmitočtem zachytíme rozsah zvuku od 0 po 22000 Hz, což je přibližný rozsah průměrného lidského sluchu. Proto podle vzorkovacího teorému respektive Shannonova teorému, musíme tedy zvolit dvakrát větší vzorkovací frekvenci, čili 44100 Hz.

Dalším důležitým faktorem k nastavení je výběr kodeku. Jelikož aplikace cílí na multiplatformnost, byl vybrán nekompresní kodek PCM<sup>1</sup> kvůli jeho podpoře napříč platformami. Jako kontejner je zvolen WAV<sup>2</sup>, který se s tímto kodekem často využívá.

Vytváření grafického rozhraní pro mikrofony je na rozdíl od většiny prvků dynamické. Zobrazení je definované v `microphoneViz.qml` a funkce `addNewMicrophone` pro vytvoření grafického objektu se nachází v souboru `main.qml`. V této funkci se tedy vytváří nový grafický objekt a nastavují se některé parametry jako, kde se má v okně zobrazit, přiřazení jména do textového pole. Také se zde napojují signály zapnutí a vypnutí nahrávání, smazání mikrofónu atd. na funkce, které tento signál obslouží.

<sup>1</sup><http://wiki.multimedia.cx/index.php?title=PCM>

<sup>2</sup><http://wiki.audacityteam.org/wiki/WAV>

### 5.2.1 Nahrávání

Po zapnutí nahrávání se zavolá funkce v QML `startBoolRecord`, která má jako parametry identifikační číslo mikrofonu, číslo řádku a barvu, kterou si uživatel nastavil. Funkce poté funguje ve dvou větvích. Jestli zapíná či vypíná nahrávání. Podle toho omezí grafické prvky nebo je opět aktivuje. K omezení dochází, aby uživatel nezavinil nekonzistenci aplikace. Pokud se zapíná nahrávání, ověří se, jestli je mikrofon stále připojen a dále, jestli už nějaký mikrofon nenahrává do řádku, který si zvolil uživatel.

Pokud je vše v pořádku, tak se nahrávání zapne a přejde se k vytváření grafického objektu nahrávky viz sekce 5.2.3. Při nahrávání se pro vytvoření vizualizace použije funkce `startRecord`, ale ještě se nezobrazí. To až po zavolání funkce `stopRecord`, kdy se doplní čas konce nahrávky, abychom mohli vypočítat délku.

V zadní části C++, kde probíhá samotná manipulace s mikrofony na základě funkcí volaných z QML, jde pouze o zapínání a stopování nahrávání s tím, že místo tvorby grafických prvků nahrávek se zde ukládají metadata do speciálního asociativního seznamu na bázi hashovací tabulky `QHash<int, t_recordDab> listOfRecords`.

### 5.2.2 Přehrávání

Okamžité přehrávání námi nahraných dabingů je řešeno následovně. Jelikož se může stát, že bude více dabingů překrytých přes sebe, tak potřebuje více objektů k přehrávání. Zde je opět přítomen seznam, jehož položky jsou objekty třídy `QtAV::AVPlayer`. Pokud měníme pozici v projektu, tak musíme opět znovu načíst, co a kdy se bude přehrávat. Na to je připravena funkce `newPosition`. Zde se zastaví časovač a stopnou všechny přehrávané nahrávky. V další funkci `playNextRecord` se projede celý seznam se strukturami nahrávek a zjistíme jestli nějaká není přítomna v nové čase. Pokud ano, nastavíme ji, že se bude přehrávat, a to od času, pozice mínus počáteční čas nahrávky. Tímto dosáhneme, že se nahrávka bude přehrávat od správné pozice. Pokud je nahrávek na nové pozici více, tak hledáme v seznamu další volné objekty, které by ji přehráli a pokud nejsou, tak vytvoříme nový a nastavíme, od kterého místa se má přehrávat.

Pokud kontinuálně bez přerušování přehráváme, tak se další spuštění následující nahrávky řeší pomocí časovače. U něho podle aktuální pozice a počátečního času další nahrávky určíme, kdy má zapnout přehrávání. Funkce, která se zavolá po vypršení časovače, je `playNextRecord_slot`. V této funkci se také volá členská funkce `setPlayNextRecord`, která ihned znovu nastaví časovač na další nahrávku. Je zde i řešeno, pokud nahrávky začínají na stejné milisekundě, ale pravděpodobnost tohoto jevu je minimální.

Při všech těchto operacích je ale diskutabilní, jak jsou rychlé, jelikož video se dále přehrává, tak pozice daná těmito funkcím jako parametr je již zastaralá. Při testování se na nějaké větší zpoždění neprojevovalo, ale zároveň se nepracovalo s desítkami či stovkami nahrávek, více v kapitole 6.

