

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

**Porovnání knihoven pro tvorbu GUI
v C/C++**

Diplomová práce

BC. JAN BARTOŠEK

Brno, jaro 2023

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

Porovnání knihoven pro tvorbu GUI v C/C++

Diplomová práce

BC. JAN BARTOŠEK

Vedoucí práce: Mgr. Luděk Bártek, Ph.D.

Katedra počítačových systémů a komunikací

Brno, jaro 2023

MUNI
FAKULTA
INFORMATIKY

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a použil jen prameny uvedené v seznamu literatury.

Bc. Jan Bartošek

Vedoucí práce: Mgr. Luděk Bártek, Ph.D.

Poděkování

Chtěl bych poděkovat vedoucímu své diplomové práce za možnost vypracovat toto téma a všechny odborné rady, kterými mě vedl při psaní této práce. A především pak své přítelkyni, ale i rodině a přátelům, kteří mě neochvějně podporovali až do samého konce.

Abstrakt

Diplomová práce pojednává o třech hlavních C++ GUI knihovnách - Qt, WxWidgets a Gtkmm. Tyto knihovny jsou demonstrovány na aplikaci hudebního přehrávače. Cílem práce je poskytnout užitečný přehled a porovnání zmíněných knihoven, které jsou často používány při vývoji desktopových aplikací s grafickým uživatelským rozhraním.

Práce začíná seznámením čtenáře s aplikací hudebního přehrávače. Definuje chování i grafické rozložení všech ovládacích prvků. Následuje praktická implementace podle popsaného návrhu a informace o fungování použité hudební knihovny i systému pro překlad a nasazení. Jsou zobrazeny rozhraní, která slouží ke komunikaci mezi oddělenými moduly aplikace. V další části je popsána implementace jednotlivých knihoven pro uživatelské rozhraní, rozdíly a překážky, kterým programátor čelil při jejich používání.

Výsledná aplikace je následně testována uživateli za účelem zjištění poznatků a dojmů od koncového uživatele. Společně s důsledným porovnáním knihoven z pohledu jejich dostupných nástrojů, podpory a podmínek je vytvořeno hodnocení pojednávající o kvalitách každé knihovny.

Výsledky této práce budou složité jako cenný zdroj informací pro vývojáře, kteří se rozhodují mezi použitím některé z uvedených C++ GUI knihoven. Práce také přispěje k lepšímu porozumění výhod a nevýhod každé knihovny a doporučí výběr knihovny pro budoucí projekty.

Klíčová slova

C++, GUI knihovny, Qt, WxWidgets, Gtkmm, hudební přehrávač, porovnání, desktopové aplikace.

Obsah

Úvod	1
1 Aplikace hudebního přehrávače	3
1.1 Popis chování	3
1.1.1 Přidání hudby do seznamu skladeb	3
1.1.2 Přehrávání hudby	3
1.1.3 Ovládání hlasitosti	4
1.2 Grafické rozhraní	5
2 Implementace aplikace	8
2.1 Architektura	8
2.2 Rozhraní pro zvukovou knihovnu	9
2.2.1 SDL_Mixer	10
2.3 Rozhraní funkční logiky	11
2.4 Rozhraní pro GUI	13
2.5 Build system	15
2.5.1 Závislosti programu	16
2.5.2 Překlad a spuštění v prostředí Linux	16
2.5.3 Překlad a spuštění v prostředí Windows	17
3 Popis implementace jednotlivých GUI knihoven	18
3.1 Qt	18
3.2 wxWidgets	21
3.3 Gtkmm	23
4 Porovnání implementace GUI	25
4.1 Grafické rozhraní	25
4.2 Podpora Cross-Platform	26
4.2.1 Qt [8]	26
4.2.2 wxWidgets [9]	27
4.2.3 Gtkmm [10]	27
4.3 Nasazení aplikace	27
4.4 Licenční podmínky	28
4.5 Kvalita dokumentace	28
4.6 Aktivita komunity	30

4.7	Historie	32
4.7.1	Qt	32
4.7.2	wxWidgets	32
4.7.3	Gtkmm	33
4.8	Známé programy využívající C++ GUI	33
4.8.1	Qt	33
4.8.2	wxWidgets	34
4.8.3	Gtkmm	34
5	Testování aplikace uživateli	36
6	Hodnocení	38
A	Příloha	41
	Bibliografie	55

Seznam tabulek

6.1	Hodnocení vybraných knihoven GUI programátorem. . .	38
6.2	Hodnocení vybraných knihoven GUI uživatelem na plat- formě Linux.	38
6.3	Hodnocení vybraných knihoven GUI uživatelem na plat- formě Windows.	39

Seznam obrázků

1.1	Mockup programu vytvořený na webu Figma	5
2.1	Schéma architektury	9
3.1	Qt Designer	20
3.2	Glade	23
4.1	Google Trends	31
A.1	Linux verze knihovny Qt	41
A.2	Linux verze knihovny wxWidgets	42
A.3	Linux verze knihovny Gtkmm	42
A.4	Linux verze knihovny Qt	43
A.5	Linux verze knihovny wxWidgets	43
A.6	Linux verze knihovny Gtkmm	43
A.7	Windows verze knihovny Qt	44
A.8	Windows verze knihovny Qt	45
A.9	Windows verze knihovny WxWidgets	46
A.10	Windows verze knihovny WxWidgets	47
A.11	Chyba načítání obrázků na Windows knihovny WxWidgets	47
A.12	Windows verze knihovny Gtkmm	48
A.13	Windows verze knihovny Gtkmm	49
A.14	UML: Diagram tříd	50
A.15	Hodnocení uživatelů na platformě Linux	51
A.16	Hodnocení uživatelů na platformě Windows	52
A.17	Preferovaná knihovna na platformě Linux	53
A.18	Preferovaná na platformě Windows	54

Úvod

Tato práce vznikla za účelem porovnání nejběžněji používaných knihoven pro tvorbu grafického uživatelského rozhraní s podporou programovacího jazyku C++. Dnes je k dispozici velké množství dostupných knihoven pro tvorbu grafického rozhraní, mezi kterými může programátor vybírat při tvorbě svého programu. Všechny knihovny mají svá specifika. Tato práce se bude zabývat rozdíly vybraných knihoven z vícero různých pohledů. A to přesněji z pohledu výkonu, efektivity a přehlednosti práce s jejími nástroji, odlišnosti jejich grafického rozhraní, historie, podpory pro různé operační systémy a dalších. U jednotlivých knihoven budou zmíněna jak fakta, která se vážou k licenčnímu použití a poplatkům za použití, ale také kvalita dokumentace, případně podpora ze strany komunity.

Zmíněné porovnání existujících rozhraní bude demonstrováno na aplikaci, která bude podporovat přehrávání zvukových souborů. Tato aplikace bude fungovat jako zvukový přehrávač pro zvukové soubory (soubory ve formátu .mp3). Součástí funkcionality budou i rozšířené možnosti přehrávání více zvukových souborů v řadě pomocí seznamu skladeb (angl. playlist) s náhodným nebo sekvenčním řazením. Rozhraní bude umožňovat přehrávání následující nebo předchozí skladby, pozastavení přehrávání, pokračování v přehrávání, zastavení přehrávání, opakované přehrávání jedné či více skladeb a operace pro zvolení hlasitosti přehrávání.

V první kapitole této práce čtenář získá podrobný náhled na budovanou aplikaci. Kapitola dále pojednává o její specifikaci i architektuře. Další částí je detailní popis implementace a její rozdělení pro nezávislé celky business logiky a grafického rozhraní. Je dán důraz na správný a efektivní návrh systému podle všech vyučovaných zásad SOLID [1]. Popsány jsou i použité závislosti této aplikace, jako jsou zvukové knihovny nebo požadované verze grafických rozhraní. Návrh implementace je vyobrazen na přehledných obrázcích podle pravidel Unified Modelling Language diagramů. Třetí kapitola obsahuje popis vybraných grafických uživatelských knihoven. Hlavní důraz je kladen na popis při jejich implementaci, konkrétní třídy a nástroje použité pro grafické rozhraní. Čtvrtá kapitola je věnována důkladnému testování aplikace s využitím nástrojů pro statickou a dynamickou analýzu

kódu. Pátá kapitola slouží k porovnání jednotlivých implementací knihoven pro grafické rozhraní z pohledu předem zmíněných specifik. V poslední kapitole bude uvedeno celkové hodnocení, jak z pohledu programátora, tak uživatelů na jednotlivé nástroje. V závěru je shrnutí všech zjištěných informací.

Součástí práce jsou veškeré zdrojové soubory, bibliografické citace a fotografie všech implementací.

1 Aplikace hudebního přehrávače

1.1 Popis chování

Aplikace umožňuje uživatelům přehrávání zvukových souborů ve formátech .mp3. Po zapnutí programu se uživateli zobrazí prázdný seznam skladeb. Tlačítka, která ovládají výběr skladby, přehrávání, pozastavení a další operace vázající se k přehrávání hudby, budou deaktivována do doby, dokud uživatel nepřidá svoji skladbu do seznamu k přehrání. Omezená funkčnost ovládacích prvků se netýká ovládání hlasitosti přehrávání ani tlačítka ztišení, která jsou k dispozici vždy.

1.1.1 Přidání hudby do seznamu skladeb

K přidání skladby do seznamu skladeb slouží funkce `Open Files` a `Open Folder`. Tyto operace se nachází v menu umístěném v hlavičce programu, v první sekci `File`. Obě operace pracují se systémovým správcem souborů a otevřou nabídku pro volbu v novém okně. První ze zmíněných funkcí, `Open Files`, slouží k vybrání konkrétního zvukového souboru nebo souborů, které chceme přidat do seznamu skladeb. Druhá možnost `Open Folder` vybere cestu ke složce, z níž budou načteny všechny podporované zvukové soubory a následně přidány do seznamu skladeb. Cesta k vybrání hudby (adresářová struktura, která se otevře zobrazí při otevření prohlížeče souborů) je shodná pro obě možnosti a její výchozí hodnota je nastavena na adresu testovacích skladeb přiložených ke zdrojovým kódům: `/sounds/`. Po vybrání hudebního souboru nebo složky s hudebními soubory z jiné adresy se tato adresa stane novou výchozí hodnotou, která se zobrazí při příštím výběru.

1.1.2 Přehrávání hudby

Přehrávání se spouští tlačítkem `Play`. Přehrávat se začne pozastavená písnička nebo posledně vybraná píseň v seznamu skladeb. Pokud uživatel nevybral žádnou píseň ze seznamu, tak se zvolí první skladba. Alternativně je možné začít přehrávat píseň dvojklikem na skladbu v seznamu. V momentě, kdy se píseň začne přehrávat, tak se nastaví aktuální čas přehrávání, délka přehrávané skladby a zpřístupní se

posuvník pro ovládání pozice v rámci skladby. Změnou pozice posuvníku lze přeskočit na konkrétní část přehrávané skladby. Tu lze pozastavit funkcí Pause nebo zastavit funkcí Stop. Zatímco pozastavená píseň v případě opětovného spuštění pokračuje v čase kdy byla pozastavena, zastavení písně znamená, že přehrávání začne od začátku skladby.

Pro posun v rámci seznamu skladeb slouží funkce vpřed Forward a zpět Backward. Podle výchozího nastavení tyto funkce odpovídají přehrávání následující nebo předchozí skladby. Toto chování je možné upravit změnou zvolené smyčky přehrávání Loop nebo pořadí přehrávání Playback. Smyčka přehrávání umožňuje tři hodnoty. Jsou to:

- přehrávání všech skladeb ve smyčce (výchozí),
- přehrávání jedné skladby ve smyčce,
- přehrávání všech skladeb pouze jednou.

Změna pořadí přehrávání má dvě dostupné hodnoty, a to buď sekvencí Sequential, a nebo náhodné Random. Při volbě náhodného řazení se při zvolení možnosti vždy vytvoří nové pořadí pro přehrávání skladeb. Zároveň se přehrají všechny skladby v seznamu neohledně na jejich pořadí. Příkladem může být přehrávání všech skladeb bez smyčky, kdy se při náhodném řazení přehraje nejdříve poslední písnička v seznamu. Přehrávání v takovém případě neskončí a bude pokračovat do doby, dokud se nepřehrají všechny písně v seznamu.

Všechny popsané funkce k přehrávání hudby jsou dostupné i v menu, v sekci Audio. Tam je na rozdíl od ovládání pomocí tlačítek možné zvolit konkrétní hodnotu pro přehrávání ve smyčce.

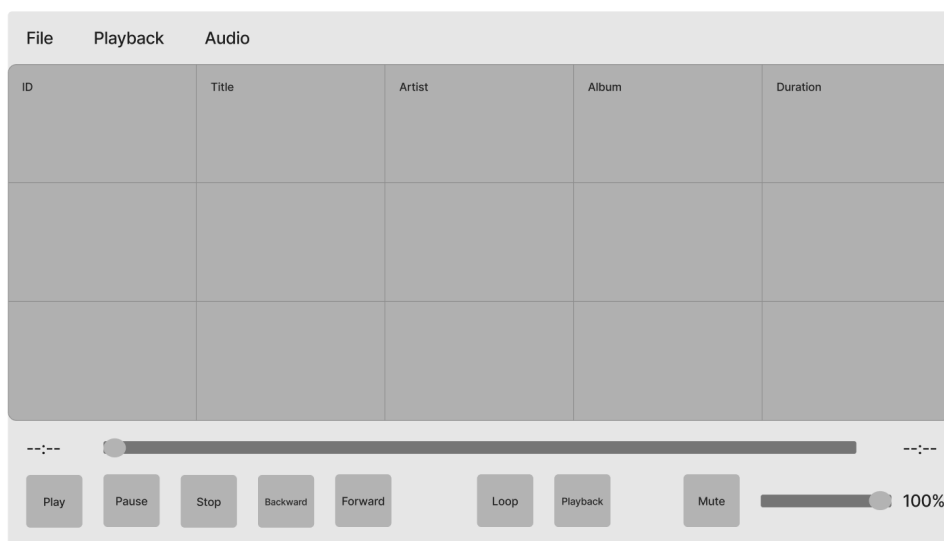
1.1.3 Ovládání hlasitosti

Pro ovládání úrovně hlasitosti slouží posuvník s minimální hodnotou 0 (zvuk je ztišený) a maximální hodnotou 100. Uživatel může využít i funkce ztlumení Mute, která funguje jako přepínač mezi aktuální úrovní hlasitosti a úplným ztišením. V menu je možné použít obdobu posuvníku funkcemi pro přidání nebo snížení hlasitosti o 20 bodů a funkci Mute.

