VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VIZUÁLNÍ EDITOR ZÁPISU BUBENICKÝCH RYTMŮ

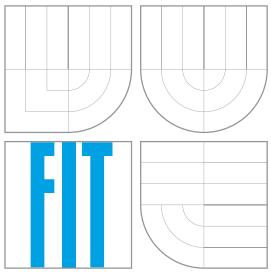
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARTIN ZELENÝ

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VIZUÁLNÍ EDITOR ZÁPISU BUBENICKÝCH RYTMŮ

VISUAL EDITOR OF DRUM RHYTHMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN ZELENÝ
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. ADAM HEROUT, Ph.D.
SUPERVISOR

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá vytvořením speciálního editoru pro zápis rytmů pro perkusní bicí nástroje. Editor musí být uživatelsky přívětivý a práce s ním musí být efektivní. Výslednou notaci je možné přímo vytisknout.

Abstract

This bachelor's thesis is about visual editor of drum rhythms. The editor can make special type of music notation. This notation is used for recording percussion drum rhythms. The editor needs to be user-friendly and effective to use. Recorded notation can be printed directly.

Klíčová slova

editor, vizuální, notace, perkusní, buben, rytmus, tisk, Qt framework

Keywords

editor, visual, notation, percussion, drum, rhythm, print, Qt framework

Citace

Martin Zelený: Vizuální editor zápisu bubenických rytmů, bakalářská práce, Brno, FIT VUT v Brně, 2011

Vizuální editor zápisu bubenických rytmů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Adama Herouta, Ph.D.

.....
Martin Zelený
17. května 2011

Poděkování

Chtěl bych poděkovat především doc. Heroutovi za mnoho usměrňujících rad a za veškerý čas, který mi během konzultací poskytl. Dále patří dík také všem, kteří přispěli radami ohledně uživatelského rozhraní a specifikovali nároky, které má program splňovat.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V době grafických uživatelských rozhraní se počítače také využívají k vytváření různých typů dokumentů, které se předtím musely vytvářet ručně nebo pomocí prostředků, které pro to nebyly primárně určeny.

V této práci jsem se zaměřil na vývoj vizuálního editoru pro speciální typ notového zápisu. Jedná se o notaci rytmů pro perkusní bicí nástroje (*bubny, na které se hraje dlaněmi, např. djembe*). V současné době neexistuje žádný editor, který by tvorbu této obecně používané speciální notace umožňoval. Klasická hudební notace se pro tento typ skladeb nepoužívá, protože je příliš univerzální a není v tomto případě dostatečně přehledná.

Cílem je vytvořit aplikaci, která umožní uživateli pohodlně zapisovat a editovat speciální notaci a bude umět tento zápis vytisknout. Pro jednoduchost obsluhy je navržený editor typu WYSIWYG[1]. Jedná se o úzce specializovaný program, který se snaží svůj účel splnit co nejpřesněji a s co nejmenším vynaloženým úsilím uživatele.

[1] What You See Is What You Get

Kapitola 2

Rozbor problematiky

Nejprve bych chtěl rozebrat důvody pro vytvoření editoru tohoto typu. Dle zjištění autora neexistuje v současnosti žádný program, který by vytvoření potřebné notace umožňoval[1]. Zároveň je však tato notace běžně používaná mnohými hudebníky. Rytmy, které skládají a šíří mezi sebou, musí psát na papír a kopírovat. Aplikace, která je předmětem této bakalářské práce, se snaží tyto podmínky zlepšit a práci s rytmy zjednodušit.

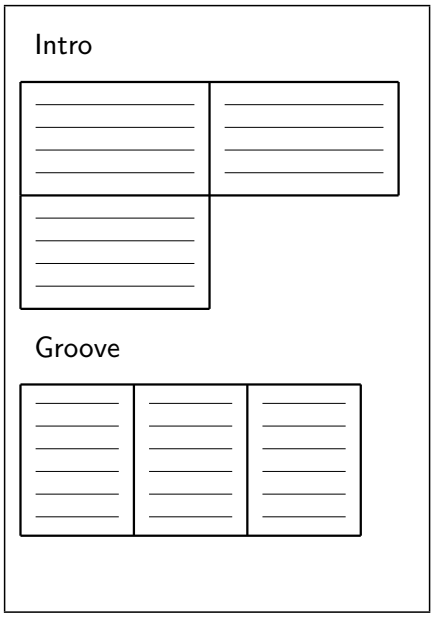
2.1 Popis notace pro perkusní bubny

Skladba pro perkusní bicí nástroje je obvykle složena z více různých částí. Jedná se například o intro ke skladbě, *groove*[2], nebo různé sólové části pro jednotlivé bubny.

U skladby je důležité její rozčlenění na jednotlivé části. Obrázek 2.1 zobrazuje zjednodušené schéma partitury pro perkusní skladby.

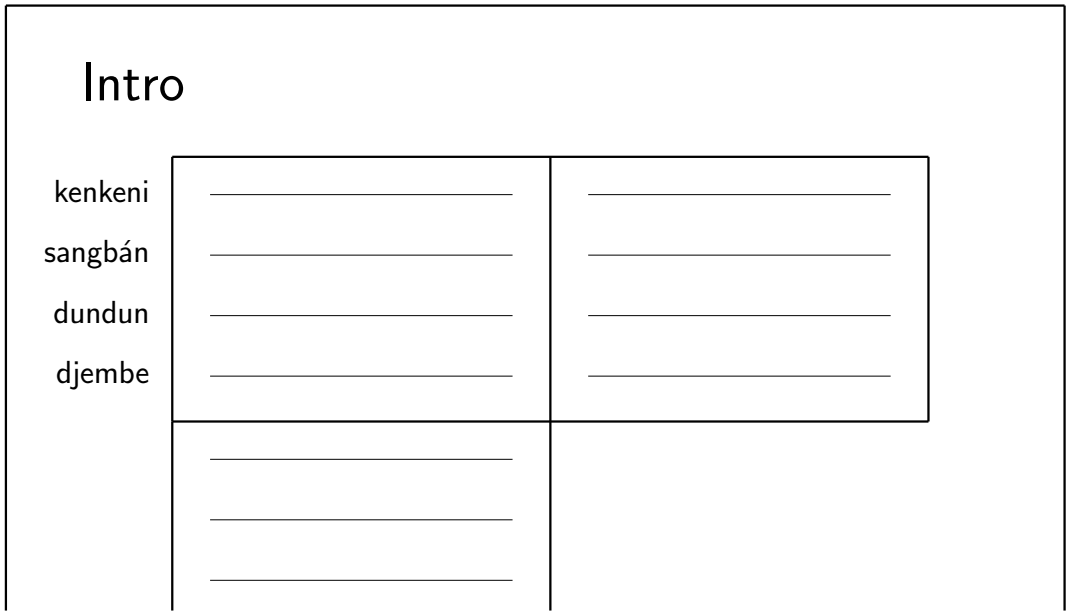
[1] perkusními bicími nástroji se sice zabývá např. program Percussion Studio, ale viz část 4.3
[2] hlavní rytmus, který prostupuje celou skladbou

Obrázek 2.1: Schéma jednotlivých částí perkusní skladby zobrazených na jedné stránce



Na začátku úseku je nadpis, který vyjadřuje jeho název a typ (*intro, groove, sólo*). Každý úsek je složen z *taktových bloků,* které je nutné vhodně zalamovat do řádků. Před prvním řádkem jsou umístěny názvy nástrojů, které danou část hrají (obr. 2.2).

Obrázek 2.2: Ukázka nadpisů jednotlivých stop



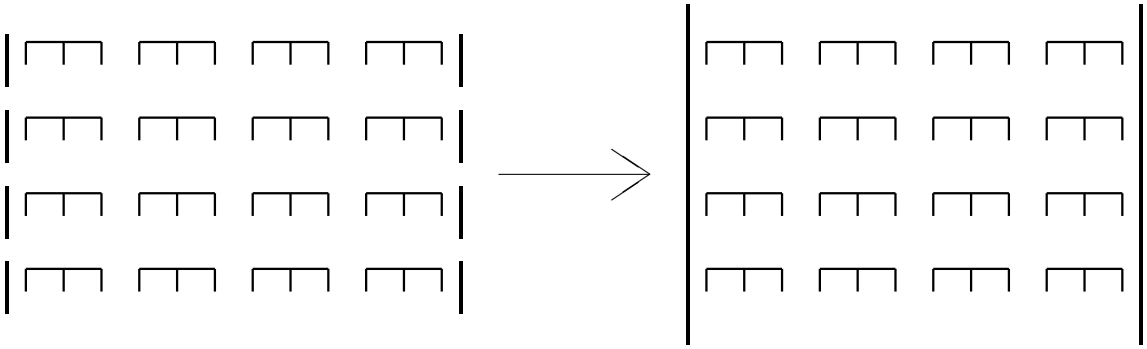
V dalších řádcích již není nutné opakovat nadpisy jednotlivých stop, ale je nutné zarovnat taktový blok na novém řádku pod první blok na řádku předcházejícím.

2.1.1 Popis osnovy pro perkusní notaci

Pro každou část skladby mohou být nastaveny jiné parametry taktu i jiný počet taktů. Každou část může hrát různý počet nástrojů. Jejich takty se vertikálně sdružují do taktových bloků tak, aby hudebníci mohli jednoduše sledovat i to, co hrají hráči na jiné nástroje.

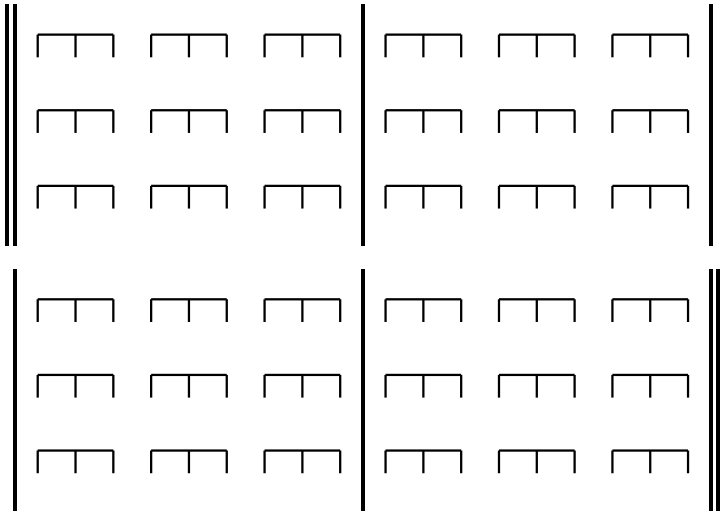
Takty se oddělují *taktovými čarami.* V případě, že je v taktovém bloku obsaženo více taktů jednotlivých nástrojů zobrazených pod sebou, je vhodné použít jednu svislou taktovou čáru místo více samostatných čar (viz obrázek 2.3).

Obrázek 2.3: Způsob zapisování taktových čar



Pro větší přehlednost je dále u každého taktu počáteční a koncová taktová čára zdvojená. Podrobně toto zobrazuje obrázek 2.4.

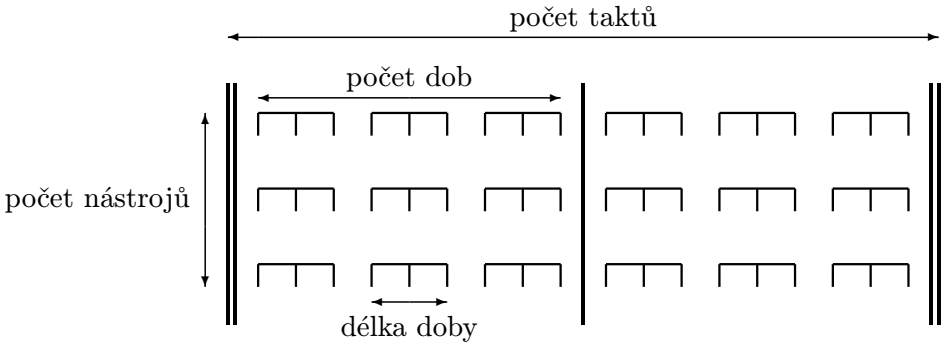
Obrázek 2.4: Zdvojení taktových čar na začátku a na konci části



Jeden takt je dále složen z jednotlivých *dob* (– doba obsahující čtyři *pulsy*). Pod pulsy se zapisují noty.