### 5.2.3 Zobrazení

To, jak vypadá vizualizovaná nahrávka, je definováno v souboru `recordViz.qml`. Jde o prvek `Rectangle`, který obsahuje prvek `MouseArea`. Díky němu lze nastavit, aby se dalo objektem táhnout po ploše k tomu určené. Vytváření těchto objektů je dynamické, jelikož nikdy nevíme, kolik jich bude, a pro každou situaci je kód trochu jiný. Při vytváření grafického objektu nahrávky se nastavuje pozice a velikost, jakožto i barva, ve kterém řádku se nachází atd. Veškeré vytvořené vizualizační objekty se v QML ukládají do asociativního

pole `listRecordItem`. Délku obdélníku vypočítáme následujícím vzorcem.

$$\frac{\text{čas konce nahrávky} - \text{čas začátku nahrávky}}{\text{počet milisekund na pixel}} \quad (5.1)$$

Objekt se zobrazuje v takovém objektu řádku, který je mu nastaven jako vizuální rodič, tedy ten ve kterém se zobrazuje a vůči kterému se poté vypočítává jeho pozice. Při vytváření také propojujeme signály objektu s funkcemi, které se při emitování signálu provedou.

Pokud s objektem táhneme, ihned se nám aktualizují jeho časy ale pouze ty, které se zobrazují. Teprve po dokončení pohybu se nám zaktualizují metadata ve struktuře dané nahrávky.

### 5.3 Metadata a ukládání

Je nutné si informace o nahrávkách nějak ukládat. Proto je vytvořena struktura `recordDab`, která obsahuje tyto položky. Dva časy, ve kterém nahrávka začíná a končí, uložené v jednotkách milisekund. Jako typ proměnné pro tyto časy byl zvolen `unsigned int`, do kterého se dá v milisekundách uložit přes 1000 hodin, což by mělo pro účely dabingu více než stačit. Dále ve kterém se nachází řádku, jakou barvu má při zobrazení, její jména na pevném disku a nakonec jednoznačné identifikační číslo. Tyto jednotlivé struktury se ukládají jako prvky `QHash<int, t_recordDab> listOfRecords` seznamu. `QHash` byl zvolen, jelikož není lineární jako klasický seznam, kde jsou položky hned za sebou, ale můžeme položky vkládat na jakoukoliv pozici a hashovací funkce zařídí její uložení na pozici. Jde vlastně o asociativní pole, kde máme klíč asociovanou k hodnotě. To je vhodné, pokud vytvoříme několik nahrávek, a poté jich pár smažeme, tak stále identifikační číslo nahrávky odpovídá indexu v `QHash` seznamu. Při manipulaci s nahrávkami, např. jejich posunutí v rámci videa, se mění pouze údaje v dané struktuře.

Tyto metadata se na vyžádání uživatele ukládají do souboru `jmenoprojektu.pDab`. Jde o soubor s XML strukturou, která je následující.

```
<?xml version="1.0" encoding="UTF-8"?>
<pDab>
  <nameOfProject>ore2</nameOfProject>
  <projectPath>D:/cesta/k/projektu</projectPath>
  <videoPath>D:/cesta/k/projektu/input.mkv</videoPath>
  <record id="0">
    <nameOfRecord>record0.wav</nameOfRecord>
    <startTime>23</startTime>
    <color>red</color>
    <row>1</row>
    <endTime>3512</endTime>
  </record>
  <record ....>
    ....
  </record>
  ....
</pDab>
```

Při načítání metadat do aplikace je to podobné jako při nahrávání. Akorát data, která načteme ze souboru, uložíme nejdříve do seznamu. Teprve potom se začne vykonávat QML funkce, která vytvoří vizualizaci těchto nahrávek pomocí načtených dat.

## 5.4 Editace

Po vybrání příslušné nahrávky či nahrávek a kliknutí na tlačítko operace, kterou chceme provést se zavolá funkce v backendové části. Ta pomocí FFmpeg rozdělí či sloučí nahrávky a reflektuje změny. Původní nahrávky smaže, ale pouze metadata, tedy položky v seznamu. Nově vzniklé nahrávky uloží a v části QML vytvoří jejich grafické objekty a původní smaže. Veškeré původní nahrávky použité při těchto operacích jsou stále přítomny na disku pro případnou negaci operace pomocí Undo.

### 5.4.1 Slučování a spojování

Slučování, když se dvě nahrávky překrývají, či spojování, kdy je mezi nimi mezera, obsluhuje členská funkce `mergeTwoFiles`. Jako parametry má dvě identifikační čísla nahrávek, které se budou slučovat nebo spojovat.