1.2 Grafické rozhraní

Rozvržení všech ovládacích prvků bylo silně inspirováno jinými, již existujícími a na trhu oblíbenými aplikacemi. Snahou při vytváření designu totiž bylo maximální porozumění uživatelem a využití běžně používaných konceptů, které mohou pomoci s jednodušší navigací při ovládání programu. Tento přístup se odráží v poskládání funkcí v menu, ovládáním posuvníků i zvolenými obrázky.

Před samotnou implementací v grafických knihovnách byl vytvořen grafický návrh aplikace, který je možné vidět na obrázku 1.1. Následně při tvorbě programu v jednotlivých knihovnách byl tento návrh vzorem pro pozicování všech ovládacích prvků. Snahou při tomto postupu bylo, aby při porovnávání vzhledu aplikací nehrálo roli jejich různorodé ztvárnění a uživatel nemusel hledat ovládací prvky na jiných místech.



Obrázek 1.1: Mockup programu vytvořený na webu Figma

Program obsahuje typicky umístěné menu pro desktopové aplikace v horní liště. Struktura tohoto menu je následující:

```
Menu
├── File
│   ├── Open Files
│   ├── Open Directory
│   └── Quit
├── Playback
│   ├── Play
│   ├── Pause
│   ├── Stop
│   ├── Backward
│   ├── Forward
│   ├── Loop
│   │   ├── None
│   │   ├── Single
│   │   └── All
│   └── Playback
│       ├── Sequential
│       └── Random
├── Audio
│   ├── Increase Volume
│   ├── Decrease Volume
│   └── Mute
└── Gui
    ├── Switch to Gui 2
    └── Switch to Gui 3
```

Pod menu se nachází tabulka zobrazující všechny přidané skladby. Ve sloupcích se zobrazují informace o čísle řádku, názvu písni, autovi, albu a délce skladby. Dalším ovládacím prvkem je posuvník času ohraničený textem z obou stran. Tato textová pole zobrazují aktuální čas přehrávané skladby vlevo a celkový čas skladby vpravo. V posledním řádku jsou seřazena tlačítka Play, Pause, Stop, Backward, Forward, Loop, Playback, Mute a posuvník hlasitosti s ukazatelem aktuální hodnoty. Tento posuvník i hodnota hlasitosti reagují na aktivaci ztišení. Když je funkce ztišení aktivní, tak se posuvník zablokuje a barva textu se změní na šedou. Dvojice tlačítek Mute a Playback jsou zaklikávací, tedy mění svůj vzhled podle toho, jestli je daná funkce aktivní nebo nikoliv.

Tlačítka jsou zobrazeny s názornými ikonami místo textu následovně:

-  Play
-  Pause
-  Stop
-  Backward
-  Forward
-  Loop All
-  Loop Single
-  Loop None
-  Playback Random
-  Playback Sequential
-  Mute

Všechny použité ikony jsou poskytnuté z webu FontAwesome. [2]

2 Implementace aplikace

2.1 Architektura

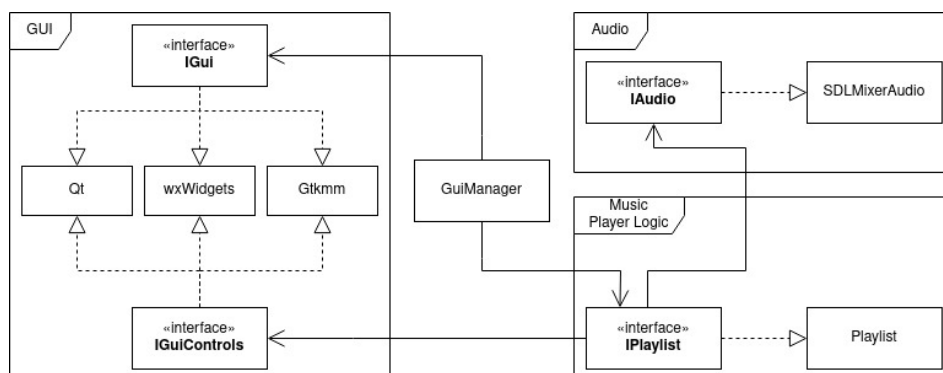
Aplikace byla navržena s cílem maximalizovat škálovatelnost a uplatnit všechny praktiky správného vývoje software vyučovaného na fakultě, jako je například dodržování SOLID pravidel. Program je rozdělen na 3 logické celky. Jsou jimi implementace zvukové knihovny, logika aplikační vrstvy a grafické rozhraní. Rozložení je viditelné na obrázku 2.1.

První ze zmíněných částí, implementace zvukové knihovny, se stará o správnou kooperaci programu a operačního systému. Její hlavní funkcí je samotné přehrávání zvukového souboru, určení hlasitosti a přečtení metadat ze souboru. Je implementováno knihovnou `SDL_Mixer` a detailněji popsána v části 2.2.1.

Ve druhé části se pod aplikační logikou rozumí backend aplikace, tedy její chování a reakce na vstupy od uživatele. Tato logika zaštiťuje komunikaci mezi grafickým rozhraním a implementací zvukové knihovny. Má v sobě uložený seznam přidané hudby, aktuální stav přehrávané písně, nastavení hlasitosti i možnosti přehrávání. Tento přístup usnadňuje přepínání mezi jednotlivými implementacemi grafických knihoven, protože při přepínání mezi knihovnami lze jednoduše nastavit všechny hodnoty z paměti.

Poslední částí je grafické rozhraní. To je implementováno hned několikrát, pro každou použitou grafickou knihovnu právě jednou. Grafické rozhraní je jedinou částí aplikace, se kterou uživatel přichází do styku. Nedrží v sobě žádnou logiku, pouze předává zprávu o určitém úkonu vrstvě aplikační logiky a čeká na její reakci. Pokud by grafická část držela data, pak by se tyto hodnoty musely kopírovat v momentě přepínání mezi různými knihovnami.

Celé schéma programu je na obrázku A.14.



Obrázek 2.1: Schéma architektury

2.2 Rozhraní pro zvukovou knihovnu

Rozhraní pro zvukovou knihovnu se nazývá `IAudio` a definuje funkce, které používá vrstva aplikační logiky k přehrávání zvukových souborů. Těmito funkcemi jsou:

- `init`
Inicializuje hudební knihovnu a nastavuje frekvenci, kanál a velikost vzorky zvukových souborů.
- `play`
Přehrává zvolený zvukový soubor.
- `pause`
Pozastaví přehrávání zvukového souboru.
- `resume`
Pokračuje v pozastaveném přehrávání zvukového souboru.
- `stop`
Zastaví přehrávání zvukového souboru.
- `isPlaying`
Vrací logickou hodnotu, jestli je právě přehrávána nějaká skladba.

- `setVolume`
Nastavuje hlasitost přehrávání.
- `setPosition`
Nastavuje posun v rámci skladby.
- `getSongDetail`
Pro zvolený zvukový soubor vrací údaje název souboru, název skladby, název alba, název umělce, dobu skladby ve vteřinách a dobu skladby zapsanou jako řetězec času, uložené ve struktuře `SongDetail`.
- `getPosition`
Vrací aktuální posuv v rámci skladby.

Instance tohoto rozhraní se vytváří v bloku hlavního souboru `main.cpp`. Používá se implementace knihovnou `SDL_mixer` [3] a vytvořená instance se posílá jako parametr konstruktoru aplikační logiky `IPlaylist`, kde se uloží a využívá po čas běhu aplikace.

2.2.1 SDL_Mixer

`SDL_Mixer` je knihovna, která poskytuje jednoduché rozhraní pro přehrávání a mixování zvukových souborů v počítačových hrách a jiných aplikacích. `SDL_Mixer` je volně dostupný open-source software licencovaný pod LGPL (Lesser General Public Licence) [4]. To znamená, že vývojáři jej mohou použít zdarma ve vlastních projektech, včetně modifikace a distribuce.

Hlavní výhodou použití knihovny `SDL_Mixer` při implementaci zvukového přehrávače je jednoduchost jejího rozhraní a podpora operačních systémů Windows a Linux.

V tomto projektu je vyžadována verze minimálně 2.6.0, z důvodu využití nových funkcí, které dovedou vyčíst metadata ze zvukového souboru. Díky nim pak může aplikace zobrazit informace o délce, názvu, umělci a albu načtené skladby. Tato verze je dostupná v linuxové distribuci Arch. V distribucích Debian se může stále nacházet starší verze 2.0 a je potřebné si stáhnout novější verzi od vývojářů a nainstalovat ručně. [5] [6]

2.3 Rozhraní funkční logiky

Logika aplikace je implementována ve třídě `Playlist`. Tato třída používá rozhraní `IPlaylist`, které slouží pro komunikaci ze strany GUI a využívá k tomu následující funkce:

- `registerGuiControls`

Funkce se volá po vytvoření nového okna aplikace. Tím o sobě okno dává vědět a `Playlist` si uloží toto nové okno pro komunikaci. Zároveň nastaví všechny hodnoty, jako je úroveň hlasitosti, seznam přidanych skladeb, zvolená skladba, čas jejího přehrávání a další.

- `removeGuiControls`

Při rušení okna aplikace se zruší její ukazatel, aby se předešlo přístupu k odstraněnému objektu.

- `playButtonClicked`

Pokud bylo pozastaveno přehrávání, tak se pokračuje od přerušené pozice. V opačném případě se začne přehrávat skladba od začátku, zvýrazní se její řádek v tabulce, nastaví se rozmezí posuvníku pozice skladby a hodnoty štítků z jeho stran.

- `stopButtonClicked`

Zastaví přehrávání skladby, zruší označení v tabulce skladeb a nastaví hodnoty pozice na výchozí.

- `nextButtonClicked`

Přehrává se následující skladba v pořadí.

- `prevButtonClicked`

Přehrává se předcházející skladba v pořadí.

- `muteButtonClicked`

Pokud byla funkce ztišení aktivní, tak je deaktivována a hlasitost nastavena na předchozí hodnotu. V opačném případě je aktivována a přehrávání hudby nepůjde slyšet, jako kdyby byla hlasitost nastavena na 0.

- `loopButtonClicked`
Změní nastavení pro přehrávání ve smyčce v následujícím pořadí:
přehrávání všech skladeb ve smyčce > přehrávání jedné skladby
ve smyčce > přehrávání všech skladeb pouze jednou.
- `pauseButtonClicked`
Pozastaví přehrávání skladby.
- `playingOrderButtonClicked`
Změní se pořadí přehrávání skladeb z posloupného přehrávání
na náhodně a naopak.
- `increaseVolumeButtonClicked`
Zvedne úroveň hlasitosti o 20 bodů, do maxima 100.
- `decreaseVolumeButtonClicked`
Sníží úroveň hlasitosti o 20 bodů, do minima 0.
- `loopNoneButtonClicked`
Nastaví hodnotu přehrávání na přehrávání všech skladeb ve
smyčce.
- `loopSingleButtonClicked`
Nastaví hodnotu přehrávání na přehrávání jedné skladby ve
smyčce.
- `loopAllButtonClicked`
Nastaví hodnotu přehrávání na přehrávání všech skladeb pouze
jednou.
- `playingOrderSequencialButtonClicked`
Nastaví hodnotu pořadí přehrávání na posloupnou.
- `playingOrderRandomButtonClicked`
Nastaví hodnotu pořadí přehrávání na náhodnou.
- `volumeSliderChanged`
Změní hodnotu hlasitosti.

- `timeSliderChanged`
Změní aktuální pozici skladby.
- `selectionChanged`
Změní výběr skladby a označí řádek v tabulce skladeb.
- `musicAdded`
Načte informace o zvukovém souboru a přidá jej do seznamu skladeb.
- `songFinishedPlaying`
Reakce na událost, že přehrávání skladby bylo dokončeno. Začne přehrávat následující skladbu v pořadí.
- `getMusicPath`
Vrátí cestu ke zvukovým souborům. Tuto hodnotu používá okno aplikace při zvolení možnosti přidání skladby. Funkce `Open Files` a `Open Folder`.
- `setMusicPath`
Nastavuje cestu ke zvukovým souborům.
- `startGui`
Volá funkci nastavenou pomocí `setGuiFunction`.
- `setGuiFunction`
Nastavuje vnitřní proměnou typu funkce, která ve třídě `GuiManager` nastaví knihovnu, která se má spustit po ukončení současného okna.

2.4 Rozhraní pro GUI

V systému existují 2 rozhraní, které jsou spojená s implementací grafického rozhraní. Tím prvním je `IGui` a slouží pro třídu, která vytváří a spouští vlákno grafické knihovny. Tohle rozhraní definuje pouze funkci `run`, jejíž parametrem jsou instance `IPlaylist` a vstupní parametry programu `argc` a `argv`. Podstatou tohoto rozhraní je, aby

třída, která spouští grafická okna, nemusela mít závislost na grafické knihovně. Proto bylo vytvořeno jednoduché rozhraní, které zaručuje inicializaci a start hlavního okna aplikace. Výjimkou je implementace grafické knihovny `wxWidgets`, protože inicializace hlavního okna probíhá v přetížené metodě `OnInit` ve třídě, která musí dědit `wxApp`. V tomto případě by musela třída implementující rozhraní `IGui` dědit i `wxApp`, čímž by se zanesla závislost do definice třídy. To by znamenalo přidání závislosti na grafické knihovně do sekce, která vytváří instanci `IGui`.

