



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Nahrazení QuickTime v rozši ujícím modulu pro FileMaker
Student:	Bc. Jan P ikryl
Vedoucí:	Ing. Tomáš Zahradnický, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2017/18

Pokyny pro vypracování

Seznamte se s programováním rozši ujících modul pro aplikaci FileMaker, multimediální knihovnou QuickTime a zdrojovými kódy zadavatelem poskytnutého rozši ujícího modulu. Prove te řešerši dostupných knihoven pro zpracování obrazových dat a knihoven pro práci s digitálním zvukem. Na základ řešerše s vedoucím vyberte vhodné knihovny. Navrh te a implementujte vybrané ásti rozši ujícího modulu tak, že volání do knihovny QuickTime budou nahrazena voláními do vybraných knihoven, a to p i zachování funk nosti rozši ujícího modulu. Implementovaný kód otestujte a srovnejte s originálem po stránce kvality generovaných dat a rychlosti.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdí k, CSc.
d kan

V Praze dne 22. ervna 2016

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Nahrazení QuickTime v rozšiřujícím modulu pro FileMaker

Bc. Jan Přikryl

Vedoucí práce: Ing. Tomáš Zahradnický, Ph.D.

29. června 2017

Poděkování

Tímto bych chtěl poděkovat vedoucímu své diplomové práce Ing. Tomáši Zahradnickému, Ph.D. za jeho odbornou pomoc při tvorbě této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel dohodu, na základě níž se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 29. června 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Jan Příkryl. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Příkryl, Jan. *Nahrazení QuickTime v rozšiřujícím modulu pro FileMaker*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Cílem této práce je nahrazení vybraných částí rozšiřujícího modulu MediaManager, jenž využívají funkce knihovny QuickTime, vlastní implementací, jenž bude pro manipulaci s médii využívat multimediální knihovny vybrané na základě rešerše a konzultace s vedoucím práce.

Klíčová slova FileMaker, MediaManager, QuickTime, ImageMagick, FFmpeg, rozšiřující modul, zvuk, obraz

Abstract

Goal of this thesis is to replace certain parts of plug-in module MediaManager, which are incorporating functionality of QuickTime library, with our own implementation that will use media manipulation libraries chosen based on research and consultation with thesis advisor.

Keywords FileMaker, MediaManager, QuickTime, ImageMagick, FFmpeg, plug-in, sound, image

Obsah

Úvod	1
1 Teorie	3
1.1 FileMaker	3
1.2 MediaManager	5
1.3 QuickTime	6
2 Analýza	11
2.1 Obraz	11
2.2 Zvuk	14
2.3 Výběr knihoven	18
3 Návrh	19
3.1 Zvuk	19
3.2 Obraz	26
4 Testování	29
4.1 Testování obrazové části	30
4.2 Testování zvukové části	31
Závěr	37
Literatura	39
A Seznam použitých zkratk	41
B Obsah příloženého CD	43

Seznam obrázků

1.1	Movie struktura	8
1.2	Struktura atomu	9
1.3	Hierarchie atomů	9
2.1	Diagram konverze multimédií pomocí ffmpeg	15
3.1	Zvuková část - soubory	20
3.2	FFMovie	22
3.3	PlaybackMapping	23
3.4	RecordingMapping	24
3.5	processSound	25
4.1	MediaManagerDemo	29
4.2	Výsledky testu ořezávání	32
4.3	Výsledky testu pootočení	33
4.4	Výsledky testu převrácení	34
4.5	Výsledky testu změny velikosti	35
4.6	Výsledky testu zkosení	36
4.7	Výsledky testu zkreslení	36

Seznam tabulek

1.1	Struktura konfiguračního řetězce	5
3.1	Třídy v rozhraní Magick++	26

Úvod

Tato práce se zabývá nahrazením částí rozšiřujícího modulu FileMaker, jenž využívají knihovnu QuickTime pro práci se zvukem a obrázky. Nová implementace bude využívat multimediální knihovny ImageMagick a FFmpeg, jejichž výběr proběhl na základě rešerše dostupných knihoven a konzultace s vedoucím diplomové práce. Teoretická část této práce se zabývá základy programování rozšiřujících modulů pro FileMaker, popisem rozšiřujícího modulu MediaManager, multimediální knihovnou QuickTime a analýzou řady multimediálních knihoven včetně zdůvodnění výběru knihoven pro implementaci řešení. V druhé části práce je představen návrh řešení a jeho implementace pomocí vybraných knihoven. V závěru práce je implementované řešení otestováno a srovnáno s originálním řešením po stránce kvality generovaných dat a rychlosti.

Teorie

Rozšiřující modul MediaManager představuje rozšíření aplikace FileMaker. Toto rozšíření umožňuje sloužit jako nástroj pro manipulaci s datovým obsahem, především však multimediálním, v aplikaci FileMaker.

1.1 FileMaker

FileMaker je cross-platformní aplikace umožňující pohodlnou a jednoduchou práci s daty uloženými v relační databázi. Tato aplikace se propojuje databázový engine s uživatelským rozhraním a bezpečnostními mechanismy, jež uživateli ve výsledku umožňuje jednoduše pozměnit databázový model pouhým přidáním prvků do formulářů, či obrazovek vytvářené aplikace.

1.1.1 Historie

Aplikace FileMaker byla původně vyvinuta z DOSové aplikace jako osobní databázová aplikace pro počítače Macintosh firmou Nashoba Systems v roce 1985 [1]. V roce 1988 byla firma Nashoba Systems koupena včetně práv na FileMaker firmou Claris, dceřinou společností Apple Computer Inc., jež měla na starosti vývoj a marketing vlastního softwaru. Následovalo vydání několika drobných verzí a v roce 1990 byl představen FileMaker Pro verze 1.0. V roce 1992 byla vydána první multiplatformní verze, jež byla dostupná pro Microsoft Windows a MacOS. Tyto verze aplikace stále neobsahovaly relační funkce, uměly pouze vyhledávat a importovat hodnoty ze souborů a umožnit jejich prezentaci. Ty byly přidány až v roce 1995, s představením FileMaker Pro 3. Dalším významným pokrokem bylo přidání architektury rozšiřujících modulů ve verzi 4.0 v roce 1997, jež umožnila vývojářům třetích stran přidávat vlastní funkcionalitu do FileMakeru.

V roce 1998 Apple přesunul nebo zrušil vývoj ostatních aplikací, jež byly v kompetenci dceřinné společnosti Claris (společnost přejmenoval na FileMaker, Inc.), aby se firma mohla zaměřit pouze na tento produkt.

Vývoj pokračoval s novou verzí přibližně každé 2 roky. Po krátkou dobu byl FileMaker server také dostupný na Linuxu, ale podpora pro tento operační systém byla zanedlouho zrušena ve verzi FileMaker Pro 7 v roce 2004. V této nové verzi byl také představen nový formát souborový formát .fp7, podporující soubory až do velikosti 8 TB, což bylo oproti předchozím verzím, jenž podporovaly soubory do velikosti 2 GB, velkým posunem. Byla také zvýšena kapacita polí, jenž mohly od této verze obsahovat až 4 GB binárních dat, nebo 2 GB 2bytového unikódového textu (navýšení z původních 64 kB). V neposlední řadě byl také rozšířen relační model, jenž nyní umožnil ukládání více tabulek v rámci jednoho souboru, a byl poskytnut grafický editor pro manipulaci s tabulkami a relacemi mezi nimi, jenž svou grafickou prezentací připomínal E-R diagram.

Verze FileMaker 8 v roce 2005 nabídla uživatelům rozšíření funkcionality, včetně vývojářských nástrojů. Mezi významnější rozšíření patřily: záložkové rozhraní, skriptové proměnné, popisky, rozšíření ladících nástrojů, možnost vytvoření vlastních menu a možnost kopírovat celé tabulky, definice polí, skripty uvnitř a mezi soubory.

Následovala další verze s drobnými vylepšeními jako je integrovaný webový prohlížeč, podmíněné formátování, hyperlinkový ukazatel do databáze, tvorba PDF, grafy, nebo tisk.

Poslední vydanou verzí byl FileMaker 16 v roce 2017 jenž například vylepšil práci s objekty, nebo přidal podporu funkcí pro manipulaci s daty v JSON formátu.

1.1.2 Tvorba rozšiřujících modulů pro FileMaker

FileMaker umožňuje integraci rozšiřujících modulů třetích stran, jenž mohou rozšířit jeho funkcionalitu [2]. Tyto rozšiřující moduly musí být napsány v C nebo C++ jazyce. Do FileMakeru se poté integrují ve formě dynamických knihoven a komunikují s FileMakerem prostřednictvím specifických funkcí a datových struktur.

FileMaker komunikuje s rozšiřujícími moduly pomocí funkce **FMExternCallProc()**, kterou FileMaker na rozšiřujícím modulu zavolá a předá jí ukazatel na datovou strukturu **FMExternCallStruct**. Na základě vyhodnocení této struktury pak funkce **FMExternCallProc()** provede jednu z následujících operací:

- **Init:** umožní rozšiřujícímu modulu provést prvotní nastavení, pokud rozšiřující modul takové nastavení vyžaduje (například rozhodnutí zda má rozšiřující modul v této verzi FileMakeru běžet)
- **Idle:** umožní rozšiřujícímu modulu využít volný čas
- **Shutdown:** umožní rozšiřujícímu modulu provést potřebné akce před jeho vypnutím

- **DoAppPreferences:** umožní uživateli nastavit parametry rozšiřujícího modulu

Externí funkce rozšiřujícího modulu se zaregistrují ve FileMakeru a dále se pak již mohou volat napřímo.

1.1.2.1 Resource soubory

Každý rozšiřující modul navíc může obsahovat resource soubory (.r pro Mac, .rc pro Windows), jenž obsahují detaily o daném rozšiřujícím modulu. V případě jejich využití jsou povinnými údaji v těchto souborech: jméno rozšiřujícího modulu, popis zobrazující se v preferencích aplikace, konfigurační řetězec (tzv. FeatureString) a jména externích funkcí (pokud chceme využít starého způsobu registrace externích funkcí). Jména rozšiřujícího modulu a externích funkcí nesmí mít prefix F, FM a Web. Tyto prefixy jsou FileMakerem rezervovány.