#### Slučování

Když se nahrávky slučují, tak FFmpeg umí pouze, že lze slučovat jednu do druhé s tím, že velikost výsledku se řídí např. podle té nejdější, a obě začínají od začátku. Proto se tato operace skládá z několika dílčích. V našem případě sloučení lze uskutečnit pouze, pokud jsou obě nahrávky stejně dlouhé a mají stejný začátek. Proto musíme analyzovat časy obou nahrávek. Po této analýze vytvoříme ticho a spojíme ho s nahrávkami tak, aby byly stejně dlouhé. Konkrétně to znamená, že výsledek má počáteční čas té nahrávky, která ho má dříve, a koncový čas té, která ho má později. Po tomto spojování s tichem máme dvě stejně dlouhé nahrávky, které není problém spojit.

#### Spojování

Při spojování, kdy mezi nahrávkami je mezera, tak tuto mezeru zaplníme tichem, a poté tyto tři soubory jednoduše spojíme.

### 5.4.2 Rozdělování

Pro rozdělení je připravena členská funkce `splitFile`, která jako parametry má identifikační číslo nahrávky, dva časy, kde bude nahrávka rozdělena a booleovskou hodnotu, jestli druhý čas použít. Časy se berou ze speciálních ukazatelů. Poté pomocí FFmpeg nahrávku rozdělíme dle časů v parametrech. Pro rozdělení musíme určit, od jakého místa v nahrávce chceme rozdělení provést a jak dlouhá bude oddělená část. Při rozdělování se vytváří úplně nové zvukové soubory, které jsou tedy vyříznuty z původního.

### 5.4.3 FFmpeg

Ukázka a popis některých parametrů použitých v projektu [6].

Vytvoření ticha:

```
ffmpeg -filter_complex "aevalsrc=0:duration=10" -ac 2 -b:a 32k cesta/silent.mp3
```

Spojení dvou souborů:

```
ffmpeg -i cesta/silent.mp3 -i cesta/record0.wav -filter_complex  
concat=n=2:v=0:a=1 cesta/record1.wav"
```

Sloučení souborů:

```
ffmpeg -i cesta/record0.wav -i cesta/record1.wav -filter_complex  
amix=inputs=2:duration=longest cesta/record.1.wav"
```

Rozdělení:

```
ffmpeg -i record0.wav -ss 0 -t 20 record1.wav
```

Vysvětlení některých parametrů:

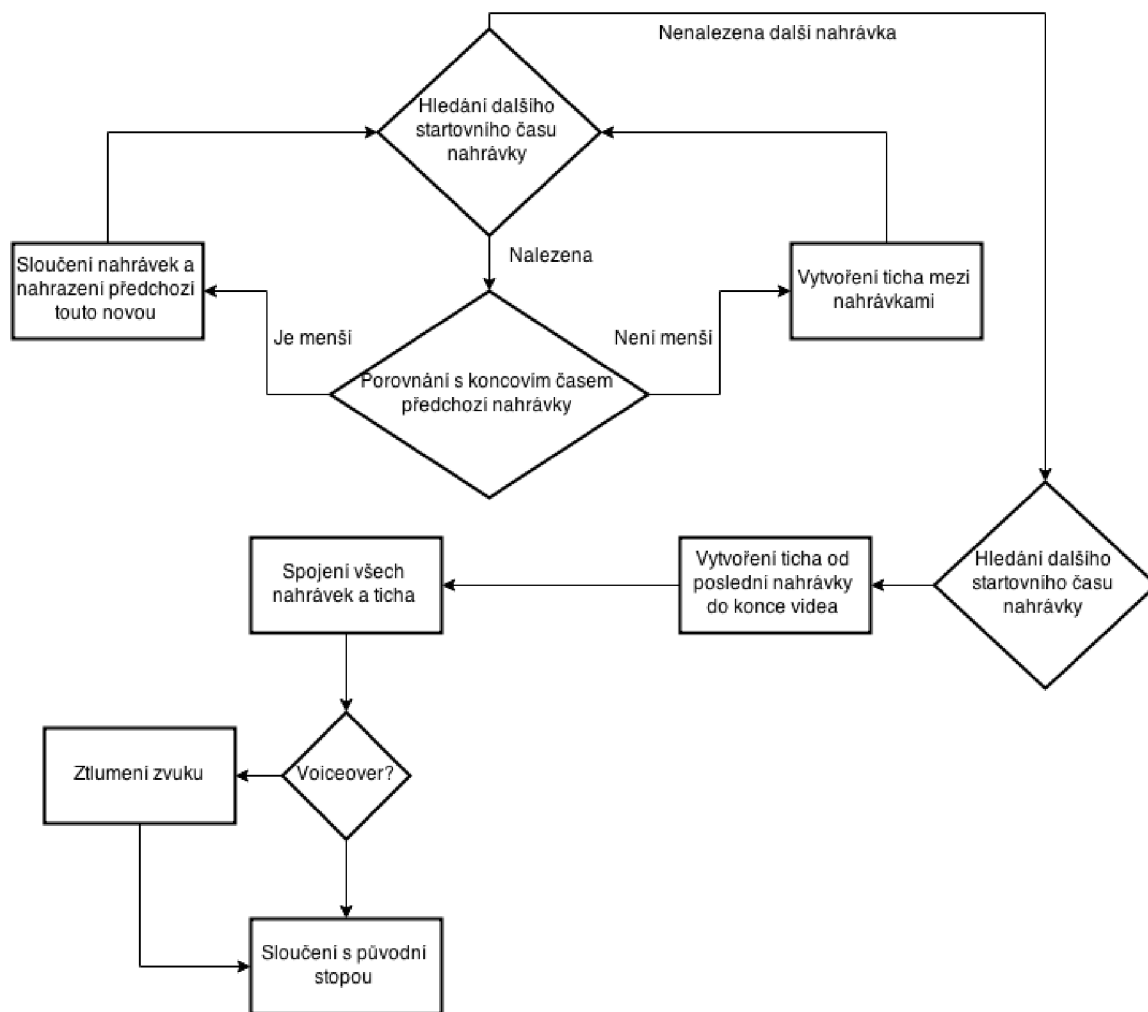
- `-i cesta` - Vstupní audio/video soubory
- `duration` - Délka generovaného ticha ve formátu HH:MM:SS[.mm] nebo v sekundách
- `-ac x` - Počet kanálů (mono, stereo), kde `x` je počet
- `-b:a x` - Bitů za sekundu pro zvukovou složku, kde `x` je počet
- `concat=n:x:v=y:a=z` - spojení vstupních souborů, kde `x` je počet, `y` je která složka videa, `z` je složka audia
- `amix=inputs=x:duration=longest` - sloučení vstupů, kde `x` je počet
- `-ss x` - Počáteční čas, kde `x` je čas ve formátu HH:MM:SS[.mm] nebo v sekundách. Používá se v kombinaci i s jinými parametry, v projektu pro rozdělování.
- `-t x` - Délka nějaké operace, kde `x` je čas ve formátu HH:MM:SS[.mm] nebo v sekundách. Používá se v kombinaci i s jinými parametry, v projektu pro rozdělování.

#### 5.4.4 Undo a Redo

Pro provádění těchto operací potřebujeme znát původní stavy metadat a zanechávat po dobu běhu operace všechny nahrané zvukové soubory i ty, které vzniknou při editacích. K tomu byly vytvořeny dva seznamy, které obsahují další seznamy. První seznam obsahuje seznamy se strukturami nahrávek. Čili každý tento vnitřní seznam představuje, co bylo smazáno a co při použití jedné z operací Undo, Redo bude opět obnoveno. Druhý seznam má v sobě opět seznamy, které obsahují číslo nahrávek, které mají být při operacích smazány. V těchto seznamech se program hýbe podle indexu `m_undoRedoPosition`. Při provedení nějaké editační operace nebo smazání nahrávky se k indexu přičte jedno a uloží se struktury smazaných a čísla nových nahrávek do seznamů. Pokud použijeme Redo operaci také přičítáme k indexu, ale prohodíme obsahy seznamů. To znamená, že struktury před smazáním uložíme do příslušného seznamu, a ty které obnovíme, tak jejich identifikační čísla také uložíme, kam patří. Při operaci Undo odečítáme od `m_undoRedoPosition` a opět prohazujeme obsahy seznamů, jako u Redo.

## 5.5 Export

Po nahrání veškerého dabingu a provedení dodatečných úprav, může uživatel projekt exportovat. Algoritmus exportu funguje, jak je popsáno v diagramu 5.1.



Obrázek 5.1: Diagram exportu

Algoritmus postupně prohledává strukturu nahrávek podle počátečních časů. Po nalezení dalšího počátečního času následující nahrávky, se porovnávají dvojice časů, počáteční nově nalezené a koncový předchozí nahrávky. Tím se zjistí, jestli se nahrávky překrývají či nikoliv. Po tomto porovnání dochází, buď k vytvoření ticha a zaplnění mezery mezi nahrávkami, nebo k sloučení dvou nahrávek. Jak se tyto operace provádějí a proč, je vysvětleno v sekci 5.4. Všechny vytvořená ticha a nahrávky vzešlé ze slučování se ukládají do seznamu postupně, aby se zachovalo pořadí, ve kterém mají být spojeny. Po vytvoření audia o délce dabovaného videa se sloučí se zvukem ve videu. Pokud si uživatel zvolí, dojde ještě k utlumení hlasitosti a vytvoření kopie videa s tlumeným audiem a s touto kopií se teprve sloučí náš dabing. Export je přítomen v novém video souboru.