Mezi volitelné parametry notace, které je nutné určit se řadí především *výběr nástrojů*, *počet taktů*, *počet dob* v taktu a *délka doby* (obr. 2.5). Počet taktů je nezávislý a značí délku úseku skladby. Počet dob a délka doby pak určují typ taktu a jsou pro celou část stejné. Např. takt $^4/_4$ má čtyři doby délky čtyř pulsů.

Obrázek 2.5: Parametry notace



2.1.2 Popis not perkusní notace

Již bylo řečeno, že noty se zapisují pod jednotlivé *pulsy* v *době*. V případě notace pro bicí nástroje znamenají každé noty určité typy úderů. Tyto noty se značí písmenem nebo značkou. Ustálený systém značek úderů pro perkusní bubny zobrazuje tabulka 2.1.

Údery *bass*, *tón* a *slep* jsou typické pro bubny *djembe*. Zbylé údery *tlumený tón* a *zvonec* v kombinaci s klasickým *tónem* se hrají na basové bubny *sangbán*, *dundun* a *kenkeni*. *Zvonec*

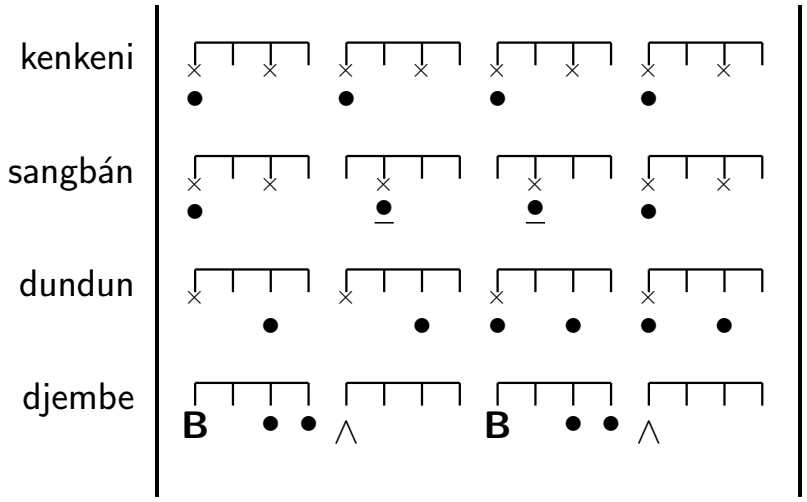
Tabulka 2.1: Typy úderů (not) v perkusní notaci

nota	název
B	bass
●	tón
●	tlumený tón
∧	slep
×	zvonec

je možné kombinovat i s jinými údery. Značka pro něj se umísťuje výš a blíže k *době* než ostatní noty, aby nezabírala jejich místo.

Kompletní ukázku jednoho taktu zobrazuje obrázek 2.6.

Obrázek 2.6: Ukázka jednoho taktu zapsaného perkusní notací



2.2 Specifikace požadavků na editor

Základní nutnost, kterou musí editor splňovat, je přesné dodržení notace pro perkusní bicí nástroje. To obnáší především pochopení všech jejich částí, aby mohl být na začátku vhodně zvolen návrh implementace. Úskalí, kterým je v vhodné se vyhnout, jsou různá upřesnění a změny v zadání.

Dále bylo s konkrétními budoucími uživateli konzultováno uživatelské rozhraní a nezbytné funkce, které má program umět. Požadavkem byl běh na různých operačních systémech Microsoftu (Windows XP až Windows 7).

Kapitola 3

Návrh a popis řešení

V následující kapitole se chci zaměřit a zcela popsat vizuální návrh aplikace a implementaci programu.

3.1 Návrh uživatelského rozhraní

Pro uživatele je nejdůležitější co nejjednodušší vytvoření osnovy pro perkusní notaci a následné pohodlné zapisování not na jednotlivé pozice. Zároveň je důležitá možnost mnoho parametrů notace upravovat později v průběhu práce s editorem.

Všechny případy užití, které musí editor splňovat, byly konzultovány s budoucími uživateli. *Use case* diagram, který tyto případy vyjadřuje je zobrazen na obrázku 3.1. Popis diagramu je součástí následujících podkapitol.

3.1.1 Hlavní okno aplikace

Po spuštění aplikace se zobrazí hlavní okno programu (obr. 3.2).

Položky, které se nachází v liště *menu* zobrazuje tabulka 3.1. Jedná se o prvky umožňující práci se souborem, tzn. jeho *otevření*, *uložení* a *tisk*. V dalším menu je možnost zvolit, jestli se má zobrazit či skrýt postranní *dokovací widget*[1]. Také je zde přítomna možnost dialogu pro editaci názvů bubnů, se kterými se v notaci pracuje (viz kap. 3.1.4).

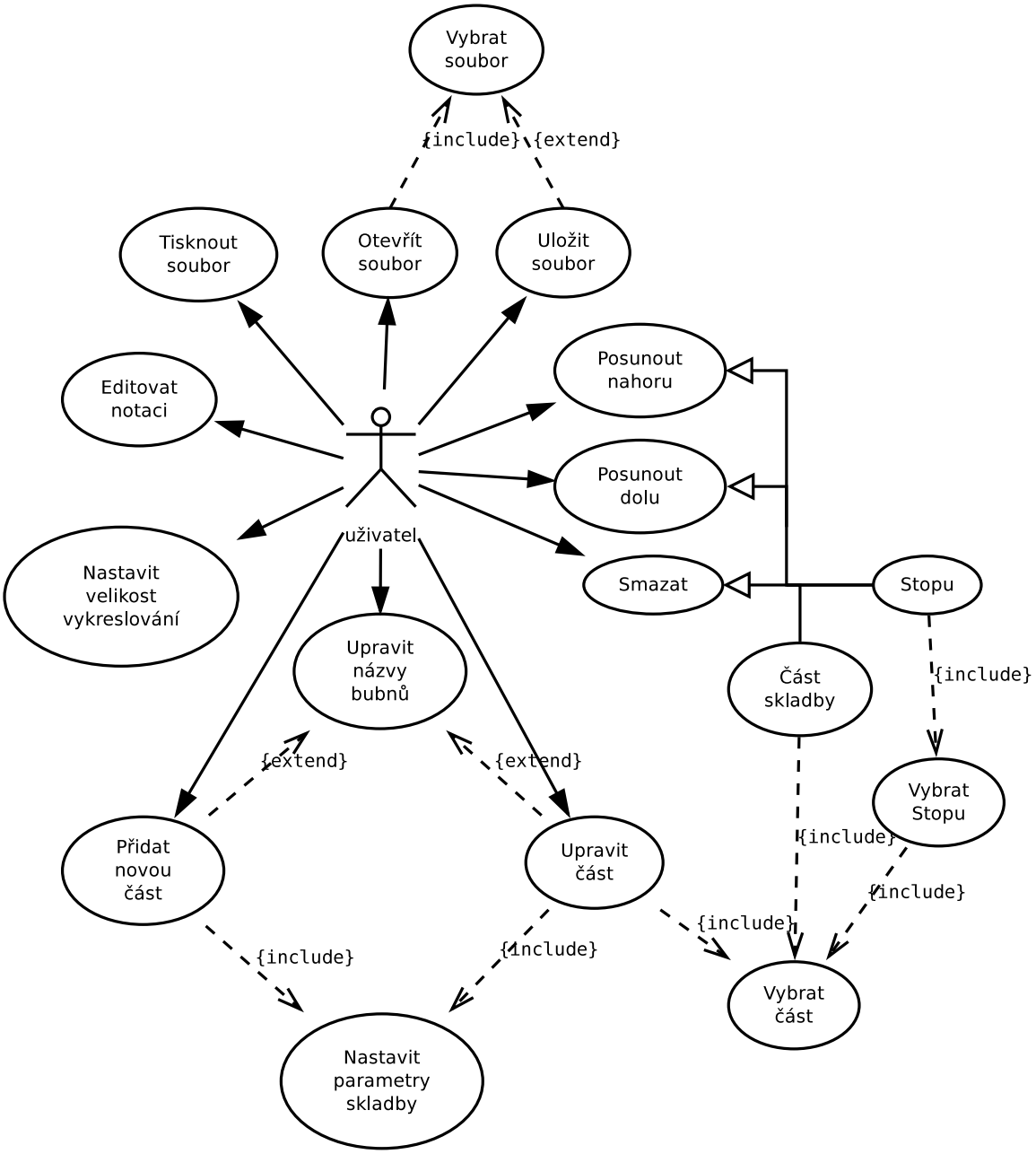
Při pravém okraji hlavního okna se nachází nástrojový panel. Ten zobrazuje tlačítka, které slouží k editaci notace. Podrobněji bude editace popsána v kapitole 3.1.6. S nástrojovým panelem je možné libovolně pohybovat. Jeho počáteční umístění při levém okraji hlavního okna se snaží reflektovat moderní styl řešení uživatelského rozhraní, který počítá především se širokoúhlými monitory. V takových případech je žádoucí vyčlenit co nejvíce vertikálního místa pro zobrazení hlavního dokumentu a různé podpůrné ovládací prvky umísťovat na levé a pravé okraje hlavního okna.

U levého okraje se v podobném duchu nese umístění *dokovacího widgetu*. Tato pozice také není fixní a kromě úplného skrytí je možné jej také vyčlenit z okna aplikace a umístit jej například na druhý monitor.

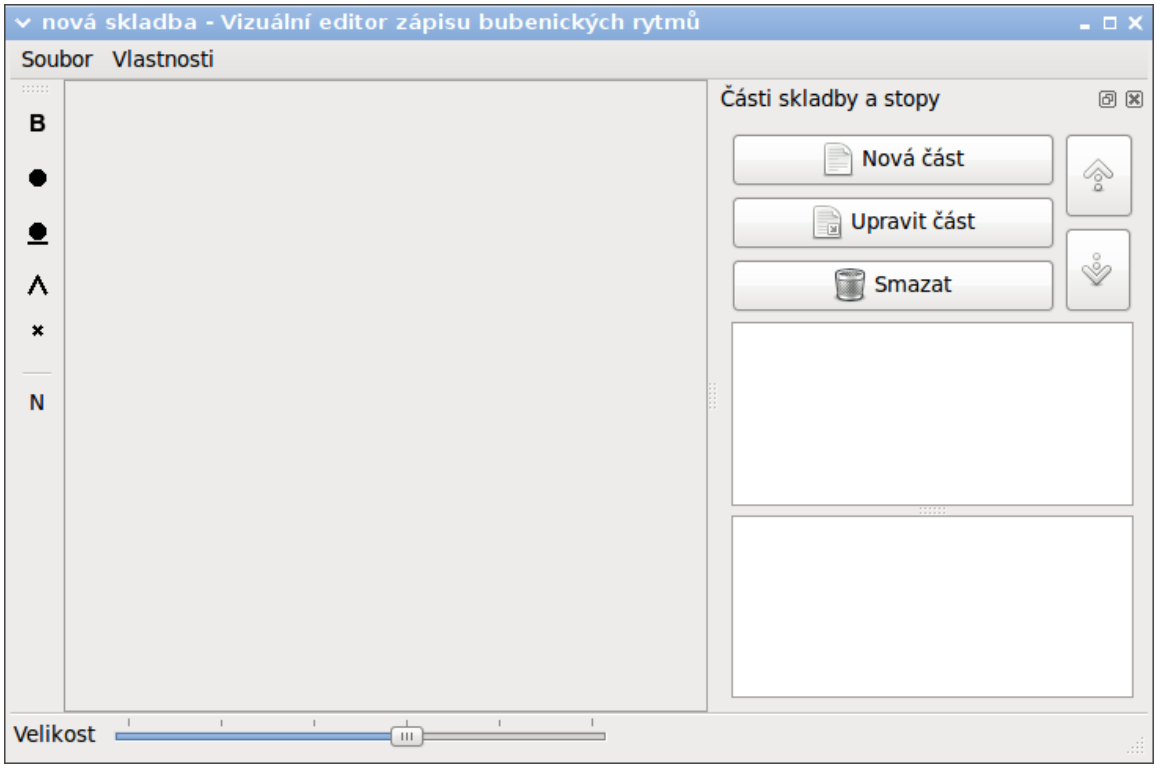
Poslední část hlavního okna aplikace je stavový řádek při dolním okraji. Zde je méně typicky umístěn posuvník, který umožňuje měnit velikost vykreslování notace. Toto umístění bylo zvoleno proto, aby byly využity všechny okraje hlavního okna. Více o této funkci je popsáno v kapitole 3.1.5.