FeatureString je 11 znaků dlouhý řetězec, který poskytuje FileMakeru základní informace o rozšiřujícím modulu. Jeho struktura a popis jednotlivých částí se nachází v tabulce 1.1.

Tabulka 1.1: Struktura konfiguračního řetězce

Pozice	Hodnota	Popis
1-4	text	jedinečné ID rozšiřujícího modulu
5	1	hodnota vždy nastavena na 1
6	Y/n	Y pokud rozšiřující modul umožňuje konfiguraci uživatelem
7	Y/n	Y pokud rozšiřující modul využívá starého způsobu registrace externích funkcí
8	n	hodnota vždy nastavena na n
9	Y/n	Y pokud rozšiřující modul vyžaduje zasílání zpráv o „idle“ čase aplikace
10	n	hodnota vždy nastavena na n
11	n	hodnota vždy nastavena na n

1.2 MediaManager

MediaManager je rozšiřující modul rozšiřující možnosti aplikace FileMaker pro zacházení se soubory a editaci zvukových a obrazových souborů [3]. Nejnovější verze rozšiřujícího modulu MediaManager 15 je kompatibilní s aplikací FileMaker verze 14 a 15.

1.2.1 Grafika

MediaManager umožňuje modifikovat, transformovat a konvertovat obrázky bez ztráty na jejich kvalitě uvnitř souborů aplikace FileMaker. Rozšiřující modul je schopen upravovat velikost, rozlišení, či kvalitu obrazu, dokáže obraz oříznout pootočít, zkosit, či převrátit. Dále také umí na obrázek aplikovat efekty jako je zaostření, úprava kontrastu, jasu, embossing, nebo úprava RGB vyvážení.

V neposlední řadě umí také obrázky konvertovat mezi mnoha formáty, mezi něž patří například JPEG, GIF, PNG, nebo TIFF.

1.2.2 Zvuk

V oblasti zacházení se zvukem a jeho zpracování MediaManager nabízí možnost přehrávání zvuku (jak synchronní, tak asynchronní), ovládání hlasitosti, tempa a vyvážení při přehrávání. Dále také poskytuje možnost vyříznutí části zvukové stopy, napojení zvukových stop za sebe nebo jejich namixování. Zvukovou stopu je také možné znormalizovat, nebo přidat efekt zesílení, či zeslabení. U MP3 souborů je zde i podpora zobrazení a úpravy metadat. Mezi podporované formáty zvukových souborů patří rozšířený MP3, WAV, nebo také MP4, či AIFF. MediaManager umí nejen s těmito formáty pracovat, ale je také schopen zvukové soubory do těchto formátů konvertovat (pro konverzi do MP3 je nezbytný externí enkodér jako je například LAME).

1.2.3 Práce se soubory

MediaManager také rozšiřuje možnosti aplikace FileMaker v oblasti práce se soubory. Umožňuje uživateli ukládat soubory přímo, nebo pomocí reference (odkazu), přemísťovat, přejmenovávat, kopírovat, mazat, vytvářet a otevírat externí soubory, a umožňuje získávat HTML text, obrázky a jiné soubory přímo z webových stránek. V neposlední řadě také umí získávat detailní informace o obrázcích, zvukových souborech a videu.

1.3 QuickTime

QuickTime je multimediální framework vyvíjený firmou Apple schopný zpracovávat širokou škálu formátů videa, obrázků, zvuku a panoramatických snímků [4]. Pro uživatele macOS je tento framework dostupný v rámci instalace operačního systému, kdežto pro uživatele operačního systému Windows je dostupný jako samostatný instalační balíček. SDK pro vývojáře není pro tento framework volně dostupné, pro přístup k němu musí být vývojář členem programu Apple Developer (dříve Apple Developer Connection).

1.3.1 Apple Developer

Apple Developer je program pro vývojáře, jenž chtějí vyvíjet své aplikace pro platformy společnosti Apple. Pro základní přístup k prostředkům pro vývoj je nutné se pouze zdarma zaregistrovat na stránkách programu. Pokud však má vývojář zájem distribuovat své aplikace skrze App Store, nebo mít přístup k rozšířeným možnostem programu, pak je zde nabízeno také placené členství, poskytující tyto možnosti, s ročním poplatkem 99 dolarů.

Placené členství v programu umožňuje uživateli přístup k širšímu množství poskytovaných technologií a služeb. Mezi výhody programu patří například:

- předčasný přístup k beta verzím operačních systémů
- rozšířenému seznamu technologií (Apple Pay, Wallet, CloudKit, ...)
- testování - aplikace TestFlight pro snadnější řízení beta testování vyvíjených aplikací
- technická podpora
- umožnění distribuce vyvíjených aplikací přes App Store
- přístup k analytickým datům pro distribuované aplikace

1.3.2 Programové rozhraní

Programové rozhraní QuickTime je rozděleno do celků nazývaných **toolsets** [5]. Mezi nejdůležitější z nich patří **Movie Toolbox**, **Image Compression Manager**, **sequence grabber** a **QuickTime streaming API**. V této práci se budeme hlavně zabývat toolsetem Movie Toolbox.

1.3.2.1 Movie Toolbox

Tento toolbox je především určen k otevírání, modifikaci a ukládání médií, jenž pracují s časovými údaji. Jedná se tedy především o práci s videem a zvukem. Aby mohl QuickTime s těmito médii pracovat, reprezentuje je a pracuje s nimi ve formě tzv. **Movie**. Tyto **Movie** struktury si můžeme představit jako kolekci stop různého typu (audio, video, text, ...). Tyto jednotlivé stopy jsou reprezentovány pomocí struktury **Track**, jenž se nadále skládá z ještě drobnějších struktur nazývaných **Media**.

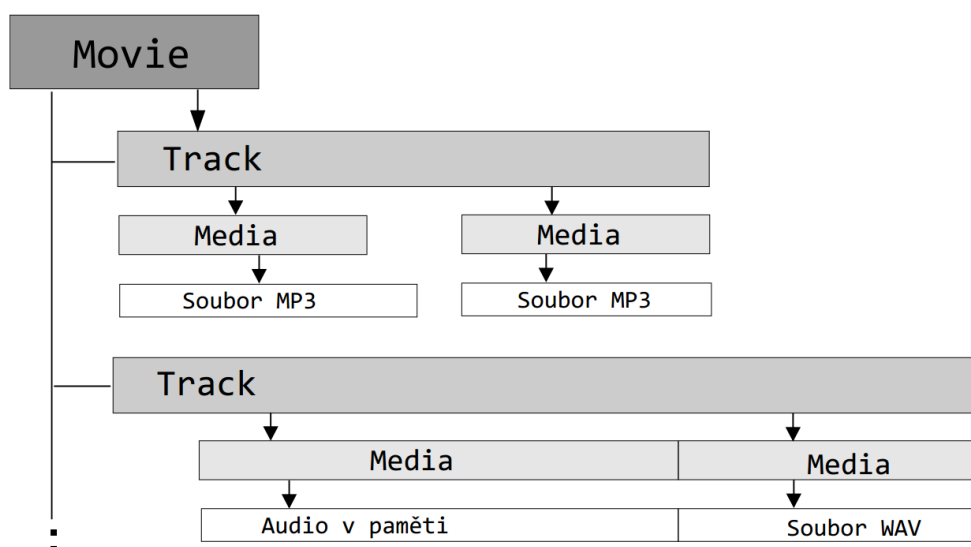
Kromě práce s časově orientovanými médii obsahuje tento toolbox také funkce pro práci s maticemi. Tato funkcionalita je především využitelná při manipulaci s obrazovými daty, jako jsou videa a obrázky.

1.3.2.2 Movie

Movie, jakožto základní jednotka pro práci s časově orientovanými médii, sdružuje dohromady všechny zdroje daných médií, popis jejich vstupů a výstupů a také rozmístění a modifikace těchto médií na časové ose. Strukturu **Movie** lze uložit do souboru, nebo ji naopak ze souboru načíst zpět do paměti. Do **Movie** lze naimportovat média různých formátů, nebo naopak vyexportovat část, nebo celou strukturu **Movie** do jiného formátu.

V rámci QuickTime může být **Movie** chápán jako formát mediálních souborů, nebo jako paměťová struktura určená pro uchování a práci s multimediálními soubory. V rámci této práce se však budeme zabývat pouze **Movie** jakožto paměťovou strukturou.

Celá základní struktura **Movie** je zobrazena na obrázku 1.1



Obrázek 1.1: Schéma struktury **Movie**

1.3.2.3 Track

Movie je složen z jednotek, které se nazývají **Track** [5]. **Track** je ve skutečnosti časovou osou, která obsahuje za sebou poskládané úseky médií. Na jeden časový bod může v rámci **Track** připadnout pouze jeden prvek **Media**. Pokud mají média být v **Movie** na časové ose umístěna paralelně, je třeba využít více paralelních časových os (prvků **Track**).

Track nemusí obsahovat pouze data, může také obsahovat modifikátory aplikované na ostatní data. Příkladem takového modifikátoru je modifikátor hlasitosti. **Track** obsahující modifikátor hlasitosti nese informaci, jakou hlasitost má mít jiný **Track** s audiem v různých časových okamžicích. Propojení mezi jednotlivými stopami je realizováno pomocí tzv. input mapy, která ob-

sahuje především informace o tom, které dvě stopy jsou spojeny a o jaký typ spojení se jedná.

1.3.2.4 Media

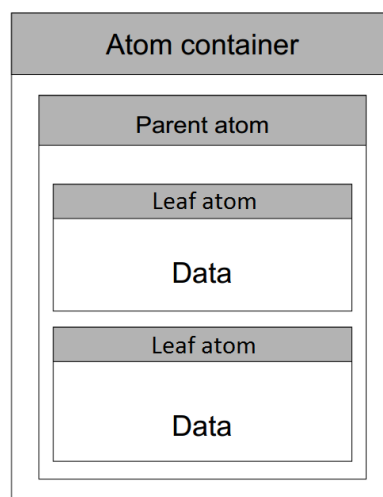
Časová osa (**Track**) je složená z úseků, jenž jsou souhrnně označovány jako **Media** [5]. Struktura **Media** obsahuje, kromě jiného, informace o délce, trvání a pozici média na časové ose, odkaz na úložiště dat a informace o typu dat. Příkladem může být odkaz na soubor MP3, nebo odkaz na paměťový blok s PCM vzorky. Další informace obsažená v **Media** je údaj o jeho měřítku, tzv. „time scale“. Tento číselný údaj říká, kolik jednotek délky připadá na jednu sekundu. Výchozí hodnotou je 600, což znamená, že jedna jednotka délky je $\frac{1}{600}$ sekundy.