Chtěl jsem vytvořit i různé možnosti pro export jako výběr zvukové stopy, pokud jich má video více. Ale QtAV s použitím komponenty v Qt Quick nedisponuje možností zjistit a volit zvukové stopy. O tomto problému jsem informoval tvůrce, který mi odpověděl, že tuto funkci klidně dodělá nebo že mám použít komponentu v Qt Widgetech, která toto podporuje.

## Kapitola 6

# Testování

Testování, které jsem uskutečnil před soutěží, probíhalo nejdříve osobně, vždy zkoušením nových naprogramovaných vlastností aplikace. Bohužel tento typ testování není nejvhodnější a dokáže odhalit pouze některé chyby, jelikož jako programátor víte, na co neklikat a v jakém pořadí vše používat a intuitivně to tak také testujete, i když se snažíte vytvářet všechny možné kombinace. Proto v rámci odhalení co možná největšího množství chyb, byla aplikace ukázána a poskytnuta k vyzkoušení několika dalším lidem. Při jejich sledování jsem si zaznamenával chyby, jež našli, nebo i rovnou opravoval na místě a opětovně nechal otestovat. Nejčastějšími problémy byly, neaktualizování proměnných při reakci na uživateli akce, přehrávání uživatelova dabingu, s čímž jsem se potýkal celkem dlouho a dále také celkové propojování Qt zadní části s QML přední části.

### 6.1 Multiplatformnost

I když veškeré technologie použité v tomto projektu jsou multiplatformní, tak při zkouškách na operačním systému Arch Linux není program funkční. Hlavní rozdíl mezi platformami je ten, že systém Windows používá při zadávání argumentů, externě spouštěných programů, jiný způsob než systém na bázi Linuxu. Další rozdílná část je různá interpretace cesty k souboru. Všechny tyto problémy lze vyřešit definováním různých kódů pro různé operační systémy.

### 6.2 Soutěž

Softwarový nástroj, jenž byl v rámci této bakalářské práce vypracován, je určen jako softwarové řešení pro soutěž v rychlodabingu na festivalu Animefest<sup>1</sup>. V minulých letech už tato soutěž jednou uskutečněna byla, ale z důvodu složitosti postprodukce byla ukončena. V letošním roce zatím nedošlo k soutěži, ale pouze k vyzkoušení, jestli je tato soutěž technicky možná a na jakém konceptu by mohla nadcházející roky fungovat. Při této testovací zkoušce jsem byl přítomen a aplikaci ovládal. Díky tomu jsem si mohl poznačit veškeré chyby a nedostatky aplikace a připomínky testujících a pořadatelů, které se musí do příštího roku vyřešit.

K dabingu byli využity dva bezdrátové mikrofóny napojené do mixážního pultu a z něho klasickým 3,5mm jackem do notebooku. Bohužel tímto způsobem jsem, ale přijímal pouze jeden zvukový proud, který mixážní pult z mikrofónů udělal. Čili, nebylo využito možnosti

---

<sup>1</sup><http://www.animefest.cz/>

nahrávat zároveň z několika mikrofonů a i editační schopnosti byli tímto omezeny, jelikož hlasy byli smíchány. Během zkoušky bylo ale zjištěno, že tato indispozice v soutěži vůbec nevadí. To proto, že hlasy se ve většině případů nepřekrývají a jdou tedy poté v aplikaci rozdělit a editovat zvlášť.

### 6.2.1 Chyby a nedostatky

Při testu bylo odhaleno i několik chyb a nedostatků. Jedna z nich je i velice zásadní. Šlo o chybu, kdy dabovaná scéna měla jenom několik sekund. Aplikace však v tu chvíli neměla implementováno ukončení nahrávání při dosažení konce videa. To samozřejmě způsobovalo nekonzistenci dat. A to tím, že bylo naprogramováno, že při přehrání videa až do konce se začne opět přehrávat znovu. Poté při zastavení nahrávání se předává čas, kde se zrovna ve videu nacházíme. Při zkoušce tento čas byl dříve než čas začátku.