[1] widget je každý prvek grafického uživatelského rozhraní ve frameworku Qt

Obrázek 3.1: Use case diagram



Obrázek 3.2: Hlavní okno aplikace



Tabulka 3.1: Obsah lišty *menu*

- Soubor
  - Otevřít
  - Uložit
  - Tisk
- Vlastnosti
  - Části skladby a stopy
  - Upravit nástroje

3.1.2 Postranní dokovací widget

Práce s celými částmi skladby a stopami jednotlivých skladeb se odehrává v postranním *dokovacím widgetu* (obr. 3.3). Zde jsou v horním seznamu zobrazeny všechny části skladby, které jsou součásti aktuálního dokumentu. Po kliknutí na určitou část se v dolním seznamu zobrazí všechny stopy, které jsou v části obsaženy.

Tento ovládací prvek umožňuje měnit pořadí úseků skladby i pořadí stop v rámci jednoho úseku. Tato činnost se ovládá tlačítky s ikonami šipek umístěnými v pravé horní části dokovacího widgetu. Pro přesun části skladby je nejprve nutné vybrat příslušnou část kliknutím myši a poté s ní pohybovat nahoru nebo dolů klikáním na tlačítka šipek.

Obrázek 3.3: Postranní dokovací widget

Přesouvání stop v rámci části skladby probíhá obdobně. Nejprve se vybere část a po zobrazení všech stop, které se v části nacházejí, je nutné vybrat stopu, kterou chceme přesunout.

K zobrazení nebo skrytí dokovacího widgetu existuje položka, která se nachází v menu *Vlastnosti* hlavního okna aplikace.

3.1.3 Formulář pro přidání a úpravu částí

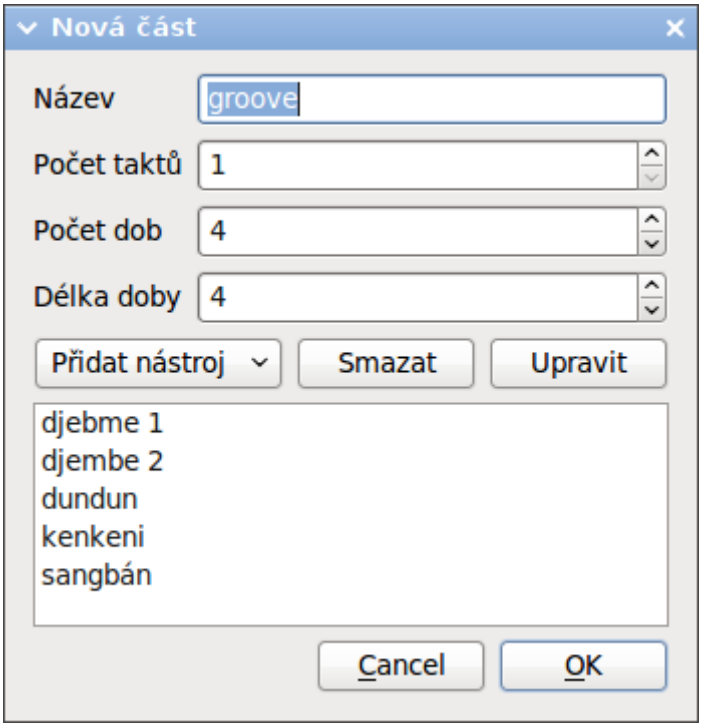
Po kliknutí na tlačítka *Nová část* nebo *Upravit část* umístěné na dokovacím widgetu se zobrazí dialogové okno (obr. 3.4), které nabízí možnost měnit parametry pro část skladby.

V závislosti na tom, jaké bylo stisknuto tlačítko, se buď otevře formulář nevyplněný nebo se do něj načtou stávající parametry části skladby, které se mohou upravit.

V otevřeném dialogovém okně se musí vyplnit povinný parametr pro název části skladby (např. *intro*, *groove* nebo *sólo*). Dále je třeba nastavit *počet taktů*, *počet dob* a *délku doby*. V poslední řadě se musí vybrat nástroje, které budou daný úsek skladby hrát.

Při stisku tlačítka *Přidat nástroj* se zobrazí menu s nabídkou názvů nástrojů, které je možno použít. Pokud si uživatel přeje přidat jiný nástroj, slouží k tomu tlačítko *Upravit*. To zobrazuje dialog pro úpravu názvů bubnů (kap. 3.1.4). Omylem zvolený nástroj je možné ze seznamu odstranit tlačítkem *Smazat*.

Obrázek 3.4: Formulář pro úpravu částí skladby



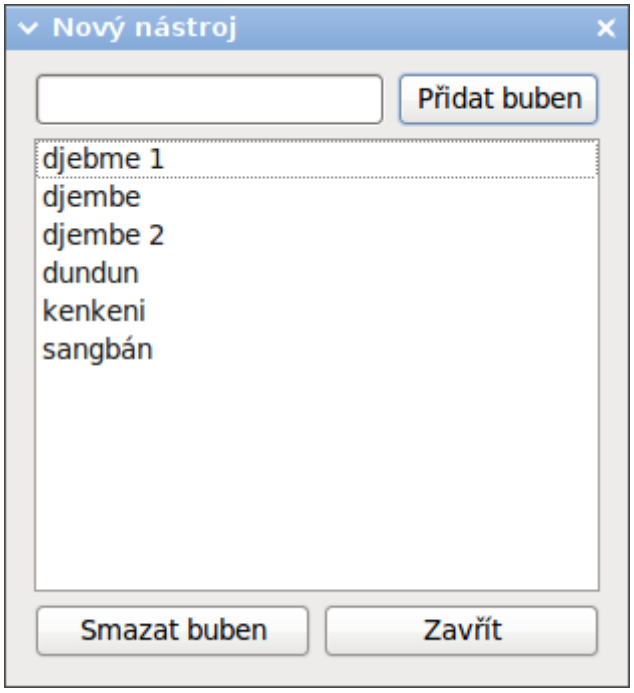
3.1.4 Formulář pro úpravu názvů bubnů

Názvy bubnů, se kterými je možné v programu pracovat se přidávají a ubírají v dialogovém okně *Nový nástroj* (obr. 3.5). Tento formulář je možné vyvolat dvěma způsoby. Klasicky z menu *Vlastnosti* položkou *Upravit nástroje* nebo z předchozího okna *Upravit část* a tlačítka *Upravit*.

Dialog zobrazuje seznam stávajících názvů bubnů, které je možné vybrat kliknutím myši a smazat stisknutím tlačítka *Smazat buben*. Nebo můžeme nový buben přidat napsáním jeho názvu do editační řádky a stiskem *Přidat buben*.

Operace, které se provádějí v tomto okně mají okamžitý účinek a je možné těchto změn ihned využívat.

Obrázek 3.5: Formulář pro editaci bubnů



3.1.5 Hlavní vykreslovací oblast

Největší část plochy okna aplikace vyplňuje hlavní vykreslovací oblast. Jedná se o prostor, kde se zobrazuje samotná notace, a kde je umožněna její editace. Screenshot této části okna zachycuje obrázek 3.6.

Parametry, které se nastavují v dialogu pro vytvoření nové části, výrazným způsobem ovlivňují velikost, kterou bude notace zabírat. Vykreslovací oblast tuto skutečnost reflektuje běžným způsobem, a to vytvořením posuvníků při pravém a dolním okraji plochy, pokud velikost notace přesahuje prostor, který je jí udělen rozměrem hlavního okna.

Další vliv, který mění velikost oblasti, je změna velikosti vykreslování. Ta se nastavuje posuvníkem umístěným ve stavovém řádku hlavního okna. Jeho manipulace má okamžitý vliv na vykreslování. Mění se zde rozteč mezi jednotlivými *pulsy* v *době*.

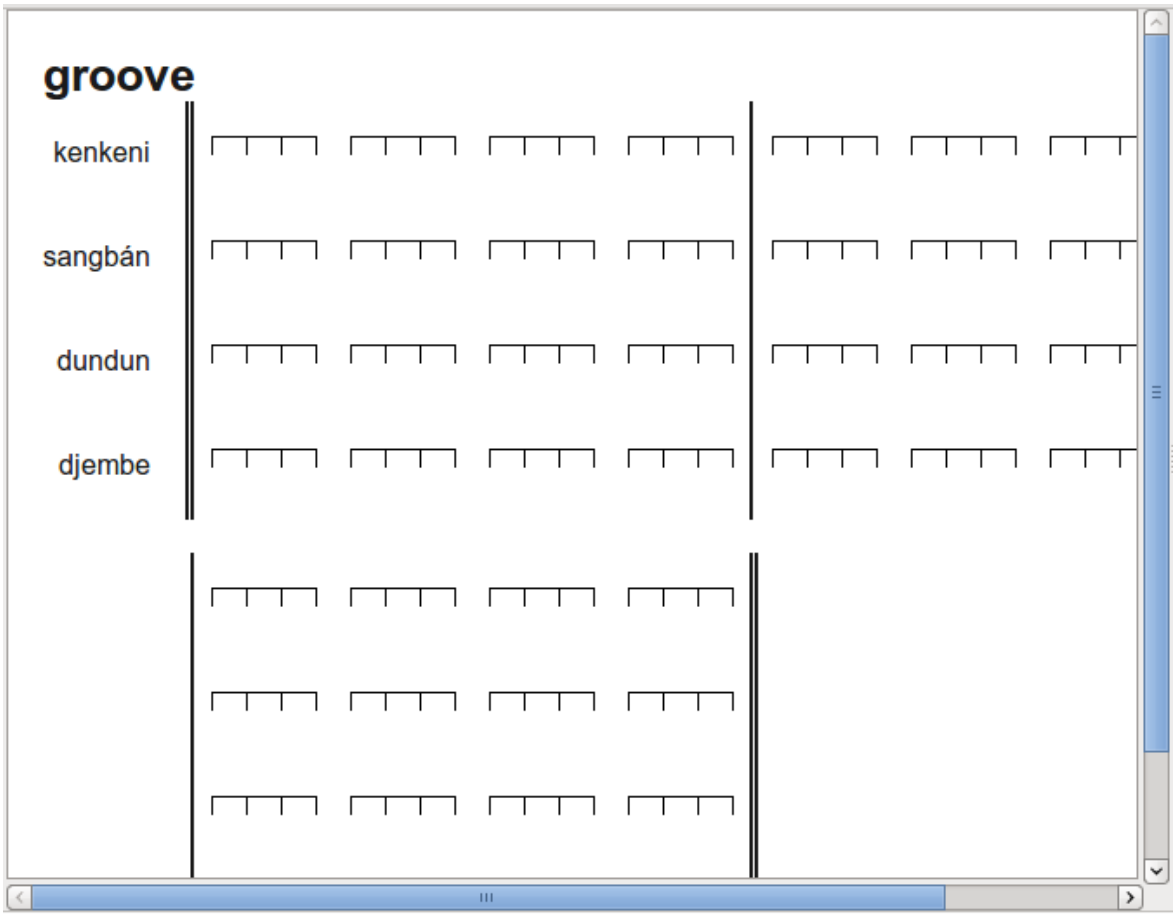
3.1.6 Popis editace dokumentu

Pokud uživatel vytvoří novou část notace, může přistoupit k samotné editaci. Ta probíhá výběrem vhodné noty z nástrojového panelu (**B**, 𝅗𝅥, 𝅘𝅥, ∧, ˣ) a kliknutím na příslušné místo v *době*. Průběh editace znázorňuje obrázek 3.7.