1.3.2.5 Atom

QuickTime často používá datového typu atom. Atom je kontejner, který umožňuje skladovat data. Každý atom obsahuje informaci o své velikosti (4 B), typ (obvykle 4 ASCII znaky), ID (4 B), počet podřízených atomů (2 B) a dále data. Struktura jednoho atomu v paměti je na obrázku 1.2. Atomy mohou být dále uspořádány do hierarchie. V tomto případě se atomy dělí na dva typy: atomy, které obsahují jiné atomy (parent atom, container) a atomy, které obsahují data (leaf atom). Příklad takové struktury je na obrázku 1.3. Atomy se používají pro uchování strukturovaných dat, například informací o kompresi, parametry efektů atp.



Obrázek 1.2: Atom v paměti



Obrázek 1.3: Hierarchie atomů

Analýza

Pro potřeby této práce bylo zapotřebí vybrat vhodné knihovny, jenž by byly svými funkcionalitami schopny nahradit doposud používaný QuickTime v rozšiřujícím modulu MediaManager. Ačkoliv se tato práce zabývá pouze implementací řešení pro platformu Microsoft Windows, vybraná knihovna by měla podporovat i platformu MacOS. Vybraná knihovna by také měla být použitelná v komerčním software. Bylo možné vybrat i placené řešení, pokud by neexistovala dostatečně kvalitní neplacená varianta, nebo pokud by tato placená varianta nabízela objektivně lepší, výhodnější řešení.

Po stránce funkčnosti se po grafické knihovně požadovala možnost konvertovat obrázky mezi jednotlivými formáty, dále pak schopnost provádět základní modifikace obrazu jako je změna velikosti, zrcadlení, pootočení, či deformace. Dále zde byly požadavky na možnost aplikovat různé vizuální efekty (zaostření, sépiový filtr, embossing, detekce hran, ...), možnost vyvážení barev, nebo také možnost kompozice obrazů.

U zvukové knihovny byl především kladen důraz na možnost převodu mezi jednotlivými zvukovými formáty a také na možnost zvuk nahrávat a přehrávat. Dalšími požadavky byla možnost zvukové stopy na sebe napojovat, nebo mixovat dohromady, možnost získávat jejich metadata a případně také aplikovat efekty jako je zesílení/zeslabení nebo normalizace.

2.1 Obraz

2.1.1 CImg

CImg je malá open-sourcová knihovna poskytující C++ rozhraní pro zpracování obrazu [6]. Tato knihovna uživatelům umožňuje:

- načítat a ukládat obrázky různých formátů
- přistupovat k hodnotám jednotlivých pixelů

- zobrazit, transformovat a filtrovat obrázky
- nakreslit text, primitivní objekty, křivky
- vypočítat statistiky

Základní funkce této knihovny nejsou závislé na žádné jiné knihovně, avšak její schopnosti se dají rozšířit použitím externích nástrojů a knihoven jako jsou **FFmpeg**, **ImageMagick**, **OpenCV**, **GraphicsMagick** a další. Pokud však ani tato rozšíření nepostačují či nevyhovují, je možné obohatit schopnosti této knihovny vlastními rozšiřujícími moduly.

Knihovna je kompatibilní s velkým množstvím C++ kompilátorů (Visual C++, g++, clang++, ...) a je přenosná na široké spektrum operačních systémů (Windows, MacOS X, Unix, ...). Díky distribuci pod CeCILL-C a CeCILL licencí je možné tuto knihovnu použít nejen v jiných open-source projektech, ale také v komerčních projektech a aplikacích.

2.1.2 ImageMagick

ImageMagick je open-source software pro manipulaci s grafickými soubory [7]. Tento software v současné době dokáže číst a zapisovat obrázky v širokém množství formátu (přes 200) mezi něž patří: PNG, JPEG, GIF, TIFF, PDF, Postscript a další, a díky vícevláknové implementaci výpočtů je schopen zpracovávat tyto obrázky efektivně i v mega-, giga- nebo i tera-pixelových velikostech. ImageMagick dále nabízí široké možnosti, jak s těmito soubory pracovat. Mezi jeho schopnosti patří:

- Konverze mezi formáty
- Základní modifikace obrázků - změna velikosti, zrcadlení, pootočení, distorze, oříznutí, ...
- Práce s barevným spektrem - úprava barev obrázků, vyvážení barev, extrakce kanálů barev
- Filtry a efekty - zaostření, rozmazání, detekce hran, kresba tužkou, olejomalba, ...
- Kreslení - text, čáry, polygony, elipsy, bézierovy křivky, ...
- kompozice obrázků, kanálů

Tento software je především využíván pomocí příkazové řádky, avšak nabízí také jednoduché grafické rozhraní pro unixové systémy a také programové interface pro širokou škálu programovacích jazyků mezi něž patří například C, C++, Pascal, .NET, Ruby nebo Python.

Díky distribuci pod Apache 2.0 licencí, jenž umožňuje bezplatné využití i v komerčních projektech, a široké škále funkcí je tato knihovna jedním z předních kandidátů.

2.1.3 OpenCV

OpenCV je open-source knihovna zaměřená na počítačové vidění (získávání informací ze zachyceného obrazu) a strojové učení [8]. Hlavním důvodem vytvoření této knihovny byla snaha poskytnout společnou infrastrukturu pro aplikace zaměřující se na počítačové vidění a urychlit tím rozšíření této techniky v komerčním prostředí. Knihovna je tedy distribuovaná pod BSD licenci, jenž umožňuje její použití, či modifikaci pro komerční produkty.

Ačkoliv je tedy knihovna zaměřena především na zpracování a analýzu obrazových dat, obsahuje též funkce pro manipulaci s těmito daty a ukládání výsledků těchto manipulací, což z ní činí jednoho z možných kandidátů pro náhradu QuickTime knihovny v rozšiřujícím modulu MediaManager.

Knihovna obsahuje přes 2500 optimalizovaných algoritmů, jenž zahrnují klasické i moderní algoritmy počítačového vidění strojového učení. Tyto algoritmy mohou být využity pro: detekci a rozpoznávání obličejů, identifikaci objektů, klasifikaci lidských činností ve videu, sledování pohybujících se objektů, odstranění červených očí z fotografií, sledování pohybu očí a mnoho dalších.

V současné době je tato knihovna využívána mnoha společnostmi jako je Google, Microsoft, Intel, IBM, Sony, Honda nebo startupy jako je VideoSurf nebo Zeitera.

Knihovna poskytuje programové rozhraní pro C++, C, Python a MATLAB a je podporována pro operační systémy Windows, Linux, Android a Mac OS.

2.1.4 Simd

Simd je open-source knihovna pro zpracování obrazu pro C a C++ programovací jazyky. Tato knihovna uživatelům nabízí výkonné algoritmy pro operace jako jsou filtrace, extrakce statistických informací, detekce pohybu, detekce objektů, nebo operace pro práci s neurálními sítěmi.

Bohužel tato knihovna svou funkcionalitou nevyhovuje požadavkům této práce.

2.1.5 GD Graphics

GD Graphics je další ze seznamu open-source knihoven pro práci s grafickým obsahem. V tomto případě se jedná o poměrně malou knihovnu jenž je především zaměřena na dynamickou tvorbu obrázků jako jsou grafy, náhledy a další. Hlavní uplatnění této knihovny našla při vývoji webového obsahu, avšak není to její jediná možnost použití.

Kromě dynamické tvorby grafického obsahu knihovna také umožňuje načítat a ukládat obrázky podporovaných formátů, mezi něž patří PNG, JPEG, BMP, GIF a další, a na tyto obrázky aplikovat základní transformace (jako je změna velikosti, zrcadlení, či rotace), nebo některé základní filtry jako je negace barev, detekce hran, embossing a další.

Knihovna je cross-platformní a díky BSD licenci je ji možné využít v komerčním software.

2.1.6 VIPS

VIPS je grafická knihovna distribuovaná pod licencí LGPL pro operační systémy Windows, Linux a MacOS X. Tato knihovna nabízí množství filtrů, aritmetických operací, zpracování barev, geometrických transformací a vytváření histogramů [9]. Kromě obvyklých formátů jako je JPEG, TIFF, PNG, PDF nebo GIF také tato knihovna podporuje vědecké formáty jako jsou FITS, OpenEXR, Matlab, PFM nebo Radiance.

Oproti podobným knihovnám se tato knihovna pyšní poměrně velkou rychlostí a efektivností (s ohledem na využití paměti).

2.2 Zvuk

2.2.1 Aquila

Aquila je drobná open-source knihovna pro zpracování digitálního signálu. Hlavní funkcí této knihovny spočívá ve výpočtu a extrakci vlastností daného signálu. Knihovna je schopna signál nahrát z WAV souboru, obyčejného textového souboru, nebo z PCM dat. Knihovna též obsahuje několik generátorů základních typů signálů jako je bílý šum, nebo sinusový signál.

Knihovna bohužel neumožňuje nahrávání zvuků (signálů), jejich přehrávání, ani konverzi mezi jednotlivými formáty. Jediná využitelná část pro tuto práci by bylo generování základních typů signálů.

2.2.2 Portaudio

PortAudio je malá open-source cross-platformní knihovna zaměřená na I/O operace se zvukovým signálem. Knihovna je určena pro psaní jednoduchých audio programů v C, nebo C++. Jejím hlavním cílem je poskytnout jednoduché, na platformě nezávislé API pro nahrávání a přehrávání zvuku využívající jednoduché callback funkce, nebo blokujícího read/write rozhraní [10]. Knihovna umožňuje jednoduchý přístup k přehrávacím i záznamovým audio zařízením připojeným k počítači. Mezi možnosti této knihovny bohužel nepatří zapisování a načítání zvukových souborů různých formátů.

PortAudio je využívána v mnoha projektech od audio přehrávačů (VLC Media Player) a editorů (Audacity) až po VoIP programy a počítačové hry. Díky své nerestriktivní licenci je možné tuto knihovnu využít i v komerčních projektech bez nutnosti zveřejnit vlastní kód.