Dalším problém byl z hlediska uživatelského prostředí. Šlo o ovládání videa, které je v okně s videem. Toto okno se však nacházelo na promítacím plátně, čili člověk který ovládá aplikaci musí na plátno vidět také, pokud, jako v testovacím případě, nemá zobrazeno to co je na plátně i na jiném monitoru, na který bez problému vidí.

Následující problém, který se vyskytl, je opomenutí přepnutí mikrofonu z interního na externí. Tím došlo ke dvěma zbytečným pokusům dabingu. Tato chyba je ale spíše uživatelská, že si uživatel při zapnutí nezkontroloval, který mikrofon je v rozhraní zobrazen a který se tedy bude zapínat. Zmatení nejspíše způsobuje že v aplikaci je již zobrazen výchozí mikrofon systému. Problém se dá řešit, že nebude zobrazený žádný mikrofon nebo si uživatel musí vybrat při zapínání aplikace nebo vytváření, načítání projektu, pouze pokud je detekováno více mikrofonů.



# Kapitola 7

## Závěr

Výsledkem této práce je funkční nástroj pro rychlodabing, který cílí na obyčejné uživatele, kteří si chtějí zkusit dabing nebo například přidat komentář k rodinnému videu, aniž by se museli učit s nějakými složitějšími programy. Je ale i vhodný k soutěžím v dabingu na tématických festivalech. Proto jsem se při tvorbě zaměřil na jednoduché uživatelské prostředí. Použití zmíněných technologií bylo celkem vhodné, hlavně kvůli jejich jednoduchému používání. Kromě kompilace QtAV pro systém Windows se kterým jsem měl problémy. Akorát použití Qt Quick a jazyka QML pro vytváření grafického rozhraní nebylo úplně to nejlepší z hlediska komunikace se zadní částí C++. Na samotnou tvorbu uživatelského rozhraní bylo velice intuitivní a výborně použitelné, ale právě ta komunikace s C++ nebyla úplně nejlepší a v klasických Qt Widgetech by to šlo nejspíš lépe. Tam by šel udělat grafický prvek, ve kterém by byl i objekt pro nahrávání z mikrofónů. Tím by odpadlo různé přeposílání dat, z QML do C++, jaký mikrofón uživatel chce použít atd. Tento názor je však osobní a hlavně způsoben počáteční nezkušeností s používáním této technologie. Proto bych do budoucna chtěl tento nástroj přepsat právě s použitím Qt Widgetů.

### 7.1 Rozšíření a budoucnost projektu

Nástroj se při testování na festivalu líbil a lidé, kteří ji testovali jako soutěžící byli s výsledky spokojeni. Příští rok bude tedy použit znovu, ale tentokrát už v klasické soutěži. Musí se tedy opravit všechny nedostatky, které byly při testování zjištěny, aby se dal nástroj plně využít. Dále se samozřejmě bude ještě testovat, aby se co nejvíce nástroj vyladil a byl co nejvíce bezproblémový.

Rozšíření by bylo vhodné např. u možnosti exportu, jako výběr zvukové stopy, kterou chceme dabovat, či vložení našeho dabingu jako úplně novou stopu. Jako další by mohla aplikace podporovat zobrazení titulků k podpoře dabingu. Uživatel by si tak nemusel tisknout scénář či si ho nemusel otevírat v nějaké další aplikaci. Navíc se musí předělat ovládání videa. Nejlepší možností bude vytvořit další sekundární ovládání např. v hlavním okně, kde se ovládají mikrofóny. Také vkládání, třeba už vytvořeného dabingu či nějakých doprovodných zvuků do projektu, by touto nástroji zajímavě rozšířilo možnosti. I vyřešení problému s funkčností na více druhů operačních systémů by mělo být vyřešeno.

# Literatura

- [1] Voice-Over Translation - Funny Way of Translation. online, 2014 [cit. 2015-05-12].  
URL <http://www.usatranslate.com/voice-over-translation-funny-way-of-translation/>
- [2] Qt 5 Tutoriáli a příklady. online, 2015 [cit. 2015-05-12].  
URL <http://doc.qt.io/qt-5/qtexamplesandtutorials.html>
- [3] Bocklage-Ryannel, J.; Thelin, J.: Qt5 Cadaques. online, 2014 [cit. 2015-05-12].  
URL <http://qmlbook.github.io/>
- [4] NOVOTNÁ, E.: *Typy filmového překladu (dabing, titulky) s příklady z animované verze filmu Robin Hood [online]*. Diplomová práce, JIHOČESKÁ UNIVERZITA V ČESKÝCH BUĎĚJOVICÍCH, Pedagogická fakulta, 2013 [cit. 2015-05-11].  
URL <http://theses.cz/id/6kuxjw/>
- [5] Pehlivanian, A.; Nguyen, D.; Baše, O.: *JavaScript okamžitě*. Brno : Computer Press, 2014, ISBN 978-80-251-4163-2.
- [6] Vývojáři FFmpeg: FFmpeg Dokumentace. online, 2015 [cit. 2015-05-12].  
URL <https://www.ffmpeg.org/ffmpeg-all.html>