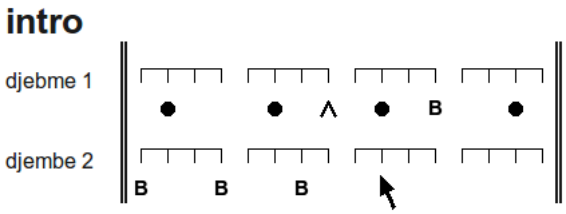
Kliknutí na *dobu* je zaregistrováno, pokud k němu dojde na oblasti, kterou zaujímá. Zároveň je přiřazeno k *pulsu*, který je k místu kliknutí nejblíž. Po kliknutí se vykreslí vybraná *nota*. Pokud již tato nota je na místě zapsána, vymaže se. Symbol *zvonec* se vykresluje o něco výše a je možné jej kombinovat s ostatními notami.

Posledním na nástrojové liště je speciální nástroj **N**. Jeho funkce nesouvisí s vykreslováním not, ale s manipulací s taktovými bloky. Přesněji slouží k zalomení taktového bloku na nový řádek. Účinek tohoto nástroje se projeví při kliknutí kamkoliv do taktového bloku, při kterém dojde k jeho posunutí na nový řádek.

Obrázek 3.6: Hlavní vykreslovací oblast



Obrázek 3.7: Zápis noty



Manuální zalamování řádků je zvoleno proto, aby měl uživatel maximální možnou kontrolu nad grafickým výstupem, který bude produktem programu.

3.2 Popis implementace uživatelského rozhraní

Z pohledu vývoje programu bylo nejprve nutné zvolit vhodnou platformu. Framework Qt jazyka C++ je moderní knihovna pro tvorbu grafických uživatelských rozhraní, která urychluje programování okenních aplikací a je multiplatformní.

Běh v různých operačních systémech byl jedním z důležitých požadavků pro výběr implementačního jazyka a frameworku. Očekává se, že výsledný program bude nejčastěji spouštěn v posledních verzích Microsoft Windows (XP až 7). Vývoj však probíhal v operačním systému GNU/Linux a směřoval k vytvoření multiplatformní kódu, který by stačilo jen přeložit pro cílovou platformu.

Samotné psaní kódu probíhalo v programu Qt Creator[2]. Jako podklad pro seznámení se s možnostmi tohoto nástroje i frameworku sloužily publikace [1] a [5]. Při programování byla konzultována dokumentace [4].

3.2.1 Třída `MainWindow`

Framework Qt pro případy vytváření hlavního okna nabízí třídu `QMainWindow`. Tato třída se při programování v C++ zdědí námi vytvořenou třídou `MainWindow`, a tak je možné využívat všechny možnosti, které Qt pro práci s hlavním oknem nabízí. Protože se jednalo o první vizuální komponentu Qt, se kterou získával autor zkušenosti, byl obsah hlavního okna programován ručně. Ostatní okna a dialogy již byly vytvářeny pomocí aplikace Qt Designer[3], který je součástí Qt Creatoru a umožňuje pohodlně editovat grafické rozhraní programu.

Hlavní okno aplikace je standardu **SDI**[4]. To znamená, že je možné v jedné spuštěné instanci programu pracovat pouze s jedním dokumentem. Opačně k tomuto typu uživatelského rozhraní existuje typ **MDI**[5], který dovoluje pracovat v jedné spuštěné aplikaci s více otevřenými dokumenty. Podrobněji je tento rozdíl popsán v publikaci [5].

Mezi využívané možnosti této třídy se řadí především metody pro přidání menu, lišty pro nástroje a status, nebo dokovací widgety. Rozložení jednotlivých části hlavního okna zobrazuje obrázek 3.8, jehož podklad vychází z [4].

Menu

Lišta s nabídkami se přidává do hlavního okna jako objekt třídy `QMenuBar`. V tomto programu jsou takto vytvořeny nabídky *Soubor* a *Vlastnosti*, jako instance `QMenu` . V těchto menu jsou dále umístěny *akce* spouštějící různé speciální funkce.

V menu *Soubor* se nachází možnosti pro otevření, uložení a tisk souboru. Akce pro otevření a uložení volají funkce `QFileDialog::getOpenFileName()`, vracející název souboru otevíraného pro čtení, resp. `QFileDialog::getSaveFileName()`, která vrací název ukládaného souboru. Obě tyto funkce otevírají speciální *dialog*, který je součásti knihovny Qt. Dialog pro otevření nebo uložení souboru nabízí možnost procházet adresářovou strukturu počítače, na kterém je program spouštěn, bez ohledu na hostitelský operační systém.

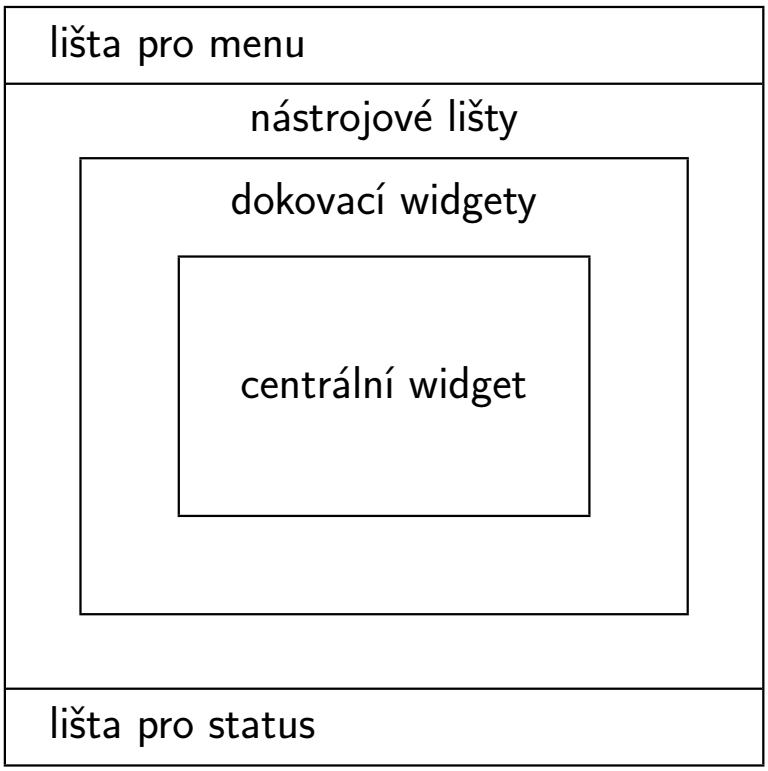
Pod položkou Tisk se ještě nachází možnost vytisknout dokument. Podrobněji bude tato část popsána v 3.2.5.

Nástrojová lišta

Další součástí hlavního okna aplikace je *Nástrojová lišta*. Tento ovládací prvek typicky uchovává tlačítka s různými funkcemi. V našem případě se zde nachází značky not, které se používají pro zápis notace.

[2] http://qt.nokia.com/products/developer-tools/developer-tools
[3] http://doc.trolltech.com/4.7/designer-manual.html
[4] Single Document Interface
[5] Multiple Document Interface

Obrázek 3.8: Schéma rozložení jednotlivých součástí hlavního okna aplikace (vychází z [4])



Všechny objekty třídy `QPushButton` na nástrojové liště jsou sdruženy do `QButtonBaru`, který nabízí možnost výlučného přístupu k tlačítkům. To znamená, že v jeden okamžik může být vybrán pouze jeden nástroj.

Tlačítka mají definovanou fixní šířku a jako obsah je nastavena ikona vytvořená z objektu třídy `QPixmap`, která slouží k manipulaci s jednoduchými obrázky. Symbol pro nástroj je vykreslen za běhu programu a je shodný se symbolem noty, který bude následně vykreslován do notace.

Nástrojovou lištu zobrazuje obrázek 3.9. V pořadí zleva doprava jsou zde tlačítka pro nástroje *bass*, *tón*, *tlumený tón*, *slep*, *zvonec* a *zalomení na nový řádek*.

Obrázek 3.9: Nástrojová lišta programu

Dolní lišta

Při dolním okraji hlavního okna je umístěn stavový řádek (obr. 3.10). Jedná se o komponentu označenou jako `QStatusBar`. Jeho úlohou v tomto případě je však pouze vytvořit prostor pro umístění posuvníku `QSlider`, který ovládá velikost vykreslování notace.

Obrázek 3.10: Spodní lišta programu

3.2.2 Třída `PropertiesDock`

Vlevo se po spuštění programu implicitně nachází `PropertiesDock`, který obsahuje zděděnou třídu `QDockWidget`. Již bylo řečeno, že zobrazení nebo skrytí tohoto prvku je možné ovlivnit položkou v menu *Vlastnosti*. Dok je možné zavřít i křížkem v jeho pravém horním rohu nebo se může vyčlenit z hlavního okna aplikace. K tomu slouží tlačítko vedle zavíracího křížku. Dále je možné dok uchopit myší za jeho horní lištu a posunout ven.

Implementace doku je samostatným modulem v programu a na hlavním okně aplikace nijak nezávisí. Ke slovu se zde dostal vizuální návrh prostřednictvím Qt Designeru, který je součásti vývojového prostředí Qt Creator. Framework zde umožňuje definovat vzhled prvků grafického uživatelského rozhraní a vnitřními mechanismy je umožněno dále s oknem pracovat v kódu programu.

Dalším používaným rysem frameworku Qt je pozicování *widgetů* pomocí *layoutů*. Tento postup je možné používat přímo v kódu i v návrhu vzhledu pomocí Designeru. V případě `PropertiesDock`u byl pro celý widget použit `QHBoxLayout` v jehož horní části je soustava dalších vnořených layoutů, ve kterých se již nachází samotná tlačítka. Zbylou část tvoří dva `QListWidgety`, které jsou zasazeny do `QSplitteru`. Tento speciální typ layoutu umožňuje svým widgetům rozdělovat podle potřeby prostor, který zaujímají.

`QListWidgety` zde slouží k zobrazení všech částí skladby a po vybrání příslušné části se zobrazí nástroje, které daný úsek hrají. Toto zobrazení dovoluje manipulovat s pořadím těchto objektů. Tlačítka s ikonami šipek po stisku posunout vybraný řádek v seznamu o jednu pozici nahoru nebo dolů. Změny pořadí probíhají v paměti, a proto se souběžně s touto manipulací mění i hlavní vykreslovací oblast a překresluje změny, které probíhají. Seznam částí skladby a stop se generují z paměťové struktury `MyData`, která je podkladem také přímo pro vykreslování notace.