Druhé rozhraní, `IGuiControls`, definuje všechny funkce, které umožňují nastavit stav ovládacích prvků grafického rozhraní. Jsou jimi funkce:

- `addSong`
Do tabulky skladeb přidává nový řádek s názvem, jménem interpreta, alba a délkou skladby.
- `enableTimeSlider`
Povoluje nebo omezuje použití posuvníku pozice skladby.
- `selectRow`
Označuje řádek v tabulce skladeb.
- `clearRowSelection`
Odstraňuje označení řádku v tabulce skladeb.
- `setMuted`
Nastavuje nebo ruší nastavení funkce `mute`.
- `setTimeSlider`
Nastavuje hodnotu posuvníku pozice skladby.
- `setVolumeSlider`
Nastavuje hodnotu posuvníku hlasitosti.
- `setTimeSliderRange`
Nastavuje rozmezí hodnot, kterých může nabývat posuvník pozice skladby.

- `setVolumeLabel`
Nastavuje hodnotu štítku vedle posuvníku hlasitosti.
- `setTimeLeftLabel`
Nastavuje hodnotu štítku vlevo od posuvníku pozice skladby.
- `setTimeRightLabel`
Nastavuje hodnotu štítku vpravo od posuvníku pozice skladby.
- `setLoop`
Nastavuje hodnotu tlačítka pro přehrávání skladeb ve smyčce.
- `setPlayingOrder`
Nastavuje hodnotu tlačítka pro řazení přehrávání.
- `updateTimeAsync`
Nastavuje hodnotu štítku aktuální pozice skladby a hodnotu posuvníku. Tato funkce se volá z druhého vlákna a musí zajistit, že widgety jsou upraveny v hlavním vlákně okna aplikace.

2.5 Build system

V rané fázi projektu vznikly experimenty s použitím C++20 modules přístupu. Snahou bylo vytvoření modulů pro jednotlivé grafické knihovny a zvukovou implementaci a jejich následné použití v hlavním souboru `main.cpp`. Vytvořený soubor `Makefile` kompiloval struktury kódu zvlášť do modulů, a poté se vzniklé moduly přiložily k překladu hlavního spustitelného souboru. Hlavní výhodou použití modulů oproti klasickému přístupu příkládání hlavičkových souborů je rychlost kompilace, lepší správa a zapouzdření různých částí logiky. Jako hlavní problém se však ukázala podpora tohoto přístupu ze strany grafických knihoven. Například hned první implementovaná knihovna Qt nedokáže v současné době s překládáním do modulů pracovat a vytvoření modulu vyžaduje vlastní implementaci celého procesu překladu.

Z výše popsanych důvodů byl zvolen meson build system. Ten podporuje všechny zvolené grafické knihovny a nabízí možnosti řazení do podprojektů (subprojects), která je svým principem podobná chování modulů ve smyslu umožnění lepšího rozložení logických částí programu na jednotlivé podprogramy. Meson build system je moderní open-source systém k překladu C++ aplikací. Byl navržen k rychlému překladu a jednoduchému a efektivnímu použití. Mezi jeho hlavní přednosti patří podpora operačních systémů Windows, Linux i macOS, výborná dokumentace se spoustou názorných příkladů použití a aktivní komunita, která pomáhá vývojářům s řešením konkrétních problémů i neustálým vývojem a inovací mesonu.

Meson build system používá k překladu aplikace stejnojmenné soubory `meson.build`. Tento soubor se nachází jak v kořenovém adresáři projektu, tak ve všech jeho podadresářích obsahujících závislosti nebo zdrojové kódy. Meson vyžaduje, aby všechny oddělené části aplikace, které jsou k hlavnímu programu přidány jako jeho závislosti, byly umístěny v podadresáři `subprojects`. Hlavní soubor `meson.build` v kořenovém adresáři definuje pouze projekt a název spustitelného souboru. Zároveň je zde uvedeno, že existuje podadresář `src`, ve kterém se nachází další soubor `meson.build`. Ten již obsahuje seznam zdrojových souborů k překladu a výčet závislostí k tomu potřebných. Tyto závislosti jsou pak definovány v jednotlivých podprojektech. [7]

2.5.1 Závislosti programu

Aplikace používá následující knihovny, které je potřeba nainstalovat před samotným spuštěním.

- Qt ve verzi ≥ 5.15
- wxWidgets ve verzi ≥ 3.0
- Gtkmm ve verzi ≥ 3.24

2.5.2 Překlad a spuštění v prostředí Linux

Pro přeložení programu slouží soubor `meson.build` v kořenovém adresáři. Příkazem `meson build` se vytvoří adresář `build`. V něm pak stačí zavolat příkaz `ninja`. Ten přeloží aplikaci a vytvoří spustitelný

soubor s názvem `main` v podadresáři `src`. K přeložení programu je zapotřebí mít nainstalované všechny závislosti. Celý postup je následující:

```
meson build
cd build
ninja
./src/main
```

Součástí zdrojových kódů je i přeložená spustitelná verze v adresáři `exe/linux/main`. K jejímu spuštění je potřeba mít v systému nainstalované všechny závislosti. Pokud i v takovém případě nelze aplikaci spustit, problémem mohou být jiné cesty ke knihovnám a je potřeba provést vlastní překlad popsaný výše.

2.5.3 Překlad a spuštění v prostředí Windows

Překlad na operačním systému Windows je komplikovanější a vyžaduje použití konzole `MSYS2`, které jako jediné je schopno správného překladu všech knihoven. V adresáři `'sources'` se nachází soubor `'README.md'`, který popisuje všechny závislosti potřebné při překladu. Celý postup překladu je potom stejný, jako v případě operačního systému Linux.

Knihovna `wxWidgets` používá pro načítání obrázků cestu s kódováním, které nepodporuje diakritiku. V případě, že cesta ke spustitelnému souboru obsahuje některý nevyklý znak (např. uživatelské jméno obsahuje diakritiku), tak se při spuštění aplikace zobrazí chybové hlášení, viz A.11. Při zvolení možnosti `continue` se program zapne bez ikon a vypadá, že reaguje správně. Při snaze přidat zvukový soubor do seznamu skladeb se ale chová neobvykle a přidá buď chybný soubor, nebo opakuje předchozí výběr. Pro správné spuštění je nutné přesunout celý adresář programu do jiného adresáře, který ve své cestě neobsahuje žádné speciální symboly.

I pro operační systém Windows je přiložená i spustitelná verze programu. Nachází se v adresáři `exe/windows/bin/main.exe`.

3 Popis implementace jednotlivých GUI knihoven

Grafické knihovny byly vybírány pro účely porovnání na základě jejich popularity a oblibě u komunity vývojářů, jejich rozšířenosti a možností, které nabízí. Snahou této diplomové práce je poskytnutí informací, které mohou pomoci při výběru knihovny pro budoucí projekty vývojářů. Právě porovnání nejčastěji používaných nástrojů může přinést největší benefity majoritě budoucích projektů.

Z těchto důvodů byly implementovány grafické knihovny Qt, wxWidgets a Gtkmm. Všechny tři se umístily na žebříčku nejoblíbenějších a nejčastěji používaných.

3.1 Qt

Qt knihovna je jednoznačně tou nejrozšířenější pro vývoj grafických aplikací v jazyku C++. Knihovna nabízí velkou řadu nástrojů, které značně usnadňují návrh, implementaci, testování i následné nasazení aplikace. K těmto nástrojům patří:

1. Qt Creator

Jedná se o integrované vývojové prostředí (IDE) navržené přímo pro vývoj aplikací v Qt. Toto vývojové prostředí nabízí správu projektů, návrh uživatelského rozhraní, editaci zdrojových kódů, ladění i testování. Usnadňuje práci pomocí generování obsahu a zvýraznění chyb.

2. Qt Designer

Qt Designer je nástrojem pro vizualizaci uživatelského rozhraní (UI) v Qt aplikacích. Nabízí většinu ovládacích prvků, které je možné umístit pomocí drag-and-drop. Následně lze upravovat jejich vlastnosti a propojovat je s funkcemi v aplikaci. Výstupem tohoto programu je soubor s příponou .ui, který lze nahrát do nezávislých projektů.

3. Qt Linguist

Tento nástroj umožňuje lokalizaci Qt aplikací. Jednoduše lze přidávat podporu pro jiné jazyky a překládat texty v aplikaci.

Takové překlady je možné ukládat do souborů a vytvářet pomocí nich vícejazyčné verze aplikací. Nabízí i možnosti pro kontrolu pravopisu.

4. Qt Assistant

Qt Assistant zobrazuje nápovědy a nabízí interaktivní prohlížeč dokumentace včetně příkladů implementace.

5. Qt Test

Nabízí nástroje pro vytváření a spouštění testů aplikace.

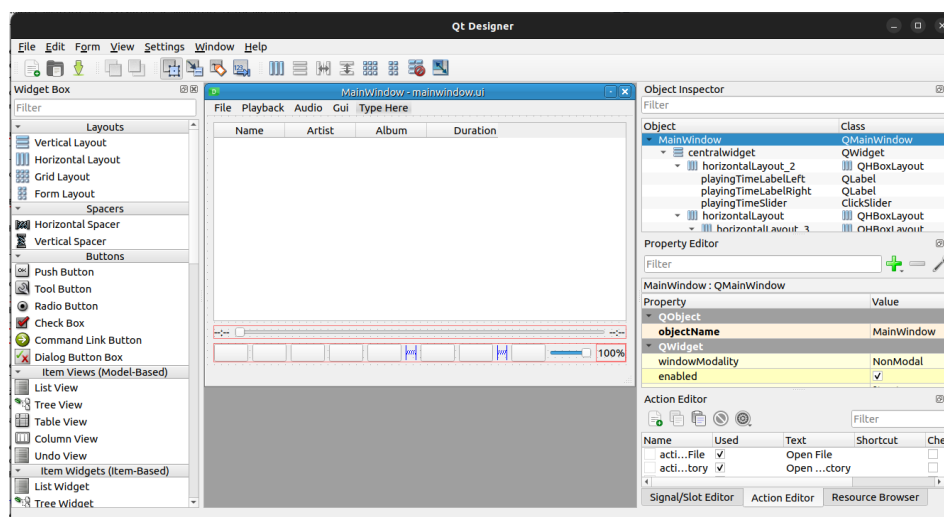
Implementace Qt byla testována na verzi 5.15. Tato verze je označena jako LTS (Long-Term-Support), tedy jako verze, která zaručuje podporu a opravu bezpečnostních chyb od vývojáře. Základem je třída nazvaná `QtGui`. Ta implementuje rozhraní `IGui` a vytváří hlavní okno aplikace `MainWindow`. Definice této třídy se nachází v podadresáři `public`, a je viditelná v hlavním souboru celé aplikace `main.cpp`. Ve funkci `run` se vytváří instance tříd `QApplication` a `MainWindow`. Právě `MainWindow` tvoří hlavní okno této Qt aplikace a implementuje rozhraní `IGuiControls`.

Qt framework používá k definování grafického rozhraní soubor typu `.ui`. Jedná se o formát `xml`, který obsahuje všechny grafické prvky a jejich atributy. K jeho vytvoření a editaci slouží program `Qt Designer`, který je popsán výše. Celý grafický návrh je ovšem možné napsat přímo v kódu. Výhodou použití přístupu s programem `Qt Designer` je vizualizace návrhu a připojení signálů a slotů. Použití programu `Qt Designer` je vidět na obrázku 3.1. Příkladem dvojice signálu a slotu může být například tlačítko `Play`, kde signálem je akce toho, že uživatel klikl na tlačítko (signál `playButton_Triggered`) a slot, který se po takové akci provede (slot `PlayButtonClicked`). Alternativně může programátor definovat vlastní signály a sloty v hlavičkovém souboru pomocí klíčových slov `public slots:`, `private slots:` a `signals:`. Tato klíčová slova se používají obdobně jako přístupy `public:` a `private:`. Všechny třídy, které obsahují implementaci nějakého Qt modulu vyžadují použití makra `QT_OBJECT` v definici dané třídy.

Framework Qt používá specifický přístup i k načítání obrázků. Jejich cesty nejsou většinou napsány v kódu absolutně, ale využívá se souboru pro definici zdrojů `qresources.qrc`. Tento soubor je opět

3. POPIS IMPLEMENTACE JEDNOTLIVÝCH GUI KNIHOVEN

ve formátu xml a definuje cesty ke zdrojům, jako jsou obrázky a jejich následné označení (alias), podle kterého je možné je použít v kódu.



Obrázek 3.1: Qt Designer

Rozšířené schopnosti frameworku Qt však vyžadují speciální postup při překladu projektu. Soubory s definicí grafického rozhraní `.ui`, seznam obrázků používaných v aplikaci `.qrc` nebo třídy definující grafické prvky s makrem `QT_OBJECT` vyžadují specifický přístup. Qt framework používá pro překlad svých aplikací program `QMake`, který je obdobou známého `Makefile` a podporuje používání těchto speciálních souborů. Tento projekt je však přeložen nástrojem `meson build` kvůli překladu ostatních knihoven. Ten obdobně podporuje přiložení všech závislostí při překladu pomocí funkce `qt5.preprocess` a jejích parametrů `moc_headers`, `ui_files` a `qresources`, kde `moc_headers` odpovídá hlavičkovým souborům s definicí tříd obsahujících makro `QT_OBJECT`, `ui_files` všem souborům typu `.ui` a `qresources` všem souborům typu `.qrc`.