# Příloha A

## Obsah CD

- Zdrojové soubory nástroje v adresáři `/src/`
- Návod (Příloha B) na použití, instalaci a kompilaci nástroje v adresáři `/manual/`
- Zkompilovaná aplikace pro systém Windows v adresáři `/tool/`
- Technická zpráva ve formátu PDF v adresáři `/thesis/`
- Zdrojové soubory technické zprávy v adresáři `/thesis_latex/`

# Příloha B

## Manual

### Použití

#### Mikrofony

Po zapnutí aplikace je načtený výchozí mikrofon systému. Pokud chcete přidat další, stačí stisknout na tlačítko Přidat mikrofon. Poté se vám otevře dialogové okno, kde si můžete nastavit vaše pojmenování mikrofonu a jaký mikrofon chcete přidat. Přidaný mikrofon se zobrazí v hlavním okně. Pokud najedete na text, který zobrazuje vaše pojmenování, zobrazí se vám systémové jméno mikrofonu. Mikrofony lze také odebrat pomocí tlačítka Smazat mikrofon.

Program reflektuje, že se připojují a odpojují mikrofony, ale není to odzkoušeno na více jak dvou, takže doporučuji mít všechny mikrofony před zapnutím připojené.

Pozor na to, který mikrofon máte nastavený jako výchozí v systému. Pokud připojujete do USB s největší pravděpodobností, pokud už jste si ho nepře-nastavovali, není výchozí. Pokud připojujete mikrofon do Jacku, tak většinou se interní vypne úplně a funguje pouze externí. Pokud máte rozdělený vstup, že interní i Jack jsou dva různé vstupy opět musíte překontrolovat.

#### Nový projekt

Při kliknutí na tlačítko New se otevře dialogové okno pro vytvoření nového projektu. Zde zadáte název, cestu ke složce, kam se bude ukládat a cestu k videu, které hodláte dabovat. Můžete také zvolit, aby se video okopírovalo do složky projektu (to ovšem nějakou dobu trvá a aplikace nijak neznázorňuje kopírování, takže vyčkejte, než zmizí dialogové okno).

#### Načtení projektu

Pokud už nějaký projekt je vytvořený stačí vyvolat okno pro výběr příslušného souboru tlačítkem Load. Soubor projektu mají tuto koncovku pDab a jako jediné se vám v dialogovém okně zobrazí.

Uložení projektu je pomocí tlačítka Save nebo při ukončení je uživatel dotázán. Žádné automatické ukládání není zatím implementováno, proto prosím periodicky ukládejte.

Aplikace podporuje video kontejnery (jakého jsou formátu video a audio v tomto kontejneru je jedno) s koncovkou avi, mp4, mkv. Je schopná podporovat i jiné formáty, ale nebyli testovány proto je to omezené pouze na výše zmíněné (pokud by byla nutnost použít jiné kontejnery, informujte mě o tom a já je doplním) Video se přehrává ve zvláštním okně a jde

o hlavní synchronizační prostředek. Nelze vypnout pouze ztlumit. Video je možné posouvat pomocí šipek, podle toho jaký máte měřítko v části, kde se zobrazují nahrané sekvence.

### **Nahrávání**

Po zmáčknutí tlačítka Record začne daný mikrofon nahrávat do zvoleného řádku. Po ukončení se nahrávka zobrazí v okně s nahrávkami a bude mít takovou barvu, jakou jste si zvolili u mikrofonu (barva lze později změnit). Lze také mikrofony hromadně ovládat, pokud u nich zaškrtnete příslušné zaškrtačací políčko. K ovládání slouží speciálně označený sloupec. Každý, mikrofon musí mít při nahrávání zvolen jiný řádek než ostatní (ty které nahrávají). Startovní čas je zvolen podle toho kde jste ve videu. (lze později také měnit)

Okno pro zobrazení nahraných sekvencí Zde probíhají editační práce na dabingu. Vidíte zde nahrané sekvence a lze s nimi manipulovat a to - Tažením po řádku (změnou pozice se mění, kde se budou přehrávat) - Změna řádku (pouze pro lepší orientaci, např. každý řádek reflektuje jednu postavu), řádky lze přidávat příslušným tlačítkem pod posledním řádkem.