Nakonec bych zmínil způsob implementace ikon, které se v programu vyskytují. Framework Qt nabízí využití systémových ikon, které jsou součásti operačního systému, ve kterém je program spuštěn. Této možnosti je však využito pouze u tlačítek šipek. Pro ostatní tlačítka jako *Nová část*, *Upravit část* a *Smazat* nebyly nalezeny vhodné možnosti, které by vypadaly na všech systémech podobně. Snahou bylo nemást uživatele při obsluze aplikace, a tak bylo od použití ikon v těchto případech upuštěno. Kvůli multiplatformnosti frameworku je nabídka základních ikon poměrně skromná a počítá se spíše se zařazením vlastní grafiky.

3.2.3 Třída `NewCollection`

Formulář pro přidání nové části nebo úpravu stávající je běžná reimplementace původní třídy `QDialog`. Tak jako všechny dialogy v této aplikaci je i tento nemodální. To znamená, že dovoluje pracovat s jinými částmi programu, i když je zobrazen.

Hlavní funkce tohoto dialogu jsou dvě. Pokud je vyvolán stiskem tlačítka *Nová část*, dojde k vymazání obsahu formuláře a vyplnění implicitními hodnotami. Při stisku tlačítka *Upravit část* musí být vybrána nějaká položka ze seznamu částí. Takto dojde k zobrazení formuláře, ve kterém jsou vyplněny parametry příslušné části a uživatel je může měnit.

Editační řádek slouží k pojmenování části skladby. Jeho vyplnění je nutné a bez něj není možné stiskem tlačítka *OK* dialog úspěšně ukončit. Zároveň musí být název části notace unikátní.

Dále následují tři prvky uživatelského rozhraní označené jako `QSpinBox`. Ty slouží k nastavení určité číselné hodnoty. Jedná se o *počet taktů*, *počet dob* a *délku doby*. Počet taktů značí počet taktových bloků v rámci jedné části skladby. Taktové bloky jsou úseky oddělené taktovými čarami. Počet dob je číslo udávající kolik jednotlivých prvků se v taktu nachází. Nakonec délka doby určuje velikost tohoto prvku, nebo-li počet jeho pulsů.

V okně dialogu následují tři tlačítka sloužící pro editaci počtu a názvů nástrojů. Tlačítko *Přidat nástroj* má přiřazené menu, které nabízí názvy nástrojů, které máme k dispozici. Menu je třídy `QMenu` stejně jako ty na horní liště programu. Vybráním některé z položek tohoto menu se příslušný název bubnu zahrne do seznamu umístěném pod tlačítky. Z tohoto seznamu lze vybraný buben dodatečně odstranit stiskem *Smazat*. Pokud nejsme s nabídkou názvů spokojeni, tlačítko *Upravit* otevře další dialog, který bude popsán v 3.2.4.

3.2.4 Třída `DrumDialog`

Dialog pro editaci názvů bubnů je možné vyvolat dvěma způsoby. Odkaz na něj je umístěn v menu *Vlastnosti* na horní liště programu. Druhý způsob se nabízí ve formuláři pro přidání nové nebo úpravu stávající notace, kde je k dispozici tlačítko *Upravit*.

Na začátku formuláře se nachází editační řádek třídy `QLineEdit`, do kterého je možné napsat požadovaný název nového bubnu. Po stisku tlačítka *Přidat buben* se tento název zařadí do seznamu `QListWidget`, který zobrazuje všechny názvy bubnů, se kterými je možné pracovat. U seznamu je nastaveno, aby se jeho obsah vždy řadil podle abecedy. Kromě přidávání nových bubnů je možné ze seznamu položky také mazat. To probíhá označením příslušného řádku a stisknutím tlačítka *Smazat*.

Po ukončení přidávání resp. odebírání bubnů ze seznamu dojde po zavření okna k uplatnění změn, které v dialogu proběhly. To obnáší aktualizaci globální paměťové struktury detailněji popsané v části 3.3.1.

3.2.5 Třída `DrawArea`

Hlavní úsilí směřovalo k tvorbě vykreslovací oblasti, kde se zobrazuje vlastní notace. Jedná se o třídu `DrawArea`, která dědí `QWidget`. Tomuto prvku je věnována největší část hlavního okna.

U objektu `drawArea`, který je instancí této třídy, je nastavena jeho velikost absolutně. Pokud není v programu zadána žádná notace, widget není zobrazen. Velikost se následně mění podle toho, jak velká notace je v něm vykreslena. Při fixním nastavování velikosti widgetu vzniká problém, pokud je widget větší než prostor, který mu rodičovský objekt nabízí.

Aby bylo zaručeno zobrazení celého widgetu zabírajícího fixně větší rozměr než jaký je mu uděleno v rámci okna programu, je tento widget umístěn do objektu třídy `QScrollArea`, který se stará o případné doplnění *scrollbarů*, aby bylo umožněno zobrazit celou plochu widgetu.

Do hlavní vykreslovací oblasti se najednou vykreslují všechny části skladby, které byly v programu vytvořeny. Každou část uvozuje její název. Dále jsou na začátku řádku nadpisy jednotlivých stop podle nástrojů hrajících daný úsek.

Při prvním vykreslení nové části notace se všechny takty umístí na jeden řádek. Na uživateli dále záleží, jak hodlá tento řádek rozdělit na další. K tomu slouží tlačítko **N** na

nástrojové liště, které se aplikuje kliknutím na určitý taktový blok. U tohoto taktového bloku se nastaví příznak zalomení řádku a po novém překreslení notace se tento takt již umístí na nový řádek.

Dalším prvkem, který se musí vykreslovat jsou taktové čáry. Pro větší zvýraznění zde byla zvolena šířka dva pixely. U prvního i u posledního taktu je nutné začátek a konec části zvýraznit zdvojením taktové čáry. Toho je docíleno vykreslením dvou taktových čar s jednopixelovou mezerou.

Mezi každými taktovými čarami nyní přichází na řadu vykreslení jednotlivých taktových bloků. Ty se prochází po řádcích a umísťují se zde dílčí widgety třídy `BeatW` popsané v 3.2.6. Tyto widgety se zde pozicují absolutně, protože je potřeba mít naprostou kontrolu nad vzhledem notace. Tradiční využití *layoutů* při práci ve frameworku Qt zde nebylo možno využít, protože bylo nutné zachovat přesné rozestupy mezi jednotlivými *dobami*.

Kromě rozsahu notace má vliv na velikost hlavního widgetu i nastavená velikost vykreslování. Ta má implicitní hodnotu 20 pixelů a její změna se uchovává i po ukončení programu (viz 3.3.1). Velikost vykreslování se nastavuje posuvníkem ve stavovém řádku aplikace.

Tisk dokumentu

S třídou `DrawArea` také souvisí implementace tisku notace. Je třeba zachovat co nejvěrohodnější podobu výsledného dokumentu s návrhem, který se v programu tvoří. Protože je editor typu WYSIWYG, snahou bylo vykreslovací oblast předat na tisk ve stejné podobě, jaká je zobrazena v okně programu.

Funkci tisku ve frameworku Qt obstarává třída `QPainter`. Pokud je v tomto objektu nastaven tiskový mód `QPrinter::ScreenResolution`, velikost objektů na obrazovce počítače přesně odpovídá tomu, co se vytiskne.

Dialog pro tisk zajišťuje třída `QPrintDialog`. K jeho zobrazení dojde po výběru položky *Tisk* z menu *Soubor* na horní liště hlavního okna programu. Podobu dialogu v různých systémech zobrazují obrázky 3.11 a 3.12.

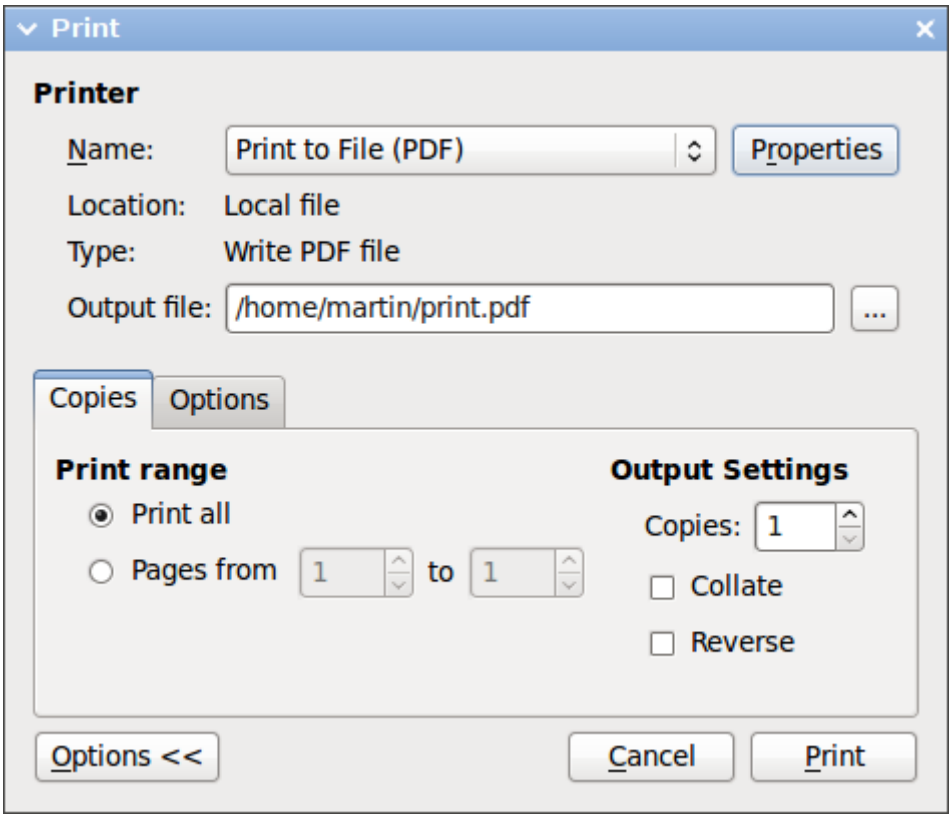
V prostředí, kde se program vyvíjel (linuxová distribuce Ubuntu s desktopovým prostředím Gnome) byla v nabídce i možnost exportovat dokument do PDF. Jedná se však o funkcionalitu operačního systému a program ani framework Qt ji v obecném použití dialogu neumožňuje. Pro stejnou možnost i v systému Windows je nutné nainstalovat nějakou implementaci virtuální tiskárny, která z daného dokumentu vytvoří PDF soubor.

3.2.6 Třída `BeatW`

Tento widget je další případ reimplementace třídy `QWidget`. Jedná se o zobrazení samostatné *doby* včetně případných not, které jsou v ní zapsány. Doba je vykreslena pomocí úseček a rozteč mezi jednotlivými svislými čarami je stanovena jako základní měřítko vykreslování v programu. Od této velikosti je odvozena i velikost widgetu, resp. šířka je rovna čtyřnásobku této velikosti a výška je rovna trojnásobku. Tuto závislost reprezentuje obrázek 3.13. Délka svislé čáry, pod kterou se zobrazují noty, je dlouhá polovinu rozteče čar. Šířka widgetu a vykreslení čar v něm bylo zvoleno tak, aby byla zachována stejná rozteč mezi jednotlivými pulsy v celém řádku.

Nad tímto widgetem je také implementována obsluha kliknutí myši a po označení příslušného místa v době dojde k vykreslení vybrané noty. Toto je jeden z hlavních důvodů, proč je *doba* reprezentována jako samostatný widget. Důležitá je pouze x-ová souřadnice kliknutí. Její hodnota se porovná, ke které nejbližší svislé čáře patří a pod tento puls se vykreslí nota. Symboly *bass*, *tón*, *tlumený tón* a *slep* se vykreslí na stejnou úroveň. Značka pro zvonec

Obrázek 3.11: Tiskový dialog Qt aplikace v Ubuntu (součást [4])



se umístí blíže k *době* tak, aby nepřekážela, protože může být zahrnuta společně i s jinou notou.