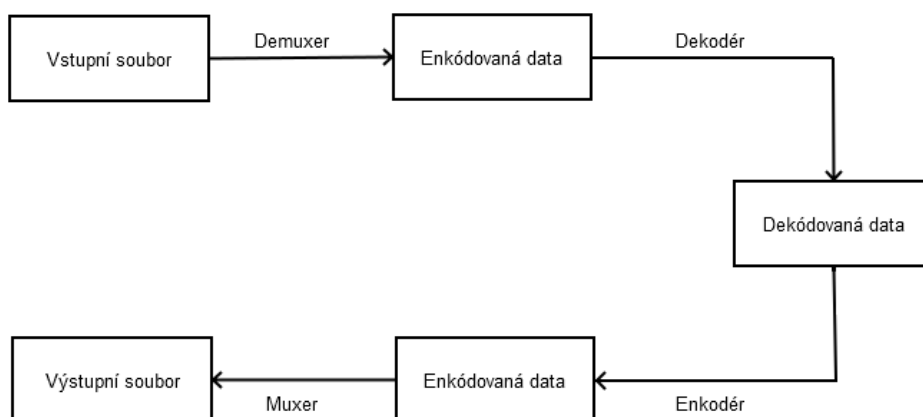
2.2.3 FFmpeg

FFmpeg je jeden z nejlepších a nejrozšířenějších multimediálních frameworků. Tento framework nabízí dekodování, enkódování, transkódování, streamování, filtraci a přehrávání velkého množství audio a video formátů [11]. Tento framework je také velice portabilní, je možné ho zkompileovat a provozovat na velkém množství platform jako jsou Linux, Mac OS X, Windows, Solaris a dalších. Není omezen ani hardware architekturou.

FFmpeg nabízí nejen možnost využití jakožto knihovny pro práci s multimédií, ale je možné jej získat také jako program pro příkazovou řádku. Díky distribuci pod GNU LGPL nebo GNU GPL licencí je též k dispozici i jeho zdrojový kód. Pro příkazovou řádku nabízí FFmpeg tři nástroje: `ffmpeg`, `ffplay` a `ffprobe`.

2.2.3.1 Nástroj ffmpeg

Tento nástroj slouží především jako rychlý konvertor audio a video formátů, jenž je též schopen jako zdroj použít vstup z nahrávacích zařízení [12]. Je též schopen konvertovat zdroj na libovolnou vzorkovací frekvenci a měnit velikost videa za běhu pomocí kvalitního polyfázového filtru.



Obrázek 2.1: Diagram konverze multimédií pomocí `ffmpeg`

`Ffmpeg` je schopen číst z libovolného počtu vstupních zdrojů (jimiž mohou být soubory, nahrávací zařízení, síťové proudy dat a další), jenž jsou specifikovány pomocí parametru `-i`, a zapisovat do libovolného počtu výstupních zdrojů. Veškeré části příkazu, jenž nelze interpretovat jako přepínače či parametry, jsou považovány za specifikaci výstupu.

Veškeré vstupní a výstupní zdroje mohou teoreticky obsahovat libovolný počet streamů různého typu (audio, video, titulky, data), avšak toto množství a typy jsou v praxi limitovány formátem daného zdroje. Pro určení jaké

vstupy (jejich streamy) se mají namapovat na dané výstupy slouží parametr `-map`. Pomocí tohoto parametru a indexace vstupů a výstupů jsme schopni namapovat libovolné vstupní streamy na libovolné výstupní streamy. Soubory jsou indexovány počínaje 0, tudíž 0:2 odkazuje na třetí stream prvního vstupního zdroje.

Obecným pravidlem pro nástroj `ffmpeg` je, že parametry a přepínače jsou aplikovány pouze na následující zdroj, tudíž je nutné věnovat pozornost posloupnosti částí příkazu. Pokud tedy chceme aplikovat jisté parametry na více zdrojů, musí se tyto parametry v příkazu opakovat před každým zvoleným zdrojem. Výjimku z tohoto pravidla tvoří globální parametry, jež se vztahují k celému příkazu a měly by být specifikovány na jeho začátku.

2.2.3.2 Nástroj `ffplay`

Nástroj `ffplay` představuje jednoduchý přenosný přehrávač médií využívající FFmpeg knihoven a knihovny SDL [13]. Pomocí parametrů příkazové řádky lze vyprat specifické streamy vstupního souboru pro přehrávání, nastavit počáteční čas, dobu přehrávání, nebo aplikovat filtry na přehrávaný zvuk/video. Po spuštění přehrávače jej lze ovládat pomocí klávesových zkratk.

2.2.3.3 Nástroj `ffprobe`

`Ffprobe` je nástroj umožňující uživateli získat informace o multimediálních souborech a jejich streamech a tyto informace zobrazit ve formátu čitelném pro člověka, nebo počítač [14].

Zobrazované informace jsou rozděleny do sekcí a podsekcí jež mohou být identifikovány pomocí obecného a unikátního jména (obecné jméno může být sdíleno několika sekcemi stejného charakteru). Výstup nástroje `ffprobe` je koncipován tak, aby byl jednoduše parsovatelný textovými filtry. Formát výstupu lze specifikovat pomocí parametru `print_format`. Mezi podporované formáty patří například formát `csv`, `ini`, `json` nebo `xml`.

2.2.4 Essentia

`Essentia` je audio knihovna zaměřená na audio analýzu a získávání informací o audio souborech. Tato knihovna obsahuje extenzivní kolekci algoritmů, jež zahrnují audio I/O funkcionalitu, statistickou charakterizaci dat a velké množství hudebních deskriptorů [15].

`Essentia` je distribuovaná pod Affero GPLv3 licencí, jež neumožňuje použití v aplikacích, jež jsou pod jinou licencí, avšak je zde možnost požádat o distribuci pod proprietární licencí.

Knihovna v současné době podporuje operační systémy Linux, MacOS X a částečně také Windows a Android.

Tato knihovna našla své uplatnění především v oblasti klasifikace hudby, vizualizace a interakce s hudbou, detekce taktu, indexace zvuků, nebo detekce podobností.

2.2.5 IRRKlang

IRRKlang je knihovna a zvukový engine zaměřující se především na přehrávání zvukových souborů a manipulaci s přehrávaným zvukem. Mezi její schopnosti patří přehrávání souborů ve formátu WAV, MP3, OGG, FLAC a další, zachycení zvukového záznamu a aplikace zvukových efektů během přehrávání.

API této knihovny je nezávislé na platformě. V současné době jsou podporovány platformy Windows, Linux a MacOS X. Jako programovací jazyk si může uživatel zvolit mezi C++ a jazyky .NET [16]. Knihovna je k dispozici zdarma pro nekomerční použití, nebo jako placená verze.

2.2.6 Juce

Juce je open-source cross-platformní aplikační framework jenž uživatelům nabízí širokou škálu možností. Ačkoliv tento framework nabízí širokou škálu funkcionalit, prominentní mezi nimi jsou funkcionality související s manipulací audio souborů. Tento framework je schopen načítat a ukládat základní audio formáty jako jsou WAV, AIFF, FLAC, nebo MP3, dále je také schopen tyto audio soubory přehrát, nebo v nich uložit nahraná data.

2.2.7 Cinder

Stejně jako u většiny zkoumaných knihoven, i u této se jedná o open-sourcový projekt s C++ programátorským rozhraním pro manipulaci s multimediálním obsahem. Tato knihovna nabízí funkcionalitu nejen pro práci se zvukem, ale také s 2D a 3D grafikou.

Zvuková část knihovny uživatelům umožňuje číst několik formátů zvukových souborů jako jsou WAV, MP3, nebo Ogg Vorbis a zapisovat prozatím pouze ve formátu WAV. Knihovna též umožňuje zvuk nahrávat a přehrávat, obsahuje několik generátorů zvukového signálu, filtrů a nástrojů pro jeho zpracování.

Knihovna je distribuována pod BSD licencí, jenž umožňuje její použití v komerčních projektech bez nutnosti zveřejnit zdrojové kódy. V současné době je však tato knihovna ve verzi 0.9.1, tudíž lze předpokládat, že knihovna může obsahovat více chyb než jiné.

2.2.8 Openframeworks

2.3 Výběr knihoven

Po pečlivém zvážení všech pozitivních a negativních aspektů všech zkoumaných knihoven a po konzultaci s vedoucím práce jsem se rozhodl pro grafickou knihovnu ImageMagick a zvukovou knihovnu FFmpeg.

ImageMagick knihovna byla především vybrána z důvodu jejích rozsáhlých možností práce s grafikou. Dalším důvodem byla její licence, jenž umožňuje použití této knihovny v komerčních programech bez nutnosti zveřejňovat zdrojový kód.

U knihovny pro práci se zvukem bylo rozhodování těžší. Knihoven, které by umožňovaly použití v komerčních programech bez zveřejnění zdrojového kódu a zároveň nabízely potřebnou funkcionalitu nebylo mnoho. Po diskuzi s vedoucím této práce jsem se rozhodl využít možností knihovny FFmpeg. Tato knihovna byla zvolena především z důvodu vyhovujícího rozsahu funkcionalit, její rozšířenosti a spolehlivosti - například oproti knihovně Cinder, jenž je teprve ve verzi 0.9. Jedinou nevýhodou této knihovny je její GPL/LGPL licence, jenž neumožňuje využití této knihovny bez publikace našeho kódu. Z tohoto důvodu jsme se rozhodli knihovnu využít externě, voláním jejích nástrojů pro příkazovou řádku. Knihovna a její nástroje tudíž nebudou přímo součástí rozšiřujícího modulu MediaManager a kód tohoto modulu tedy nebude muset být zveřejněn.

Návrh

Stejně jako analýza, i návrh implementace změn v rozšiřujícím modulu MediaManager můžeme rozdělit do dvou oblastí - úprava zvukové části a úprava části pro manipulaci s obrázky. V každé z těchto částí bude nutné především navrhnout v jaké podobě bude rozšiřující modul pracovat s potřebnými soubory/daty.

3.1 Zvuk

Nejdůležitějšími funkcionalitami zvukové části jsou nahrávání a přehrávání zvuku, konverze mezi zvukovými formáty a dále také několik funkcí pro manipulaci zvuku jako je postupné ztišení/zesílení hlasitosti zvuku, nebo jeho normalizace. Ve všech těchto částech původní implementace rozšiřujícího modulu MediaManager využívá strukturu `Movie` a její dílčí části, `Media`, `Track` a `Atom`, nad kterými následně provádí příslušné operace.