### **Mazání**

Po kliknutí na nahrávku je vybrána a lze ji tlačítkem Delete smazat. Pokud máte vybrané dvě nahrávky pomoci kliknutí a klávesy shift smaže se poslední vybraná.

### **Spojování**

Pomocí kliknu s držením klávesy Shift lze vybrat dvě nahrávky a pomocí tlačítka Sloučit je spojit. Pokud se překrývají, budou se překrývat i ve výsledném spojení. Pokud je mezi nimi místo, bude vyplněno tichem

### **Rozdělování**

Pro rozdělování musíte na osu času (pruh jiné barvy nad řádky s nahrávkami) umístit značky. Ty se umísťují pomocí klávesy shift a kliknutí na osu. Lze aktivovat až dvě a pokud chcete jednu smazat, stačí na ní znovu s klávesou shift zmáčknout. Rozděluje se pouze poslední vybraná nahrávka a to na místech kam směřují značky.

### **Zoom**

V okně s dabingem lze nahrávky přibližovat pomocí klávesy ctrl a kolečka na myši. Jde o měřítko - 1pixel = 1s - 1px = 100ms - 1px = 10ms - 1px = 1ms

Podle toho jaký máte měřítko, můžete s vybranou nahrávkou hýbat pomocí tlačítek (nebo táhnutím).

### **Pohyb ve videu**

Posun ve videu a tím pádem v celém projektu lze, buď pomocí posuvníku u videa, nebo kliknou na osu, která je v okně pro zobrazení nahrávek. Po této ose jezdí malí čtvereček s čárkou (červený), se čtverečkem lze posouvat stejně jako s posuvníkem u videa.

## Undo, Redo (Zpět a Dopředu)

Klasická funkce, která je téměř ve všech aplikacích. V této aplikaci reflektují jenom akce mazání, spojování a rozdělování

Nahrané sekvence se přehrávají zároveň s videem. Tuto funkci lze vypnout odškrtnutím políčka „Přehrávat nahrávky“.

## Export

Tlačítkem Export vyvoláte vytvoření koncového produktu. Dojte ke spojení vašich nahrávek a zvuku z videa. Bohužel nelze vybrat, se kterým zvukovým streamem se budou vaše nahrávky spojovat, takže video soubor se kterým chcete pracovat, by měl mít jenom jednu audio stopu, jinak nevím, jak se to bude chovat. Zvuk z videa bude ve výsledku plně přítomen, takže doporučuji nahrávku předem připravit (vystříhnou hlasy nebo potlačit zvuk (potlačení mohou dodělat)). Export trvá celkem dlouhou dobu (s 20 nahrávky řady minut) a celá aplikace zamrzne (může začít i neodpovídat). Proto prosím, vyčkejte (export vždy skončí). Výsledný produkt bude uložen jako output.mkv.

K exportu byla přidána možnost tzv. voicover. Což znamená utlumení původní stopy, podle nastavení od 10% po 100%. Je to řešeno tak, že se video i s audiem okopíruje a utlumí. Bohužel to trvá značnou dobu (zkoušeno na skoro 4 minutovém videu a trvalo kolem 30 sekund (bude se lišit výkonem procesoru)).

## Instalace a kompilace

### Instalace

Zkompilovaný nástroj, který se nachází na CD, není nutné nijak instalovat. Stačí pouze okopírovat a spustit. Veškeré nutné knihovny jsou přibalené nebo jsou součástí systému. Ovšem při zkouškách se ukázalo, že často chybí knihovna MSVCR120.dll. To se vyřeší instalací balíku <http://www.microsoft.com/en-us/download/details.aspx?id=40784>.

### Kompilace

Pokud chcete nástroj zkompilovat sami, potřebuje nainstalované Qt a to verze 5.x (Osobně zkoušeno verzi 5.4 je možné, že některé prvky nejsou ve starších verzích). Jelikož aplikace využívá externí rámec QtAV, je ho také nutnost nainportovat do adresářů Qt. Osobně jsem QtAV musel ještě kompilovat sám, ale nyní už je možnost si už zkompilovanou knihovnu stáhnout. Nyní je dostupná verze 1.6.0 <https://github.com/wang-bin/QtAV>, kde naleznete i návod na nainportování to Qt. Pokud je QtAV správně vloženo mezi ostatní knihovny Qt tak poté stačí již jenom projekt přeložit. Buď pomocí Qt Creatoru, který si najde kompilátory sám nebo si je můžete nastavit nebo můžete rovnou použít qmake a make. Přeložení jde ovšem zatím pouze na systémech Windows. Po přeložení musíte k exe souboru aplikace přikopírovat program FFmpeg <https://www.ffmpeg.org/download.html>, který aplikace využívá k exportu a dílčím editacím.