Vytvoření hlavního okna aplikace je s pomocí všech nástrojů a souborů `.ui` a `.qrc` velice jednoduchá. Práce s nástrojem Qt Designer je intuitivní a vývojář dokáže vytvořit rozhraní během chvilky. Při implementaci zdrojového kódu je základem vytvoření instance `QApplication`, která vytvoří hlavní smyčku grafického okna. Zároveň se vytvoří instance třídy `MainWindow`, která v sobě obsahuje implementaci celého

rozhraní. Ve dvou případech bylo nutné přetížení základních komponentů k dosažení požadované funkcionality, a to pro použití posuvníku času přehrávané skladby `QSlider` a hlavičky seznamu skladeb `QHeaderView`. V případě posuvníku, výchozí chování neumožňuje zvolení konkrétní hodnoty kliknutím na konkrétní pozici. V takovém případě by se posuvník nastavil na hodnotu vyšší nebo nižší o 20 bodů. To je přijatelné u ovládání hlasitosti, avšak ne pro posuvník s časem. Kýženého chování bylo docíleno změnou chování po obdržení `mousePressEvent` signálu, který nově vypočítá pozici kliknutí a promítne ji na konkrétní hodnotu podle nastaveného minima a maxima. Tato hodnota se následně posuvníku nastaví. V případě hlavičky seznamu skladeb se jednalo o blokování signálu, který je obdržen při kliknutí na sloupec tabulky. Třída `QTable`, která realizuje seznam skladeb totiž umožňuje i selekci sloupce. Opět se tak jednalo o metodu `mousePressEvent`, která nově neoznačuje zvolený sloupec. S výjimkou těchto dvou funkcí byla implementace v prostředí Qt přímočará a jednoduchá. Používané metody všech grafických prvků jsou deskriptivní a programátor ve většině případů ani nemusí procházet dokumentaci.

3.2 wxWidgets

Grafická knihovna `wxWidgets` je ve srovnání s Qt menší a nenabízí tolik nástrojů pro pomoc při vývoji. Co se samotné implementace týče, tak si jsou obě knihovny dosti podobné. Obě nabízejí velkou škálu ovládacích prvků, jako jsou tlačítka, menu, seznamy, tabulky a mnoho dalšího. Obdobně jako Qt, i `wxWidgets` nabízí aplikaci pro tvorbu uživatelského rozhraní ve vizuální podobě. K tomuto slouží program `wxFormBuilder` a jeho podstata je stejná, jako tomu bylo u Qt Designeru. Tentokrát byl pro implementaci grafického rozhraní však použit druhý z možných přístupů, a to vytvoření celého rozhraní přímo v kódu.

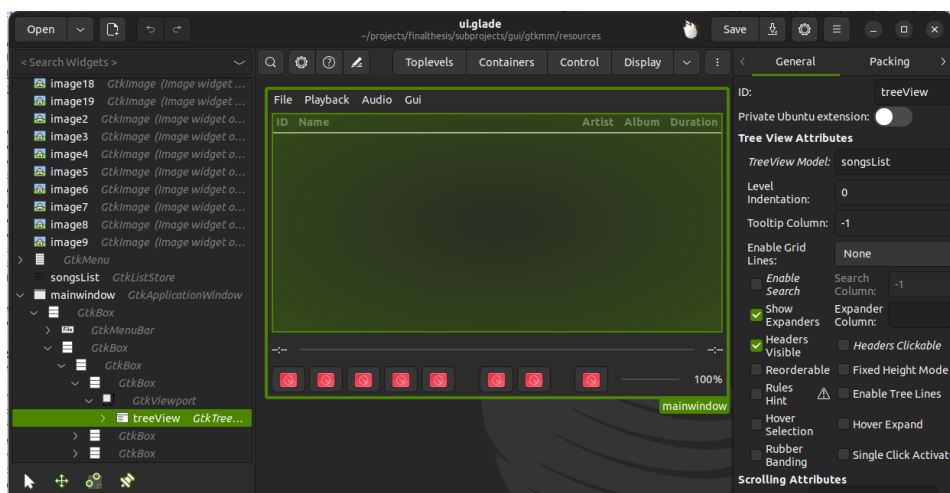
Při porovnání zdrojových souborů si lze všimnout, že implementace `wxWidgets` je oproti Qt násobně větší. Důvodem je implementace rozhraní v kódu, namísto vytvoření programem `wxFormBuilder` a následné nahrání souboru. Vytvoření rozhraní ve zdrojovém kódu je jednoduché a základem je vytvoření kontejnerů pro rozložení ovládacích prvků, pak jejich naplnění a následné napojení pro příjem událostí. Na

druhou stranu samotná inicializace a zobrazení grafického okna byla složitější. WxWidgets k tomuto účelu přistupuje vytvořením instance `wxApp`, která ve své přetížené metodě `OnInit` následně vytváří okno aplikace. To v případě WxWidgets dědí od `wxFrame`. Problémem však bylo správné zrušení všech vytvořených závislostí, protože při opětovném spuštění hlavního okna nebyly některé objekty správně ukončeny a aplikace se s chybou ukončila. Dokumentace v tomto případě nebyla moc nápomocná a chyběly především praktické ukázky spuštění aplikace různými způsoby, které WxWidgets nabízí. Mimo zmíněné problémy s nastartováním aplikace byla samotná implementace dost podobná té předchozí v knihovně Qt. K dispozici jsou všechny požadované ovládací prvky a funkce k jejich ovládání jsou pojmenována obdobně.

Implementace hudebního přehrávače používá verzi WxWidgets 3.0, opět označenou jako LTS. I zde nenabízela grafická knihovna veškerou funkcionalitu v základu a k docílení stejného chování bylo potřeba přetížit dohromady dva ovládací prvky a jeden event. Tentokrát tím byla absence zaklikávacího tlačítka `toggle button`, tedy tlačítka, které by bylo označené jako zvolené nebo nezvolené. Tuto funkci nabízí knihovna pouze ve smyslu zaškrťovacího políčka `checkbox`. Proto byla přetížena třída `wxBitmapButton` s hodnotou tlačítka, která se po zmáčknutí změnila a odpovídajícím způsobem upravila vzhled tlačítka. Druhým upraveným ovládacím prvkem byla opět hlavička seznamu skladeb, tentokrát ne z důvodu označení sloupce, ale z důvodu roztahování jednotlivých sloupců. Tomu bylo zamezeno přetížením metody `OnColBeginDrag`, který nově nepředává zprávu o eventu. Poslední zmíněnou úpravou bylo vytvoření události s hodnotou o změně aktuálního času právě přehrávané skladby. Tato událost slouží k nastavení hodnoty štitku a posuvníku způsobem bezpečným pro mezipřevodovou komunikaci. Knihovna WxWidgets nemá nativní podporu pro obrázky ve vektorovém formátu `.svg`. Takové obrázky je možné použít pouze s využitím knihoven třetích stran, které dovedou rasterizovat vektorový obrázek, a pak vytvořit objekt `wxBitmap`, který knihovna používá. Z tohoto důvodu byly obrázky převedeny do formátu `.png`.

3.3 Gtkmm

Knihovna Gtkmm poskytuje objektově orientované rozhraní pro GTK+. Svými nástroji se velice podobá knihovně wxWidgets. Opět i knihovna Gtkmm nabízí nástroj pro vizuální vytvoření rozhraní, které se stejně jako v případě ostatních knihoven překládá do formátu .xml a načítá v kódu aplikace. Tento nástroj se jmenuje Glade a ukázkou z jeho použití je možné vidět na obrázku 3.2. Jeho rozložení a ovládání je velmi podobné zkušenostem s Qt Designer. V levé části se nachází seznam vložených widgetů, uprostřed je interaktivní náhled okna aplikace a v pravé části se nachází pole atributů označeného widgetu. Nové widgety jsou seřazeny v sekcích Toplevels, Containers, Control a Display uprostřed, v horní části programu a vkládají se do okna přetažením. Dvakrát během návrhu se aplikace nečekaně ukončila, z tohoto důvodu je vhodné si postup průběžně ukládat.



Obrázek 3.2: Glade

Verze použitá při implementaci aplikace je 3.24. V současné době je již dostupná i verze 4, ale zatím není součástí většiny balíčků, a tak bylo rozhodnuto ponechat implementaci s předešlou verzí. Implementace knihovny Gtkmm byla nejjednodušší ze všech, protože k dosažení požadovaných vlastností nebylo nutné přetížít žádný ovládací prvek. Všechny widgety totiž nabízeli možnosti nastavení, které se dali přímo aplikovat. Nelze podle toho však usuzovat, že knihovna nabízí nejlepší

možnosti přizpůsobení. Příklad demo aplikace vyžadoval funkčnost, která je nativně podporována, nicméně se může zkomplikovat použití jiných widgetů. Posuvník času, ukazující aktuální pozici, není schopen nastavit hodnotu větší než 100. U všech verzí aplikace se posuvník času nastavuje na rozmezí 0-n, kde n je počet vteřin skladby. Při přehrávání se pak každou vteřinu zvyšuje hodnota posuvníku. Tři minuty dlouhá píseň má rozmezí 0-180, ale při nastavení hodnoty vyšší než 100 se posuvník zasekl. Takové chování nebylo pozorováno u žádné jiné knihovny, a i když by podle dokumentace hodnota být omezena neměla, tak žádné nastavení posuvníku vzniklý problém neopravilo. Výsledkem je tedy to, že posuvník má vždy nastavené rozmezí 0-100 a hodnota posuvníku se vypočítá podle rovnice 3.1, kde neznámá x je výslednou hodnotou posuvníku v rozmezí 0-100, neznámá y je původní hodnota v rozmezí 0-n a n je počet vteřin přehrávané skladby.

$$x = \frac{y}{n} * 100 \quad (3.1)$$

4 Porovnání implementace GUI

Tato kapitola se zaměřuje na porovnání vybraných grafických knihoven z pohledu vizuální stránky, licenčních podmínek, kvality dokumentace, aktivity komunity, historie, podpory pro nasazení na různých operačních systémech a hodnocení od uživatelů. V závěru kapitoly je uveden seznam populárních aplikací používajících vybrané knihovny.

4.1 Grafické rozhraní

Při pohledu na vizuální stránku aplikace je zjevné, že všechny knihovny aplikace jsou ovlivněny prostředím, ve kterém běží. Knihovny pro zobrazení grafického uživatelského prostředí využívají nástrojů pro zobrazení grafických prvků v operačním systému, a proto je jejich výchozí vzhled odlišný. Avšak knihovny wxWidgets a Gtkmm vypadají dost podobně jak na operačním systému Ubuntu s grafickým prostředím GNOME, tak na Windows. Aplikace jsou zobrazené na obrázcích A.2 a A.3 pro systém Linux a na obrázcích A.9 a A.12 pro systém Windows. V tomto konkrétním případě aplikace vypadají výrazně podobně, protože obě dvě využívají stejné widgety nainstalované v systému. Vzhled widgetů je možné stylizovat a určit svou vlastní paletu barev a stylů, lze tak dosáhnout podobného vzhledu na různých zařízeních.

Mezi knihovnami jsou ale i další rozdíly, které nejsou ovlivněny pouze operačním systémem a prostředím, ve kterém běží. V knihovně wxWidgets nejsou k vidění obrázky z levé strany každé volby v menu hlavičce A.10. Jedná se zřejmě o chybu v knihovně wxWidgets, protože obrázky se nezobrazily ani při použití postupu popsaného v dokumentaci. Černé obrázky jsou pak v menu špatně viditelné při použití tmavého schématu v prostředí GNOME a knihovny Gtkmm (viz obr. A.6). Další detail, kterého si lze na obrázku všimnout, je nižší kvalita obrázků v knihovně wxWidgets. Toto je způsobené jejich rasterizací. Knihovny Qt a Gtkmm totiž používají k zobrazení obrázků vektorový formát, kdežto knihovna wxWidgets používá raster ve formátu .png. Existují nástroje třetích stran, které mají za úkol načtení ikony ve vektorovém formátu. Jejich funkce je však na pozadí taková, že pouze převedou ikonu na raster a použijí ji, přičemž kvalita zobra-

zení je stejná jako použití vlastního .png. Liší se i zobrazení tabulky skladeb. Ta v aplikaci Qt nezobrazuje levý sloupec s označením ID do té doby, dokud se nepřidá první skladba. V knihovně Qt je tento sloupec k tabulce generován automaticky, kdežto v ostatních knihovnách je sloupec přidán manuálně a jeho hodnota explicitně nastavena. Porovnání je v příloze na obrázcích A.1, A.2 a A.3.

4.2 Podpora Cross-Platform

Podporou cross-platform se rozumí schopnost programu fungovat na různých operačních systémech a zařízeních bez nutnosti vytvářet a udržovat verzi programu pro konkrétní zařízení. V dnešní době je velký zájem o podporu cross-platform z důvodu velkého počtu přístupů z různých typů zařízení, jako jsou počítačové systémy Windows, Linux a macOS, ale zároveň i stále se rozšiřující trend používání mobilních zařízení Android, iOS a spousta dalších.

V rámci zadání této diplomové práce byla vyžadována ukázka implementované aplikace na platformách Linux a Windows. Požadavkem pro výběr grafických knihoven byla podpora alespoň těchto dvou operačních systémů. Všechny tři vybrané knihovny však podporují mnohem více než jen dva operační systémy.

4.2.1 Qt [8]

- Windows (včetně Windows 7, 8, 10, 11)
- Linux (včetně Ubuntu, Fedora, Debian, openSUSE)
- macOS (včetně macOS X 10.12 a vyšší)
- Android
- iOS
- BlackBerry
- ...

4.2.2 wxWidgets [9]

- Windows (včetně Windows 7, 8, 10, 11)
- Linux (včetně Ubuntu, Fedora, Debian, openSUSE)
- macOS (včetně macOS X 10.9 a vyšší)
- Android
- iOS
- Raspberry Pi
- ...

4.2.3 Gtkmm [10]

- Windows (včetně Windows 7, 8, 10, 11)
- Linux (včetně Ubuntu, Fedora, Debian, openSUSE)
- macOS (včetně macOS X 10.9 a vyšší)
- Android
- iOS
- ...

4.3 Nasazení aplikace

Pro hlavní desktopové platformy Linux a Windows většinou knihovny nabízejí postup pro nasazení uvedený v dokumentaci. Uvedený postup ale nezaručuje jednoduchý průběh instalace. Navíc v případě této demo aplikace, která používá všechny grafické knihovny naráz, byl překlad velice obtížný, protože nástroje podporované knihovnami nejsou kompatibilní mezi sebou. Zvlášť instalace programu na systému Windows byla velmi náročná a jedinou možností se ukázalo použití konzole MSYS2. Při překladu a instalaci programu na systému Linux se nevyskytly žádné větší obtíže. Po instalaci všech závislostí je překlad s použitím meson build jednoduchý.

Nevýhodou knihoven jsou i dynamické závislosti, které musí mít uživatel nainstalované v momentě, kdy chce aplikaci spustit. Tyto závislosti jsou větší obtíží na systému Linux.

4.4 Licenční podmínky

Licenční podmínky se týkají práv a povinností, které programátor musí dodržovat při užívání knihovny. Vývojář se používáním knihovny zavazuje, že je obeznámen s licenčními podmínkami a za jejich nedodržení může hrozit postih.

Knihovna Qt byla od začátku svojí existence nabízena s dvojitou licenci a programátor se může rozhodnout, kterou z licencí použije. První je open-source licence a její použitím pro vývojáře plyne povinnost zveřejnit všechny zdrojové kódy. Druhá licence je placená, komerční. Pokud tedy projekt, který používá knihovnu Qt, není uveřejněn, je programátor povinen zajistit si komerční licenci. Výše ceny se odvíjí od různých kritérií. Qt například nabízí levnější plán komerční licence pro menší firmy a start-upy, jejichž roční obrát nepřesáhne 250 tisíc dolarů (zhruba 5 350 000 korun) a současně je počet programátorů pracujících na projektu maximálně čtyřčlenný. Každý programátor musí mít svoje vlastní předplatné komerční licence. [11] [12]

Knihovny wxWidgets a Gtkmm jsou oproti tomu obě vedeny pod volnou licencí LGPL.¹ Pro programátory z toho neplynou žádná omezení, jako tomu je u licence Qt. Jedinou podmínkou použití licence LGPL je povinnost uvedení zdroje. [13] [14]

4.5 Kvalita dokumentace

Kvalitní dokumentace je důležitým aspektem rozsáhlé knihovny. Grafické knihovny, kterým se tato práce věnuje, jsou velmi obsáhlé a zahrnují spoustu tříd, dědičností, funkcí a vlastností. Správně napsaná a především přehledná dokumentace je potom nenahraditelnou mapou programátora, s níž lze rychleji a snadněji pochopit, jak a kde používat konkrétní nástroje knihovny. Dokumentace by měla obsahovat přesné a aktuální informace, včetně příkladů použití a doporučení,

1. V případě wxWidgets se jedná o licenci svými podmínkami velice podobnou LGPL.

jak používat konkrétní struktury, případně varování, jakým praktikám se vyhnout. Dokumentace, která toto vše obsahuje, má značný přínos pro knihovnu, protože:

- Usnadňuje začátky a pomáhá novým uživatelům knihovny. Návodů ukazující start aplikace, nastavení hlavního okna nebo práce se základními widgety výrazně zlepšují učící křivku.
- Umožňuje snadnou orientaci, s níž je možné rychle najít funkce, které uživatelé potřebují.
- Zvyšuje produktivitu programátorů, kteří se mohou zaměřit na řešení problémů a vývoj aplikací namísto hledání odpovědí na internetu.
- Podporuje správné použití funkcí a předchází chybám, které mohou vést ke špatnému chování aplikací.
- Zvyšuje spokojenost uživatelů, kteří ocení snadnou dostupnost informací. Ta vede ke snazšímu a plynulejšímu vývoji aplikace.

Pro porovnání dokumentace byl vybrán základní widget, který v každé knihovně slouží k zobrazení textu. Pro Qt se jedná o třídu `QLabel` [15]. U knihovny `wxWidgets` to je `wxStaticText` [16] a v dokumentaci `Gtkmm` je k nalezení pod jménem `Gtk::Label` [17].

V dokumentaci knihovny Qt je k vidění vše, co bylo popsáno jako příklad kvalitní dokumentace. Na prvním místě se zobrazí hlavičkový soubor, kde je definice třídy spolu s informací pro kompilaci programu. Dále se nachází seznam atributů, následovaný všemi podporovanými funkcemi, signály a sloty. Hned pod tímto seznamem jsou praktické příklady použití třídy `QLabel`, jeho vytvoření, nastavení textu i stylu. Níže jsou pak další ukázky použití obrázků v kombinaci s textem. Dalším důležitým aspektem je i jednoduchá možnost přepnutí aktuální verze v levé části obrazovky, stejně jako poznámka u všech novějších funkcí, od jaké verze je dostupná k použití.

Dokumentace knihovny `wxWidgets` obdobně zobrazuje informaci o hlavičkovém souboru, který definuje danou třídu. Je zde vyobrazen graf dědičnosti, a pak seznam stylů, které třída může používat. K těmto stylům bohužel není žádná praktická ukázka použití, a tak jsou nováčky bez zkušeností s aplikací stylů odkázáni na následné hledání,

jak některý z uvedených stylů použít. Následuje seznam funkcí bez příkladu vytvoření nebo ovládání widgetu. Funkce obsahují informace o verzi, ve které byly do knihovny přidány. Verze je také zobrazena v pravém horním rohu obrazovky, ale na rozdíl od knihovny Qt mezi verzemi není možné přepínat.

Dokumentace knihovny Gtkmm je ve všech ohledech stejná, jako dokumentace knihovny wxWidgets. S rozdílným přístupem aplikování stylů widgetu, který je řešen voláním konkrétních funkcí místo nastavení bitové hodnoty. Tento přístup je lépe zdokumentován a použití funkce k nastavení stylu je více přímočaré.

4.6 Aktivita komunity

Aktivita komunity je důležitou složkou při výběru knihovny pro vývoj aplikace. U většiny projektů mohou vývojáři narazit na problém, který pozastaví vývoj a vyžaduje pomoc či podporu od zkušenějších programátorů. A právě knihovny s aktivní komunitou často řeší podobné případy a navzájem si pomáhají v online prostředí při řešení zapeklitých situací. Zároveň má programátor přístup k již vyřešeným problémům, které mohou nastat i v jeho případě.

Jak knihovna Qt, tak i wxWidgets mají pro účely komunikace mezi vývojáři vytvořené vlastní internetové fórum, kde se může každý zeptat nebo naopak reagovat na položenou otázku. Z oficiálních zdrojů Gtkmm pak pro takové účely slouží fórum, které patří pod GNOME.

První z knihoven, Qt, se pyšní největší aktivitou příspěvků i reakcí na fóru. Každý den tam přispívá mnoho programátorů a celkem je možné hledat ve více než 740 tisíci příspěvcích. [18]

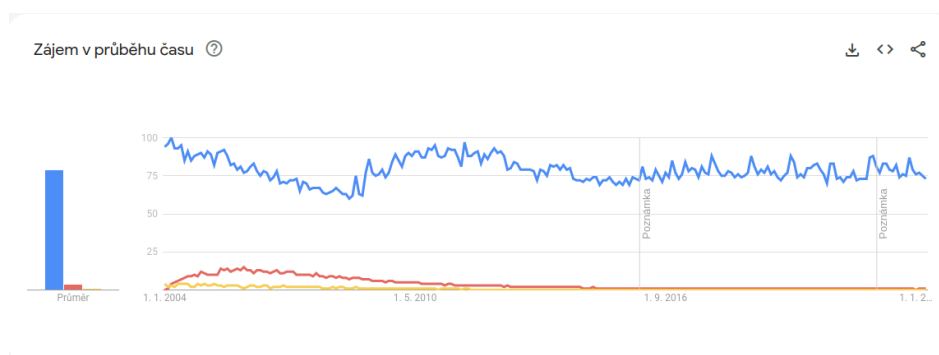
Knihovna wxWidgets má také velice populární fórum, i když co do počtu pokulhává za Qt. Obsahuje více než 168 tisíc příspěvků, a stejně jako tomu bylo u Qt, i zde aktivně odpovídají vývojáři z celého světa každý den. [19]

Poslední knihovna Gtkmm za svými konkurenty mnohonásobně zaostává. Na oficiálních stránkách je možné se dočíst, že lze používat diskuzní fórum GNOME s využitím štítku cplusplus. Pod tímto štítkem je však v současné době na fóru 25 příspěvků, naopak desítky dalších jsou vytvořeny pod štítkem "gtk". [20] [21]

Podstatná část komunity k diskutování nad implementováním knihoven nepoužívá oficiální fórum. Velké oblibě se u programátorů těší web stackoverflow, a proto je důležité zahrnout i výsledky tohoto fóra k porovnání. Pro knihovnu Qt je na tomto fóru vytvořených vícero štítků, pod kterými se nachází více než 100 tisíc diskuzí. Štítek pro knihovnu wxWidgets obsahuje necelých 5 tisíc příspěvků. Pro knihovnu Gtkmm se nachází 3 štítky rozlišující se podle verze knihovny. Dohromady čítají přes tisíc příspěvků.

Knihovny wxWidgets a Gtkmm mají i oficiální repozitář poskytovaný na webu Github.com. I zde je možné vnímat podporu a aktivitu komunity. Oba repozitáře jsou velice aktivní, ale stejně jako tomu bylo v předchozích příkladech, i zde lze vidět převahu wxWidgets nad Gtkmm. Hlavní větev projektu je aktualizována každých pár dní a v případě Gtkmm to je každých pár týdnů. Zároveň je repozitář pro wxWidgets mnohem rozsáhlejší a čítá přes 74 500 commitů, kdežto Gtkmm má pouze 3 450 commitů. Oba dva repozitáře jsou dostupné v následujících citacích. [22] [23]

Na obrázku 4.1 je možné vidět vyhledávání klíčových slov na populárním vyhledávači Google. Ten nabízí nástroj Google Trends, který umožňuje zobrazit četnost vyhledávání. Z uvedeného grafu je zřejmé, že knihovna Qt mnohonásobně předbíhá knihovny wxWidgets a Gtkmm. Graf je zobrazen od roku 2004 do současnosti, ale ke stejným výsledkům se dostaneme i při omezení na kratší časový úsek, jako je poslední rok nebo měsíc. [24]



Obrázek 4.1: Google Trends

4.7 Historie

4.7.1 Qt

Knihovna Qt byla původně založena dvojicí programátorů norského původu. Haavard Nord a Eirik Chambe-Eng se potkali na univerzitě v Trondheimu, kde oba dva dokončili magisterské studium v oboru počítačové vědy. Vývoji objektově orientovaného grafického rozhraní v jazyku C++ se začali věnovat v roce 1988. V té době vyvíjeli základy první verze Qt frameworku jako součást práce na jiném projektu. Založili společně firmu Quasar Technologies, později přejmenovanou jako Troll Tech, dnes známou pod jménem Trolltech. V roce 1995 byla vydána první oficiální verze Qt. Od prvního dne vydání byla knihovna Qt nabízena ve dvou licencích, a to komerční a volné. Knihovna Qt zažívala ve svých počátcích pomalý start a nízký zájem o komerční licenci. Vše ze změnilo po prvním roce a s přibývajícími zákazníky se rychle rozšiřovaly dostupné funkce a nástroje, které Qt nabízelo. V roce 2008 převzala všechna práva knihovny Qt společnost Nokia a vlastnila je až do roku 2011. Poté byla knihovna krátce vlastněna společností Digia a v současné době je spravována společností The Qt Company. [25] [26]

4.7.2 wxWidgets

Počátky knihovny wxWidgets sahají do roku 1992. Postaral se o to Julian Smart, který působil na univerzitě v Edinburghu. Základy knihovny vznikly jako součást jiného projektu, který vyžadoval zobrazení grafického rozhraní na obou platformách Windows i Unix. V té době již existovaly konkurenční, komerční knihovny, které takovou funkci poskytovaly, ale pro použití na menším projektu byly dost nákladné, a tak Julian začal s vývojem wxWidgets, kde počáteční písmena w a x znamenají Windows a Unix. Postupem času se do projektu přidali vývojáři, které zaujal koncept projektu, a tak pomáhali s jeho vývojem. Během následujících let začalo mít o projekt zájem více komerčních firem i vlád. [27]

4.7.3 Gtkmm

Původním jménem Gtk-, které se později přejmenovalo na Gtkmm (Gtk minus minus), protože adresace se znaménky '-' byla občas matoucí. Gtkmm bylo vyvinuto jako C++ wrapper nad C knihovnou GTK+. První verze byla vydána v roce 1998.

4.8 Známé programy využívající C++ GUI

Zde je uveden seznam známých a oblíbených programů, které byly vytvořeny grafickými knihovnami Qt, wxWidgets nebo Gtkmm.

4.8.1 Qt

- **KDE Plasma desktop environment**

Populární linuxové uživatelské prostředí. [28]

- **VirtualBox**

Nástroj pro virtualizaci, který umožňuje uživatelům spouštět více operačních systémů na jednom zařízení. [29]

- **Autodesk Maya**

3D počítačový grafický systém, který je hojně využíván při tvorbě filmů, televizních pořadů a videoher. [30]

- **Google Earth**

Virtuální globus, mapa a geografické informace o planetě. [31]

- **VLC media player**

Přehrávač hudby, který dokáže přehrát většinu multimediálních souborů a podporuje různé streamovací protokoly. [32]

- **Skype**

Program pro videohovory a posílání zpráv. [33]

- **Autodesk AutoCAD**

Program pro návrh designu používaný ve strojírenství, architektuře a stavebnictví. [34]

- **Mathematica**

Nástroj pro výpočet matematických úloh používaný ve vědě a strojírenství. [35]

4.8.2 wxWidgets

- **Audacity**

Open-source aplikace pro úpravu a stříhání audia. [36]

- **Code::Blocks**

Velice rozšířené open-source vývojové prostředí (IDE) pro vývoj programů v jazyku C++. [37]

- **FileZilla**

Jedná se o open-source pro služby FTP (File Transfer Protokol) klienta a serveru. [38]

- **OpenShot Video Editor**

Open-source program pro editaci videa s grafickým rozhraním pro tvorbu. [39]

- **Celestia**

Open-source program, který nabízí 3D vizualizaci a prozkoumávání vesmíru. [40]

- **CodeLite**

Další open-source vývojové prostředí pro tvorbu C++ programů. [41]

4.8.3 Gtkmm

- **GIMP**

GIMP je velice rozšířený a populární open-source nástroj pro úpravu obrázků a grafiky. Bývá základem většiny linuxových distribucí. [42]

- **Inkscape**

Velmi oblíbený open-source program, tentokrát pro tvorbu vektorové grafiky. [43]

- **Transmission**

Jedná se o klienta BitTorrentu pro Linux a Unix systémy. [44]

- **GNOME desktop environment**

Populární linuxové uživatelské prostředí. [45]

- **GNOME Calculator**

Jednoduchá aplikace kalkulačky, která je zahrnuta v prostředí GNOME. [46]

- **Dia**

Open-source program pro vytváření grafů, diagramů a další typy vizualizací dat. [47]

5 Testování aplikace uživateli

K testování aplikace bylo vybráno deset uživatelů, a to jak zástupce ze skupiny IT specialistů, tedy lidí, kteří používají počítače a techniku na denní bázi, tak i starší uživatele, kteří s výpočetní technikou nemají tolik zkušeností. Pro všechny uživatele byl vytvořen stejný testovací scénář, jehož účelem bylo vyzkoušení všech funkcí hudebního přehrávače s postupným přepínáním mezi grafickými knihovnamí a operačními systémy. Uživatelé komunikovali svoje dojmy z používání jednotlivých ovládacích prvků a hodnotili jejich vzhled, chování, ovládání i celkový dojem z prostředí. Výsledná tabulka hodnocení je zobrazena v poslední kapitole Hodnocení. 6

Testovací scénář byl následující:

1. Přidat zvukový soubor do seznamu skladeb.
2. Spustit přehrávání přidané skladby.
3. Posunout pozici skladby.
4. Pozastavit přehrávání skladby.
5. Přidat více zvukových souborů.
6. Pokračovat v přehrávání skladby.
7. Přepnout pořadí přehrávání.
8. Přepnout na následující skladbu.
9. Změnit hlasitost zvuku.
10. Přepnout přehrávání ve smyčce.
11. Přepnout na následující skladbu.
12. Přepnout na jiné zobrazení Gui.
13. Opakovat akce 1-11.
14. Přepnout na poslední zobrazení Gui.

15. Opakovat akce 1-11.

Ovládání hudebního přehrávače nedělalo problémy žádnému uživateli, který jej testoval. Jak lze očekávat, knihovny wxWidgets a Gtkmm u uživatelů vyvolávaly stejné dojmy, protože jejich vizuální stránky jsou velice podobné. Uživatelské prostředí těchto dvou knihoven bylo hodnoceno lépe než prostředí Qt. Gtkmm lehce předčilo wxWidgets z důvodu kvalitnějšího zobrazení ikon a hezčího provedení nadpisů v tabulce skladeb. Většina uživatelů si všimla i chybějících ikon v menu nabídky. Tento problém byl popsán v předchozí kapitole 4.1. Při hodnocení chování jednotlivých ovládacích prvků si uživatelé všimli dvou zvláštností v implementaci Qt. První bylo přebliknutí řádku v tabulce skladeb při zvolení jiné písně a druhou pak posouvání pozice skladeb, u kterého nefunguje možnost drag-and-drop. Z tohoto důvodu jej také jako jediný hodnotili hůře než zbývající knihovny. Při hodnocení plynulosti nebyly výtky na žádnou z implementací. V poslední kategorii hodnocení celkového dojmu se všechny knihovny umístili dosti podobně, pouze Qt zaostávalo za svými konkurenty.

6 Hodnocení

Hodnocení vybraných knihoven je založeno na zjištěných informacích popsaných v předchozích kapitolách. Pro účely porovnání je v každé kategorii knihovna hodnocena na stupnici od 1 do 10, kde 1 znamená nejhorší a 10 nejlepší. Hodnocení je rozděleno na sekci z pohledu programátora v tabulce 6.1 a z pohledu uživatele v tabulkách 6.2 a 6.3. Pro hodnocení uživatele byly vytvořeny 2 hodnocení, protože aplikace se chová a vypadá odlišně na platformách Windows a Linux.

Tabulka 6.1: Hodnocení vybraných knihoven GUI programátorem.

	Qt	wxWidgets	Gtkmm
kvalita dokumentace	10	7	6
dostupné nástroje	10	8	8
implementace	9	8	8
aktivita komunity	10	6	3
licenční podmínky použití	8	10	10
podpora cross-platform	10	10	10
náročnost při nasazení	7	6	3
hodnocení	9.1	7.9	6.9

Tabulka 6.2: Hodnocení vybraných knihoven GUI uživatelem na platformě Linux.

	Qt	wxWidgets	Gtkmm
grafické prostředí	8.7	7.7	9.3
chování ovládacích prvků	9.3	10	10
plynulost ovládání	10	10	10
celkový dojem	9.6	8.1	9.9
hodnocení	9.4	8.8	9.8

Tabulka 6.3: Hodnocení vybraných knihoven GUI uživatelem na platformě Windows.

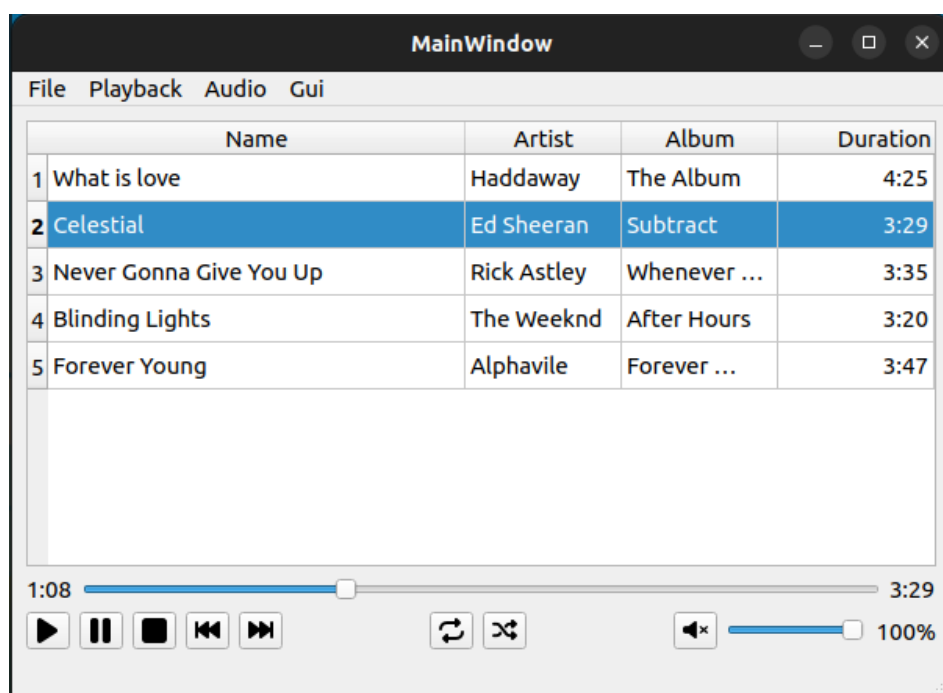
	Qt	wxWidgets	Gtkmm
grafické prostředí	8.3	7.9	9.3
chování ovládacích prvků	10	10	10
plynulost ovládání	10	10	10
celkový dojem	9.6	8.8	9.9
hodnocení	9.5	9.2	9.8

Při pohledu na hodnocení jsou vidět jasné rozdíly mezi hodnocením programátorem a uživatelem. Uživatelé vnímají pouze výsledný vzhled a chování aplikace, které dost často nekorelují s nástroji a možnostmi knihoven použitých při vývoji. Rozdíly mezi jednotlivými knihovnami jsou však pro uživatele minimální a většinou je možné vzhled i chování přizpůsobit v každé knihovně. Pohledem na vývoj aplikace nabízí největší možnosti knihovna Qt. Jako jediná z testovaných však naráží na omezení komerčního použití licenčními podmínkami. Pokud toto není omezením pro projekt, jedná se o nejlepší volbu z důvodu všech nástrojů, které nabízí, a zároveň je zde významná aktivní komunita vývojářů, kteří se snaží radit a řešit problémy při používání. Pokud je vyžadována komerční licence a náklady na pořízení Qt jsou vysoké, tak je volba mezi Gtkmm a wxWidgets vyrovnaná. Lepší variantou se spíše zdá použití wxWidgets, z důvodu rozšířenější komunity a snadnější práce s nasazením aplikace.

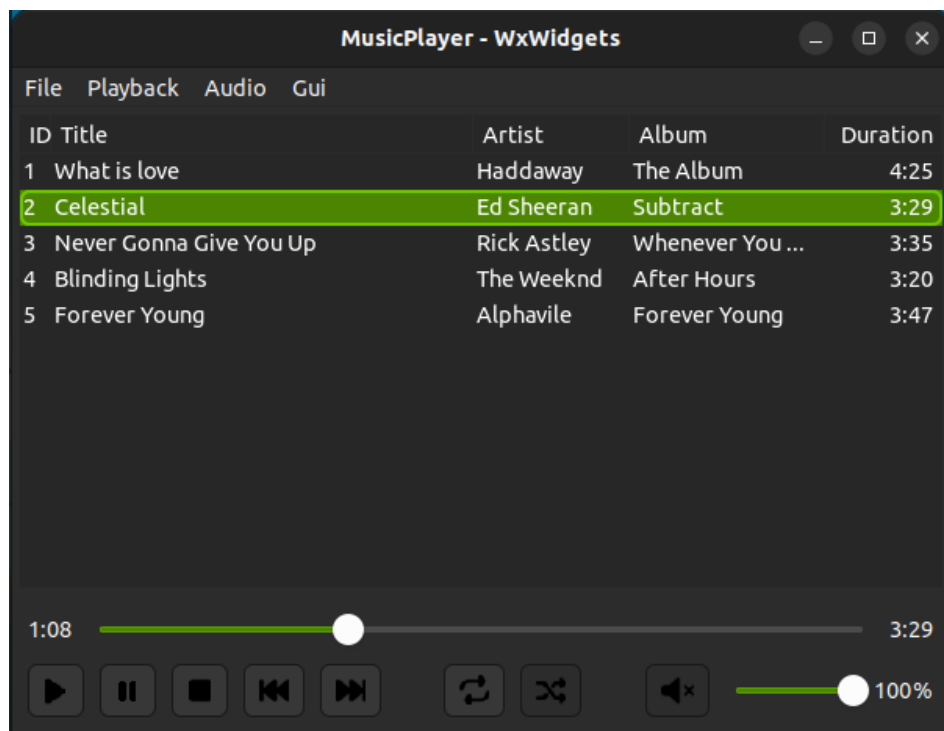
Závěr

V této práci byl popsán návrh a vytvoření aplikace hudebního přehrávače, který používá hned tři různé knihovny pro zobrazení grafického rozhraní v programovacím jazyku C++. Aplikace byla navržena s využitím všech znalostí nabytých během studia a zahrnuje použití praktik SOLID. Implementace je rozdělena na dílčí oddíly, které oddělují logiku aplikace od její grafické stránky. Následně byla hotová aplikace testována uživateli pro pozorování jejich dojmů z používání prvků různých grafických knihoven. Tato pozorování potom v kombinaci s detailním prozkoumáním nástrojů, kvality dokumentace, podpory a dalších vlastností tvořila finální hodnocení jednotlivých knihoven. Práce předává poznatky nabyté při implementaci aplikace a upozornění na nedostatky každé knihovny, včetně jejich omezení při nasazení na jiných operačních systémech a obsahuje finální hodnocení vybraných knihoven.

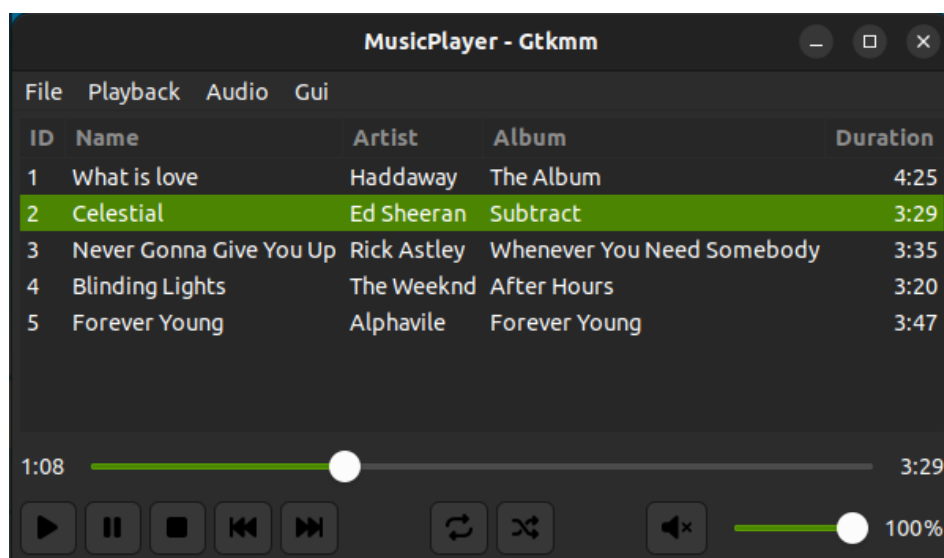
A Příloha



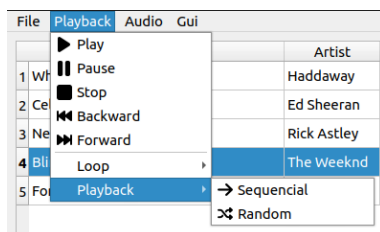
Obrázek A.1: Linux verze knihovny Qt



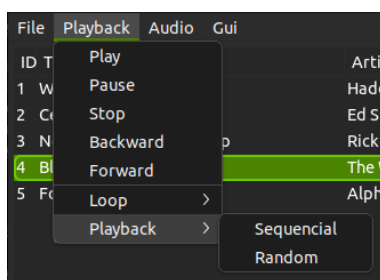
Obrázek A.2: Linux verze knihovny wxWidgets



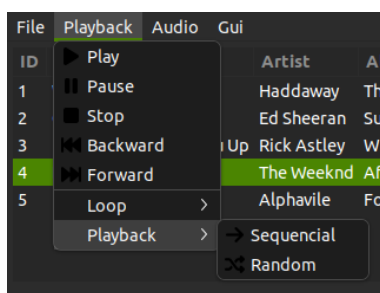
Obrázek A.3: Linux verze knihovny Gtkmm



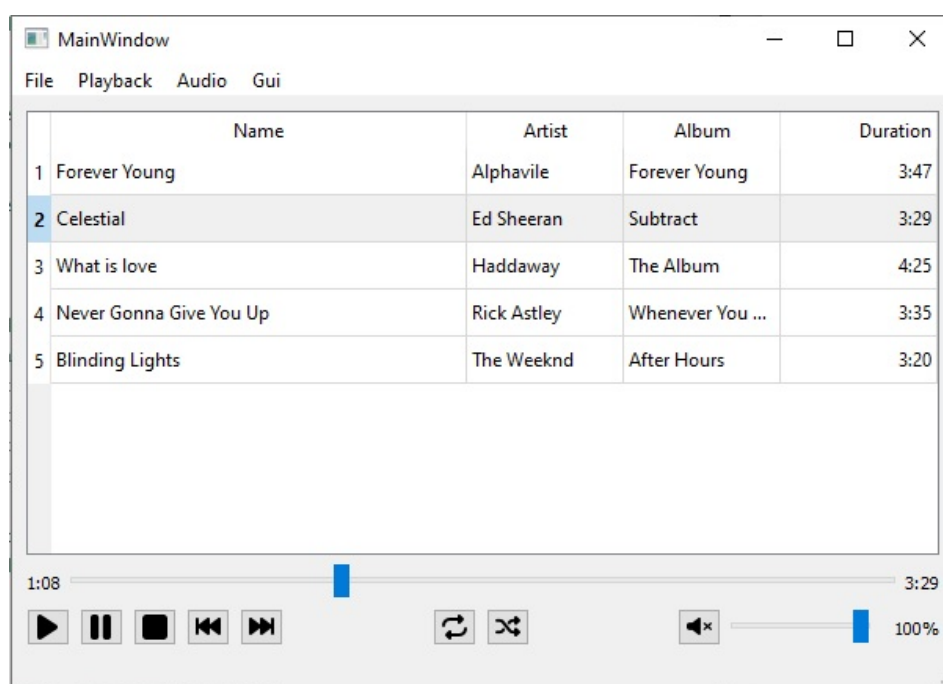
Obrázek A.4: Linux verze knihovny Qt



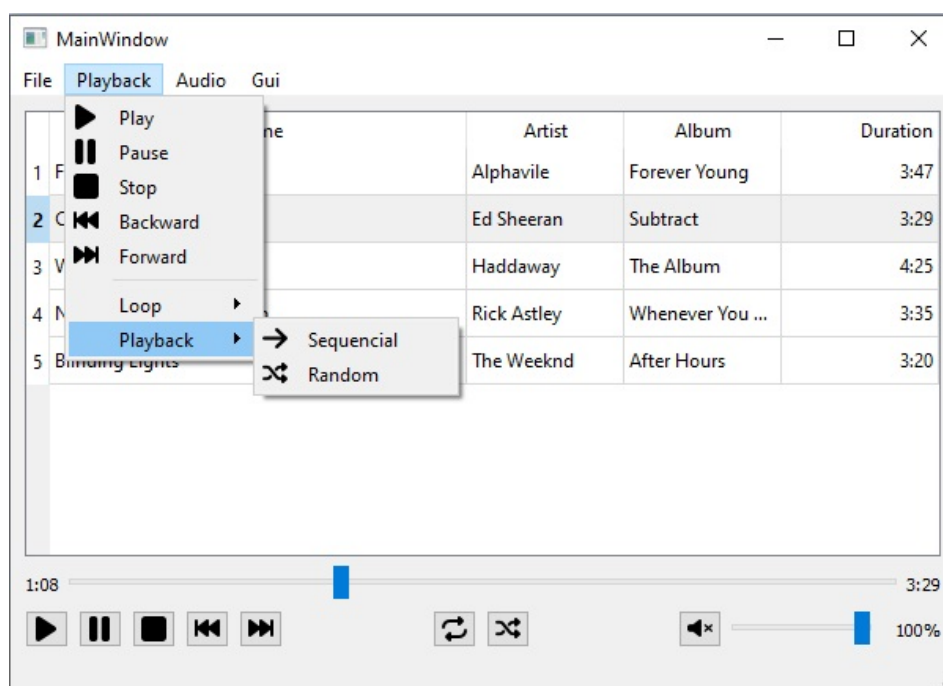
Obrázek A.5: Linux verze knihovny wxWidgets



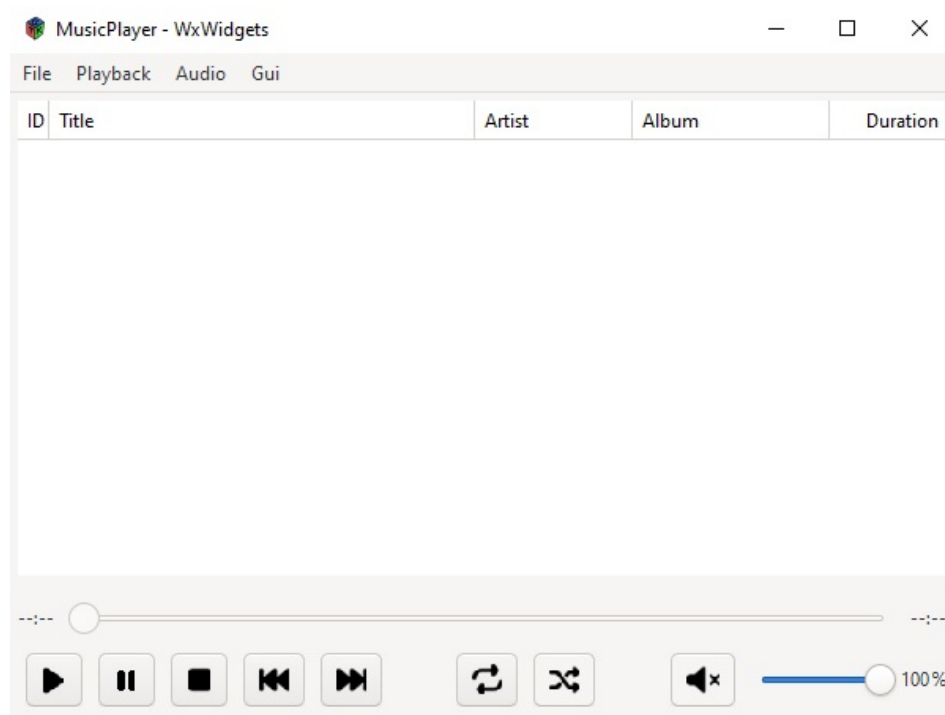
Obrázek A.6: Linux verze knihovny Gtkmm



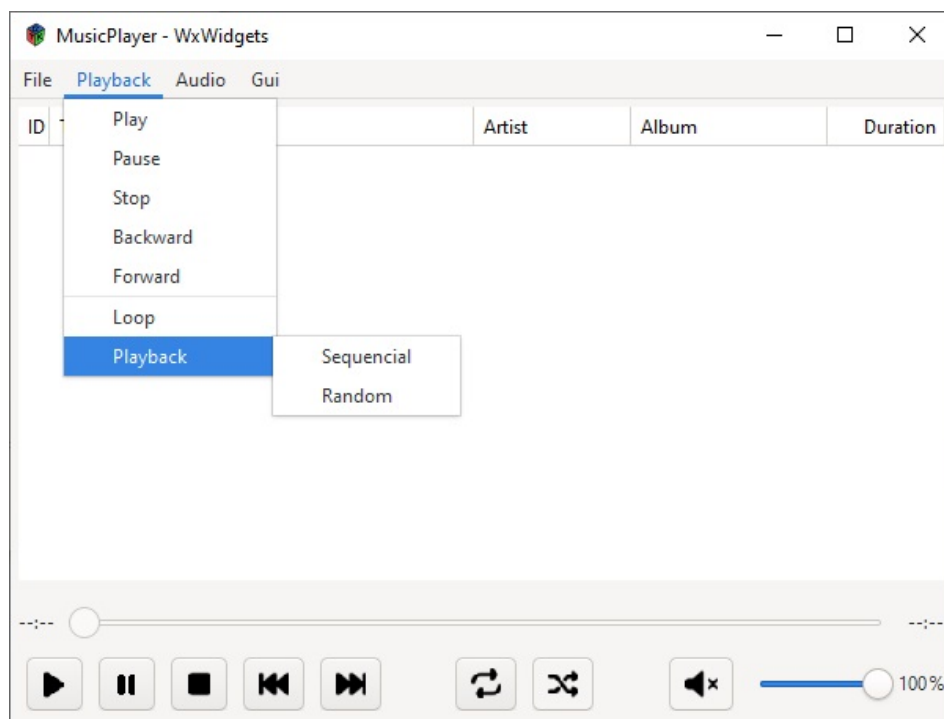
Obrázek A.7: Windows verze knihovny Qt



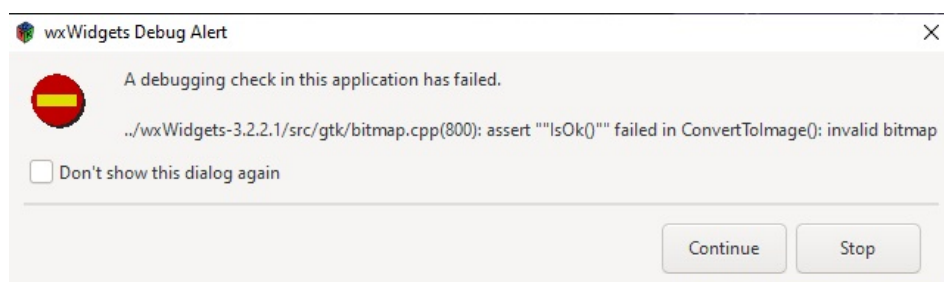
Obrázek A.8: Windows verze knihovny Qt



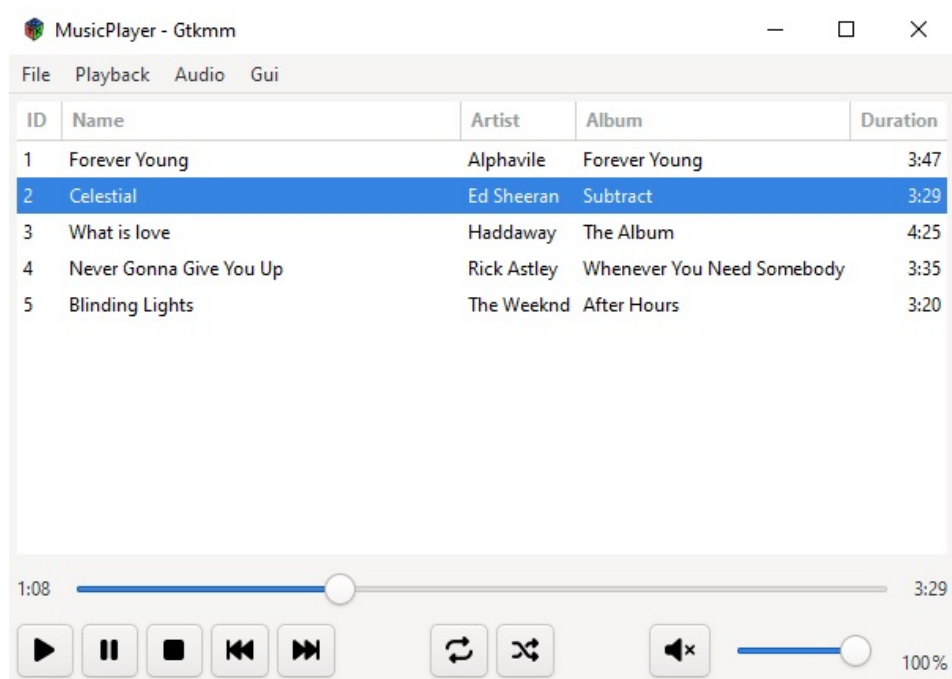
Obrázek A.9: Windows verze knihovny WxWidgets



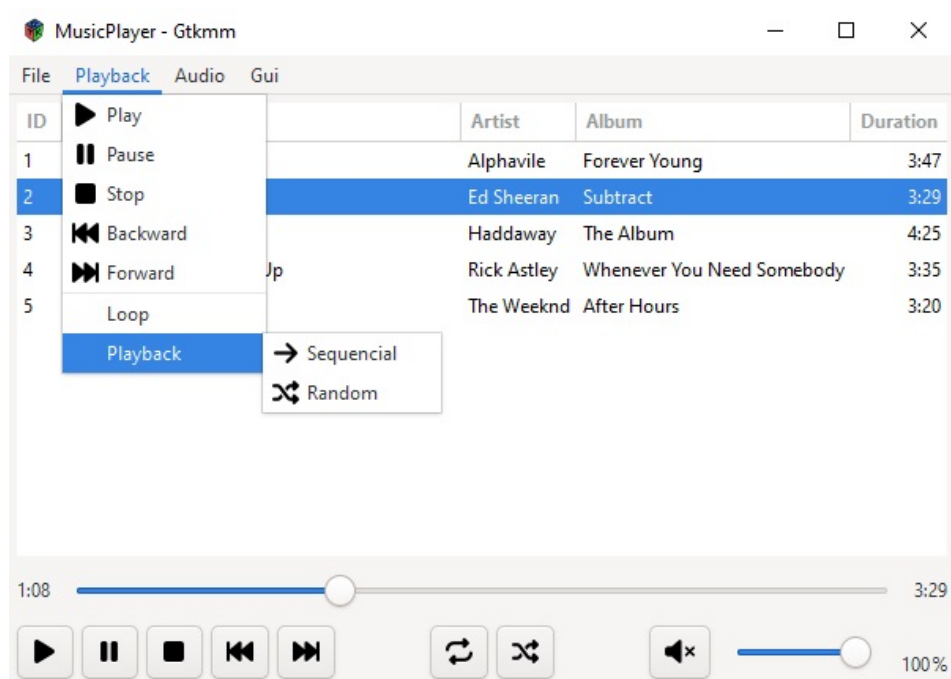
Obrázek A.10: Windows verze knihovny WxWidgets



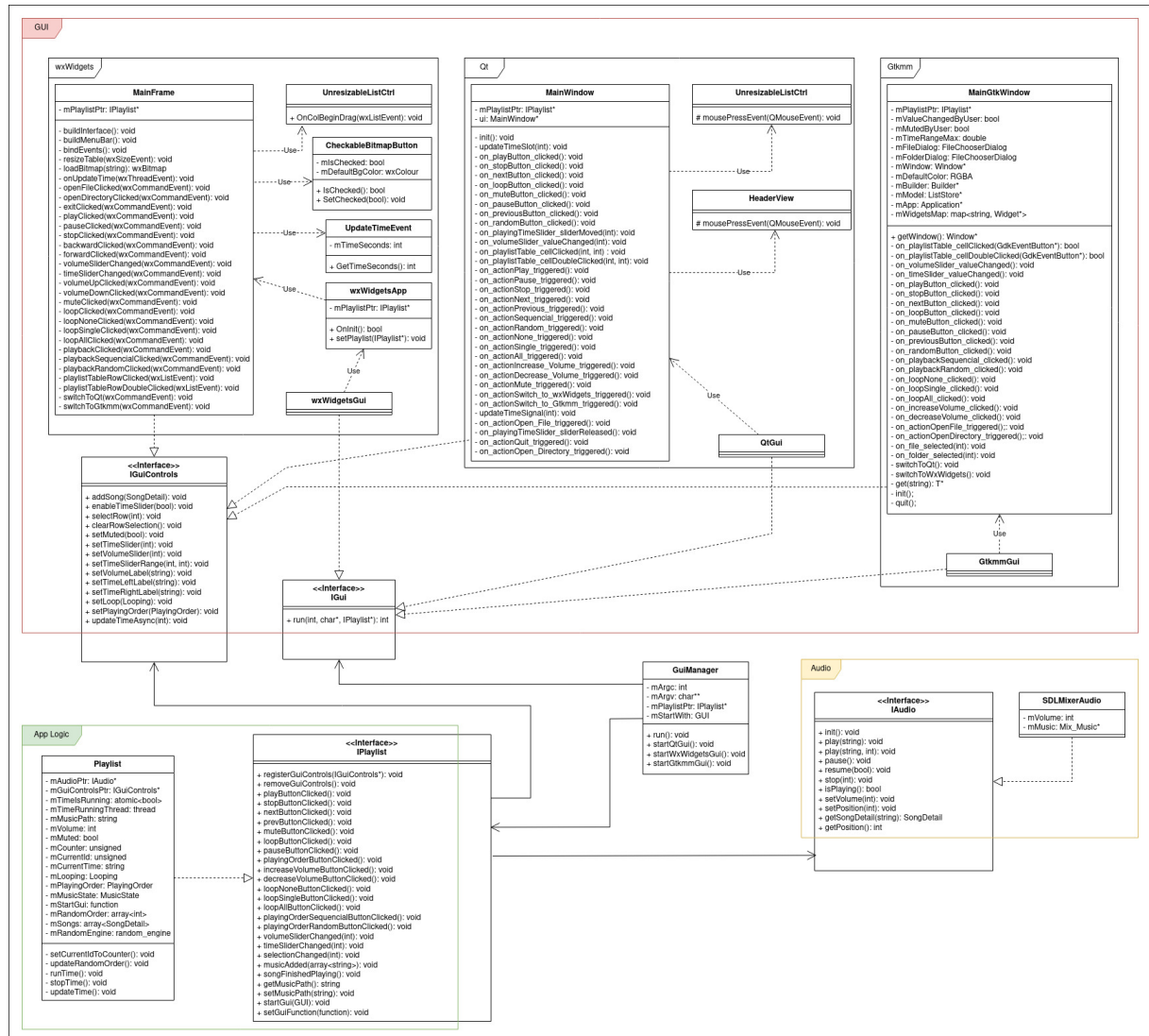
Obrázek A.11: Chyba načítání obrázků na Windows knihovny WxWidgets



Obrázek A.12: Windows verze knihovny Gtkmm

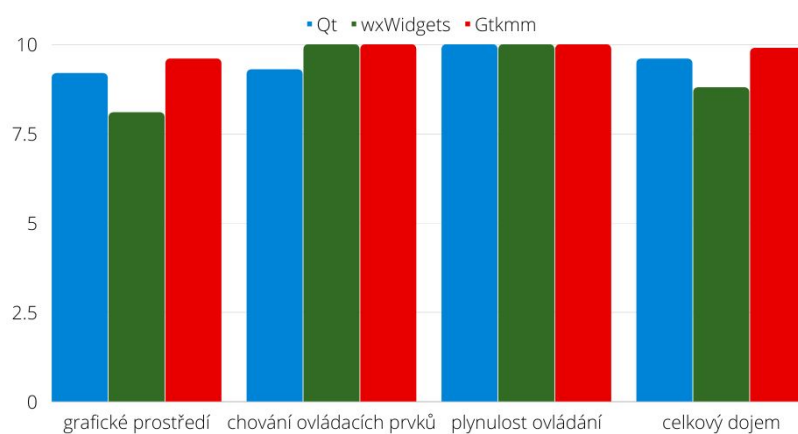


Obrázek A.13: Windows verze knihovny Gtkmm



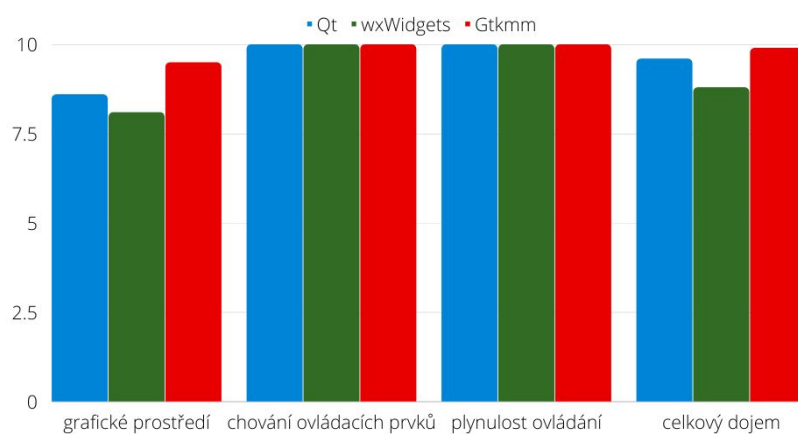
Obrázek A.14: UML: Diagram tříd

Chování na Linuxu



Obrázek A.15: Hodnocení uživatelů na platformě Linux

Chování na Windows



Obrázek A.16: Hodnocení uživatelů na platformě Windows



Obrázek A.17: Preferovaná knihovna na platformě Linux



Obrázek A.18: Preferovaná na platformě Windows

Bibliografie

1. NOBACK, Matthias. *Principles of Package Design: Preparing your code for reuse*. Matthias Noback, 2015. ISBN 978-9082120141.
2. *FontAwesome Icons* [online]. FontAwesome [cit. 2023-04-23]. Dostupné z: <https://fontawesome.com/icons>.
3. *SDL_mixer 2.0* [online]. Meson [cit. 2023-04-25]. Dostupné z: https://wiki.libsdl.org/SDL2_mixer/FrontPage.
4. *GNU Lesser General Public License* [online]. GNU [cit. 2023-04-25]. Dostupné z: <https://www.gnu.org/licenses/lgpl-3.0.html>.
5. *SDL2_Mixer Repository* [online]. libsdl-org [cit. 2023-04-23]. Dostupné z: https://github.com/libsdl-org/SDL_mixer.
6. *SDL2_Mixer Wiki page* [online]. libsdl [cit. 2023-04-23]. Dostupné z: https://wiki.libsdl.org/SDL2_mixer/FrontPage.
7. *Meson Subprojects* [online]. Meson [cit. 2023-04-25]. Dostupné z: <https://mesonbuild.com/Subprojects.html>.
8. *Qt Supported Platforms* [online]. Qt [cit. 2023-04-23]. Dostupné z: <https://doc.qt.io/qt-5/supported-platforms.html>.
9. *wxWidgets Supported Platforms* [online]. wxWidgets [cit. 2023-04-23]. Dostupné z: https://wiki.wxwidgets.org/Supported_Platforms.
10. *Gtkmm Supported Platforms* [online]. Gtkmm [cit. 2023-04-23]. Dostupné z: <https://www.gtkmm.org/en/index.html>.
11. *Qt FAQ Why and when do I need a license* [online]. Qt [cit. 2023-04-23]. Dostupné z: <https://www.qt.io/faq/2.5.-why-and-when-do-i-need-a-license>.
12. *Qt FAQ Commercial licensing* [online]. Qt [cit. 2023-04-23]. Dostupné z: <https://www.qt.io/faq/tag/qt-commercial-licensing>.
13. *wxWidgets Licence* [online]. wxWidgets [cit. 2023-04-23]. Dostupné z: <https://www.wxwidgets.org/about/licence>.
14. *Gtkmm Licence* [online]. Gtkmm [cit. 2023-04-23]. Dostupné z: <https://gtkmm.org/en/license.html>.

15. *Qt Documentation QLabel Class* [online]. Qt [cit. 2023-04-24]. Dostupné z: <https://doc.qt.io/qt-5/qlabel.html>.
16. *wxWidgets Documentation wxStaticText Class* [online]. wxWidgets [cit. 2023-04-24]. Dostupné z: https://docs.wxwidgets.org/3.0/classwx_static_text.html.
17. *Gtkmm Documentation Gtk::Label Class* [online]. Gtkmm [cit. 2023-04-24]. Dostupné z: https://gnome.pages.gitlab.gnome.org/gtkmm/classGtk_1_1Label.html.
18. *Qt Forum* [online]. Qt [cit. 2023-04-23]. Dostupné z: <https://forum.qt.io/>.
19. *wxWidgets Discussion Forum* [online]. wxWidgets [cit. 2023-04-23]. Dostupné z: <https://forums.wxwidgets.org/viewforum.php?f=36>.
20. *gtkmm Discussion* [online]. gtkmm [cit. 2023-04-23]. Dostupné z: <https://gtkmm.org/cs/maillinglist.html>.
21. *GNOME Discourse* [online]. GNOME [cit. 2023-04-23]. Dostupné z: <https://discourse.gnome.org/tags/c/platform/language-bindings/11/cplusplus>.
22. *wxWidgets Github Repository* [online]. Github [cit. 2023-04-23]. Dostupné z: <https://github.com/wxWidgets/wxWidgets>.
23. *gtkmm Github Repository* [online]. Github [cit. 2023-04-23]. Dostupné z: <https://github.com/GNOME/gtkmm>.
24. *Google Trends* [online]. Google [cit. 2023-04-23]. Dostupné z: <https://trends.google.com/trends/explore?date=all&q=Qt,wxWidgets,Gtkmm>.
25. *Qt History* [online]. Qt [cit. 2023-04-23]. Dostupné z: https://wiki.qt.io/Qt_History.
26. *A Brief History of Qt* [online]. O'Reilly [cit. 2023-04-23]. Dostupné z: <https://learning.oreilly.com/library/view/c-gui-programming/9780137143979/pref05.html>.
27. *History* [online]. wxWidgets [cit. 2023-04-23]. Dostupné z: <https://www.wxwidgets.org/about/history>.
28. *KDE Plasma Desktop* [online]. KDE [cit. 2023-04-23]. Dostupné z: <https://kde.org/plasma-desktop>.

29. *VirtualBox* [online]. VirtualBox [cit. 2023-04-23]. Dostupné z: <https://www.virtualbox.org/manual/ch01.html>.
30. *Autodesk Maya* [online]. Autodesk [cit. 2023-04-23]. Dostupné z: <https://www.autodesk.cz/products/maya/overview>.
31. *Google Earth* [online]. Google [cit. 2023-04-23]. Dostupné z: <https://www.google.cz/intl/cs/earth>.
32. *VLC Media Player* [online]. Videolan [cit. 2023-04-23]. Dostupné z: <https://www.videolan.org/vlc>.
33. *Skype* [online]. Microsoft [cit. 2023-04-23]. Dostupné z: <https://www.skype.com/cs/>.
34. *Autodesk AutoCAD* [online]. Autodesk [cit. 2023-04-23]. Dostupné z: <https://www.autodesk.com/products/autocad/overview>.
35. *Wolfram Mathematica* [online]. Wolfram [cit. 2023-04-23]. Dostupné z: <https://www.wolfram.com/mathematica>.
36. *Audacity* [online]. Audacity [cit. 2023-04-23]. Dostupné z: <https://www.audacityteam.org>.
37. *Code::Blocks* [online]. Code::Blocks [cit. 2023-04-23]. Dostupné z: <https://www.codeblocks.org>.
38. *FileZilla* [online]. FileZilla [cit. 2023-04-23]. Dostupné z: <https://filezilla-project.org>.
39. *OpenShot Video Editor* [online]. OpenShot [cit. 2023-04-23]. Dostupné z: <https://www.openshot.org>.
40. *Celestia* [online]. Celestia [cit. 2023-04-23]. Dostupné z: <https://celestia.org>.
41. *CodeLite* [online]. CodeLite [cit. 2023-04-23]. Dostupné z: <https://codelite.org>.
42. *GIMP* [online]. GIMP [cit. 2023-04-23]. Dostupné z: <https://www.gimp.org>.
43. *Inkscape* [online]. Inkscape [cit. 2023-04-23]. Dostupné z: <https://inkscape.org/>.
44. *Transmission* [online]. Transmission [cit. 2023-04-23]. Dostupné z: <https://transmissionbt.com/>.

BIBLIOGRAFIE

45. *GNOME* [online]. GNOME [cit. 2023-04-23]. Dostupné z: <https://www.gnome.org/>.
46. *GNOME Calculater* [online]. GNOME [cit. 2023-04-23]. Dostupné z: <https://wiki.gnome.org/Apps/Calculator>.
47. *Dia* [online]. GNOME [cit. 2023-04-23]. Dostupné z: <https://wiki.gnome.org/Apps/Dia>.