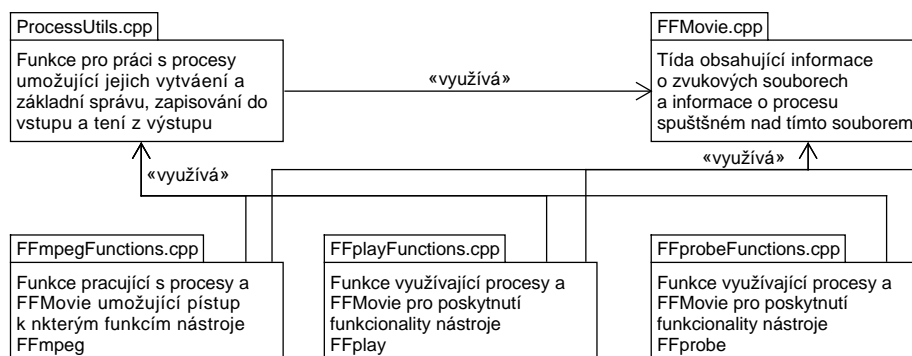
Jelikož byla pro práci se zvukem vybrána knihovna FFmpeg, jenž bude využívána externě, bude nutné navrhnout funkce pro práci s jejími nástroji příkazové řádky. Tohoto by se mělo dosáhnout vytvářením nových procesů a jejich následnou manipulací. Dále by se měla vytvořit sada funkcí jenž by zaobalovala nízkoúrovňovou práci s procesy a poskytovala tak již určitou funkčnost knihovny FFmpeg. Díky této sadě funkcí se již poté při vlastním nahrazování jednotlivých funkčních částí rozšiřujícího modulu nebudeme muset zabývat voláním procesů a jejich nezbytnou obsluhou.

Nová funkcionalita pro zvukovou část rozšiřujícího modulu MediaManager by měla být rodělena pro přehlednost do jednotlivých souborů dle obrázku 3.1.

3.1.1 Zacházení s procesy

Pro zpřístupnění funkcionality knihovny FFmpeg bude třeba vytvořit sadu funkcí, jejichž pomocí budeme vytvářet a spravovat přístup k procesům spouš-

3. NÁVRH



Obrázek 3.1: Schéma souborů pro zvukovou část

tějším nástroje příkazové řádky. Bude také potřeba implementovat funkce, jenž nám umožní zapisovat do standardního vstupu a číst standardní výstup takto vytvořených procesů.

Seznam funkcí, jenž bude nutné naimplementovat pro správu procesů je následující:

- **executeCommand** - vytvoří proces, jenž vykoná zadaný příkaz příkazové řádky
- **terminateProcess** - funkce, jenž vynutí okamžité ukončení procesu
- **suspendProcess** - pozastaví běh daného procesu
- **resumeProcess** - obnoví běh daného procesu
- **closeProcessHandle** - bezpečně uvolní `Handle` daného procesu
- **waitForProcess** - čeká na ukončení daného procesu po určitou dobu
- **waitAndCloseProcess** - čeká dokud se daný proces neukončí a poté bezpečně uvolní `Handle` na tento proces

Pro práci se standardním vstupem a výstupem pak bude nutné naimplementovat funkce:

- **createPipes** - vytvoří `pipe` jenž nám umožní číst ze standardního vstupu a zapisovat do standardního vstupu procesu
- **readStandardOutput** - funkce jenž vrátí data přečtená ze standardního vstupu daného procesu
- **writeStandardInput** - funkce jenž zapíše daná data do standardního vstupu daného procesu

- **closeReadPipes** - uzavře a uvolní **pipe** určené pro čtení z výstupu daného procesu
- **closeWritePipes** - uzavře a uvolní **pipe** určené pro zápis do vstupu daného procesu

Tyto funkce společně s třídou **FFMovie** představují základní jádro řešení náhrady knihovny QuickTime nástroji knihovny FFmpeg ve zvukové části rozšiřujícího modulu.

3.1.2 FFMovie

Jelikož rozšiřující modul **MediaManager** využívá pro práci se zvukovým obsahem strukturu **Movie**, budeme tuto strukturu muset nahradit vlastní třídou jejímž prostřednictvím budeme pracovat se zvukovými soubory. Na rozdíl od QuickTime knihovny nebudeme ve většině případů vyžadovat přístup přímo k datům zvukového souboru. Jelikož budeme využívat nástroje příkazové řádky knihovny FFmpeg, bude nám ve valné většině případů postačovat název a cesta ke zvukovému souboru.

Abychom práci se zvukovým souborem pomocí procesů ulehčili, bude také naše třída **FFMovie** obsahovat proměnné pro práci s procesy. Tyto proměnné nebudou využívány přímo při implementaci změn ve stávajícím kódu rozšiřujícího modulu, nýbrž v implementaci navrženého rozhraní pro práci s procesy. Díky tomuto budeme moci pracovat při vlastní náhradě funkcionalit knihovny QuickTime pouze s třídou **FFMovie** a funkcemi pracujícími s touto třídou. Budeme tak odstíněni od složitější logiky práce s procesy.

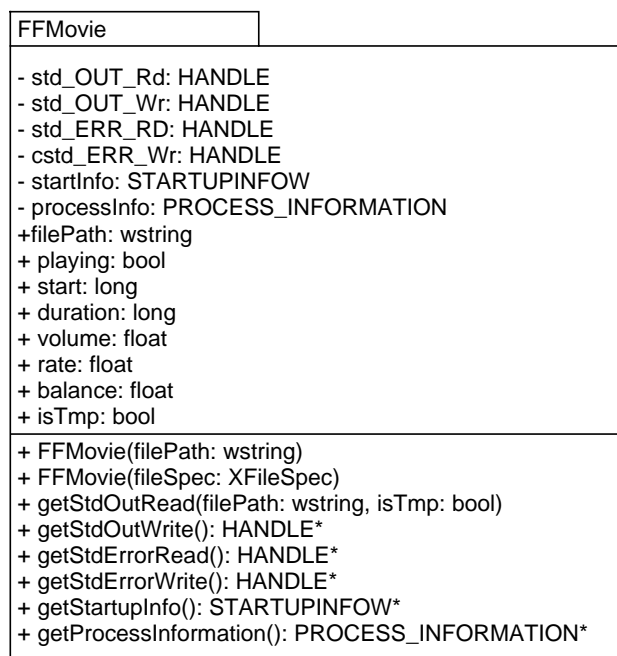
Pro práci s procesy budeme v třídě **FFMovie** potřebovat proměnou typu **STARTUPINFO** pro uchování informací o nastavení spouštěného procesu, jenž bude pracovat s daným **FFMovie**, a také proměnnou typu **PROCESS_INFORMATION** jenž v sobě uchovává informace nezbytné k ovládní spuštěného procesu, jako je odkaz na spuštěný proces a jeho příslušné vlákno. Proměnnou **PROCESS_INFORMATION** využijeme především pro kontrolu, zda spuštěný proces již došel, nebo pro jeho pozastavení či opětovné spuštění.

Dalšími nezbytnou součástí **FFMovie** jsou 4 proměnné typu **HANDLE**, jenž odkazují na **Pipe** (roury) napojené na standardní a chybový vstup a výstup procesu fpojeného s daným **FFMovie**. Díky těmto **HANDLE** budeme například moci předávat pomocí standardního vstupu zvuková data jenž budeme chtít enkódovat a zapsat do souboru, nebo číst ze standardního výstupu metainformace informace získané ze zvukového souboru prostřednictvím nástrojů FFmpeg.

Posledními, ale stejně tak důležitými, proměnnými v **FFMovie** jsou proměnné pro uložení informací o zvukovém souboru. Tyto proměnné zahrnují cestu k danému souboru, příznak, zda je tento soubor pouze dočasný, příznak zda je tento soubor aktuálně přehráván a také informace od jakého časového

3. NÁVRH

úseku se má zvukový soubor přehrávat, po jak dlouhou dobu se má přehrávat a s jakou hlasitostí, rychlostí a vyvážením.



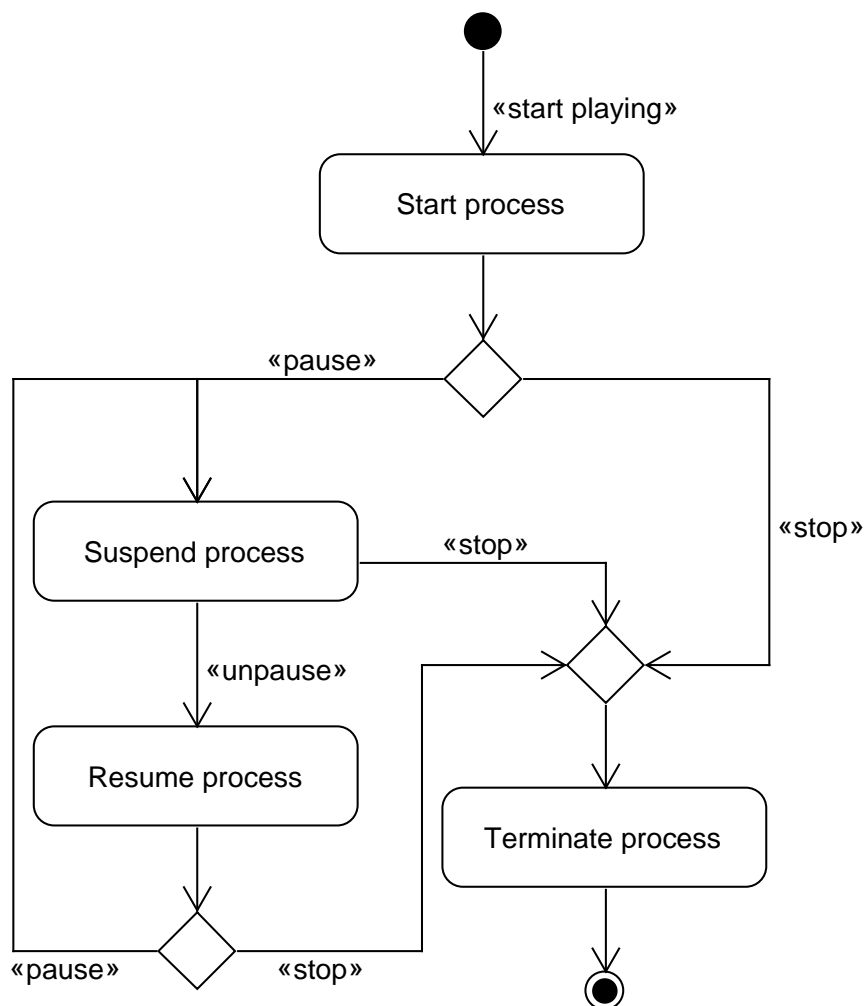
Obrázek 3.2: Třída FFMovie

3.1.3 Funkce pro přehrávání

Pro přehrávání zvukových souborů se vytvoří sada funkcí využívající nástroj `ffplay` a manipulaci s procesy. Funkce, jež budou využívány přímo v kódu rozšiřujícího modulu budou mít jako parametr pouze `FFMovie`, jež bude obsahovat všechny potřebné informace pro manipulaci procesů. Vlastní práce s procesy bude probíhat uvnitř příslušných funkcí a tím se tedy zjednoduší práce s nimi.