Symboly pro noty

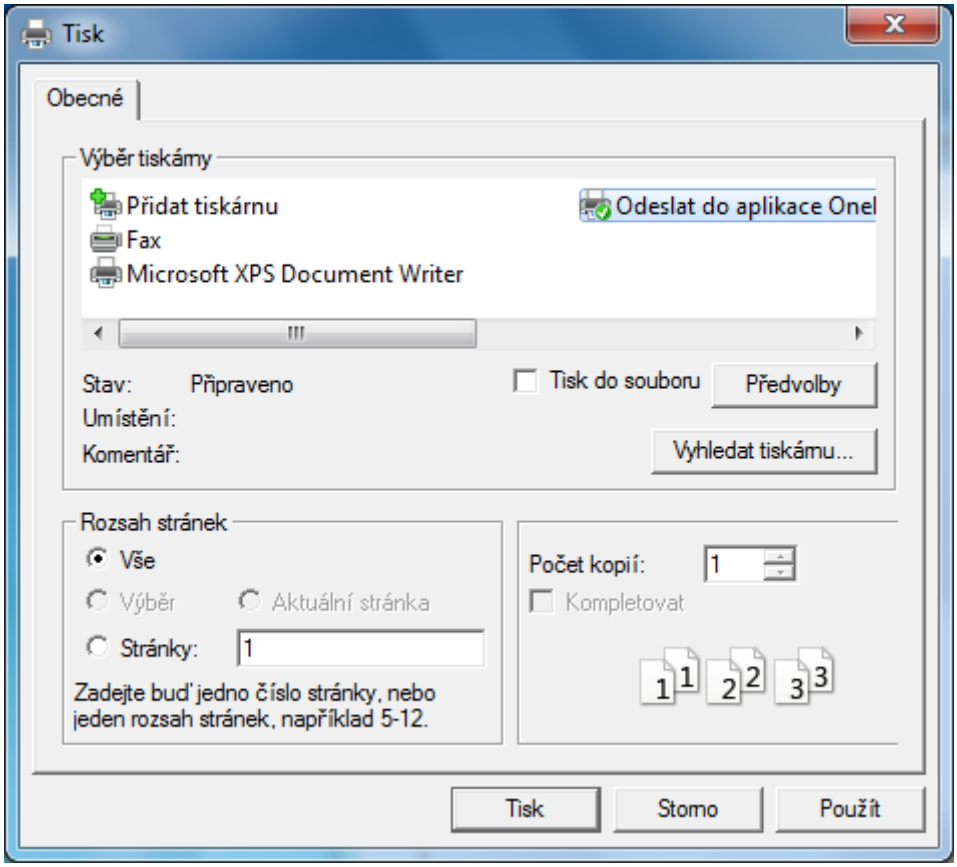
Podle vizuální specifikace not byla navržena i jejich počítačová reprezentace. Vhodné symboly bylo třeba umístit na tlačítka nástrojů i zahrnout je do vykreslování notace, resp. jednotlivých dob.

V původní verzi se diskutovala možnost vyjádřit všechny noty jako symboly ve znakové sadě UTF-8. Kromě písmena **B** se zde především jednalo o znaky $\bullet$, $\wedge$ a $^{\times}$. Všechny případy mají v UTF-8 své zastoupení a použití znaků nabízelo mnohé výhody. Znaky jsou reprezentovány vektorově a při zobrazení v Qt aplikaci u nich bylo použito vyhlazování hran. Tyto kvality byly velmi znatelné při zvětšení vykreslovaní pomocí posuvníku. Velikost fontu se měnila podle této závislosti.

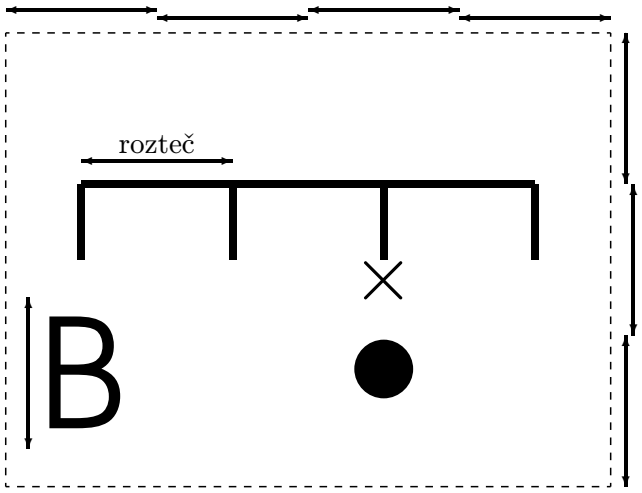
Bohužel se při testování aplikace na jiných platformách ukázal tento návrh jako nespolehlivý. Při testování ve Windows XP, Windows 7 i v jiných desktopových prostředích linuxových distribucí (KDE 4) často nebyly některé UTF-8 znaky dostupného fontu přítomny. Jejich místo nahrazoval prázdný obdélníček, což byla chyba, kterou bylo nutno vyřešit.

Přistoupilo se k vykreslování symbolů pomocí základních grafických primitiv, podobně jako byla při tvorbě *doby* využita přímka. Framework Qt pro tyto činnosti nabízí třídu `QPainter`, které se nastaví widget kam má kreslit a metody takto vytvořeného objektu již

Obrázek 3.12: Tiskový dialog Qt aplikace ve Windows 7



Obrázek 3.13: Schéma jednoho elementu zobrazujícího *dobu*



umožňují přímo měnit obsah widgetu.

Znak pro *tón* se nakreslil jako kruh vhodně zvolené velikosti. Ve variantě *tlumeného tónu* byl symbol doplněn o spodní vodorovnou čáru značící podtržení. Zbylé znaky jako *slep* a *zvonec* byly také vytvořeny vhodně nakreslenými úsečkami.

3.3 Popis vnitřní části implementace

Vnitřní implementací se zde rozumí části aplikace, které nesouvisí přímo s grafickým uživatelským rozhraním. Jedná se především o způsob uložení vlastních informací o částech skladby a notovém zápisu, který je oddělený od tříd starajících se o grafickou reprezentaci.

Dále zde přichází na řadu diskuse vhodného formátu pro uložení dat do souboru tak, aby je bylo možné zapsat na harddisk. Jako vhodný formát pro uložení notace bylo vybráno XML.

3.3.1 Třída `MyData`

Hlavní datový objekt, který prostupuje celým programem je instance třídy `MyData`. Jedná se o strukturu, která uchovává informace o zpracovávané notaci. Její umístění je globální, protože se využívá v celém programu. Deklarována je v modulu `main.c` a v ostatních modulech je specifikována klíčovým slovem `extern`.

Kromě uchování aktuální notace slouží objekt `myData` i k dalším činnostem. Jsou zde umístěna mnohá data, která se využívají v programu na více místech a tudíž mají také globální charakter. Jedná se například o seznam názvů bubnů, které mohou být využity při návrhu notace. Tento seznam spravuje okno `DrumDialog` a využívá jej formulář `NewCollection` pro přidání nové části. Ve třídě, která jej uchovává, jsou implementovány metody, které seznam bubnů načítají ze souboru při spuštění programu a po ukončení tento seznam opět do souboru ukládají. Pro jednoduchost se do tohoto souboru zapisuje i aktuální hodnota velikosti vykreslování notace. Soubor má název `drumfile.txt`. Pokud není v aktuálním pracovním adresáři nalezen, automaticky se vytvoří s implicitními hodnotami. V těchto případech je základní vykreslovací konstanta nastavena na 20 pixelů a jako seznam bubnů jsou uvedeny *djembe*, *dundun*, *kenkeni* a *sangbán*.

Dále se zde nachází proměnná udávající velikost vykreslování notace, objekt uchovávající aktuálně vybraný nástroj pro zapisování not, nebo název právě editovaného souboru. S ním souvisí i metody pro ukládání a načítání dat, které budou popsány v kapitole 3.3.2.

Poslední část, která je nutná na více místech programu, obstarává třída `Note`. Jedná se o třídu zprostředkující grafickou reprezentaci not (**B**, ●, ∧ a ˣ). Ty bohužel nebylo možné vyjádřit jednoduše pomocí UTF-8 znaků, jak bylo zmíněno v kapitole 3.2.6, ale musely se vykreslit samy. Třída `Note` obsahuje metody, které navrací objekty třídy `QPixmap`, které již obsahují nakreslené symboly. Toto řešení umístěné globálně bylo nutné, protože symboly se využívaly jak při vykreslování v hlavním okně, tak při vytváření tlačítek na nástrojové liště programu.

V další části je uvedena hierarchie tříd, ve které je uložena samotná notace.

Třída `Collection`

V hlavní datové struktuře je především umístěn seznam všech částí skladby. Každá část je dále reprezentována třídou `Collection`, která uchovává název části, seznam všech ná-

zvů stop a nakonec seznam taktových bloků, které přísluší do této kolekce. Nezbytné jsou metody pro nastavování různých vnitřních parametrů objektu.

Třída `Bar`

Jednotlivý taktový blok je reprezentován třídou `Bar`. Nachází se v něm příznak zalomení taktového bloku a seznam vnořených seznamů jednotlivých *dob*. Použití této konstrukce abstrahuje každý taktový blok na dvourozměrné pole, ve kterém řádky znamenají takty, které hrají jednotlivé nástroje. Toto řešení umožňuje potřebné slučování taktů do taktových bloků.

Zalomení znamená, že při vykreslování notace do hlavního okna aplikace se takto označený blok přesune na nový řádek. Tato činnost je čistě v rukou uživatele a je dostupná přes příslušné tlačítko v nástrojové liště. Dále je zde připraveno rozšíření do budoucna v podobě příznaků repetice (bude dále rozebráno v 4.4.2).

Třída `Beat`

Poslední logickou částí pro uchování záznamu notace je samotná *doba*. Je jí třída `Beat`, obsahující pro tuto dobu seznam *not* a seznam příznaků přítomnosti *zvonce*. Nota je reprezentována jako výčet (`enum`) a zvonec je značen logickou hodnotou. Délka těchto seznamů určuje délku doby. Pokud není pod daným pulsem zapsána nota, obsahuje seznam pod tímto indexem nulovou hodnotu.

3.3.2 Ukládání dat s využitím XML

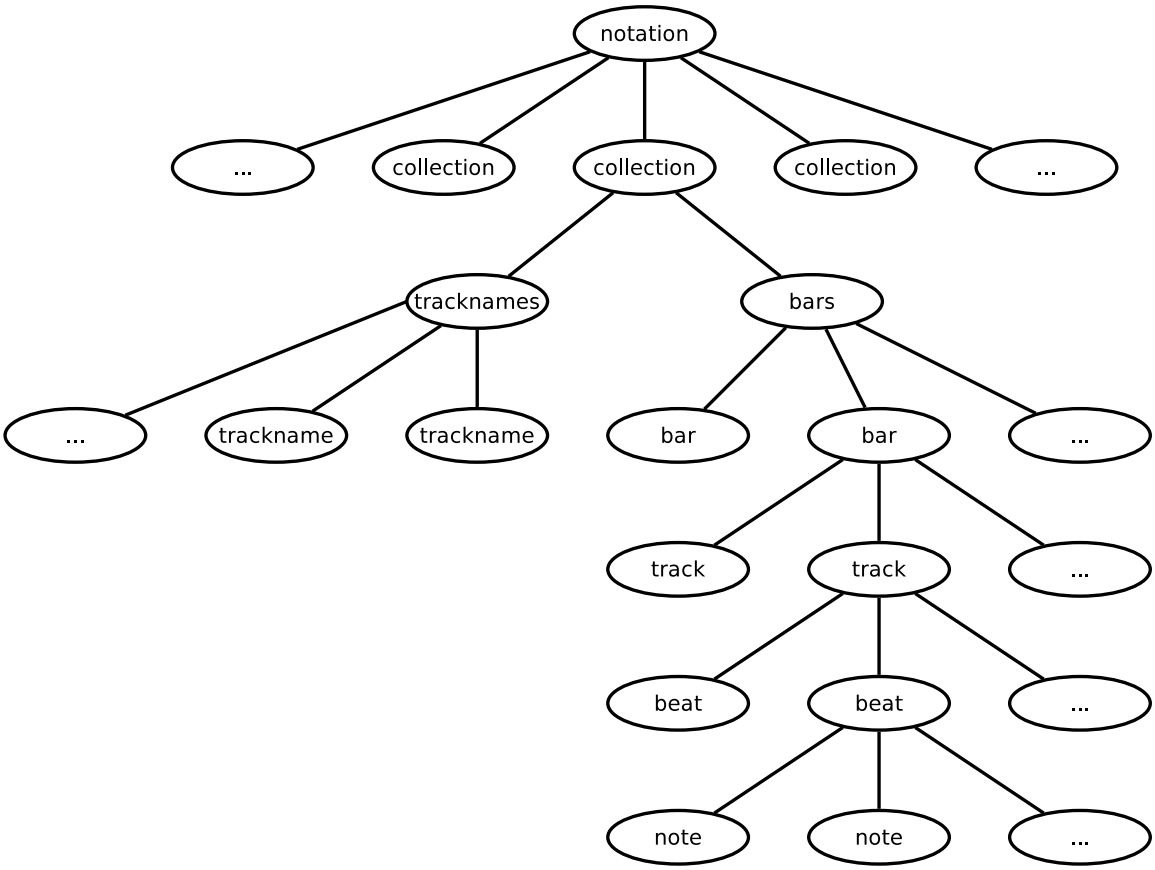
Objekt `myData`, se kterým se v programu pracuje, obsahuje části, které přímo popisují editovanou skladbu. Rozvržení notace a pozice jednotlivých not jsou dostačujícími daty, podle kterých je možné záznam obnovit z uloženého souboru.

Jedná se o stromovou hierarchii, ve které je kořenem celá notace. Ta obsahuje seznam částí skladeb označených jako kolekce. Toto označení spočívá v tom, že kolekce sdružuje jednotlivé stopy do společného řádku. Každá kolekce dále obsahuje seznam názvů stop a seznam taktových bloků. Pro každý blok obsahující dvojrozměrnou strukturu jednotlivých *dob* se do stromu zařazuje úroveň reprezentující řádek v taktovém bloku. Tento *track* již obsahuje konkrétní *doby* a každá *doba* obsahuje seznam *not*. Nota je listem tohoto stromu a má dva parametry. První z nich je logická hodnota přítomnosti úderu *zvonce* v tomto pulsu a dále je specifikován typ noty, který zde je zahrán. Zobrazení této struktury v podobě grafu se nachází na obrázku 3.14.

Tento stromový objekt je nutné transformovat do podoby XML dokumentu. Protože se zde s pracuje s celými objekty v paměti, framework Qt nabízí k této činnosti `QDomDocument`. Instance této třídy je reprezentací XML dokumentu v paměti a je možné s tímto dokumentem jakkoliv manipulovat. V programu je toto využito v metodách, které serializují důležité části struktury `myData` a provádějí její zápis do souboru nebo čtení z něj.

Serializace dat spočívá v průchodu objektu v několika vnořených cyklech a postupném budování XML dokumentu. Následně je tento soubor zapsán na disk. Podobu těchto dat zobrazuje 3.2. Čtení probíhá opačně. Ze souboru se načte instance `QDomDocumentu` a z jeho podoby je nutné obnovit datový objekt. O samotné parsování textu se stará kód knihovny Qt.

Obrázek 3.14: Grafové znázornění XML dokumentu



Tabulka 3.2: Struktura XML dokumentu

```
<notation>
    <collection name="název části">
        <tracknames>
            <trackname name="djebme"/>
            ⋮
        </tracknames>
        <bars>
            <bar postrep="0" beakline="0" prerep="0">
                <track>
                    <beat>
                        <note bell="1" type="2"/>
                        ⋮
                    </beat>
                    <beat>
                    ⋮
                    </beat>
                </track>
                <track>
                ⋮
                </track>
            </bar>
            <bar>
            ⋮
            </bar>
        </bars>
    </collection>
    <collection>
    ⋮
    </collection>
</notation>
```

Kapitola 4

Zhodnocení dosažených výsledků

Přínosem projektu je skutečnost, že aplikace bude mít po svém dokončení jistou uživatelskou základnu a program tedy bude někým používán. Vznikla tak náhrada zápisu notace na papír. Svým uživatelům také zjednodušuje možnost sdílení zapsaných skladeb nebo jejich publikaci.

4.1 Vyhodnocení zpětné vazby od uživatelů

Program je primárně určen pro hráče na perkusní bicí nástroje, kteří jej budou využívat na zaznamenávání vlastních rytmů. Během vývoje byly funkce programu konzultovány s různými bubeníky a vývoj je směřován tak, aby potřeby uživatelů byly naplněny co nejlépe. Snahou je neklást skladateli rytmu žádné omezující podmínky, které by zabraňovaly vyjádřit rytmus tak, jak si jej autor přeje zapsat. Nutné je ovšem dodržet obecně užívanou notaci pro perkusní bubny.

Protože počet uživatelů testujících software v průběhu jeho vývoje byl v řádu jednotek, probíhala komunikace osobně. Veškerým připomínkám, které vznikaly se vyhovělo co nejlépe a nejrychleji.

4.2 Parametry programu

V této části bych chtěl shrnout informace týkající se technické stránky samotného programu. Jedná se o specifikaci minimálních požadavků, které musí být dodrženy. Dále jsou zmíněny metriky tohoto software.

4.2.1 Nároky na hardware a operační systém

Program má pro své spuštění minimální hardwarové nároky. Jeden z počítačů, kde byla aplikace také testována, měl 32-bitový procesor AMD Athlon XP a běžely na něm Microsoft Windows XP Professional.

Ovládání programu včetně editace rozsáhlejších notací probíhalo naprosto plynule. Toto nabízí úsudek, že aplikace bude spustitelná na většině současných počítačů.

4.2.2 Softwarové metriky

Program byl kompletně psán ve vývojovém prostředí Qt Creator. Tato možnost nabízí rozmístění ovládacích prvků budované aplikace v editoru Qt Designer. Díky tomu je samotný

kód oproštěn o část definující pozice jednotlivých elementů.

Celkový počet řádků ve všech ručně psaných souborech je 1855. Kromě souboru `main.c` bylo vytvořeno sedm dalších modulů a k nim odpovídajících sedm hlavičkových souborů. Většina modulů obsahuje jednu třídu grafického uživatelského rozhraní. Třídy vnitřní implementace jsou obsaženy v modulu `mydata`. Zdrojové soubory programu včetně souborů vytvořených Qt Designerem mají 83,7 KB. Ručně psaná část zaujímá 57,9 KB. Přeložený program pro 64-bitovou architekturu v operačním systému GNU/Linux je velký 2,4 MB. Kód programu je celkem rozložen do dvanácti tříd.

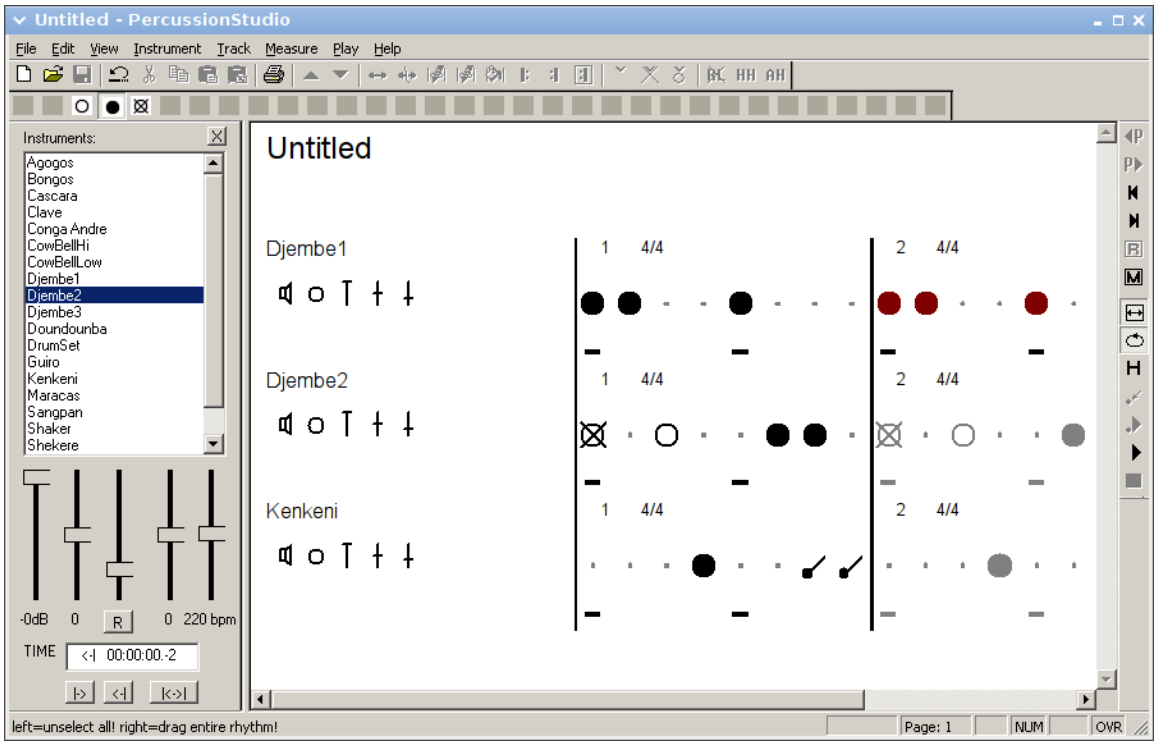
4.3 Srovnání s existujícími programy

Přesnou podobu notace pro perkusní bubny nenabízí žádná z prozkoumaných aplikací. Protože vyvíjený program je zaměřen poněkud odlišně od jiných testovaných aplikací, je vzájemné srovnání komplikovanější.

4.3.1 Percussion Studio

Software Percussion Studio, které vyvíjí firma Mooseware [3] je nástroj pro skládání rytmů pro perkusní bicí nástroje. Umožňuje rytmus přehrát, ale nabízí notaci, která je komplikovanější a méně přesná. Uživatelům zde chybí rozdělení taktů na jednotlivé *doby*, a tak se výtisk této notace nedá příliš využít při výuce hraní.

Obrázek 4.1: Screenshot Percussion Studia



4.3.2 FL Studio

Program FL Studio se řadí do jiné kategorie než předchozí případ. Jedná se o komplexní nástroj pro tvorbu hudby a obsahuje mnoho předpřipravených *samplů*. Je zde zařazen, protože mimo jiné nabízí i skladbu rytmů pro perkusní bubny. Výstupem programu je však zvuková nahrávka a vytisknout notaci v požadované podobě není možné.

Obrázek 4.2: Screenshot aplikace FL Studio, převzato z [2]

4.4 Možnosti budoucího rozšíření

Zvolená problematika editoru pro speciální hudební notaci skýtá mnoho možností budoucího rozšíření. V této kapitole chci přiblížit, kam bude v budoucnu vývoj editoru směřovat. Tyto nápady uvádím včetně odhadované náročnosti dané změny.

4.4.1 Přehrávání zaznamenaných rytmů

V prvé řadě na mysl přichází schopnost skládaný rytmus přehrát. Tato funkce by velmi zlepšila možnosti skladby pro méně zkušené hudebníky. Současný stav předpokládá, že hráči jsou již natolik zkušení, aby si mohli rytmus představit a editor tak slouží pouze jako nejjednodušší způsob, jak tuto myšlenku rychle zaznamenat.

Rozšíření na tuto funkci zjednodušuje naprosté oddělení tříd uchovávající zaznamenanou skladbu od widgetů starajících se o jejich grafickou reprezentaci. Protože je objekt obsahující notaci v programu globální, je možné jej využít i při přehrávání skladby.

Framework Qt nabízí i třídy pro práci se zvukem, a tak by toto rozšíření obnášelo nastudování této části dokumentace a pořízení vhodných zvukových nahrávek úderů vybraných nástrojů. Rozsah implementace tohoto rozšíření se již jeví jako pokročilý.

4.4.2 Vykreslování repetic

Další budoucí práce na programu spočívá v rozšíření možností notace. Tak jako v klasické hudební notaci se i v notaci pro perkusní bicí nástroje uplatňují některé prvky, které je vhodné poskytnout navíc.

Součást notace, která se také objevuje ve skladbách pro bubny, je způsob vyjádření repetice. Jedná se o zvýraznění úseku o délce jednoho nebo více taktů, který se hraje beze změny dvakrát po sobě. Otevírací repetice má podobu zdvojené taktové čáry doplněné o dvojtečku na pravé straně (‖:). Uzavírací část je vykreslena opačně (:‖).

Podoba repetice ve variantě pro perkusní bubny může mít více možností a její vyjádření mezi hudebníky není jednotné. Zvláště se zde rozumí to, jestli mají být tečky u každé linky taktového bloku, nebo stačí vykreslit tyto tečky společné pro celý taktový blok.

Programové řešení tohoto problému bylo navrženo přidáním dalších vlastností taktového bloku. Jedná se o příznak, jestli je levá nebo pravá taktová čára repeticí nebo není. Příslušná datová reprezentace je již v programu napsána včetně možnosti ukládání a načítání ze souboru. Ovládat se bude tento prvek vhodnými tlačítky z nástrojové lišty a kliknutím do prostoru taktového bloku. Po ujasnění, jak přesně má být repetice graficky znázorněna, bude přidání této funkcionality rychlé.

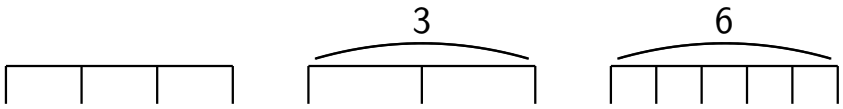
4.4.3 Trioly, kvartoly a sextoly

Speciální prvek, který se také objevuje v některých skladbách, je nepravidelný počet *pulsů* v určité *době*. Tyto speciální doby mají stejnou fyzickou šířku jako ostatní, ale mají jiný počet pulsů. Rozteč mezi jednotlivými svislými čarami je jiná. Navíc je tato *triola* (*kvartola*, *sextola*) označená i obloučkem nad vykreslenou *dobou* a nad ním je uvedeno číslo kolik *pulsů* v tomto případě *doba* obsahuje.

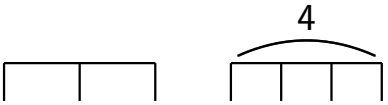
Úpravu *doby* délky čtyř *pulsů* na *triolu* a *sextolu* ukazuje obrázek 4.3. Rozšíření tří pulsů na *kvartolu* zobrazuje 4.4.

Jako řešení tohoto problému se jeví možnost nabídnout uživateli další tlačítka na nástrojové liště, které po aplikaci na určitou *dobu* nastaví tomuto prvku zvláštní počet *pulsů*, který bude odlišný od zbytku notace. Dále zde bude nastaven speciální příznak, který zajistí, že při vykreslování *doby* bude brán na toto nastavení zřetel a bude zajištěno i vizuální odlišení této *trioly*, *kvartoly* nebo *sextoly*.

Obrázek 4.3: Způsob zapisování *trioly* a *sextoly*



Obrázek 4.4: Způsob zapisování *kvartoly*



4.4.4 Rozdělování pulsů

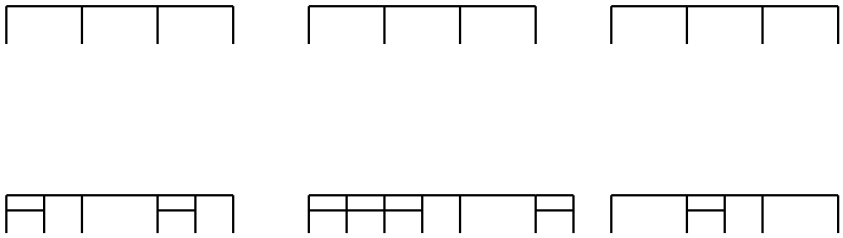
V některých případech může nastat situace s nevyhovujícím počtem pulsů, která není řešitelná změnou celé *doby* na *triolu*, *kvartolu* nebo *sextolu*. V tu chvíli je potřeba určitý puls rozdělit na dva poloviční. Dva údery zapsané do toho rozděleného pulsu jsou zahrány rychle za sebou. Ukázky dob s rozdělenými pulsy zobrazuje obrázek 4.5.

Jednotlivý *puls* je rozdělen tak, že se na jeho pravé straně vykreslí další *puls* a oba dva se spojí krátkou čárkou. Pokud se takto rozdělí dva sousední *pulsy*, všechny čtyři čárky se spojí dohromady. Další sousední *puls* však v případě jeho rozdělení zůstává samostatně.

Implementační návrh počítá s tím, že každý *puls*, pro který bylo nutné doposud ukládat jen samotný typ *noty* a příznak *zvonce*, bude dále obsahovat dodatečnou dvojici *nota-zvonec* a příznak samotného rozdělení *pulsu*. Při vykreslování se dále bude hlídat posloupnost pulsů a hledat příslušné shluky, které se musí speciálně zvýraznit.

O funkci rozdělení noty se bude také starat tlačítko na liště a tento nástroj se bude aplikovat na jednotlivé pozice v *době*, podobně jak probíhá zápis samotných *not*. Na nově vzniklou pozici bude možné zapsat notu nebo symbol zvonce běžným způsobem.

Obrázek 4.5: Ukázky dob s rozdělovanými pulsy



4.4.5 Optimalizace uživatelského rozhraní

Poslední část, která si bude žádat ještě určité rozšíření programu, se týká běžných standardů při práci s okenními aplikacemi. Tím se rozumí například zachování stejných rozměrů a pozice okna při dalším spuštění nebo rozmístění ovládacích prvků v programu. To se týká především pozice nástrojové lišty nebo *dokovacího widgetu*.

Framework Qt pro tuto úlohu nabízí třídu `QSettings`, která se snaží uložit v systému různé informace, které jsou pro běh programu potřeba nebo je vhodné je zachovat a využít při dalším spuštění. Toto uložení dat zprostředkovává Qt multiplatformně, a tudíž se jednotlivé implementace pro různé systémy mohou lišit. Třída `QSettings` však abstrahuje tento přístup na jednoduché uložení položek *klíč-hodnota* a tyto data dovoluje kdykoliv přečíst. Tento princip je popsán v oficiální dokumentaci [4].

Kapitola 5

Závěr

Práce si kladla za cíl naprogramovat editor pro vytváření zápisů rytmů. Této činnosti bylo dosaženo a výsledná aplikace je funkční a uživateli používaná. Prioritou pro aplikaci je grafický výstup v podobě vytisknutého notového zápisu, který slouží hráčům jako partitura při bubnování.

Během programování byly autorem nastudovány potřebné části frameworku Qt a prameny, ze kterých bylo čerpáno jsou uvedeny v sekci Literatura. Bohužel není možné citovat žádnou publikaci, která by byla zdrojem informací o samotné notaci. Jedná se o nepsaný standard mezi hráči a veškeré podrobnosti potřebné pro implementaci byly autorovi sděleny ústně s detailními příklady vzorových skladeb.

Dále je cílem autora přidat většinu prvků uvedených v kapitole 4.4. Především je vhodné doplnit možnosti rozšiřující základní notaci. Nutné bude také poskytovat podporu uživatelům, optimalizovat uživatelské rozhraní, případně opravovat chyby, které mohly v programu vzniknout.

Literatura

[1] Blanchette, J.; Summerfield, M.: *C++ GUI programming with Qt 4*. Trolltech Press, 2006, iSBN 0-13-187249-4.

[2] FL Studio: FL Studio web [online]. http://flstudio.image-line.com/, [cit. 2011-04-09].

[3] Mooseware.net: Percussion Studio web [online]. http://www.moosware.net/PercussionStudio, [cit. 2011-04-09].

[4] Nokia Corporation: Qt Reference Documentation [online]. http://doc.qt.nokia.com/latest/index.html, 2008-2010 [cit. 2011-04-05].

[5] Thelin, J.: *Foundations of Qt Development*. Apress, 2007, iSBN 978-1-59059-831-3.

Příloha A

Obsah CD

- `src` – zdrojové soubory programu
- `bin` – přeložený program
- `lib` – Qt framework
- `tex` – technická zpráva
- `poster` – plakát
- `readme.txt`

Příloha B

Plakát

Náhled plakátu, který je součástí zadání bakalářské práce se nachází na straně 36. Originál ve formátu A1 byl využit také při soutěži Student EEICT 2011[1], kde byla tato práce zařazena do posterové sekce bakalářských projektů.

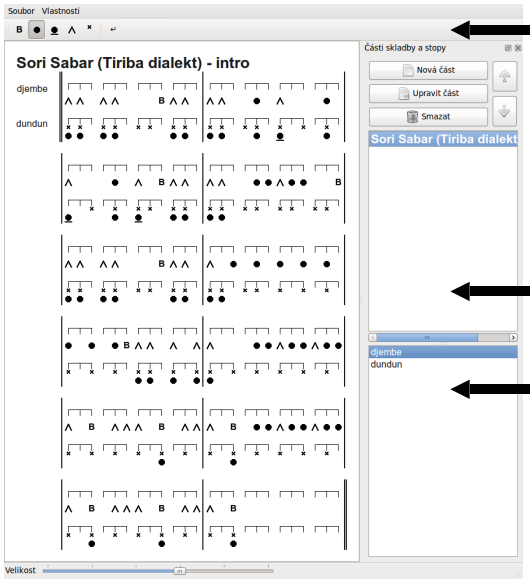
[1] http://www.feec.vutbr.cz/EEICT

Visual Editor of Drum Rhythms

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Martin Zelený
Bachelor Degree Programme, FIT BUT
xzelen00@stud.fit.vutbr.cz

Supervised by:
doc. Ing. Adam Herout, Ph.D.



nástrojová lišta — standardní notace pro perkusní bicí nástroje

šipky pro změnu pořadí

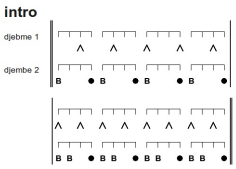
- B — bass
- ● — tón
- ◉ — tlumený tón
- Λ — slep (název speciálního úderu)
- × — zvonec
- ↵ — přesun taktu na nový řádek

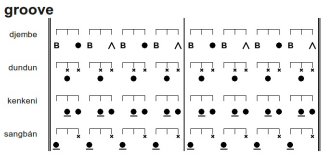
seznam všech částí skladby

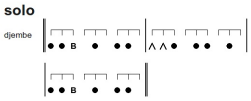
seznam nástrojů (stop)

velikost vykreslování notace

tisk zaznamenané notace







Implementace:

- napsáno v C++ s využitím frameworku Qt 4
- ukládání souborů do struktury XML
- hlavní vykreslovací plocha složena z dílčích widgetů
- zápis not vybráním nástoje (typu úderu) a kliknutím na příslušné místo v notaci
- WYSIWYG - přímý tisk vykreslovací plochy v nezměněné podobě

Bakalářská práce
Grafika a multimédia
2010/2011