Do funkcí, jež budou mít na starosti práci s procesy zajišťujícími přehrávání, budou předávány objekty třídy `FFMovie`, jež v sobě ponese všechny nezbytné informace k samotnému přehrávání zvuku. Samozřejmostí je využití parametru `filePath`, jakožto parametru obsahujícího cestu k přehrávanému souboru. Pro přehrávání samotné jsou dále důležité atributy `start` a `duration`, jež udávají od jakého časového okamžiku se má zvukový soubor začít přehrávat a jak dlouhý časový úsek se má přehrát, pokud nebude přehrávání přerušeno. Dalším významným parametrem pro přehrávání je `PROCESS_INFORMATION`, jež bude využíván k uchování identifikátoru vytvořeného procesu. Právě díky tomuto identifikátoru budeme schopni pozastavovat, obnovovat

a ukončovat proces uskutečňující přehrávání a efektivně tak ovládat přehrávání daného zvukového souboru.



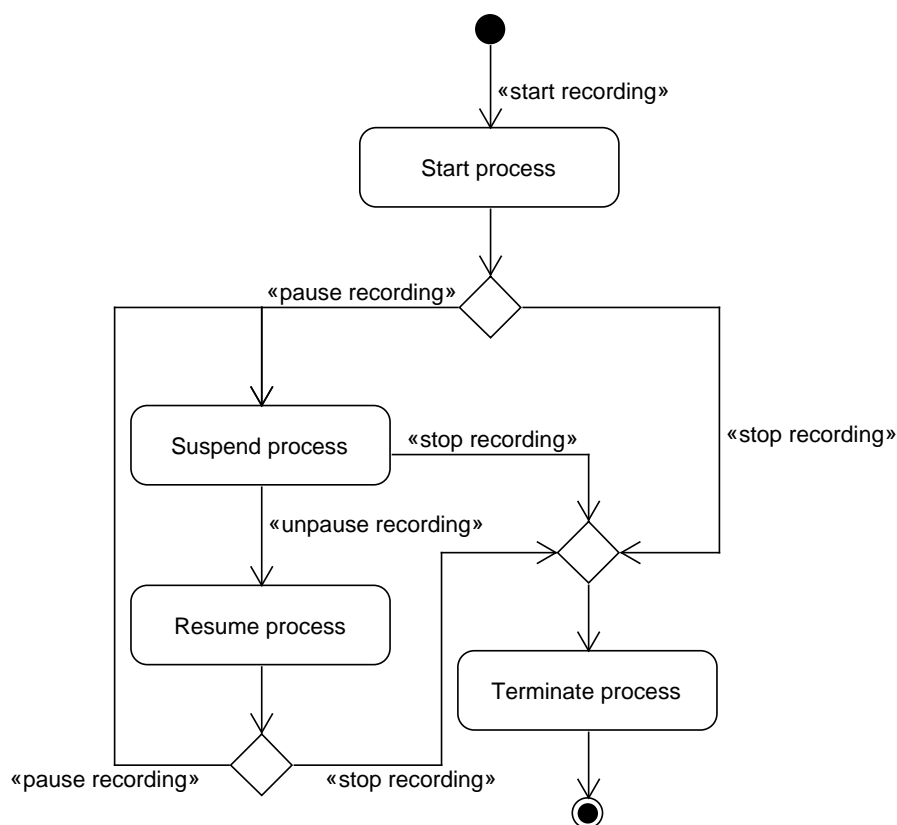
Obrázek 3.3: Namapování přehrávání zvuku na práci s procesy

3.1.4 Funkce pro nahrávání

Nahrávání zvuku bude prováděno pomocí nástroje `ffmpeg`. Pomocí tohoto nástroje zahájíme nahrávání do dočasného souboru a průběh nahrávání budeme opět řídit pomocí manipulace procesů. Bude nutné implementovat pozastavení, znovuspuštění a úplné zastavení nahrávání. Tohoto se opět docílí pomocí pozastavení procesu, opětovného rozběhnutí procesu a ukončením procesu 3.4.

3. NÁVRH

I v této části implementace bude využito objektů třídy `FFMovie`, avšak zde budou využity pouze pro uchování cesty k vytvářenému souboru v atributu `filePath` a k uchování identifikátoru běžícího procesu v atributu `PROCESS_INFORMATION`.



Obrázek 3.4: Namapování nahrávání zvuku na práci s procesy

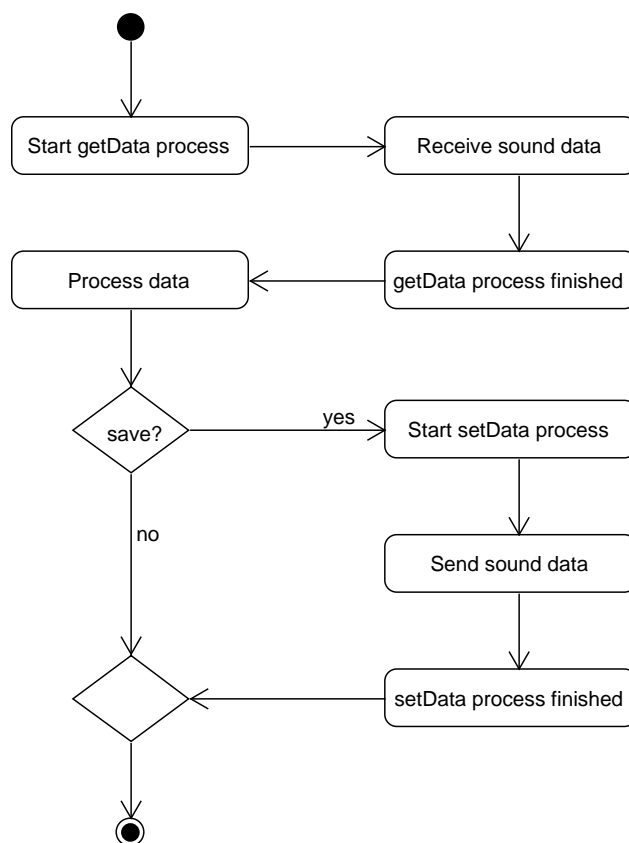
3.1.5 Práce se zvukem

Pracovat se zvukovými soubory se bude dvěma způsoby. Prvním způsobem je práce se zvukovými daty přímo v rozšiřujícím modulu, kdežto druhý způsob využívá pro nakládání se zvukem schopnosti nástroje `ffmpeg`.

První způsob využívá faktu, že práce se zvukovými daty je již implementována v rozšiřujícím modulu `MediaManager`. V tomto případě tedy naimplementujeme funkci umožňující dostat surová zvuková data ze souboru a funkci umožňující opačný postup, tedy zapsání surových zvukových dat do souboru v požadovaném formátu. Požadované funkčnosti ve funkcích docílíme spuštěním procesu s nástrojem `ffmpeg` a připojením `rour` (pipe) ke standardnímu

výstupu a vstupu. Při získávání zvukových dat nasměrujeme výstup nástroje ffmpeg na standardní výstup, odkud již data získáme pomocí roury. V případě zápisu dat do souboru nastavíme nástroj ffmpeg tak, aby namísto vstupního souboru četl data ze standardní vstupu. Na standardní vstup poté již jen zašleme daná data pomocí připravené roury.

Tento způsob práce se zvukem bude využit například pro normalizaci zvuku, nebo pro vykreslení zvukového signálu. V případě vykreslování pak budeme potřebovat pouze první z výše zmíněných funkcí, jelikož výsledkem této operace je grafický, nikoliv zvukový výstup. Obecný případ využití je znázorněn v diagramu 3.5



Obrázek 3.5: Obecný případ práce se zvukovými daty

Druhý způsob práce se zvukovými soubory, využívající funkčnosti nástroje ffmpeg, bude podstatně jednodušší. V tomto případě stačí vytvořit proces spouštějící nástroj ffmpeg s potřebnými parametry a cestou k vstupním a výstupním souborům. Tento proces pak vykoná veškerou práci a v aplikaci pouze počkáme na jeho ukončení.

Tento způsob se dá využít například při slučování několika zvukových souborů dohromady, vystržení části zvukového souboru, nebo při změně formátu zvukového souboru.

3.2 Obraz

V obrazové části se zaměříme především na základní funkcionalitu, jako je změna velikosti, pootočení, převrácení, zkosení, nebo deformace. Další významnou částí je také převod mezi obrazovými formáty.

Jelikož v této části využíváme přímo knihovny `ImageMagick`, není zde potřeba vytvářet funkce pro práci s procesy. Budeme zde využívat přímo funkcionalitu této knihovny prostřednictvím jejího rozhraní `Magick++` pro programovací jazyk `C++`. Dostupné třídy tohoto rozhraní jsou v tabulce 3.1 včetně jejich popisu

Tabulka 3.1: Třídy v rozhraní `Magick++`

Třída	Popis
<code>Blob</code>	Binary Large Object kontejner - třída pro uchování obrázků v podobě binárních dat
<code>CoderInfo</code>	Třída poskytující informace o podporovaných obrázkových formátech
<code>Color</code>	Třída umožňující specifikaci barev
<code>Drawable</code>	Třídy umožňující vykreslování primitivních tvarů
<code>Exception</code>	Třídy vyjímek specifických pro <code>ImageMagick</code> knihovnu
<code>Geometry</code>	Třída umožňující specifikaci geometrických argumentů
<code>Image</code>	Hlavní třída knihovny. Reprezentuje objekt obrázku
<code>Montage</code>	Třída pro vytváření seznamů miniatur
<code>Pixels</code>	Třída umožňující přístup k obrázkům ve formě pixelů
<code>STL</code>	Sada algoritmů pro práci s obrázky v STL kontejnerech
<code>TypeMetric</code>	Třída umožňující specifikaci metriky pro textové fonty

3.2.1 Repräsentace obrázků

Pro reprezentaci obrázků v rozšiřujícím modulu budeme využívat tříd poskytnutých knihovnou `ImageMagick`, a to tříd `Image` a `Blob`. Třída `Image` obsahuje velké množství metod, jež budou využity pro manipulaci s obrázky, kdežto třída `Blob` bude sloužit pro přístup k obrázkům v binární podobě.

3.2.2 Transformace obrázků

Knihovna `ImageMagick` obsahuje široké spektrum funkcionality. Díky tomuto faktu bude vcelku snadné implementovat nahrazované transformace. U ně-

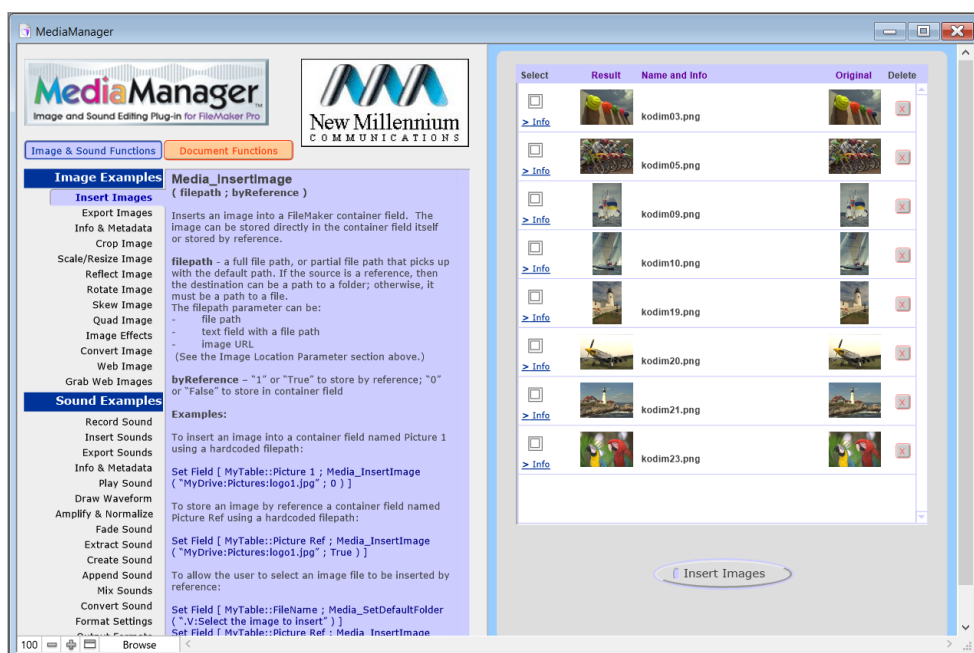
kterých transformací poskytovaných knihovnou jsou vyžadovány parametry odlišně od parametrů vyžadovaných v původním rozšiřujícím modulu. Avšak tyto parametry jsou navzájem převoditelné a nebude tak problém přizpůsobit získávané parametry požadavkům knihovny.

3.2.3 Převod mezi formáty

V případě obrázků bude převod mezi formáty velice jednoduchý. Pro uskutečnění převodu mezi formáty postačí když obrázek nahrajeme od objektu `Image`, nastavíme výstupní formát tohoto objektu pomocí metody `magick()` a objekt následně zapíšeme do souboru, nebo paměti pomocí metody `write()`. Výstupní formát je definován jako textový řetězec obsahující název formátu (JPEG, PNG, GIF, ...).

Testování

Pro účely testování bylo využito zkušební rozhraní rozšiřujícího modulu MediaManager v aplikaci FileMaker 4.1, jenž je volně ke stažení na stránkách společnosti New Millennium. Díky tomuto rozhraní bylo testování změn v rozšiřujícím modulu ulehčeno, jelikož rozhraní umožňuje vyzkoušet veškerou nahrazenou funkcionalitu rozšiřujícího modulu.



Obrázek 4.1: Zkušební rozhraní pro rozšiřující modul MediaManager

Testování kvality manipulace s mediálním obsahem bylo prováděno na základě subjektivního zhodnocení výstupů upraveného rozšiřujícího modulu, nebo na základě srovnání s výstupem původního rozšiřujícího modulu. Tento způsob testování byl zvolen vzhledem k povaze testované funkcionality a tes-

tovaných dat, jenž by bylo velmi obtížné testovat objektivní formou.

4.1 Testování obrazové části

Pro testování obrazové části byly využity obrázky od společnosti Eastman Kodak Company, která tyto obrázky uvolnila pro volné použití. Tyto obrázky jsou používány v mnoha testovacích sadách, převážně pak pro testování komprese. Celkem bylo využito 8 obrázků formátu PNG.

4.1.1 Oříznutí, pootočení a převrácení

Funkce pro oříznutí obrazu, jeho pootočení a převrácení byly testovány jako první vzhledem k jednoduchosti jejich testování. U každého testování byl využit pouze jeden z testovacích obrázků, na který byla použita každá z testovaných funkcí několikrát s různými parametry.

Testování oříznutí bylo provedeno na 70 a 30 % obrázku od horního levého okraje a oříznutí 20 a 30 % z každé strany. Ačkoliv jsou zde hodnoty uvedeny v procentech, pro testování musely být přepočítány na pozici levého horního a pravého dolního okraje, jelikož v rozšiřujícím modulu se pro ořezávání definuje vyřezávaný region. Výsledky jsou viditelné na obrázku 4.2.

Testování pootočení bylo testováno pro úhly: 30°, 45°, 90° a 180°. Byly zvoleny takové úhly, aby výsledky transformací s použitím čtvercového obrázku bylo možné spolehlivě zhodnotit při pouhém pohledu na ně. Výsledky těchto transformací jsou zobrazeny na obrázku 4.3.

U funkce převrácení byly vyzkoušeny všechny tři možnosti: horizontální a vertikální převrácení a kombinace obou předešlých. V tomto případě bylo vyhodnocení výsledků jednoduché, jelikož se jedná o velice přímočarou transformaci. Výsledky jsou vidět na obrázku 4.4

Testování této části se obešlo bez problémů a bylo vyhodnoceno jako úspěšné.

4.1.2 Změna velikosti, zkosení a zkreslení

U těchto funkcí bylo již testování těžší a to hlavně proto, že tyto funkce obsah obrazu deformují. U změny velikosti se tato deformace projevuje hlavně při zmenšování, jelikož je funkce nucena vměstnat stejné informace o obrazu do menšího prostoru. Tím u obrazu dochází ke ztrátě detailů.

U funkce pro změnu velikosti byl vybraný obrázek nejdříve zmenšen na 50, 25 a 10 %. Tyto zmenšeniny byly poté opět převedeny na původní velikost a porovnány s originálem. Míra ztráty detailů oproti originálu je dle mého názoru v očekávaných mezích. Výsledky změn velikosti jsou k porovnání na obrázku 4.5.

U testování funkce pro zkosení byla sledována míra deformace. Čtvercový obrázek byl zkosen ve směru osy x o 50, 75 a 100 % své velikosti. Výsledky

transformací v porovnání s originálem jsou na obrázku 4.6. U žádné z transformací jsem nepozoroval výsledky, jež bych považoval za chybné, obrázky si zachovávají míru detailů.

Funkce pro zkreslení obrázku umožňuje uživateli definovat nové pozice rohových pixelů a na základě těchto informací transformovat celý obrázek. Pro účely testování vyzkoušíme pouze zmenšení pravé hrany obrázku na 50 %. Výsledek je vidět na obrázku 4.7. Obrázek si zachovává míru detailů při transformaci, jež nevyžaduje výrazné zmenšení obrázku v některé z dimenzí.

4.2 Testování zvukové části

Testování zvukové části probíhalo na deseti vybraných hudebních souborech a dvou zvukových záznamech vytvořených pomocí rozšiřujícího modulu.

Hlavní testování se týkalo funkcionality přehrávání a zaznamenávání zvukových souborů. Za účelem otestování nahrávání byly vytvořeny 2 nahrávky, které byly následně také přehrány pomocí rozšiřujícího modulu. Zvuk se zaznamenal v pořádku a v ucházející kvalitě. Pro otestování přehrávání byly přehrány testovací soubory a porovnány s přehráním v původním rozšiřujícím modulu. Při porovnání nebyly zaznamenány rozdíly.

Dále také byly otestovány funkce pro spojování zvukových souborů, vystřížení části záznamu, amplifikaci, normalizaci a funkce pro vytvoření efektu postupného ztišení, či náběhu. Opět ani zde nebyly nalezeny žádné zásadní rozdíly oproti původnímu rozšiřujícímu modulu.

4. TESTOVÁNÍ



(a) Originál



(b) 70 % obrázku od levého horního okraje



(c) 30 % obrázku od levého horního okraje



(d) Ořiznuto 20 % z každé strany



(e) Ořiznuto 30 % z každé strany

Obrázek 4.2: Výsledky testu ořezávání



(a) Originál



(b) Pootočeno o 30°



(c) 30 % obrázku od levého horního okraje



(d) Ořiznuto 20 % z každé strany



(e) Ořiznuto 30 % z každé strany

Obrázek 4.3: Výsledky testu pootočení

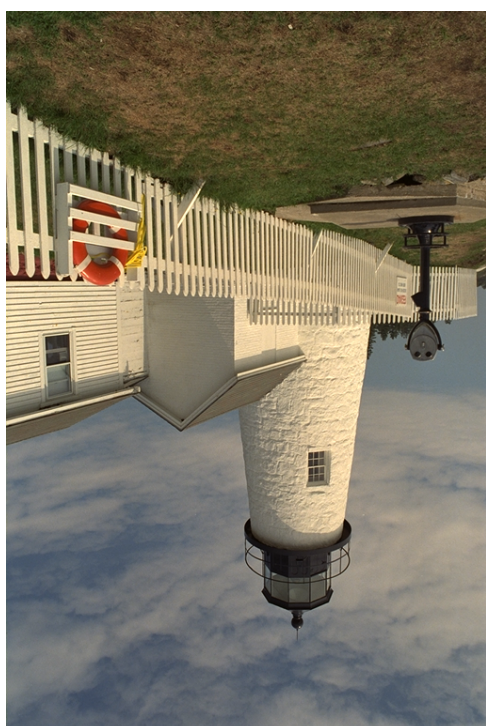
4. TESTOVÁNÍ



(a) Originál



(b) Horizontální převrácení



(c) Vertikální převrácení



(d) Kombinace horizontálního a vertikálního převrácení

Obrázek 4.4: Výsledky testu převrácení



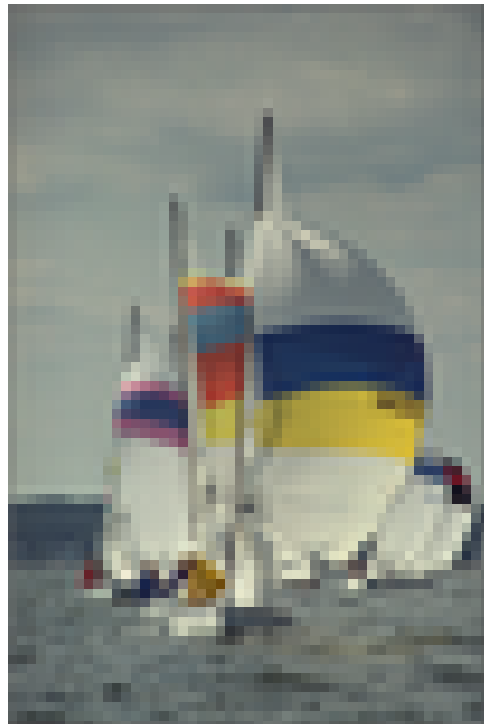
(a) Originál



(b) 50% velikost



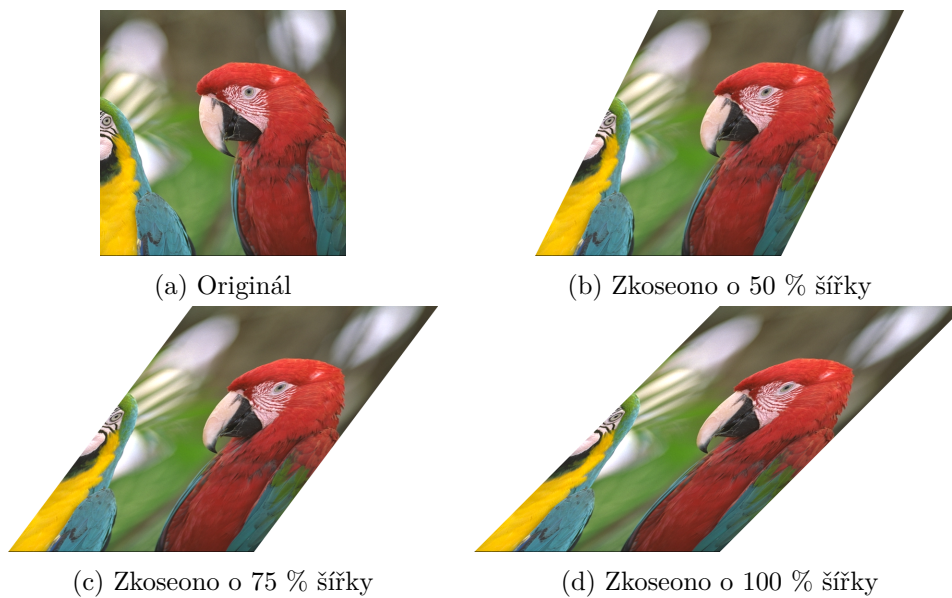
(c) 25% velikost



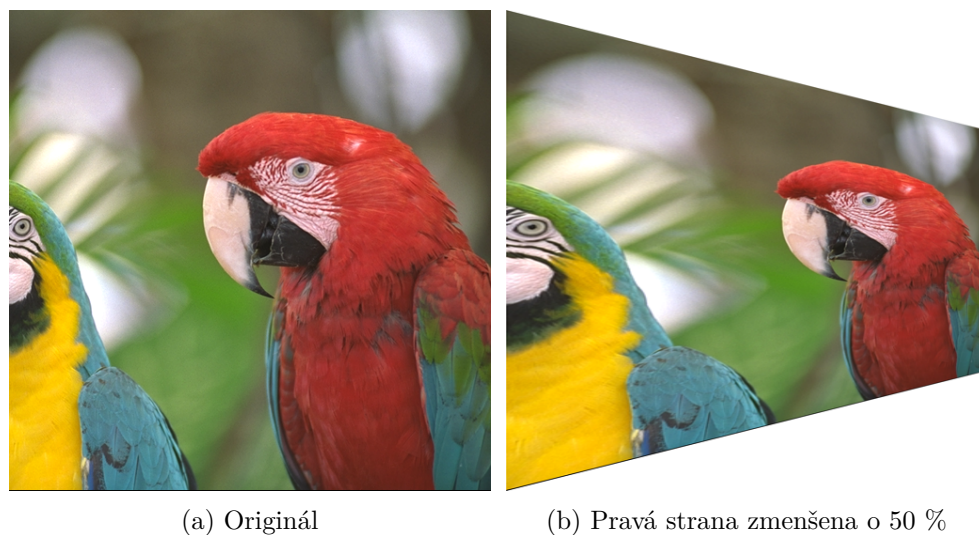
(d) 10% velikost

Obrázek 4.5: Výsledky testu změny velikosti

4. TESTOVÁNÍ



Obrázek 4.6: Výsledky testu zkosení



Obrázek 4.7: Výsledky testu zkreslení

Závěr

Práce se zabývala nahrazením částí funkcionality rozšiřujícího modulu MediaManager pro aplikaci FileMaker, jež využívaly knihovny QuickTime. Jednalo se o funkcionality související s manipulací s multimediálním obsahem - obrázky a zvukem. Pro tento účel byly vybrány knihovny ImageMagick a FFmpeg. Knihovna ImageMagick byla použita v části pro manipulaci s obrázky, kdežto FFmpeg byl využit pro manipulaci se zvukovým obsahem. Jelikož licence FFmpeg nedovolovala využití ve formě knihovny v uzavřeném komerčním programu, jako je právě rozšiřující modul MediaManager, byl FFmpeg využit externě za pomoci vytváření a správy externích procesů.

Ačkoliv testování nahrazované funkčnosti probíhalo na základě subjektivního hodnocení výstupů, domnívám se, že vzhledem k povaze testovaných funkčností byla volba tohoto způsobu testování adekvátní.

Nahrazení knihovny QuickTime v rozšiřujícím modulu, jež bylo provedeno v rámci této práce, nebylo úplné, avšak položilo solidní základy pro nahrazení zbytku funkcionalit rozšiřujícího modulu.

Literatura

- [1] Sam: The History of FileMaker. [online], [cit. 2017-03-31]. Dostupné z: <https://filemakerexamples.co.uk/blog/2016/04/04/the-history-of-filemaker/>
- [2] CUMMING, M.: Writing FileMaker Plug-ins. [online], [cit. 2017-04-08]. Dostupné z: <http://www.murrayc.com/permalink/2002/03/13/writing-filemaker-plug-ins/>
- [3] Milenium, N.: MediaManager. [online], [cit. 2017-04-08]. Dostupné z: <http://www.newmillennium.com/mediamanager/>
- [4] Wikipedia: QuickTime. [online], [cit. 2017-04-08]. Dostupné z: <https://en.wikipedia.org/wiki/QuickTime>
- [5] NOVÁK, J.: Zvukový mixer použitelný místo QuickTime. 2004, [cit. 2017-04-09].
- [6] TSCHUMPERLÉ, D.: CImage. [online], [cit. 2017-04-22]. Dostupné z: <http://cing.eu/>
- [7] ImageMagick Studio LLC: ImageMagick. [online], [cit. 2017-04-22]. Dostupné z: <https://www.imagemagick.org/script/index.php>
- [8] OpenCV team: OpenCV. [online], [cit. 2017-04-22]. Dostupné z: <http://opencv.org/about.html>
- [9] CUPITT, J.; MARTINEZ, K.: VIPS. [online], [cit. 2017-04-22]. Dostupné z: <http://www.vips.ecs.soton.ac.uk/index.php?title=VIPS>
- [10] BURK, P.: PortAudio. [online], [cit. 2017-04-23]. Dostupné z: <http://www.portaudio.com/>
- [11] FFmpeg: FFmpeg. [online], [cit. 2017-04-09]. Dostupné z: <http://ffmpeg.org/about.html>

LITERATURA

- [12] FFmpeg: ffmpeg documentation. [online], [cit. 2017-04-22]. Dostupné z: <http://ffmpeg.org/ffmpeg.html>
- [13] FFmpeg: ffmpeg documentation. [online], [cit. 2017-04-22]. Dostupné z: <http://ffmpeg.org/ffplay.html>
- [14] FFmpeg: ffmpeg documentation. [online], [cit. 2017-04-22]. Dostupné z: <http://ffmpeg.org/ffprobe.html>
- [15] Universitat Pompeu Fabra: Essentia. [online], [cit. 2017-04-23]. Dostupné z: <http://essentia.upf.edu/documentation/>
- [16] Ambiera: IRRKlang. [online], [cit. 2017-04-23]. Dostupné z: <http://www.ambiera.com/irrklang/>

Seznam použitých zkratek

- API** Application programming interface
- ASCII** American standard code for information interchange
- BSD** Berkeley Software Distribution
- CeCILL** CEA CNRS INRIA Logiciel Libre
- CSV** Comma-separated values
- E-R** Entity-relationship
- GPL** General public license
- HTML** Hypertext markup language
- JSON** Javascript object notation
- LGPL** Lesser general public license
- PCM** Pulse code modulation
- STL** Standard template library
- VoIP** Voice over internet protocol
- XML** Extensible markup language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe	adresář se spustitelnou formou rozšiřujícího modulu
	MediaManager.zip...	zabalená spustitelná forma rozšiřujícího modulu
	src	
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF