



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Nástroj pro vizualizaci datového toku formátu MIDI pro živá vystoupení
Student:	Vadim Petrov
Vedoucí:	Ing. Adam Sporka, Ph.D.
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Proveďte rešerši existujících softwarových nástrojů pro vizualizaci hudby, zejména v kontextu použití standardu MIDI. Realizujte systém, který bude generovat vizuální odezvu na vstupní proud dat ve formátu MIDI. Typickým případem užití bude vizualizace živé produkce hudebníka během koncertu. Navrhněte estetický a přitom technicky netriviální princip vizualizace. Systém by měl pracovat s vaše nou znalostí interpretované skladby, aby bylo možné tento systém použít v rámci definované dramaturgie koncertu, aby konkrétní skladby mohly být vizualizovány předešle definovanou sadou vizuálních prvků a tyto prvky odpovídaly určitým pasážím dané skladby. Systém by během použití měl vyžadovat minimální vstup od uživatele, ale tento vstup bude možný. Systém otestujte například živě ve stejném vystoupení.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
ředitel katedry

V Praze dne 17. února 2016

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



Bakalářská práce

Nástroj pro vizualizaci datového toku formátu MIDI pro živá vystoupení

Vadim Petrov

Vedoucí práce: Ing. Adam Sporka, Ph.D

16. května 2016

Poděkování

Ing. Adamu Sporkovi Ph.D za vedení a konzultace během přípravy práce, MgA. Michaeli Frančemu za 3D model medových pláství, Gurkanu Te za ilustrace veverek, Davidu Čajčíkovi za organizaci a spoluúčast při testování, zvukovému techniku z Kulturního centra Klubovna za záchranu testování, členům kapely Human Ketchup za hudební podporu při testování a všem účastníkům veřejného testování

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Vadim Petrov. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Petrov, Vadim. *Nástroj pro vizualizaci datového toku formátu MIDI pro živá vystoupení*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato práce se zabývá tematikou hudebních vizualizací. Analyzuje předchozí technická řešení a jejich synchronizační možnosti a implementuje systém generující vizuální odezvu na datový proud formátu MIDI v reálném čase. Tento systém je realizován s pomocí jazyka C++ a knihovny OpenGL. Systém byl úspěšně testován během živého veřejného vystoupení.

Klíčová slova MIDI, hudba, vizualizace, C++, OpenGL, 3D, počítačová grafika

Abstract

The topic of this work is music visualization. It analyzes previous technical solutions and the corresponding synchronisation possibilities and implements a system designed to generate real-time visual feedback based on a MIDI data stream. This system makes use of the C++ programming language and the OpenGL environment. The system was successfully tested by the means of a live public performance.

Keywords MIDI, music, visualization, C++, OpenGL, 3D, computer generated imagery

Obsah

Úvod	1
Cíle práce	2
1 Analýza	3
1.1 Hudební prostředí	3
1.2 Technické možnosti	8
1.3 Možnosti synchronizace	12
1.4 Shrnutí	17
2 Návrh	19
2.1 Obecný návrh	19
2.2 Technické požadavky	19
2.3 Workflow	20
2.4 Technické náležitosti	22
3 Implementace	25
3.1 Architektura	25
3.2 Použité technologie	27
3.3 Diagram tříd	28
3.4 GLSL shadery	35
3.5 Implementované efekty	35
3.6 Technická řešení	38
4 Testování: Realizované hudební vystoupení	41
4.1 Testovací prostředí	41
4.2 Test I: Zpoždění odezvy	43
4.3 Test II: „Skála“	43
4.4 Test III: „Definice“	44
4.5 Test IV: „Veverka“	45
4.6 Průběh testování	47

4.7 Shrnutí	48
Závěr	49
Splnění cílů práce	49
Budoucnost	50
Použité zdroje	51
A Seznam použitých zkratk	53
B Obsah přiložené SD karty	55

Seznam obrázků

1.1	Plachty s logem kapely	4
1.2	Kulisy na pódiu při koncertě Inquisition	5
1.3	Kulisy na pódiu při koncertě Amon Amarth	6
1.4	Vizualizace v programu Winamp	9
1.5	Mapping a VJing	11
1.6	Živé kódování	11
1.7	Graf spektrální analýzy skladby s ženským vokálem	13
1.8	Graf spektrální analýzy skladby s kytarovými vyhrávkami	14
3.1	Architektura systému	26
3.2	Globální diagram tříd	29
3.3	Diagram tříd objektů ve scéně	30
3.4	Diagram tříd testovacích skladeb	33
3.5	Diagram ostatních objektů	34
3.6	Náhodné texturovací souřadnice	36
3.7	Trhání obrazovky	39
3.8	Tunel z bloků stavebnice	40
4.1	Závěrečná fáze vizualizací skladby „Skála“	44
4.2	Osvícený válec ve skladbě „Definice“	45
4.3	Veverka s oříškem ve skladbě „Veverka“	46

Seznam tabulek

4.1	Test zpoždění odezvy	47
-----	--------------------------------	----

Úvod

Hudba je v současné době jedno z nejvýznamnějších uměleckých médií. V kontextu technologického pokroku se rychle vyvíjí a pružně reaguje na konkrétní nové technologie. Ty jednak umožňují vznik dalších a dalších žánrů a podžánrů, jednak mění formy, ve kterých je hudba prezentována. Běžně tak hudební vystoupení přebírají vliv z jiných uměleckých forem a s vlivem i technologie blízké dané cizí formě. Vznikají pak více či méně multi- a intermediální díla charakteristická pro dnešní dobu.

Hudbě se aktivně věnuji od studia gymnázia, kde jsem se spolustudenty založil nyní již neaktivní black metalovou kapelu. Dlouhodobě se také věnuji výtvarnému umění, původně klasické kresbě, později počítačové grafice. Přirozeným vývojem jsem došel k průniku obou médií, který umožňují právě dnešní technologie.

Aktivní tvorba se málokdy obejde bez pohybu v lokální, případně globální scéně. Pro mne je domácí pražská klubová scéna, na které působí značné množství kapel, od studentských až po zavedené, napříč žánry a styly. Od jejího založení působím v pražské promotérské agentuře BrainWave Production, která zajišťuje koncerty zejména zahraničních metalových kapel. Dále sleduji vývoj elektronické hudby, a to zejména té evropské – jak anglické, tak kontinentální.

Během mého působení na pražské hudební scéně jsem zaznamenal řadu originálních vystoupení, ať už samotným hudebním projevem, použitými hudebními nástroji a dalšími technologiemi, či právě vizuální stránkou. Z takto nasbíraných zkušeností pramení inspirace pro tuto práci a konkrétní případy budou v patřičných kapitolách podrobněji rozebrány.

Všechny vyjmenované vlivy chci ve své práci spojit v jedno dílo, kde se elektronická a metalová hudba spolu s vizualizacemi nebudou přebíjet, ale splynou v jeden intermediální celek s čitelnými souvislostmi, a to jak vnitřními – jednotlivé hudební či vizuální prvky si budou navzájem odpovídat, tak vnějšími – odkazy na současnou nebo minulou popkulturu a další různě známé prvky.

Krom rozhraní MIDI jsem jako hlavní technické prostředí zvolil grafickou knihovnu OpenGL a programovací jazyk C++. Pro MIDI mluví hlavně jeho zásadní rozšíření na jinak dost nesourodém poli elektronických hudebních nástrojů. OpenGL a C++ svou mírou abstrakce umožňují poměrně pohodlnou práci pro vývojáře a zároveň dostatečně vysoký výkon pro pokud možno co nejmenší zpoždění za relativně vysoké složitosti vykreslované scény. Tyto technologie jsem doplnil ještě dalšími knihovnami.

Cíle práce

1. **Provést rešerši existujících softwarových nástrojů pro vizualizaci hudby**, zejména v kontextu použití standardu MIDI. Problém vizualizace je s hudbou úzce spjatý a technologický vývoj na něj má zásadní vliv. Cílem je zmapovat předchozí řešení, jejich přednostmi se inspirovat a nedostatkům se vyhnout.
2. **Navrhnout technicky netriviální princip vizualizace** Technický princip zahrnuje rozšiřitelný návrh systému a netriviální řešení vzniklých problémů.
3. **Navrhnout esteticky čitelný vztah zvuku a obrazu** Ve vizualizacích různých skladeb budou patrné odlišnosti, skladby mohou mít jedinečné i společné prvky, aby bylo možné tento systém konzistentně použít v rámci definované dramaturgie koncertu.
4. **Navrhnout systém vhodný pro využití během živého vystoupení** Systém by během použití měl vyžadovat minimální vstup od uživatele, ale tento vstup bude možný. Systém bude možné konfigurovat pro rozdílnou dramaturgii či pro rozdílné technické vybavení. Během vizualizací bude primárním vstupem datový proud MIDI.
5. **Realizovat systém generující vizuální odezvu na vstupní proud dat ve formátu MIDI** Systém bude odpovídat poznatkům získaných rešerší a provedeným návrhům. Systém bude pracovat s dostatečně nízkým zpožděním a výstup bude v dostatečně vysoké technické kvalitě.
6. **Otestovat systém při živém veřejném vystoupení** Jako intermediaální dílo bude systém veřejně prezentován. Z prezentace bude pořízen audiovizuální záznam a tento záznam bude součástí této práce.

Analýza

Tato práce navazuje na předchozí díla podobného charakteru. Vychází z kontextu jak kulturně-hudebního, tak čistě technického. Historie hudebních vizualizací by se dala chápat již od začátku dvacátého století (případně i dříve), historická analýza ale není předmětem této práce a proto se soustředí na vybrané případy, ze kterých může přímo čerpat inspiraci.

1.1 Hudební prostředí

1.1.1 Dva hudební proudy

V současné době lze pozorovat dva hlavní hudební proudy (viděno jednou z mnoha možných optik), pro účely této práce je nazývejme hudbou tradiční a hudbou elektronickou. Za tradiční hudbu považujeme takovou, která je produkována zejména na hudební nástroje v tradičním slova smyslu, tedy nástroje perkusní, žesťové, strunné apod. Zahrňme do této kategorie i nástroje s elektrickými snímači či zesilovači, tedy zejména elektrické kytary, tolik charakteristické pro rockovou hudbu. Za hudbu elektronickou považujeme hudbu produkovanou zejména pomocí čistě elektronických nástrojů, typicky různých druhů syntezátorů a samplerů. Toto dělení je poměrně vágní a v mnoha případech těžko aplikovatelné, jak si ostatně ukážeme dále, nicméně hraje svou zásadní roli. Oba hudební proudy mají své zastánce a smutnou realitou zůstává, že zastánci jednoho proudu často odmítají ten druhý. V některých případech se jedná o generační rozdíl, v jiných o osobní názor na konkrétní hudební estetiku.

1.1.2 Vizualní stránka hudebních proudů

Pro tradiční hudbu je vizualní součást vystoupení obvykle méně významná (pokud tedy pomineme dnes již historické žánry jako balet nebo opera, kde navíc silně figuroval příběh a celé dílo mělo blíže divadlu). Pro začínající kapelu

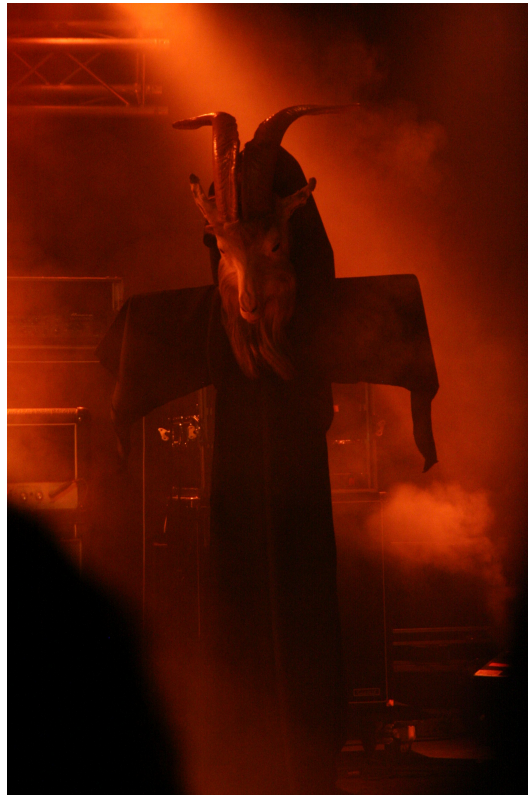


Obrázek 1.1: Kytarista kapely Sólstafir s plachtou s logem v pozadí, Brutal Assault 2015, foto autor

je nejvýznamnější prioritou opatřit si hudební nástroje a dosáhnout dostatečně vysoké úrovně ve hře na ně. Vrcholem mnoha kapel je pak pořízení několika-metrové plachty s logem či ilustrací stylově odpovídající danému žánru, viz obrázek 1.1.

Při koncertech nelze opomenout světelnou show. Tu si obstarávají samotné kapely poměrně zřídka, typicky se jedná o světelný park lokace, kde se koncert odehrává, tedy hudební klub či venkovní pódium. Z toho plyne jeden z hlavních problémů, které má tato práce za cíl řešit: nízká souvislost a nárazově vysoké zpoždění ve vztahu zvuku a obrazu. Pokud se na jednom pódiu po sobě během klubového koncertu vystřídají tři kapely, během open-air festivalu třeba deset, a všechny osvětluje jeden technik s jedním světelným parkem, výsledný vizuál působí genericky a okoukaně až nudně. Nejedná se jen o omezené množství světelných efektů, ale jejich potenciálně nevhodný vztah k hudbě. Zejména pokud je hudba velice dynamická a pracuje s ostrými zvraty, neobeznámený světelný technik nemá šanci dostatečně rychle reagovat a výsledkem je zpoždění někdy až několika dob.

Nejhorším případem vůbec jsou kluby s kapacitou do přibližně sta návštěvníků (ve kterých se ale odehrává velké množství hudebních vystoupení), které mají základní světelný park sestávající z několika málo levných světel a jednoho éru s presety. Přednastavenou sekvenci světel pak pustí zvukový technik



Obrázek 1.2: Kulisy na pódiu při koncertě Inquisition, Brutal Assault 2014, foto autor

a dál se o světla nestará. Podobnou variantou jsou pak světla se zabudovaným mikrofónem, která během vystoupení nepředvídatelně blikají a celkový dojem je obdobně obludný.

Jedním řešením je vlastní světelná show. Jedná se o řešení finančně nákladné a pro tradiční hudbu spíše atypické, proto se k němu vrátíme později. Obvyklejším řešením je použití různých kulis a objektů přímo na pódiu. Nejodvážnějšími jsou v tomto smyslu asi kapely spadající do různých podžánrů metalové hudby, jako příklad poslouží satanistický black metal Inquisition (obrázek 1.2) či folk metal (fúze metalu s tradičními, typicky pohanskými nástroji, melodiemi a tématy) Amon Amarth (obrázek 1.3).

Zcela jinak situace vypadá v případě elektronické hudby a dá se předpokládat, že v podstatě z historických důvodů: během koncertu skupiny tradičních hudebníků se na pódiu pořád něco děje, různé nástroje vyžadují různé ovládání, někdo sedí, někdo stojí, v zásadě je neustále co pozorovat. V případě elektronické hudby je na pódiu jen jeden člověk – DJ, který buď otáčí potenciometry, na které není pořádně vidět, nebo v klasické době vinylových desek hledá tu vhodnou desku někde v krabici pod stolem.



Obrázek 1.3: Kulisy na pódiu při koncertě Amon Amarth, Brutal Assault 2014, foto autor

Upoutat zrak posluchačů tedy muselo něco jiného. V případě psychedelického trance se jedná často opět o kulisy, tentokrát však pomalované fluorescentními barvami a nasvícené ultrafialovým světlem. Běžnější jsou video projekce, opět v kombinaci se světly. Video typicky zpracovává opět další osoba, tzv. VJ, a pokud nemá na starosti i světla, jedná se opět o dalšího člověka. Na rozdíl od nenahraditelných zvukových techniků VJové nepracují dlouhodobě v klubech, ani nespolupracují s konkrétními hudebními projekty, místo toho se sdružují do vlastních skupin, které se krom VJingu často věnují i video mappingu a příbuzným aktivitám. Zejména díky těmto aktivitám jsou projekce od mladých VJů často kreativní, VJing ale opět naráží na problém, na který naráží i světelný design: asynchronnost hudebního a vizuálního projevu. Elektronická hudba bývá typicky repetitivní, což role VJe i světelného technika často usnadňuje, dochází ale k singulárním momentům, kdy DJ využije zaběhlou repetitivnost a udělá něco naprosto nečekaného, a stává se tak častěji, než by se mohlo zdát. Volba vhodného vizuálu na sebe pak dá čekat.

Tradiční kapely typicky hrají podle předem domluvené dramaturgie, zmíněnému problému se tak dá do celkem velké míry předcházet. Jinak je tomu u DJů, kteří typicky během celého setu improvizují a běžně si udělají představu o tom, co vlastně budou hrát za skladby, až během setu DJe, který je předchází – odhadnou náladu a vkus davu a přizpůsobí se mu. Existují i DJové, kteří mají své sety do velké míry předem připraveny, většinou je ale

takové chování nahlíženo jako jistý způsob podvodu a běžně se setkává se silně negativními ohlasy.

1.1.3 Fúze hudebních proudů

Jak bylo již dříve naznačeno, rozdělení hudební tvorby na proudy tradiční a elektronický má značné nedostatky. Tyto nedostatky se nejsilněji projevují v případech, kdy se oba světy střetávají, a zejména během posledních deseti až dvaceti let se tak stává stále častěji.

Naprosto zásadní je v tomto směru původně australská kapela Pendulum. Původně duo DJů působící ve vodách undergroundového elektronického žánru drum & bass [1] se později rozšířilo do kapely a pomohlo definovat živá elektronická vystoupení [2]. Pendulum Live i přes svůj několikaletý hiatus dosud zůstávají vůdčím projektem na poli živé elektroniky s plným nástrojovým arzenálem: kytara, baskytara, bicí, vokály a množství syntezátorů. Za manifest živé elektroniky by se dal považovat moment z vystoupení v londýnské Brixton Academy, kde v jeden celek spojili thrash metalovou hymnu Master Of Puppets legendární Metallicity se svou vlastní skladbou Slam – moment, ve kterém se úspěšně střetávají dva světy.

Pendulum nejsou ovšem zdaleka sami, mezi další podobné hudební projekty patří např. The Qemists, Trentemøller nebo The Glitch Mob. Z českých pak zejména electroswingová Mydy Rabycad či Vložte Kočku.

U The Glitch Mob se ještě zastavíme. Nejen, že se jedná o originální hudební projekt, zajímavější je technické řešení celého jejich vystoupení. The Glitch Mob spojili umění a technologii v ještě větším rozsahu, než má za cíl tato práce, nicméně v jiném směru. Všichni tři členové během vystoupení v podstatě hrají dohromady na jeden jediný hudební nástroj pojmenovaný The Blade. Jedná se o systém sestávající z tabletů, MIDI kontrolérů a elektronických bicích, které jsou přímo napojeny na světelný park a zasazeny do kulis vytvořených filmovým studiem. Prostředníkem jsou dva výkonné počítače se softwarovými nástroji jako Ableton Live nebo Max MSP [3]. Jedná se o propracované řešení zmíněného problému asynchronnosti. Výsledkem je nevídaně konzistentní vystoupení, kde každý pohyb muzikantů má význam a dopad na celkovou scénu.

1.1.4 Inspirace

Úspěch The Glitch Mob je důkazem, že snaha o co nejbližší spojení technologií a umění má pro oba obory význam. Tam, kde se The Glitch Mob věnují světlu, má tato práce za cíl se zabývat komplexními vizualizacemi za použití podobných technických postupů (viz dále).

Trend živých elektronických kapel poskytuje právě pro takové spojení vhodné prostředí – fúze dvou proudů najde své místo v obou, přitom v každém bude působit jedinečně. Je možné navíc předpokládat, že se dosavadní

trend bez (zatím) konkrétní formy vyvine v hudební platformu s čistě vlastními událostmi a festivaly. Proto bude i hudba použitá pro testování systému čerpat inspiraci z obou proudů.

1.2 Technické možnosti

1.2.1 Vizualizace v hudebních přehrávačích

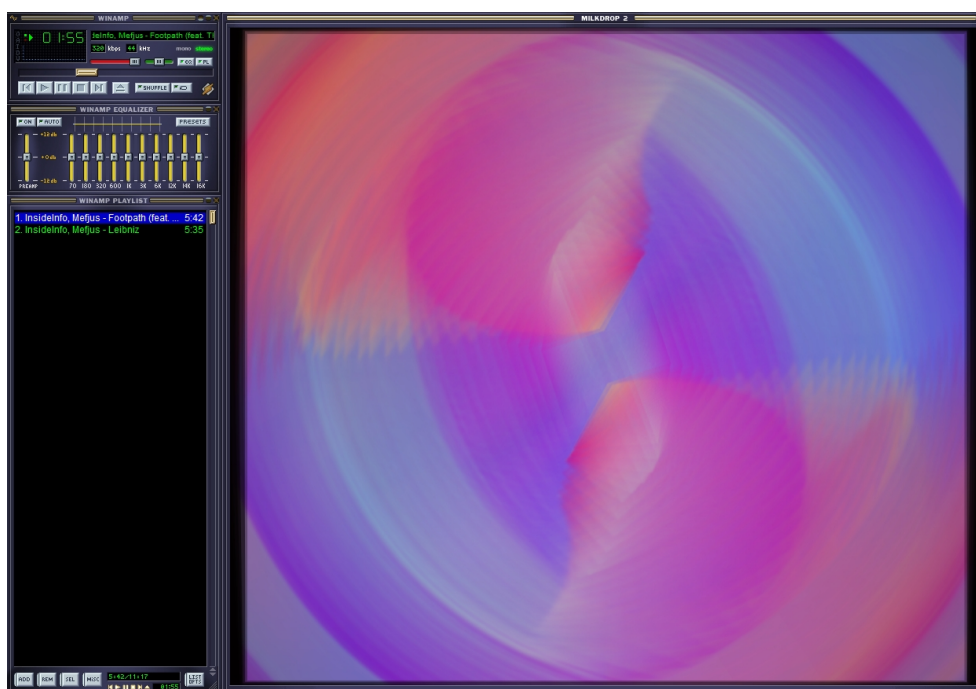
S tou nejvíce primitivní formou hudebních vizualizací se asi setkala většina běžných počítačových uživatelů. Jedná se o vizuální pluginy v hudebních přehrávačích z dob, kdy nebylo standardem hudbu streamovat, ale přehrávat z lokálního úložiště. Takovým přehrávačem byl například Windows Media Player běžně dodávaný se systémem Windows, nebo jeho populární alternativa Winamp firmy Nullsoft.

Vizualizace v obou programech byly v zásadě srovnatelné – abstraktní vzory, fraktály, průlety barevnými tunely apod. (viz obrázek 1.4). Byť vizuálně stimulující, obrazové prvky měly nízkou nebo žádnou souvislost s přehrávanou hudbou a bylo tak snadné v ně brzo ztratit zájem. Jediný typ vizualizací, které s hudbou měly jasnou souvislost, byly varianty grafu spektrální analýzy přehrávaných tónů. Jedná se však o technickou a vizuálně nepřilíš zajímavou záležitost, a informace v grafu obsažené jsou pro běžného uživatele zbytečné a nejasné. Jinak tomu je v případě hudebních producentů, pro něž je spektrální analýza nedocenitelný nástroj – např. typická začátečnická technika je analyzovat oblíbené skladby, sledovat, v jakých frekvencích se pohybují různé prvky, a následně získané znalosti aplikovat ve vlastních projektech.

1.2.2 Světla

Nepostradatelným prvkem při jakémkoli dnešním koncertě je patřičné osvětlení. Světla a svícení je mnoho druhů, koncertní světla původně vyšla ze světla divadelních a některé typy divadelních světla se dodnes používají. Jedná se zejména o světlo parabolické (PAR) a klasický spot light pro osvětlení úzkého kruhu světla. Světla PAR s technologií LED místo lampy a zrcadla sice již neodpovídají svému názvu, nicméně jsou dnes nejlevnější variantou a proto jsou také nejobvyklejší. Při produkci elektronické hudby jsou velice populární lasery, které v kombinaci se strojem na mlhu umí vytvářet abstraktní světelné mapy, dříve se paprsek také nechával odrážet od lesklého zakřiveného povrchu (např. diskokoule). V krátkých záblescích se používají osleповací světla s mnoha výbojkami umístěnými v maticích nebo stroboskopy s jednou výbojkou a zrcadlem. Existují barevné světelné hlavy na kloubech a mnoho variant LED v různých kvalitách, intenzitách a s různými možnostmi.

Tyto světla je možné vzdáleně ovládat pomocí asynchronního sériového protokolu DMX-512, běžně přezdívanému pouze DMX. Používá 512 kanálů, přičemž každý kanál ovládá jeden parametr jednoho světla, RGB PAR si tak



Obrázek 1.4: Snímek programu Winamp se zapnutými vizualizacemi

vezme tři kanály a intenzity jednotlivých barev jsou definovány jednobajtovou hodnotou. Žádané hodnoty se nastaví v kontroléru, který bývá umístěn vedle pozice zvukového technika. Do kontroléru se ještě před vystoupením naprogramují tzv. scény a jejich sekvence, které je pak možné vhodně kombinovat. Pro složitější scény se používají jednak masivní kontroléry, jednak různé splittersy a opakovače, díky kterým lze získat kanálů daleko víc.

1.2.3 VJing a projekce

Nejprogresivnějším typem hudebních vizualizací je v dnešní době VJing v kombinaci s dalšími technologiemi. VJ pomocí dedikovaného softwaru či hardwaru (to je spíše retro přístup) mixuje dohromady dvě nebo více video stop, podobně jako DJ mixuje zvukové stopy. Je možné zvolit různé módy prolínání či video efekty, video je možné pouštět ve smyčkách, jejichž délku lze dynamicky měnit, používají se masky, které zobrazí jen část daného videa a mnoho dalšího.

Nejjednodušší variantou je promítání obrazu na zadní stěnu pódia, to sebou však zejména v menších prostorách nese jisté problémy. Projektor je těžké vhodně umístit, pokud je na pódiu, je příliš blízko, pokud je až u stánku zvukového technika, je příliš daleko. Pokud je přibližně uprostřed a v klubu je nízký strop, lidé vzadu přes projektor nevidí dopředu a lidé vepředu překáží ve světle. Ať je kdekoli, tak navíc ještě osvětluje samotné hudebníky, což na-

konec působí celkem zajímavě a na plátno za hudebníky se promítanou jejich stíny. Dalším problémem je, že v cestě světla krom lidí stojí ještě technika, většinou světla.

Z těchto důvodů pražské hudební kluby Storm a Roxy v současné době na zadní stěnu pódia nepromítají. Místo jednoho projektoru používají dva nebo více a promítají na boční stěny. Největší výhodou je decentralizace projekcí, takže pokud jedné projekci v cestě překáží procházející návštěvníci, je možné sledovat druhou.

Další variantou je nepoužívat projekce, ale obrazovky. V současné době se jedná zejména o LED matice s různou hustotou, pro větší, zejména open air pódia se používá více matic a hustota nižší, pro pódia menší naopak. Ještě před několika lety byla hustota LED běžně tak nízká, že i ze vzdálenosti několika desítek metrů bylo lze rozeznat jednotlivé obrazové body. Rapidní vývoj a snižující se cena umožnily použití matic daleko hustších, nestačily ale zabránit proniknutí dalšího fenoménu do světa hudebních vizualizací.

Jedná se o video mapping. V Česku je nejnámější díky světelnému festivalu Signal, kde se běžně mapuje na budovy. Byť hudební kluby často mívají značně složité dispozice plné zákoutí, chodeb a kleneb, takže by interiér klubu byl vhodný pro mapping podobně jako exteriéry historických budov, stísněné prostory naplněné návštěvníky neposkytují ani zdaleka dostatek místa pro projekci, které by obsáhla větší část interiéru. Proto se běžně mapuje na tvary (např. logo pořádané akce), 3D objekty jako různě natočené kostky, organicky prověšená plátna, jednoduše zavěšené poloprůhledné materiály v řadě za sebou nebo dokonce zvukové soustavy samotné.. Jelikož tyto povrchy většinou nenabízí žádné konkrétní zachytné body jako fasády budov, mapping je omezen na pouhé vymaskování a warping obrazu (deformace obrazu původně čtvercového či obdélníkového tvaru, typicky použitá pro dorovnání perspektivy), samotné vizualizace jsou pak opět v rukou VJe.

Mapping a VJing k sobě mají blízko i použitým softwarem, například Resolume Arena umožňuje mixování videa, ale právě i warping obrazu charakteristický pro mapping. Ne všechny softwarové nástroje umožňují obojí, pak je nutné je kombinovat.

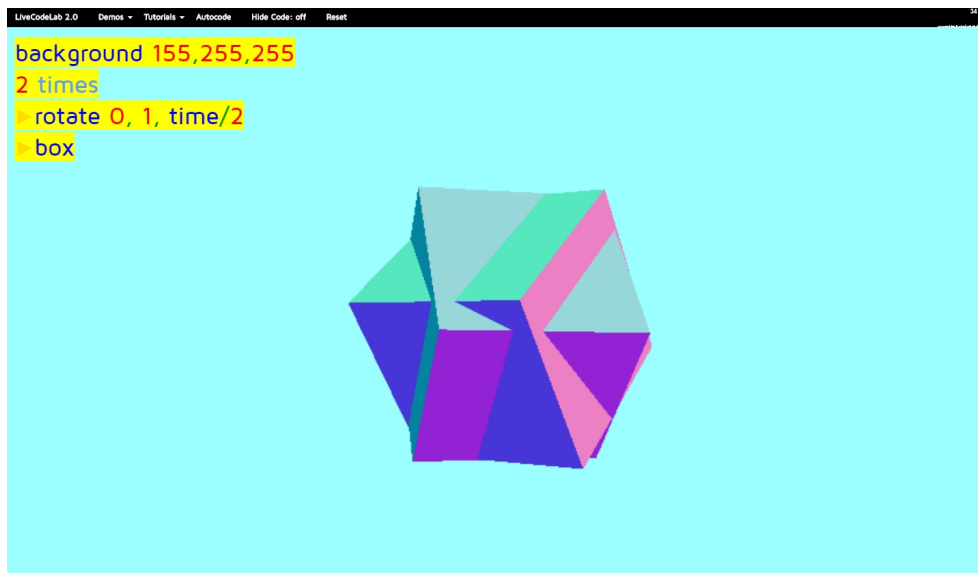
1.2.4 Ostatní

Živé kódování Na elektronické scéně Algorave se místo VJingu běžně používá živé kódování. Jedná se o programování v reálném čase za použití rozhraní umožňujících rychlé a jednoduché vykreslování tvarů a manipulaci s obrazem. Tento obraz je pak často poslán opět do softwaru určeného primárně pro VJing a posléze na plátna nebo obrazovky.

Virtuální realita, 360° Novinkou na poli hudebních vizualizací je použití vybavení pro virtuální realitu. Spoluprací hudebních vydavatelství Bad Taste a Eat Brain a umělecké agentury MethLab vznikl projekt Terminal – první



Obrázek 1.5: Kombinace mappingu a VJingu během vystoupení Changing Faces v pražském klubu Cross. V pozadí svítící zadní stěna pódia [4].



Obrázek 1.6: Snímek demonstračního programu prostředí Live Code Lab

kanál na Youtube poskytující hudební vizualizace ve 3D pro virtuální realitu. Všechny tři společnosti se zaměřují zejména na elektronický žánr neurofunk charakteristický pro svůj technický a temný zvuk, proto i vizuální stránka působí industriálně a kontrastně. Jediným prvkem je zde sám pozorovatel – sám si vybírá, kterou část imerzivního prostředí bude sledovat a jakým způsobem, v míře nesrovnatelné s ostatními přístupy (např. stroboskopy a oslepuvací světla snadno prosvítí i zavřená oční víčka, světlům tedy v podstatě není úniku).

Hudební videa Poslední variantou hudebních vizualizací jsou hudební videa. Jedná se však o předem nastříhaný materiál a míra závislosti na hudbě je zcela v rukou filmařů. Jde tak o poněkud jinou disciplínu a dále se tato práce bude zabývat pouze vizualizacemi generovanými v reálném čase.

1.2.5 Shrnutí

Většina uvedených technických řešení sdílí problém asynchronnosti obrazu a zvuku. Jediný pokus o technické spojení byl v případě vizualizací v hudebních přehrávačích, jako naprosto vedlejší funkcionalita ale nikdy nebyla dotažena do esteticky přijatelného stavu. Ostatní řešení spoléhají na profesionalitu techniků a v případě špičkových hudebních projektů na jejich sešranost s muzikanty. Jak již bylo řečeno, zde lidský faktor působí spíše potíže, kreativní prvky (světelné scény, předtočená videa) jsou beztak připraveny dávno před samotným vystoupením. Tomuto problému se chce tato práce pomocí použitých technologií zcela vyhnout.

1.3 Možnosti synchronizace

1.3.1 Ovládání technikem

Již zmíněným přístupem je správa synchronizací příslušným technikem, ať už světelným, VJem, či např. kodérem. Jedná se o přístup v dnešní době naprosto dominantní, přístupy dále popsané se objevují jen zřídka nebo vůbec. I přes zmíněný problém asynchronnosti dosahuje tento přístup úctyhodných výsledků a své místo na scéně si určitě zaslouží, bylo by ale nerozumné nezkoušet přístupy komplexnější.

1.3.2 Analýza zvukového signálu

Pokud mají vizualizace souviset s hudbou, je přirozené je na hudbu navázat přímo. Na hudbu se fyzika dívá jako na souhrn periodických funkcí v čase, je na ni proto možné aplikovat různé matematické aparáty. Jedním takovým je například spektrální analýza, často implementována pomocí Rychlé Fourierovy transformace (FFT). Pomocí FFT je možné akord rozložit na zastoupení



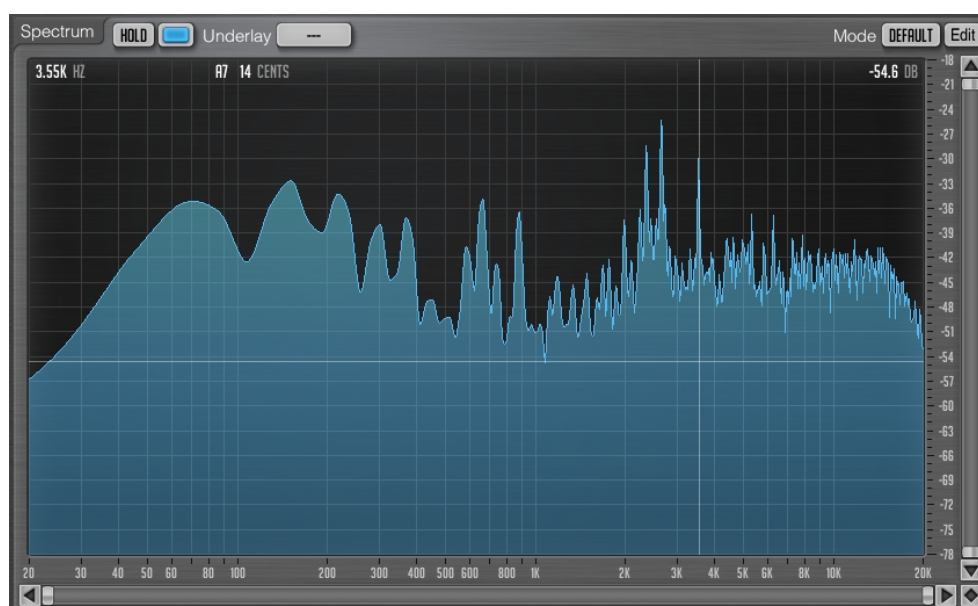
Obrázek 1.7: Snímek grafu spektrální analýzy skladby Etherwood – Light My Way Home [5]. Zaměřený peak odpovídá vokálu, stejně jako dva výraznější peaky napravo od něj. Dále jsou ve skladbě výrazně zastoupeny elektronické bicí a hutné basy charakteristické pro elektronický žánr drum & bass.

předem daných frekvencí a pokud tyto frekvence zvolíme vhodně, na jednotlivé tóny.

Zastoupení tónů ovšem ani zdaleka neposkytuje kompletní informace potřebné pro správné pochopení hudby, data je zapotřebí ještě vhodně interpretovat, což přináší další komplikace. Pro jednoduchost nyní uvažujme analýzu pouze jednoho nástroje. Dalo by se předpokládat, že jediný tón zahráný na analyzovaný nástroj vyvolá jediný peak. Opak je však pravdou – k zahránému tónu se přiřadí ještě tóny alikvótní, jež jsou jeho celočíselnými násobky. Počet a intenzita alikvótních tónů záleží zcela na nástroji, a nejen na jeho typu, ale i na materiálu a zpracování. Alikvótní tóny jsou (alespoň u tradičních nástrojů) hlavním důvodem rozdílné barvy zvuku rozdílných nástrojů. Tón zahráný na cedrovou akustickou kytaru se zobrazí odlišně od stejného tónu zahráného na mahagonovou elektrickou – a to ani nebereme v potaz použité snímače. Správně identifikovat jeden tón a přiřadit ho k danému nástroji může být proto náročné, záleží na zvolených nástrojích.

Je tu však ještě větší problém – během hudební produkce se téměř výhradně kombinuje více hudebních nástrojů. Rozsah lidského sluchu je přibližně 20 Hz – 20 KHz, přičemž dolní hranice odpovídá vlnám, které spíše fyzicky cítíme jako vibrace než slyšíme jako zvuk, a horní hranice odpovídá téměř neslyšitelnému pískání. V tomto rozsahu se pohybují veškeré nástroje a mnoho

1. ANALÝZA



Obrázek 1.8: Snímek grafu spektrální analýzy skladby Toundra – Danubio [6]. Zaměřený peak a dva podobné nalevo od něj odpovídají sólové kytáře. Ve skladbě jsou dále přítomné další kytara, baskytara a bicí.

z nich ho je schopno zastoupit celkem velkou část, což sebou nese i fakt, že se nástroje ve svém frekvenčním zastoupení značně překrývají. Např. baskytara a basový buben (a vlastně všechny basové nástroje), kytara s pianem, trubka s činely apod. Peak v dané frekvenci tedy může odpovídat klidně i všem nástrojům zastoupeným v hudebním uskupení a jeho bližší určení vyžaduje analýzu alikvótních tónů – která je problematická, krom důvodů uvedených výše i proto, že opět mohou příslušet nástroji zcela jinému.

Dalším problémem je samotný sběr dat. Koncerty jsou hlučná prostředí a úkol zvukového technika je náročný. Není těžké celou atmosféru zničit špatným nazvučením, což je situace, která běžně nastává (ať už se jedná o nekvalitní techniku, nedostatečně kvalifikovaného technika nebo jiné příčiny). I při nahrávání ve studiu klade vhodné zpracování zvuku na technika mnoho požadavků a ten je nucen parametry během vystoupení dynamicky měnit. Chybovost a šum jsou další faktory, které interpretaci spektrální analýzy činí složitou.

Tato data by použitím dnešních technologií jistě šla zpracovat, nejspíš i s přijatelnými výsledky. Zcela určitě by ale dané zpracování přesáhlo rozsah této práce a na samotný cíl – vizualizace – by nezbylo dostatek prostředků.

1.3.3 CV/Gate

Existuje varianta někde na pomezí mezi variantou předchozí a následující – pracuje s analogovým signálem a zároveň připomíná variantu MIDI. Jedná se o CV/Gate signály používané v modulárních syntezátorech. Modulární syntezátory jsou syntezátory nejstaršího typu, postupně je nahrazovaly syntezátory digitální, hybridní a v dnešní době virtuální VST pluginy. Modulární syntezátory však nikdy nebyly z trhu zcela eliminovány, jednak díky robustnosti svého zpracování a celkové kvalitě jak mechanické, tak zvukové, jednak kvůli svobodě projevu a specifickému workflow, které nabízejí. Modulární syntezátor sestává z několika modulů, triviálním příkladem budiž oscilátor a low pass filtr (filtr, který podle daného parametru ze signálu odstraní vyšší frekvence). Pro komunikaci modulů se používá právě CV/Gate, kontrolní napětí pro výšku tónu a gate pro rytmiku. Jedná se tak v podstatě o předchůdce MIDI.

1.3.4 MIDI

MIDI je rozhraní používané pro komunikaci elektronických hudebních nástrojů. Od svého uvedení v osmdesátých letech zaujalo naprosto dominantní postavení a je k nalezení v mnoha profesionálních hudebních aplikacích. Komunikace probíhá pomocí digitálních zpráv, zpráva sestává ze dvou nebo více bajtů podle typu zprávy. První bajt označuje status neboli typ zprávy, např. zapnutí nebo vypnutí noty na daném kanálu nebo synchronizační tik. Další bajty jsou datové, v případě zapnutí noty jeden bajt označuje o kterou notu se jedná a druhý jakou silou byla aktivována (velocity). MIDI zprávami lze dále oznámit změnu skladby, změnu programu, postupné uvolnění klávesy apod. Je také možné specifikovat MIDI zprávy exkluzivně pro daný hardware, které bude cizí hardware ignorovat.

MIDI nástroje jsou typicky sériově zřetězené pomocí pětipinových MIDI kabelů (poslední dobou má navíc mnoho MIDI nástrojů USB výstup). Na počátku bývá master keyboard nebo sekvencer, který sám zvuk negeneruje, ale posílá MIDI zprávy dál. Každý nástroj v řetězu má přiřazen jeden kanál, na kterém zprávy přijímá, a zbylé nebo všechny zprávy posílá dál, záleží na implementaci. Nástroje přijímající zprávy jsou již zdrojem tónů, jedná se o syntezátory, které zvuk přímo generují (ať už analogovým, digitálním, nebo hybridním způsobem), nebo samplery, které z úložiště (v dnešní době se typicky jedná o SD kartu) přehrávají předem připravené zvukové samplery, případně je dodatečně modifikují. Předem naprogramovaný sekvencer umožňuje hudebníkovi soustředit se na parametry a barvu zvuku, čímž vzniká estetika opakujících se motivů, které se v dalších iteracích postupně rozvíjí – estetika charakteristická pro techno a technem inspirované žánry, což zahrnuje značnou část elektronické hudby. Je patrné, jak MIDI workflow ovlivnil hudební vývoj.

Existují pokusy extrahovat MIDI zprávy z tradičních hudebních nástrojů, například z kytar. Jedná se o přídavné snímače, které monitorují jednotlivé struny a s různou mírou úspěšnosti detekují přehraný tón. Typické problémy jsou nepřesnosti v detekci tónu a snímání ruchů, které tóny vůbec nejsou. Tyto snímače nejsou příliš rozšířeny, za zmínku stojí jen snímače a převaděče od firmy Roland, v kontextu rozpočtu na testování této práce se však jedná o příliš nákladná zařízení. Další variantou jsou MIDI kytary od firmy Starr Labs, mající na hmatníku tlačítka schopná detekovat i jemnější techniky hry, jsou však ještě nákladnější. O něco lepší situace je u tzv. dechových syntezátorů, jež už několik generací vyprodukovali různí zavedení výrobci.

MIDI se jeví jako nejlepší, nebo aspoň nejjednodušší způsob, jak získat přesné informace o hudbě, přestože zdaleka nepokrývá celou šíři hudebních nástrojů. Několik aplikací schopných propojit MIDI a hudební vizualizace už tedy existuje.

OpenTZT Opensourcový VJ nástroj pro platformu Windows postavený na nyní již opuštěném VJ nástroji TransZendent a platformě DirectX. OpenTZT je ovládaný primárně přes klávesnici a funguje podobně jako hudební nástroj – uživatel na klávesnici „hraje“ a tím vytváří vizualizace. Podobným způsobem je možné nástroj ovládat pomocí MIDI, tato kapitola dokumentace ale nebyla dokončena a od roku 2013 nebyl nástroj aktualizován. Od roku 2010 dosáhl přes 20000 stažení [7] a soudě podle této statistiky, jeho popularita výrazně neklesla až do května 2015.

MIDI2OpenGL Vedlejší projekt sanfranciského hudebníka s pseudonymem Chance’s End. Podle stručné dokumentace se jedná o jednoduchou aplikaci pro konkrétní MIDI kontrolér, ve které měly prostorem létat koule podle hraných tónů. Stránka s dokumentací je špatně naformátovaná a plná mrtvých odkazů, projekt byl tedy pravděpodobně brzy opuštěn.

MIDI.live Projekt z Vienna University of Technology mající za cíl demonstrovat použitelnost MIDI zpráv pro hudební vizualizace. Využívá knihovnu OpenGL a programovací jazyk Java. Primárním prvkem v architektuře projektu byla rozšířitelnost pomocí pluginů, dva ukázkové pluginy byly zahrnuty v beta verzi. Původní webové stránky projektu nyní již nejsou dostupné, spustitelná verze a dokument popisující vlastnosti a vývoj projektu byl dostupný na osobní stránce jednoho z autorů. Během psaní této práce byla ale stránka v procesu rekonstrukce a není jisté, zdali budou spustitelná verze nebo zdrojové kódy dostupné později. Jak se uvádí v dokumentu, bez rekompilace není v poslední verzi aplikace možno přidat plugin, přímé navázání na tento projekt tedy nebylo v této práci možné.

Gameboy MIDI Chrome visuals Projekt vzniklý v době psaní této práce, jehož autorem je elektronický muzikant Or4. V principu se jedná o aplikaci podobného charakteru jako v této práci, s hlavním rozdílem v použitých technologiích – je určena pro webové prohlížeče. Po vytvoření demoverze byl vývoj pozastaven. Budoucí spolupráce je možná.

The Blade Jak již bylo dříve zmíněno, elektronická skupina The Glitch Mob používá nástroj vlastní produkce The Blade, který komunikuje právě pomocí MIDI. Krom krátkého dokumentárního filmu [3] nebyly zveřejněny podrobnější informace.

1.4 Shrnutí

Je patrné, že myšlenka použití MIDI pro hudební vizualizace je atraktivní. Většina zmíněných projektů sice byla opuštěna, zato ten, který byl dotažen až do konce, slaví notné úspěchy. Použití MIDI není pochopitelně jediná příčina, zjevně ale splňuje požadavky na jednoduchost, efektivitu a spolehlivost a je tedy vhodným prostředkem pro komunikaci mezi hudební a vizuální částí implementovaného systému. Co se týče vizuální stránky, měla by se odprostit od generických abstraktních vzorů či světelných tunelů a použít jasný vizuální jazyk, kde konkrétní prvky a jejich animace budou zřetelně odpovídat hudbě.

Návrh

2.1 Obecný návrh

Primárním vstupem bude datový proud formátu MIDI a primárním výstupem budou obrazová data vhodná pro projekci. Obrazová data budou přímo odpovídat zvoleným parametrům zvolených MIDI zpráv, aby byla čitelná estetická souvislost mezi zvukem a obrazem. Různé MIDI zprávy budou ovlivňovat různé vizuální prvky. Vizuální prvky se budou skládat z objektů ve scéně a efektů na objekty nebo na celou scénu aplikovaných. Vizuální prvky se budou kombinovat – budou zobrazeny vedle sebe i přes sebe, aby vznikla komplexní proměnlivá scéna vhodně reprezentující hudební skladbu, kterou bude doprovázet. Během průběhu skladby se mohou jednotlivé prvky i celá scéna vyvíjet, závěrečná scéna tedy může být zcela odlišná od scény úvodní.

Pro co největší svobodu výtvarného projevu bude vizuální jádro systému implementováno s pomocí knihovny OpenGL. Svoboda je sice vykoupena náročnější implementací, ze solidního technického základu ale v budoucnu vyplyne výhodnější pozice pro další rozšiřování.

2.2 Technické požadavky

Systém bude splňovat následující technické požadavky:

Výkon Systém využije vhodné algoritmy, postupy a knihovny odpovídající verze, aby bezproblémově fungoval na testovacím zařízení.

Správa prostředků Systém bude vhodně spravovat prostředky a v rozumné míře minimalizovat požadavky na paměť.

2. NÁVRH

Komunikace Systém bude informovat o svém stavu, proběhlých událostech či případných chybách. Podle potřeby bude komunikace obousměrná. Komunikace nebude rušit vizualizace.

Konfigurovatelnost Systém bude možno manuálně konfigurovat pro podmínky, ve kterých je spuštěn. Systém bude podporovat běžné i širokoúhlé zobrazení.

Stabilita Systém bude dostatečně stabilní pro použití během živých vystoupení.

Platforma Systém bude vyvinut pro platformu Windows, aby jej bylo možné testovat i bez hardwarových zařízení pomocí běžně užívaných DAW.

Struktura Zdrojový kód bude mít rozšiřitelnou strukturu a bude přehledný, ne však na úkor výkonu.

Testování Systém bude testován konkrétními hudebními skladbami. Systém bude schopný vhodně reagovat i na chyby, nepřesnosti či hudební improvizaci během vystoupení.

Workflow Systém bude zapadat do níže popsaného tvůrčího workflow.

2.3 Workflow

Systém bude testovaný za tím účelem vytvořeným hudebním projektem. Tento projekt si vezme inspiraci z hudebního prostředí popsaného v kapitole 1 a bude kombinovat prvky tradiční a elektronické hudby. Každý proud bude reprezentován jedním hudebníkem, pro podrobnosti viz kapitolu 4.

Pro živou elektronickou hudbu (a obecně pro hudebníky s „organickým“ přístupem) jsou charakteristické výrazné odlišnosti mezi verzí živě prezentované skladby a studiovou nahrávkou. Vlastnostmi všech hudebních vystoupení jsou obecně nižší zvuková kvalita než v nahrávacím studiu a nepřesnosti v přehrávání – běžným jevem zejména u nepřiliš vyhraných kapel je neschopnost udržet tempo, typicky zrychlování. V živé elektronické hudbě se dále odstraňují prvky, které by v hlučném prostředí dostatečně nevyzněly, zvuky a prvky, které se ve skladbách opakují jen málokdy se omezují a generalizují se nastavení syntezátorů. Důvodem je zejména minimalizace režie nutné pro přepínání mezi jednotlivými skladbami. Tento jev bude patrný i v implementovaném systému, vizualizace budou vhodně sdílet znovupoužitelné prvky a tyto budou vhodně uloženy v paměti, aby se mezi jednotlivými skladbami nemusely znovu

inicializovat. Dalším důvodem pro generalizaci skladeb je minimalizace potřebného vybavení a tím snížení pořizovacích nákladů i nákladů na přepravu. Tento vliv bude také během testování systému přítomen.

Posledním rozdílem mezi živými a studiovými verzemi skladeb je jejich vývoj. Tím se konečně dostáváme ke tvůrčímu workflow, ve kterém bude systém figurovat. Testovací skladby jsou nejprve zhruba navrženy: je načrtnuta základní myšlenka či abstraktní poselství, které se krom názvu skladby a vokálních samplů či zpívaných partů promítne i do celkové atmosféry. Dále žánrový rámec, případně syntéza vícero žánrových vlivů, a několik beatů (rytmických smyček), jejichž repetice a variace určí základní strukturu. V tuto chvíli je připravena kostra, kterou je nutno obalit a vyplnit hudebním materiálem. Dále se připraví syntezátorové smyčky nebo se alespoň navrhne improvizací rámec. První demoverze se nahraje a pošle dalším členům, aby ji opřipomínkovali a připravili si návrhy svých partů. V tuto chvíli jsou také navrženy základní stavební kameny vizualizací, typicky ve formě jednoduchých náčrtů na papíře.

Dále dojde ke střetu ideí na hudební zkoušce. Již během prvních několika přehrání vyplynou na povrch hudební konflikty a estetické nesmysly. Struktura je více či méně přeskládána, hudební party přeladěny, rytmicky upraveny, nahrazeny či úplně odstraněny. Během iterativního procesu na hudební zkoušce se skladba postupně přibližuje veřejně prezentovatelné formě. Také proběhne diskuze o vizualizacích.

Mezi zkouškami je čas získat odstup a nad skladbami přemýšlet. Je možné samostatně improvizovat a skladbu mírně upravovat, také je možné začít s implementací vizualizací. Na další zkoušce se mezitím nashromážděné úvahy opět střetnou a iterativním procesem se skladba posune dál. V závislosti na komunikačních schopnostech a vzájemné sešranosti hudebníků je po několika zkouškách připravena první veřejně prezentovatelná verze skladby. V tu chvíli by měly všechny zásadní hudební prvky mít svou reprezentaci ve vizualizacích.

Během prvního živého vystoupení je skladba podrobena prvnímu ostrému testu. Typickými výstupy jsou celkový úspěch skladby u publika (který je nutno brát s rezervou, publikum může být zmateno, překvapeno, nebo třeba jen může nedávat pozor, navíc si skladbu beztak pamatovat nebude, takže konkrétní zpětná vazba po koncertě není možná) a zkušenost muzikantů s hraním svých partů pod stresem. Po koncertě se jedná opět o iterativní proces, mnohdy zdlouhavý. Skladba se nemění tolik během jednotlivých zkoušek, spíše mezi vystoupeními. V různých podmínkách (zvukových, světelných, nebo v kontextu dalších hudebních projektů) je možné ověřit funkci všech prvků z různých pohledů. Nefungující detaily (přechody mezi beaty, technika hry konkrétních partů) jsou odstraněny a postupně nahrazovány takovými, které v sobě zahrnují předchozí zkušenost. Stejně tak se vyvíjí i vizualizace, prvky jsou obměňovány a variovány.

Další fází je finalizace skladby pro studiovou nahrávku. Během zkoušek se klade větší důraz na detail, až se nakonec dojde verze vhodné pro vydání. Pokud projekt funguje v rámci vydavatelství nebo na nahrávce spolupracuje

s hudebním producentem, ti do finalizace také vstupují, typicky s požadavkem na pochopitelnost hudebního projevu cílovou skupinou – publikem.

Po vydání nahrávka často „zamrzne“ a dále se již nevyvíjí, záleží ovšem na přístupu hudebníků. Existují takoví, kteří se ke starým skladbám vrací a omlazují je, občas k nelibosti publika, které si zapamatuje verzi vydanou a novější verzi se cítí zmateno. Nejběžnější formou úpravy již vydaných skladeb jsou v elektronické hudbě takzvané remixy a VIP (nikoli very important person, ale variation in production). Remix je zpracování originálních partů jiným producentem, často pohybujičím se v jiném žánru, skladba tak získá možnost prosadit se v širším žánrovém okruhu. Druhou formou je VIP, remix od původního autora. Ten většinou přichází přibližně rok po vydání skladby, zahrnuje osvěžení zvuku, reformulaci některých motivů a je typicky mířený ne na poslech, ale přímo na taneční parkety. VIP často nebývají oficiálně vydány a DJové je mezi sebou sdílí na základně přátelských vztahů.

Pro úplnost nutno poznamenat, že popsané workflow není jedinečné, dalo by se skoro říci, že každý hudební projekt má workflow vlastní. Krom patologických případů (studiové projekty, které nikdy nekonzertují, jednočlenné projekty. . .) budou ale různá workflow v zásadě podobná – budou sdílet alespoň některé prvky. Např. elektronická hudba je často produkována jediným autorem, budou tak zcela vynechány hudební zkoušky, a komunikaci s členy kapely nahradí komunikace s nejbližšími dalšími umělci, typicky v rámci hudebního vydavatelství. Ti budou také nedokončenou skladbu živě testovat, více méně nezávisle na autorovi.

2.4 Technické náležitosti

Běh systému Po startu systému proběhne počáteční konfigurace. Ta bude zahrnovat volbu MIDI portu a monitoru pro vizualizace. Tyto budou požadovat uživatelský vstup. Poté budou inicializovány potřebné knihovny. Pokud bude konfigurace nebo inicializace knihovny neúspěšná, systém nahlásí chybu a ukončí svou činnost. V opačném případě budou načteny sdílené prostředky (shadery, modely, textury. . .) a posléze prostředky užité první písní. Pokud nastane chyba, systém ji ohlásí a nastaví příslušné indikátory (např. model nebyl úspěšně načten a proto se nebude vykreslovat). V opačném případě bude systém zcela připraven interpretovat vstupní proud formátu MIDI. V momentu přepnutí skladby na další se uvolní prostředky ukončené skladby a načtou prostředky skladby následující. Přepínání skladeb bude cyklické, tzn. z poslední skladby povede přepnutí na skladbu první. Pokud přijme systém během závěrečné skladby MIDI zprávu oznamující ukončení vizualizací, zobrazí odpovídající výstup, dále však svůj stav nemění – MIDI zpráva mohla být způsobena chybou hudebníka a je na něm, aby ji stornoval. Pokud se skutečně jedná o závěr vystoupení, systém vyčkává, dokud nebude manuálně ukončen, kdy uvolní všechny alokované prostředky a ukončí svou činnost.

Programovací koncepty Systém využije vhodných programovacích konceptů. Mezi hlavní se bude řadit objektové programování, dědičnost a polymorfismus. Dále abstrakce nad rutinními činnostmi jako načítání modelů či textur, toto bude přenecháno externím knihovnám.

Množina vnitřních stavů Systém bude mít množinu vnitřních stavů, která bude ovlivňovat chování a výstup systému. Po úvodní konfiguraci bude stav systému měněn výhradně zprávami MIDI s výjimkou manuálního přepnutí na další skladbu a ukončení běhu systému. Celkový stav systému bude kombinací pravdivostních hodnot indikátorů zahrnujících konfiguraci systému (např. širokoúhlé zobrazení nebo režim celé obrazovky (fullscreen)), stisknuté klávesy, a aktivované vizuální prvky (např. zelená kostka vlevo nahoře má rotovat či zobrazení barevných doplňků).

Uživatelské rozhraní Uživatelské rozhraní bude rozděleno do dvou oken. V hlavním okně budou probíhat ničím nerušené vizualizace. V případě dvou či více monitorů uživatel během počáteční konfigurace jeden monitor zvolí a v něm se vytvoří okno na celou obrazovku, ve kterém budou probíhat vizualizace. Druhé okno bude kontrolní a bude se jednat o konzoli. Komunikace během konfigurace bude probíhat v konzoli, další komunikace (přepnutí skladby, ukončení běhu systému) bude probíhat pomocí klávesových zkratk v okně vizualizací.

Implementace

3.1 Architektura

Základem architektury systému je komunikace vstupu a výstupu. Vstupem je proud MIDI, jehož zdrojem jsou hardwarové zařízení ovládaná hudebníkem. Výstupem jsou obrazová data zobrazená datovým projektorem na plátno.

Architektura je rozdělena na dvě vrstvy, a to vrstvu datovou a vrstvu prezentačně-kontrolní. Datová vrstva zahrnuje příjem a základní zpracování MIDI zpráv a načítání modelů, textur, shaderů a dalších prostředků. Datová vrstva je z velké části realizována pomocí přídatných knihoven. Prezentačně-kontrolní vrstva spravuje tyto prostředky, reaguje na vstupní data, v různých místech je vyhodnocuje a komunikuje s knihovnou OpenGL, která zajišťuje přípravu samotných obrazových dat na GPU.

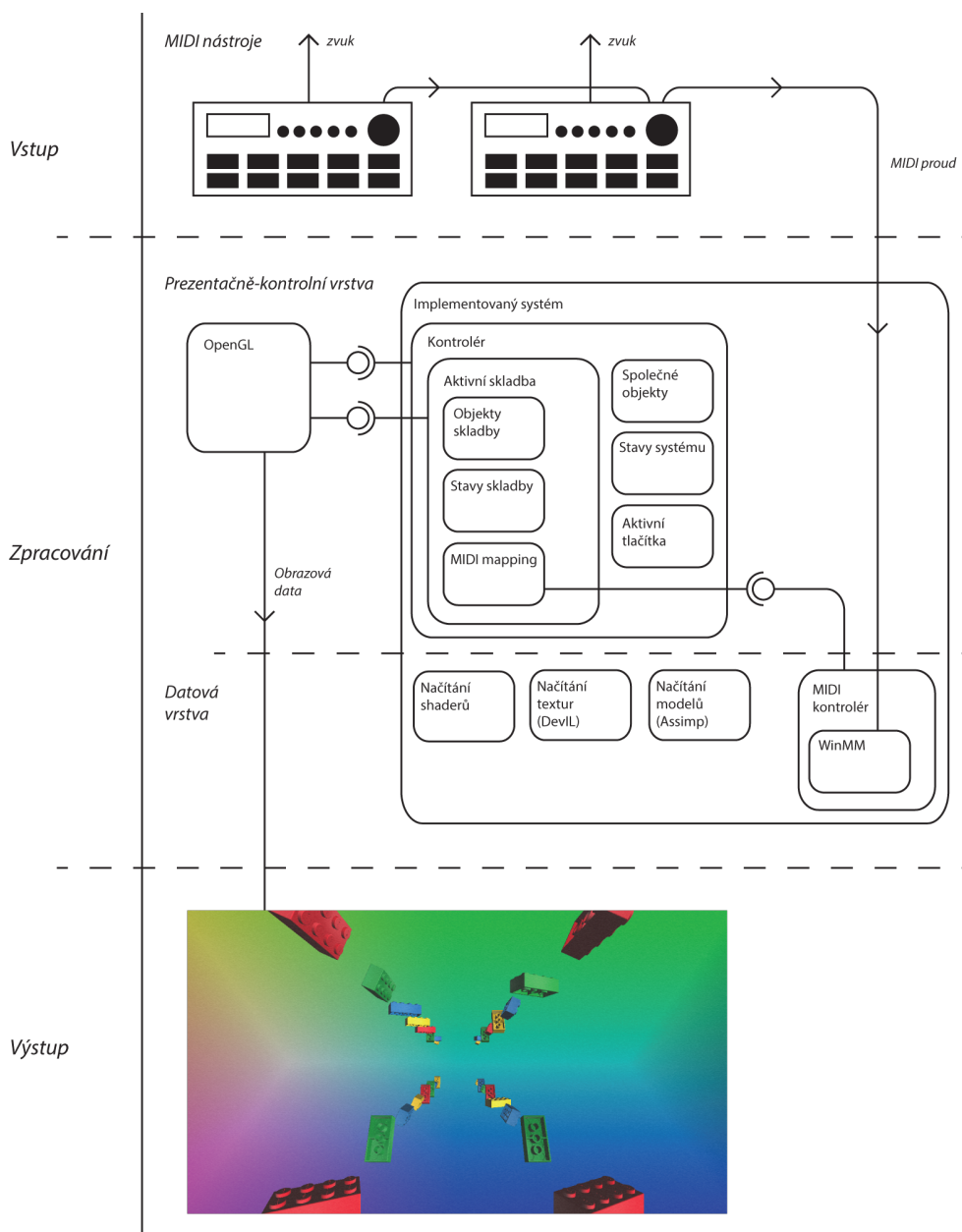
Pro komunikaci s MIDI zařízeními byla zvolena knihovna WinMM. Ta je v systému obalena objektem MIDI kontrolér, který zajišťuje inicializaci, konfiguraci a samotnou MIDI komunikaci. MIDI komunikace je realizována pomocí zpětného volání při přijetí MIDI zprávy. Tato zpráva je MIDI kontrolérem předzpracována, což zahrnuje vyhodnocení typu zprávy a případné předání zprávy hlavnímu objektu Kontrolér.

Kontrolér je objekt zajišťující hlavní funkcionalitu systému. Kontrolér spravuje informace o hlavních stavech systému, např. typ zobrazení (běžné či širokoúhlé), stav inicializace či informaci o načtení skladby. Dále reaguje na vstup z klávesnice a pomocí mapy kláves realizuje odezvu. V Kontroléru jsou přítomny dva typy objektů: jedním je Aktivní skladba a dalším jsou objekty společné pro všechny skladby.

Jednotlivé skladby evidují svůj vnitřní stav, jedná se zejména o aktivované vizuální prvky. Obsahují objekty ve scéně, pomocí kterých jsou tyto vizuální prvky realizovány. Významnou součástí skladby je také mapování MIDI zpráv na vizuální prvky.

Podle přichozích MIDI zpráv a vnitřních stavů připravují objekty Kontrolér a Aktivní skladba podklady a volání pro knihovnu OpenGL.

3. IMPLEMENTACE



Obrázek 3.1: Architektura systému se zobrazenou primární komunikací

3.2 Použité technologie

MIDI Pro komunikaci s hudebníky byl zvolen standard MIDI. Jeho hlavní výhodou je masové nasazení, je implementován téměř ve všech elektronických hudebních nástrojích vyrobených během posledních třiceti let. Způsob implementace se zejména u levnějších zařízení od standardu občas odchyľuje, např. místo běžných samostatných konektorů pro vstup a výstup (případně třetí konektor MIDI Through, který pošle na výstup i zprávy, které byly určeny pro dané zařízení a to je přijalo) je přítomen pouze konektor pro vstup apod. Některá zařízení pak mají implementovanou funkcionalitu nad rámec standardu (např. specifické zprávy pro zálohu uložených dat) a zejména větší a dražší zařízení zahrnují rovnou několik párů/trojic pro vstup a výstup.

Hlavní nevýhodou standardu MIDI je jeho zastaralost. V roce 2002 byl proto uveden standard OSC, který využívá síťovou architekturu klient-server, komunikaci pomocí paketů a 32bitové proměnné, jeho nasazení je však oproti MIDI naprosto minimální.

WinMM API operačního systému Windows pro práci s multimediálním materiálem. Součástí jsou i nízkoúrovňové funkce a struktury pro práci s MIDI. Nad MIDI funkcionalitou WinMM existuje několik veřejně dostupných abstrakcí, kvůli konfliktům s dalšími knihovnami byly ale během vývoje odstraněny a MIDI funkcionalita WinMM byla obalena vlastním objektem. WinMM je jediná z použitých technologií specifická pro jeden operační systém, v budoucím vývoji je tedy možné její vyřazení a nahrazení.

C++ Systém je implementován v programovacím jazyce C++. Zvažovanou variantou byl Processing postavený na programovacím jazyku Java, nakonec byla ale zvolena kombinace OpenGL a C++, ve kterém se pro OpenGL nejčastěji vyvíjí.

OpenGL Knihovna umožňující vykreslování 3D modelů na GPU. Přímou alternativou je proprietární DirectX společnosti Microsoft, zejména během svého raného vývoje často kritizovaná. OpenGL svou otevřeností však více odpovídá charakteru této práce i fakulty, na které je publikovaná.

GLEW Asi nejběžnější knihovna používaná spolu s OpenGL. Umožňuje jednak načtení samotné grafické knihovny, jednak načtení případných rozšíření.

GLFW Framework pro práci s OpenGL. Umožňuje zejména tvorbu grafických kontextů či oken a množství různorodých zpětných volání – změna rozměrů okna, stisknuté tlačítko, pohyb myši. . .

GLM Knihovna matematických funkcí pro C++, která svou syntaxí odpovídá GLSL – jazyku, ve kterém se v OpenGL zpracovává obraz samotný. Zahrnuje zejména struktury pro dvou- až čtyřsložkové vektory, matice a funkce, které tyto snadno vygenerují, např. funkci pro matici rotace s parametry úhel a osa rotace.

DevIL Knihovna pro práci s obrazovými daty. Je schopná zpracovat velké množství obrazových formátů, přičemž práce s knihovnou je pořád jednoduchá. Knihovna DevIL je rozdělena do několika modulů, pro účely toho systému byl použit pouze modul první, zajišťující základní funkce pro nakládání s obrazovými daty, tedy jejich načítání, ukládání a konvertování. Další moduly jsou pro manipulaci a zobrazení obrazových dat.

Assimp Knihovna pro načítání, a v poslední verzi také ukládání a konvertování 3D modelů. 3D modely použité v této práci jsou v jednoduchém formátu `.obj`, takže by pro ně mohl stačit i jednoduchý vlastní mechanismus pro načítání, specifikace formátu `.obj` je ale dost vágní a různé nástroje pro 3D modelování generují soubory s odlišnou syntaxí.

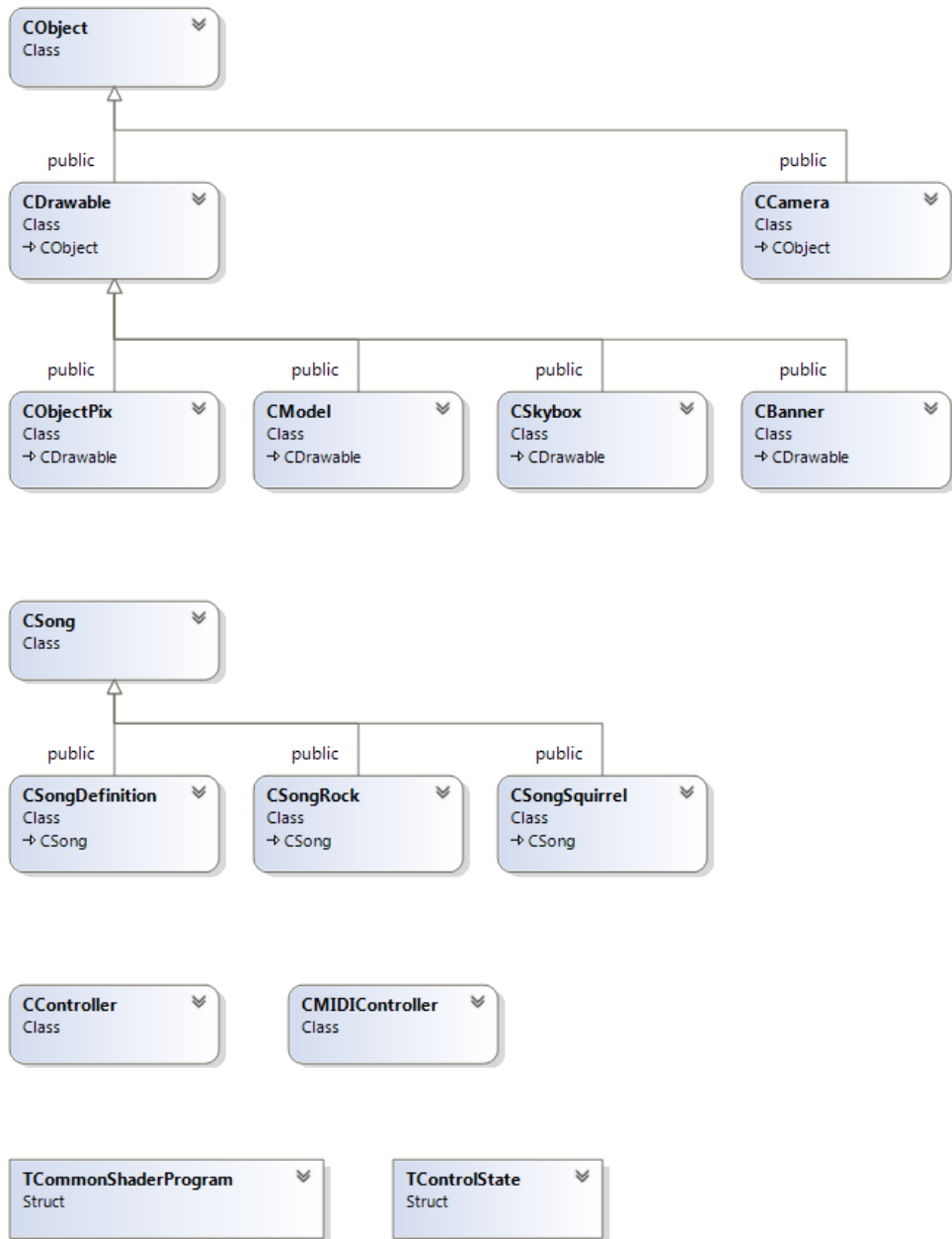
3.3 Diagram tříd

Diagram tříd 3.2 zobrazuje objektovou strukturu systému. Ta zahrnuje dvě skupiny dědičnosti, dvě samostatné třídy a dvě struktury. Další struktury jsou definovány v některých třídách, vždy se však jedná o pomocné struktury pro snadnější orientaci v členských proměnných třídy. Jedna taková je např. struktura `TTempMatrixess` ve třídě `CDrawable` pro dočasné uložení modelové, zobrazovací, normálové a PVM matice, než se odešlou jako uniform proměnné do shaderů.

3.3.1 Objekty ve scéně

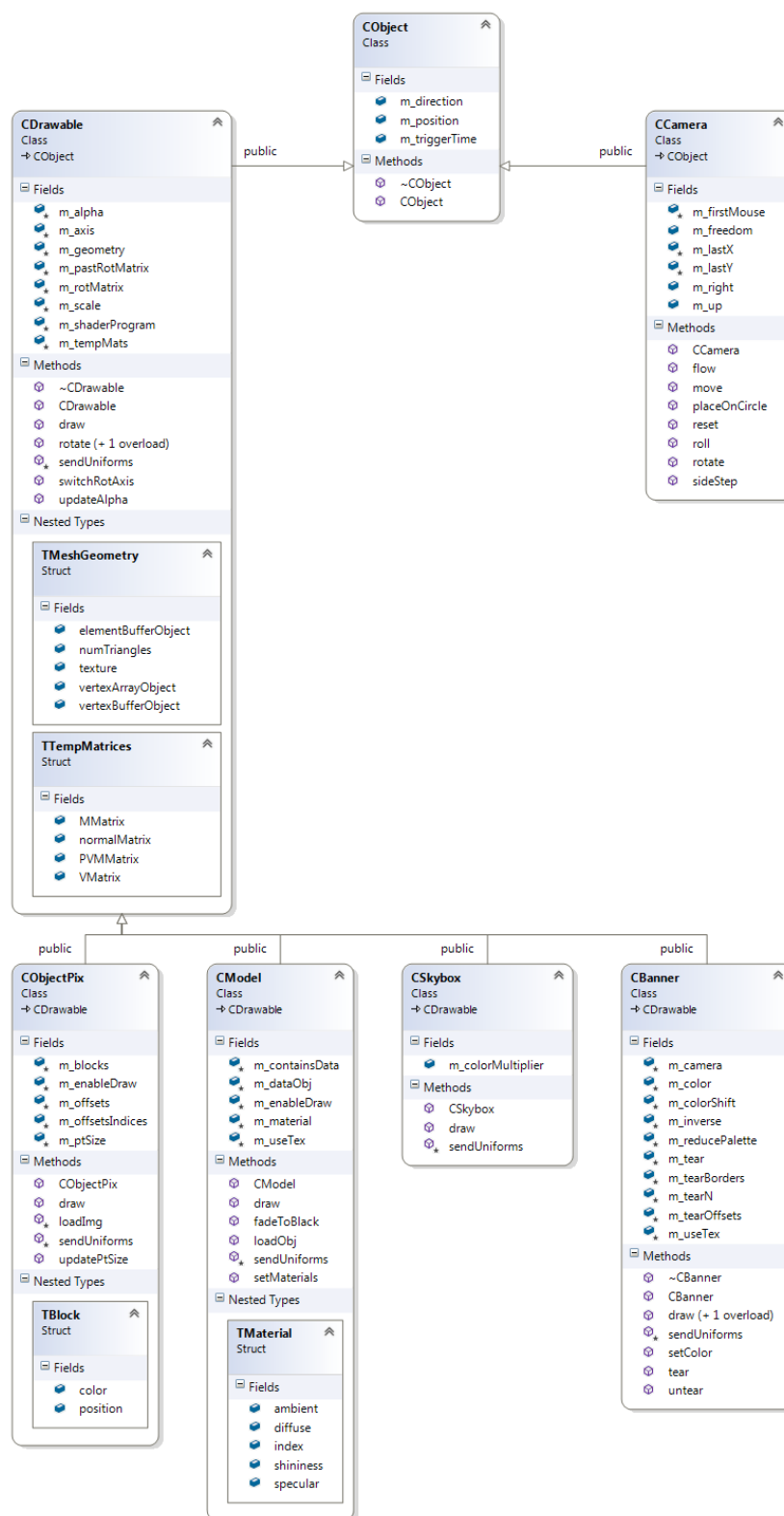
CObject První dědičnou skupinou jsou objekty ve scéně, podrobněji zobrazeny na diagramu 3.3. Hlavním předkem je abstraktní třída `CObject`, jež definuje proměnné sdílené ve všech potomcích. Jedná se například o pozici ve scéně `m_position` ve světových souřadnicích. Tato třída má dva přímé potomky.

CCamera Jedním z přímých potomků třídy `CObject` je `CCamera` pro objekt kamery. Metody `move()`, `sidestep()`, `roll()` a `rotate()` jsou pro zcela svobodnou 3D kameru ovládanou klávesnicí (pohyb pomocí WSAD či šipek, otáčení pomocí Q a E) a myši, metody `flow()` a `placeOnCircle()` pro pohyb ovládaný pomocí MIDI zpráv. Kamera často využívá tří vektorů udávajících



Obrázek 3.2: Globální diagram tříd

3. IMPLEMENTACE



Obrázek 3.3: Diagram tříd objektů ve scéně

její postavení – `m_direction` (z předka), `m_right` a `m_up`. Další členské proměnné jsou pomocné proměnné pro metody.

CDrawable Dalším přímým dědicem třídy `CObject` je abstraktní třída reprezentující všechny kreslitelné prvky ve scéně `CDrawable`. Ta předepisuje zejména čistě virtuální metody `draw()` a `sendUniforms()` a krom nich metody pro animace opakovaně využívané potomky. Významnými členskými proměnnými jsou `m_shaderProgram`, ukazatel na zkompileovaný OpenGL shader program předaný v konstruktoru a trojsložkový vektor `m_scale` určující zvětšení ve třech směrech, také zadaný v konstruktoru. Krom těchto se jedná opět o pomocné proměnné. Třída také zavádí struktury `TMeshGeometry`, jež sdružuje identifikátory OpenGL bufferů a objektů, identifikátor textury a počet trojúhelníků v meshi, a již zmíněnou strukturu `TTempMatrixess`.

CObjectPix Třída `CDrawable` má čtyři potomky. Prvním je `CObjectPix` reprezentující obrázek načtený do OpenGL bufferů po jednotlivých pixelech, které se následně vykreslí jako body. Výsledný objekt je pak 2D obrázek ve 3D prostoru s některými zajímavými odlišnostmi oproti klasickému přístupu zobrazení obrázku jako textury na čtverci nebo obdélníku (viz dále). Je možné měnit tvar a velikost bodů, je možné jednotlivými body dynamicky pohybovat ve třech rozměrech (u textury nerealizovatelné), navíc při rotaci objektu si body zachovávají tvar a orientaci, perspektivně se překrývají a vytváří falešný dojem objemu.

Zásadní je ve třídě `CObjectPix` metoda `loadImg()`, která pomocí knihovny DevIL načte a konvertuje zadaný obrázek do vhodného barevného formátu a s pomocí struktury `TBlock` jej uloží do OpenGL bufferu. Třída také implementuje zděděné čistě virtuální metody `draw()` a `sendUniforms()`, s jejichž pomocí vykreslí obsah bufferu jako body (oproti běžně kresleným trojúhelníkům).

CModel Druhým potomkem třídy `CDrawable` je třída reprezentující načtený 3D model (v současné době je podporován pouze typ souboru `.obj`). Jedná se o třídu `CModel`. Přidává strukturu `TMaterial` reprezentující materiály pro Phongův osvětlovací model. Obsahuje také ukazatel na případný datový objekt pro šetření paměti, více viz sekci 3.6.

CSkybox Dalším potomkem třídy `CDrawable` je třída reprezentující tzv. skybox `CSkybox`. Jedná se o objekt běžně používaný v realistické 3D grafice – krychli s texturami nebe, případně pohoří v dálce apod. V případě této implementace slouží skybox jako abstraktní barevné pozadí, kde je barva reprezentací pozice fragmentu v modelovacích souřadnicích objektu. Skybox se pohybuje s kamerou, není tedy možné z něj uniknout.

CBanner Posledním potomkem třídy `CDrawable` je třída reprezentující banner či ceduli `CBanner`. Jedná se o třídu v systému hojně využívanou, slouží k vykreslení obrázků jako textur, k efektu barevných filtrů a pro víceprůchodové vykreslování (viz. sekci 3.5). Jedná se o dvourozměrný 3D model složený ze dvou trojúhelníků, jehož modelové souřadnice se vygenerují podle poměru stran obrazovky během inicializace. Pro účely víceprůchodového vykreslování zavádí množství členských proměnných a metod, jako např. `tear()` a `untear()` pro trhání obrazovky.

3.3.2 Objekty testovacích skladeb

CSong Abstraktní třída `CSong` pro obecný objekt testovací skladby. Předepisuje řadu čistě virtuálních metod, jednak inicializačních – `shadersInit()`, `modelsInit()` a jednak metod zajišťujících odezvu – `redraw()`, `update()` a `midiIn()`. Obsahuje ukazatele na sdílené objekty uložené v objektu třídy `CController`. Implementuje také víceprůchodové vykreslování a obsahuje objekt `m_innerMap` pro množinu vnitřních stavů skladby.

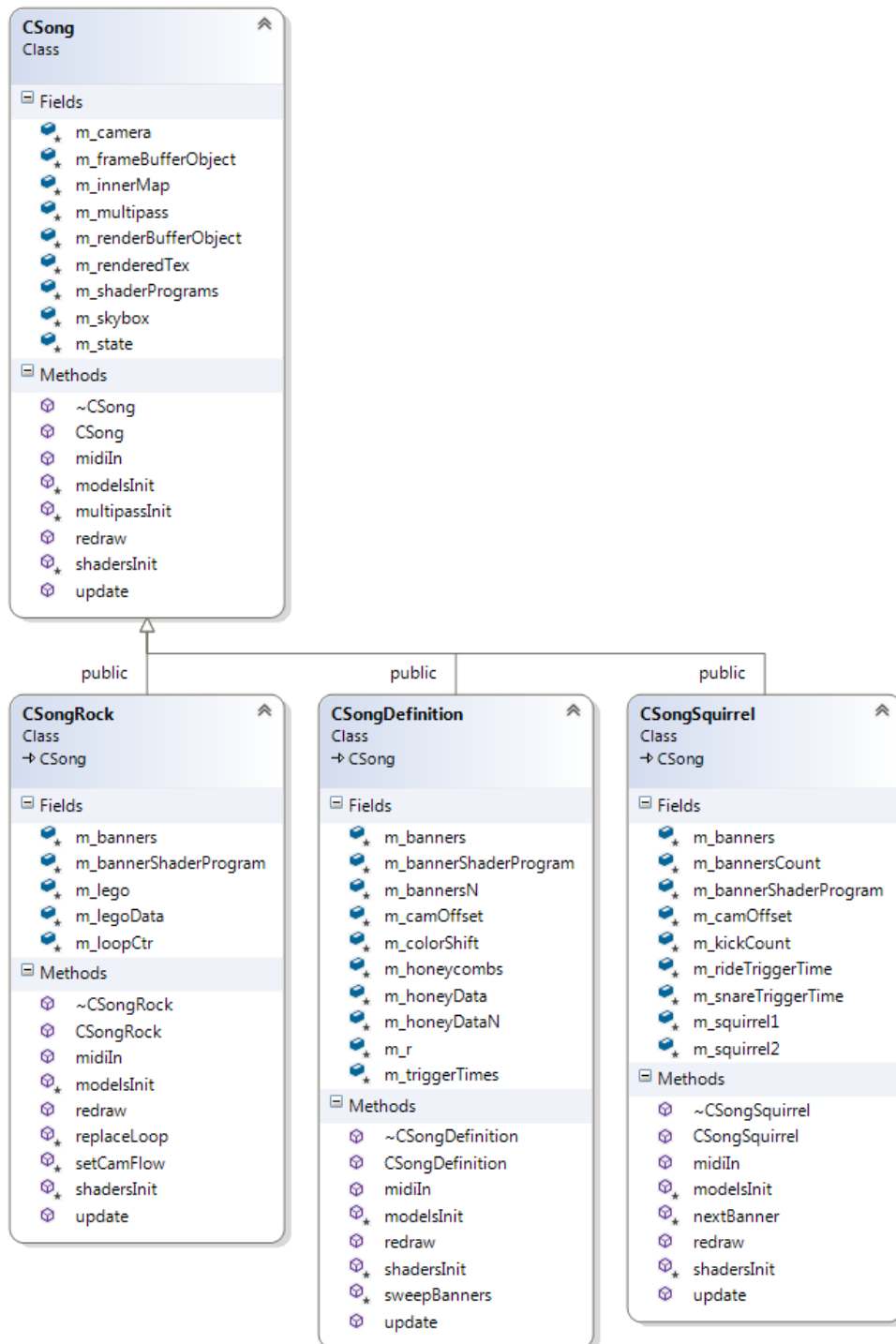
CSongRock Třída implementující testovací skladbu s názvem *Skála, na které postavím svoje lego*, popsanou v sekci 4.3. Hlavním prvkem je nekonečný tunel bloků stavebnice (pro nekonečnost viz sekci 3.6), které se podle přijatých MIDI zpráv otáčejí. Bloky, instance třídy `CModel`, jsou uloženy v poli `m_lego`, datové objekty (viz opět sekci 3.6) v poli `m_legoData`. Třída také ovládá kameru.

CSongDefinition Třída implementující testovací skladbu s názvem *Definice limity v bodě smrti*, popsanou v sekci 4.4. Hlavním prvkem je válec složený z modelů medových pláství, nahlížený z jiného pohledu než tunel ve třídě `CSongRock`. Válec je implementován podobně, není však nekonečný – kamera zda válcem neprochází, ale točí se kolem něj. Dále třída implementuje sadu instancí třídy `CBanner`, které v rychlém sledu mění a vytváří efekt stroboskopu.

CSongSquirrel Třída implementující testovací skladbu s názvem *Zmutovaná veverka*, popsanou v sekci 4.5. Využívá dvou instancí třídy `CObjectPix`, které synchronně animuje a dynamicky zobrazuje jednu nebo druhou. Dále implementuje barevné filtry s pomocí instancí třídy `CBanner` a trhání obrazovky s pomocí víceprůchodového vykreslování.

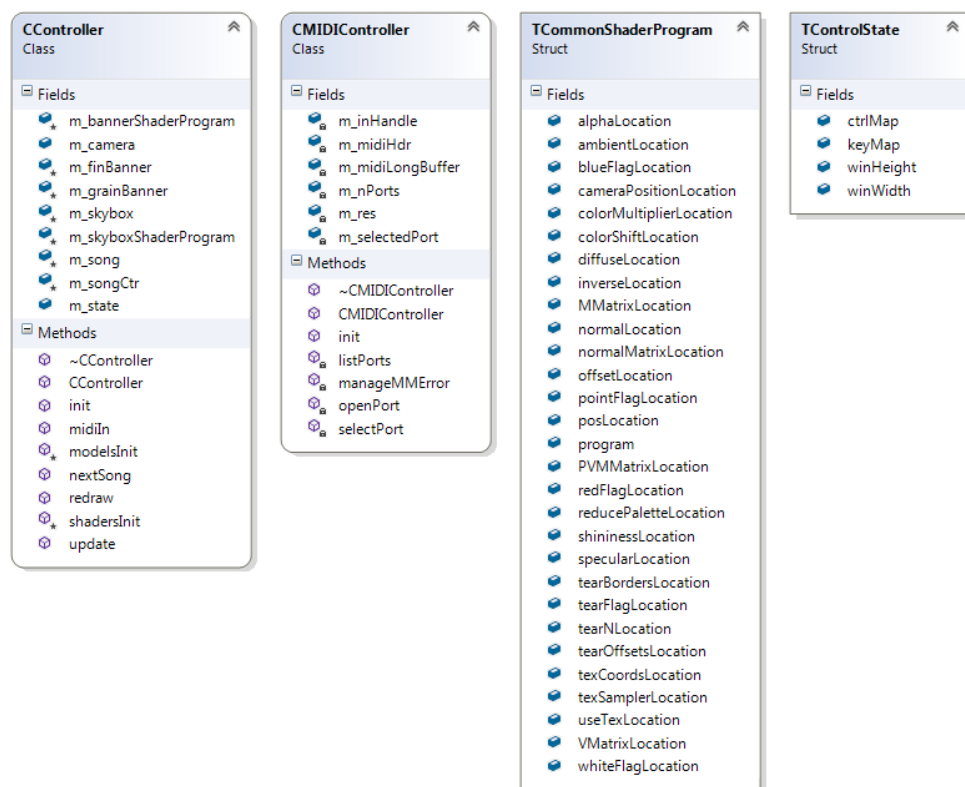
3.3.3 Ostatní objekty

CController Asi nejvýznamnější třída přímo nebo zprostředkovaně spravující instance většiny ostatních tříd. Jedná se o objekt inicializovaný v metodě `main()` a jediný objekt, který s ní přímo komunikuje. V hlavním cyklu, který běží až do svého ukončení frameworkem GLFW, jsou volány metody `redraw()` a `update()`, které dále volají stejnojmenné metody aktivní skladby. Při správě



Obrázek 3.4: Diagram tříd testovacích skladeb

3. IMPLEMENTACE



Obrázek 3.5: Diagram ostatních objektů

objektů dědicích ze třídy `CSong` je využito polymorfismu (ukazatel `m_song`). Třída dále spravuje všechny objekty ve scéně společně dvěma nebo více testovacím skladbám, zejména kameru (`CCamera`) a skybox (`CSkybox`).

CMIDIController Třída zajišťující MIDI komunikaci. Obaluje a zprostředkovává funkcionalitu knihovny WinMM, umožňuje konfiguraci MIDI zařízení a předzpracování MIDI zprávy, které dále předává metodě `midiIn()` třídy `CController`.

TCommonShaderProgram Struktura pro identifikátory (lokace) proměnných v GLSL shaderech. Vysoké množství identifikátorů je pro větší přehlednost kódu – sdílení proměnných pro různé efekty v různých shaderech by bylo matoucí.

TControlState Struktura shromažďující základní informace o běhu systému – šířku a výšku okna, stisknutá tlačítka, množinu vnitřních stavů. Přítomná ve třídě `CController` a předávaná třídám dědicích ze třídy `CSong`.

3.4 GLSL shadery

Systém využívá OpenGL verze 4.0 a odpovídající grafickou pipeline, která zahrnuje vertex a fragment shadery. Systém implementuje rovnou čtyři dvojice těchto shaderů. Jedná se o jednoduchý `skyboxShader`, využívaný objekty třídy `CSkybox`, který fragmentu přiřazuje barvu jeho pozice v modelových souřadnicích, a podobně jednoduchý `pixShader`, využívaný objekty třídy `CObjectPix`, který fragmentu přiřadí zadanou barvu. Tím výčet triviálních shaderů končí. Objekty třídy `CModel` využívají `commonShader` implementující Phongův osvětlovací model a objekty třídy `CBanner` využívají `bannerShader`, který krom přiřazení barvy či textury také implementuje veškeré efekty využívající víceprůchodové vykreslování. Veškerá odpovídající implementace, u které to bylo možné, je provedena v fragment shaderech pro vizuální plynulost.

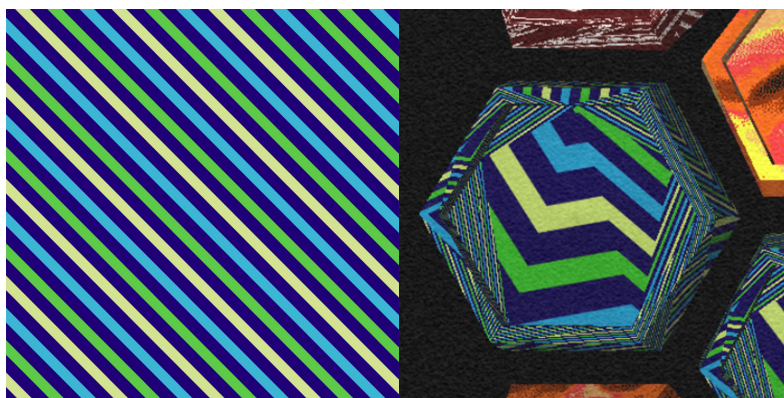
3.5 Implementované efekty

Vizuální prvky odpovídající zahraným notám sestávají ze dvou primárních částí, jednou jsou již popsané objekty ve scéně, další jsou efekty aplikované na tyto objekty, případně na celou scénu. Množství efektů je realizováno pomocí víceprůchodového vykreslování – běžným způsobem je do bufferu vykreslena celá scéna, ale místo aby se buffer zobrazil na obrazovku, použije se jako textura, která se zobrazí na obdélník ve scéně. S texturou lze v shaderech i mimo ně pracovat mnoha různými způsoby, které by jinak na celou scénu nebyly aplikovatelné. Textura se následně opět vykreslí a tentokrát se již zobrazí. Tento proces je možné iterovat dle výkonových možností, v tomto systému se však nepoužívá více než dvou průchodů.

Níže jsou stručně popsány vizuální efekty implementované v systému.

Náhodné texturovací souřadnice Během načítání modelu jsou typicky načteny i texturovací souřadnice udávající, která část textury odpovídá které části modelu. U některých načtených modelů jsou texturovací souřadnice generovány náhodně, jednotlivé prezentace vizualizací se tak díky prvku náhody od sebe liší a získávají živější charakter. Textury s jednoduchými vzory fungují pro tento efekt nejlépe, viz obrázek 3.6.

Rotace objektů, kamery Vděčným efektem je rotace, je proto v systému použita hned několikrát. Jedná se o rotaci objektů v čase, rotaci objektů kolem osy, rotaci kamery s objektem navázaným přímo před kameru či rotaci kamery uvnitř nebo vně sady objektů ve scéně. Zejména rotace spojená s pohybem v 3D prostoru je pro bytosti pohybující se většinou po ploše (ulice, podlahy...) zajímavá a dokáže vyvolávat až lehké závratě.



Obrázek 3.6: Vlevo textura, vpravo objekt s náhodnými texturovacími souřadnicemi.

Pohyb Pohyb je vedle rotace dalším jednoduchým, ale vděčným efektem. Implementován je pohyb kamery, byť v některých případech se může pozorovateli zdát, že se pohybuje zbytek scény.

Barevný doplněk Barvy jsou v OpenGL chápány v normalizovaném barevném prostoru RGB(A), kde každá složka je reálné číslo od nuly do jedné, získat doplněk je tedy snadné. Doplněk alfa není využit. Barevný doplněk je realizován s pomocí víceprůchodového vykreslování.

$$(r, g, b)' = (1 - r, 1 - g, 1 - b)$$

Zaokrouhlení barvy Pro snížení barevného rozsahu a deformaci 3D prostoru do barevných ploch lze barvu po složkách zaokrouhlit na danou přesnost. V systému je použito zaokrouhlení na celá čísla, výsledný barevný prostor se tedy skládá pouze z bílé (1, 1, 1), černé (0, 0, 0), červené (1, 0, 0), zelené (0, 1, 0), modré (0, 0, 1), žluté (1, 1, 0), růžové (1, 0, 1) a tyrkysové (0, 1, 1). Efekt zaokrouhlení barvy je realizován s pomocí víceprůchodového vykreslování.

Barevné filtry Použité barevné filtry jsou objekty třídy `CBanner` s nastavenou barvou a průhledností, posouvají tak barevné spektrum vykreslené scény blíže k barvě filtru.

Blending Blending není jeden efekt, ale způsob vykreslování v OpenGL, kde se překrývající objekty částečně vykreslí všechny, záleží na tzv. blending funkci. V systému je použit blending s alfa průhledností, blending funkcí je ale více a s použitím vhodných doplňků se dají snadno získat i funkce známé z nástrojů pro DTP apod.

Stroboskop Stroboskop se využívá pro své silné záblesky, ať už jednorázové, nebo opakující se v rychlém sledu. Tento efekt byl implementován rovnou dvěma různými způsoby, oba pomocí třídy `CBanner`: jednou objektem s bílou barvou a objektem s černou barvou v rychlém sledu, a jednou objektem s převážně bílou texturou, na který byl po krátkém časovém úseku aplikován efekt barevný doplněk. První varianta je teoreticky rychlejší, druhá poskytuje širší možnosti a zajímavější možnosti.

Barevný posun Posunem složek RGB lze jednoduše docílit silných barevných kontrastů a v případě fotografií přírodních objektů (lidí, zvířat, rostlin. . .) až mimozemskou vizáž. Jedná se v podstatě o dvě lineární zobrazení:

$$\begin{aligned}\mathcal{A}_1(r, g, b) &= (b, r, g), \\ \mathcal{A}_2(r, g, b) &= (g, b, r)\end{aligned}$$

Barevný posun je realizován s pomocí víceprůchodového vykreslování.

Zrno Pro přirozenější dojem je přes celý vykreslený obraz přidán ještě efekt zrna inspirovaný analogovou fotografií. Jedná se o objekt třídy `CBanner` s texturou mírně rozmazaného šumu a alfa průhledností.

Změna velikosti Všechny objekty tříd dědicích ze třídy `CDrawable` během vykreslování zohledňují zadané měřítko, které je možné dynamicky měnit a zejména u objektů třídy `CObjectPix`, kde se mění velikost jednotlivých bodů, lze dosáhnout efektních výsledků.

Trhání obrazovky Tzv. glitch, který nastává při špatně provedené synchronizaci obnovování obrazovky (typicky 60 Hz) a generování nových dat pro překreslení obrazovky – vertikální synchronizaci (VSync). Správné fungování VSync zajišťuje framework GLFW, trhání obrazovky bylo proto nutno simulovat, pro podrobnosti viz sekci 3.6. Trhání obrazovky je realizováno s pomocí víceprůchodového vykreslování.

Násobení barvy Efekt podobný efektu zaokrouhlení barvy, ale citlivější. Implementován pomocí multiplikátoru, kterým se v shaderu násobí barevné složky a lze jej dynamicky měnit. Čísla menší než jedna celý objekt ztmaví a barvy zešednou, čísla větší než jedna zostřují barevné přechody. Záporná čísla nemají smysl. Nejlépe funguje pro vícebarevné objekty s pozvolnými přechody, například objekty třídy `CSkybox`.

Šoková změna parametrů Obecný koncept efektu bez nutné konkrétní implementace, hlavní myšlenka tkví v současném aplikování několika efektů se silně odlišnými parametry od předchozího stavu. Např. stroboskop s pomalým

zhasínáním (pomalu se snižující alfa průhledností), který scénu zcela zakryje, a po vyblednutí záblesku jsou změněny materiály objektů či vynásobené barvy.

Světla Světla jsou v OpenGL mocný nástroj. Implementace využívá bodová a směrová světla Phongova osvětlovacího modelu. Světla se mohou rozsvěcet či zhasínat, pohybovat, kombinovat a mnoho dalšího.

3.6 Technická řešení

Níže jsou popsána některá řešení technickým problémů, které se během vývoje naskytly.

Běžné a širokoúhlé zobrazení Během poslední dekády došlo k rapidnímu vývoji obrazovek a s ním se často měnil i poměr stran obrazovky. Dá se říct, že dodnes zůstávají dva poměry stran – 4:3 a 16:9. Reálná situace ale není tak jednoduchá, např. během vývoje systému byly běžně používány dva různé monitory a ani jeden uvedeným poměrům neodpovídal. Jeden odpovídal poměru 16:10, druhý dokonce 683:384 (původní rozlišení 1366x768 px).

Dalším problémem bylo, jak vlastně poměr stran jednoduše určit? I kdyby existovaly pouze dva uvedené poměry stran, vydělení šířky a výšky obrazovky by nebylo spolehlivé kvůli nepřesnosti datových typů s plovoucí čárkou.

Řešením prvního problému byla generalizace. Textury a další materiály budou připraveny v obou poměrech stran, pokud bude detekován poměr 4:3 tak se použijí odpovídající, a pokud jakýkoli jiný, použijí se textury širokoúhlé a budou mít malou rezervu (zobrazí se blíže kameře a budou mít vždy nezobrazený okraj, ze kterého zobrazení podobná 16:9 budou čerpat).

Řešením druhého problému bylo použití vlastnosti racionálních čísel chápaných jako třídy ekvivalence párů celých čísel:

$$((m, n) \equiv (m', n')) \Leftrightarrow (n' \cdot m = n \cdot m')$$

Mějme šířku obrazovky w a výšku obrazovky h , pokud platí

$$((w, h) \equiv (4, 3)) \Leftrightarrow (3 \cdot w = h \cdot 4)$$

pak se jedná o klasický poměr stran 4:3, v opačném případě volíme širokoúhlý poměr stran 16:9.

Třída CObjectPix Pro originální zpracování 2D obrázků byla implementována třída CObjectPix. Stejně jako při načítání textury využije knihovny DevIL, načtená data ale zcela jinak interpretuje. Každý pixel převede do objektu struktury TBlock, do kterého uloží normalizovanou barvu a spočítá pozici v modelových souřadnicích. Nejlépe funguje s obrazovými formáty využívajícími alfa průhlednost (např. PNG) – zcela průhledné pixely tak vůbec neukládá a šetří tím paměť.



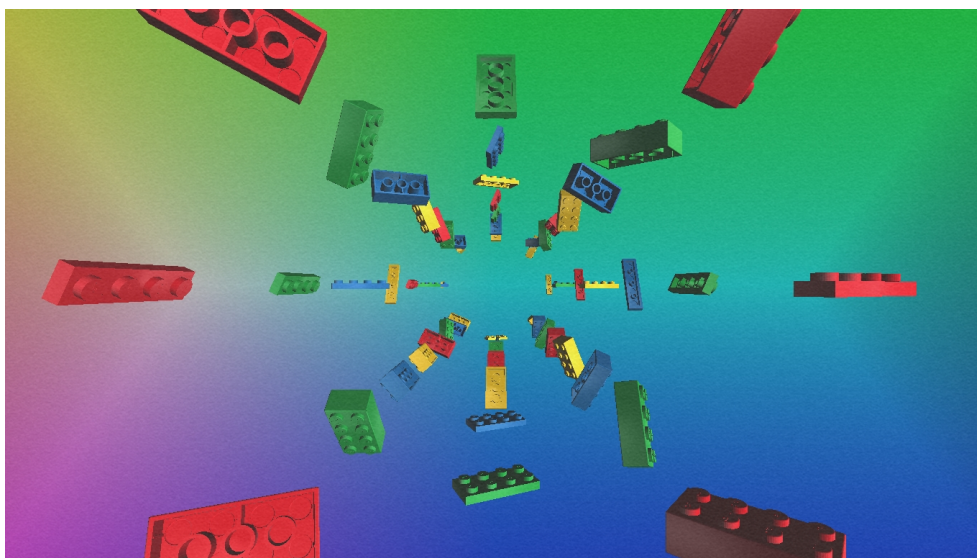
Obrázek 3.7: Trhání obrazovky. Je vidět natočený objekt třídy `CObjectPix`, jehož jednotlivé čtvercové body zůstávají kolmo na kameru, a přes něj zelený filtr. Na větších barevných plochách je patrné i zrno.

Pole objektů struktury `TBlock` pak třída uloží do paměti grafické karty běžným způsobem používaným v OpenGL. Navíc za každý blok přiloží nulový trojsložkový vektor posunu a data označí jako `GL_DYNAMIC_DRAW` – připravená pro častou změnu. Během vykreslování se vektor posunu přičte k pozici vertexu a data se kreslí jako body. Vektory posunu lze pak dynamicky měnit a celý obrázek tak ve třech směrech „trhat“.

Trhání obrazovky Jak bylo uvedeno v sekci 3.5, trhání obrazovky je nutné zcela simulovat. Oproti chybě vznikající při špatném VSyncu, kdy na jedné části obrazovky je ještě snímek minulý a na další už snímek současný, bylo řešeno rozdělením obrazovky do několika vertikálních částí a jejich horizontálním posunem.

Trhání obrazovky je implementováno ve fragment shaderu `bannerShader` s pomocí víceprůchodového vykreslování. Do shaderu jsou skrze uniform proměnné poslány počet tržení, hranice vertikálních oblastí, posuny jednotlivých oblastí a pravdivostní identifikátor, zdali efekt aplikovat. Funkce `getTear()` pak horizontální texturovací souřadnici přiřadí posun podle vertikální oblasti, ve které se nachází.

Počet tržení a posuny jsou generovány náhodně, počet však nepřesáhne deset a posun je v intervalu od nuly do jedné – odpovídá tak rozsahu texturovacích souřadnic. Při přetečení se souřadnice sníží o jedna.



Obrázek 3.8: Tunel z bloků stavebnice. Při pohybu vpřed nebo vzad se prstence postupně přesouvají.

Šetření paměti u opakujících se modelů Pokud je pomocí třídy `CModel` použit opakovaně ten samý model s tou samou texturou, bylo by zbytečným plýtváním paměti ho skutečně znovu načítat. Místo toho je možné třídě v konstruktoru předat ukazatel na objekt, který obsahuje načtená data, a při vykreslování se použijí data z datového objektu a matice objektu, na který bylo vykreslování voláno. Jedná se v podstatě o primitivní implementaci principu hluboké a mělké kopie, vzhledem k tomu, že se modely během průběhu už nijak nemění, nenastane při běhu systému situace vyžadující hlubokou kopii.

Nekonečný tunel bloků stavebnice Ve třídě `CSongRock` projíždí kamera „nekonečným“ tunelem složeným z prstenců tvořených bloky stavebnice. Nekonečno modelů by pochopitelně zabralo příliš paměti, proto je prstenců omezené množství a při pohybu vpřed se prstence, které kamera nechává za sebou, přesouvají zcela dopředu, a při pohybu zpět naopak.

Testování: Realizované hudební vystoupení

Vzhledem k charakteru práce je testování pojato jako veřejné hudební vystoupení. Pro účely obhajoby práce bude z vystoupení pořízen obrazový a zvukový záznam.

4.1 Testovací prostředí

4.1.1 Technické prostředky

Systém bude spuštěn na osobním počítači s procesorem Intel Core i5 generace Ivy Bridge s integrovanou grafickou kartou Intel HD 4000 a 8 GB RAM. Pro účely vytvoření záznamu byly z Institutu intermédií při Fakultě elektrotechnické zapůjčeny zvukový rekordér Zoom H6 a dvě kamery GoPro Hero 3+.

Vystoupení bylo naplánováno na 9. 5. 2016 v kulturním centru Klubovna v pražských Dejvicích. Kulturní centrum Klubovna je zavedeným prostorem, který spravuje spolek Povaleč zodpovědný za stejnojmenný alternativní hudební festival. Hudební sál v kulturním centru Klubovna má mezi pražskými sály s obdobnou kapacitou (přibližně 70 návštěvníků) jedny z nejlepších podmínek pro kvalitu zvuku i promítání – to je jeden z hlavních důvodů pro jeho volbu. Dalšími jsou profesionální personál a popularita prostoru mezi veřejností.

Hudební vystoupení bude první koncert nového hudebního projektu. Jedná se o dvoučlenný projekt inspirovaný fúzí tradiční a elektronické hudby, konkrétní žánry, ze kterých byla čerpána primární inspirace, jsou punk, black metal, noise, rave, drum & bass, techno a hardtek. Ve výsledku se jedná o expresivní hudební projev s důrazem na rytmiku, repetici a náhodu. Oba dva hudebníci také mají relativně široký prostor pro improvizaci.

Využity budou následující hudební nástroje.

4. TESTOVÁNÍ: REALIZOVANÉ HUDEBNÍ VYSTOUPENÍ

- MIDI zařízení
 - sampler *Akai Professional MPX16*. Přestože se jedná o výrobek od renomovaného výrobce, MIDI implementace nezahrnuje MIDI Through a i po aktualizaci firmwaru působí jen v módu, kdy přijímá zprávy na kanálu 1, které dál neposílá, stejně jako neposílá zprávy na ostatních kanálech. Proto je tento sampler v sériovém zapojení umístěn jako první.
 - Bicí automat *Alesis SR-16* je legendou ve svém oboru, která koncem devadesátých let minulého století značně napomohla rozvoji hip hopu pro svou nízkou pořizovací cenu a kvalitní zvuk. Dodnes se jedná o oblíbený bicí automat, přestože bicí automaty byly již technologicky překonány. Jednotka použitá během testování byla přibližně pět let stará.
 - Převodník *M-Audio Midisport 1x1* zprostředkovává převod z pětipinového MIDI kabelu do USB. Umožňuje samostatný vstup i výstup.
- Ostatní
 - *Korg Monotron Duo* je malý analogový syntezátor se dvěma oscilátory a dolní propustí. Jeho přednostmi jsou kvalitní zvuk v malém formátu a otevřenost – technické nákresy byly oficiálně zveřejněny a syntezátor je přizpůsoben pro snadné hackování. Během testování byl však ještě v původním stavu.
 - *Distant Voices Theremin* je syntezátor od české firmy Widara. Jedná se o variaci na původní elektronický nástroj navržený v roce 1919 Lvem Sergejevičem Těrmenem. Vyniká svým vysokým zvukem a bizarním ovládáním – v originálním návrhu se o jedná o dvě antény, z nichž jedna ovládá výšku tónu a jedna hlasitost.
 - Druhý hudebník bude hrát na elektrickou kytaru *Fender Stratocaster* zapojenou do standardního kytarového zesilovače. Také pracuje s množstvím kytarových efektů, mezi hlavní patří delay a distortion, pomocí kterých vytváří hlukové stěny.

Systém bude testován na rychlost odezvy a následně během veřejného vystoupení třemi rozdílnými hudebními skladbami, jejichž hudební a vizuální charakteristiky následují. Z vystoupení bude pořízen audiovizuální záznam, který bude součástí této práce.

4.2 Test I: Zpoždění odezvy

Prvním testem je test rychlosti odezvy. Z běhu bude pořízen záznam s vysokou snímkovací frekvencí zabírající jak MIDI zařízení, tak vizualizace v jednom záběru. Zpomalením záznamu bude zkoumáno zpoždění odezvy, přičemž přesnost měření bude \pm perioda snímání.

4.3 Test II: „Skála“

4.3.1 Hudební charakteristika

První testovací skladba nese název *Skála, na které postavím svoje lego*. Hudebně se jedná, alespoň oproti dalším testovacím skladbám, o pomalé (120 BPM) a citlivé dílo inspirované současným berlínským technem a francouzským housem.

Jako taková stojí skladba na jednoduchém beatu tzv. „four on the floor“ – jedná se o basový buben na každou těžkou dobu ve čtyř čtvrtovém taktu. Ten je doplněn činelem hi-hat na vedlejší osminy a v závěrečné části skladby samplem připomínajícím virbl na druhou a čtvrtou dobu. Ve střední části je beat rozložen do podoby bez basového bubnu.

Dva minimalistické basové motivy jsou hrány na kytaru s efektem oktáveru, který přidá tón o oktávu níže, případně výše. Většinou se však využívá první varianta.

Primární basová linka vychází ze sampleru a jedná se opět o rozdílné linky pro první a druhou část skladby. Dále ze sampleru vychází dva vokální motivy recitující „Král Midas má oslí uši“ mužským a ženským hlasem.

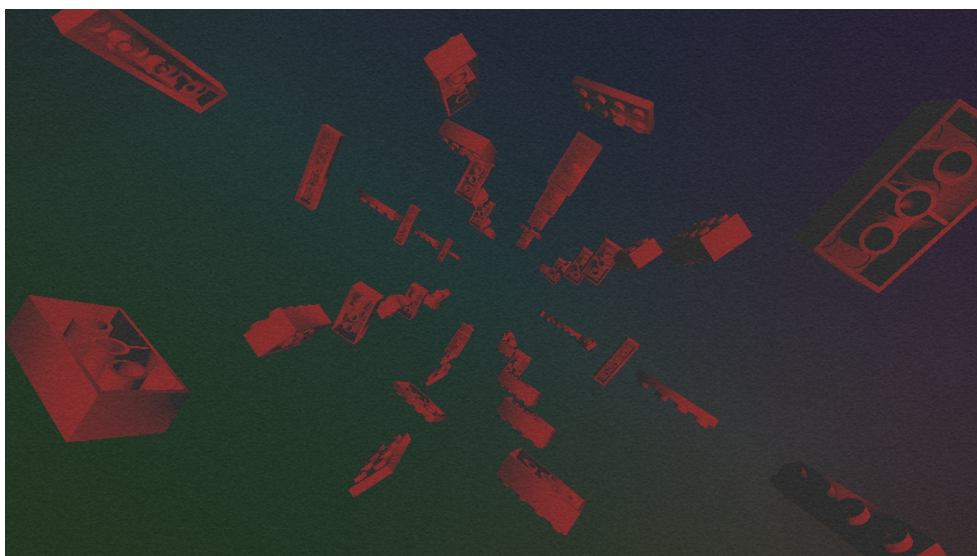
4.3.2 Využití vizuální prvky

Základní strukturou je tunel složený z prstenců tvořených bloky stavebnice Lego, viz obrázek 3.8. Každý prstenec sestává z osmi bloků: čtvercově jsou umístěny vysoké bloky s 4x2 piny, na vodorovné ose jsou na obou stranách umístěny nízké bloky s 4x1 piny a na svislé ose je dole i nahoře po jednom nízkém bloku s 4x2 piny. Tyto prstence se opakují do dále, jejich počet byl přizpůsoben výkonnostním možnostem zařízení, na němž systém během testu běžel. Pro fungování prstenců viz sekci 3.6.

Bloky stavebnice odpovídají bicím a při každém úderu se otočí kolem náhodné osy. Vysoké bloky s 4x2 piny reagují na basový buben, bloky s 4x1 piny reagují na hi-hat a nízké bloky s 4x2 piny reagují na virbl.

Tunelem projíždí kamera, podle tónu zahraného samplerem se pohybuje buď dopředu nebo dozadu, během některých se navíc otáčí kolem směru pohybu.

Vokální sampl oba spouští záblesk stroboskopu, mužský spouští stroboskop bílý a ženský stroboskop žlutý. Mužský vokál je navíc pouštěn v pivotním



Obrázek 4.1: Závěrečná fáze vizualizací skladby „Skála“

momentě skladby, po vyhasnutí stroboskopu se tedy změní materiály všech bloků na červené, zhasne bílé světlo a rozsvítí se červené a sníží se barevný multiplikátor skyboxu, viz obrázek 4.1.

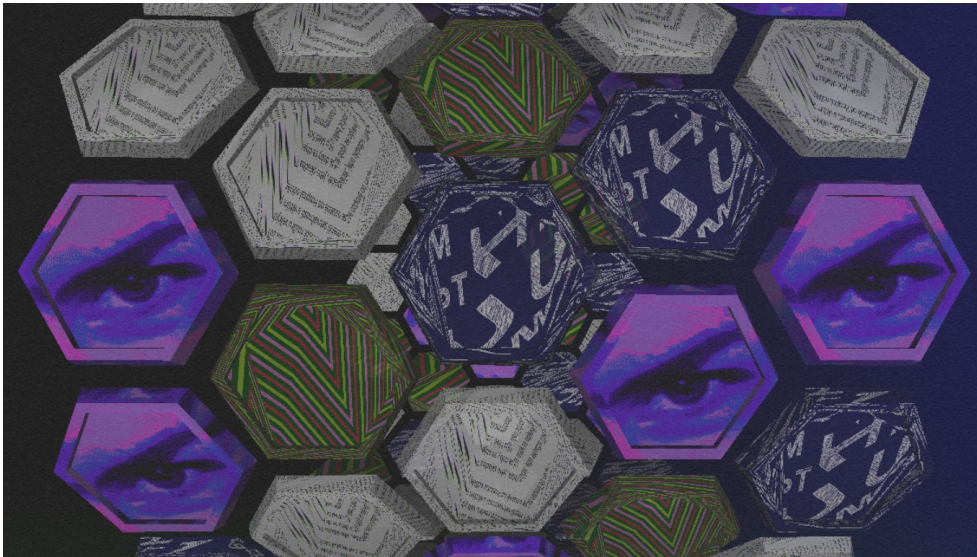
4.4 Test III: „Definice“

4.4.1 Hudební charakteristika

Druhou testovací skladbou je *Definice limity v bodě smrti*. Jedná se o skladbu s nejdelší historií, první varianta byla zveřejněna koncem roku 2012. Tehdy se jednalo o Lo-Fi experimentální elektronickou hudbu, v další iteraci o fúzi drum & bass a thrash metalu a v prezentované verzi o fúzi drum & bass a black metalu. Tyto žánry přibližně sdílí tempo a skladba byla prezentována na 170 BPM.

Skladba je opět rozdělena na tři části. První vede agresivní synkopovaný beat postavený na tzv. amen breaku charakteristickém pro drum & bass, druhá připomíná beat skladby první – basový buben a činely hi-hat na vedlejší osminy, a ve třetí se jedná o sérii tzv. blast beatů charakteristických pro black metal. V typickém blackmetalovém blast beatu se ve vysokém tempu střídají šestnáctiny basového bubnu, virblu a hi-hat doplněné činelem crash. Vznikají hlukové stěny, ve kterých lze sledovat repetitivní vzory, často synkopované.

Beat poskytuje značné množství MIDI informací, je proto doplněn pouze improvizovaným rytmickým samplem připomínajícím zvuk filmového laseru a jedním vokálním samplem.



Obrázek 4.2: Osvícený válec ve skladbě „Definice“ s barevným posunem (pro srovnání viz obrázek 3.6)

Kytarista během hry používá silně efekt distortion pro metalový zvuk kytary a věnuje se zejména metalovým riffům.

To vše je doplněno improvizovanou hrou na theremin.

4.4.2 Využité vizuální prvky

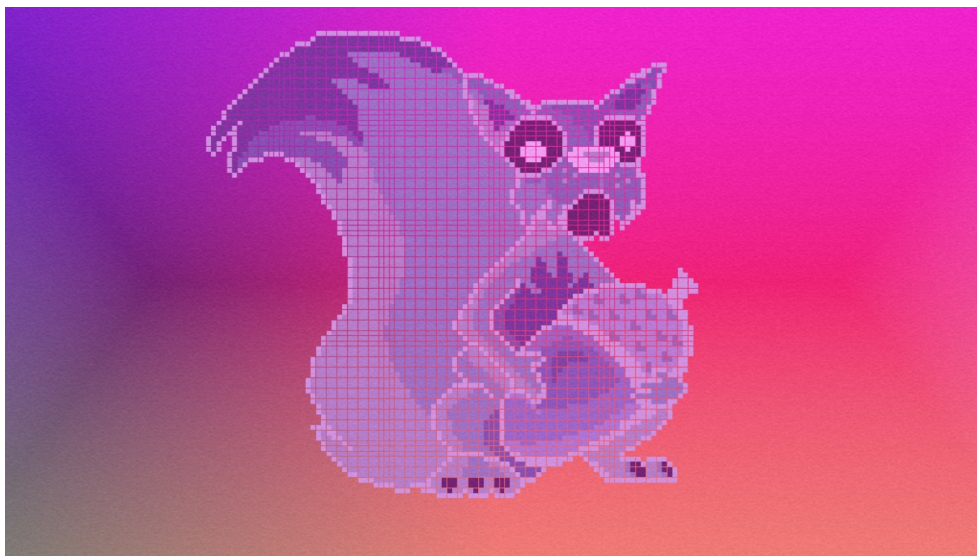
Základním prvkem je točící se válec sestavený z modelů medových pláství. Tyto modely jsou poloprůhledné a mají náhodné textury (z výběru čtyř), některé texturovací souřadnice jsou náhodné. Standardně není osvětlen, světla se rozsvítí jen na krátký okamžik při samplu laseru. Bicí přechody (tomy) navíc plástve osvětlují červeným či modrým světlem. Na basovém bubnu, hi-hat a virblu jsou namapovány stroboskopy s texturami, na všech činelech krom otevřené hi-hat je namapován třístupňový barevný posun (žádný, posun o jedna, posun o dva).

4.5 Test IV: „Veverka“

4.5.1 Hudební charakteristika

Poslední testovací skladbou je *Zmutovaná veverka*. Jedná se o ze začátku přímočarý hardtek, který se v závěrečné pasáži změní na hutný noise. Při 175 BPM se jedná o nejrychlejší testovací skladbu.

Skladbě dominuje repetitivní vokál podložený rychlým hardtekovým beatem (opět připomínající techno) se silnými výplněmi a přechody mezi jed-



Obrázek 4.3: Veverka s oříškem ve skladbě „Veverka“. Jsou patrné efekty zelený filtr, barevný doplněk a změna měřítka – vykreslované body se zmenšují a jako příjemný vedlejší efekt tvoří moaré.

notlivými smyčkami. Možnosti bicího automatu jsou naplno využity a ostatní hudební prvky ustupují do pozadí.

Kytara se věnuje kombinaci efektů a ostrým přechodům mezi jejich parametry. Na analogovém syntezátoru je využito dolní propusti a vysoké rezonance pro efekt zvaný „filter sweep“.

4.5.2 Využití vizuální prvky

Hlavním prvkem je pár objektů `CObjectPix` – ilustrace roztomilé a zmutované veverky ve stylu pixel art, které se s úderem basového bubnu střídají. Ty jsou napevno umístěny přímo před kamerou a když kamera náhodně rotuje, jsou perspektivně deformovány. Systém počítá doby a na každou první nastaví velikost vykreslovaných bodů na právě tak velkou, aby byly veverky dobře patrné. Během taktu se pak velikost bodů lineárně snižuje až klesne na pevně stanovené minimum.

Třem různě laděným přechodům (tomům) odpovídá přepínání barevných filtrů základních barev (jedná se v podstatě o kombinaci stroboskopu a barevných filtrů):

$$R \rightarrow G$$

$$G \rightarrow B$$

$$B \rightarrow R$$

Dva činely crash zapínají a vypínají barevný doplněk, činel ride pak dočasně aktivuje zaokrouhlení barev. V závěrečné pasáži se přidá virbl s aplikovaným kytarovým efektem distortion (tvořící zvuk charakteristický pro extrémní elektronickou hudbu), jež aktivuje trhání displeje.

4.6 Průběh testování

4.6.1 Test I

Pro test byla použita kamera GoPro Hero 3+, která umožňuje snímání frekvencí až 60 fps, přesnost měření bude tedy $\pm 1/60$ s.

Zaznamenáno bylo deset testů s bicím automatem *Alesis SR-16* a deset testů se samplerem *Akai Professional MPX16*. Pro konkrétní data viz tabulku 4.1.

Sampler se projevil jako vhodnější – jako novější zařízení má nejen svítící displej, ale také svítící pady (tlačítka), takže je daleko snazší vizuálně detekovat, kdy byly přesně aktivovány. U bicího automatu vedla nejistota k více rozptýleným výsledkům a pravděpodobně k příliš brzké detekci aktivace padu.

Po zaokrouhlení průměrů (desetinná část nemá v dané přesnosti význam) získáváme zpoždění odezvy 2–5 snímků, tedy $1/30$ – $1/12 \pm 1/60$ s. Zpoždění odezvy systému je tak přibližně stejně dlouhé, jako perioda mezi snímky filmu točeného při standardním snímkování 25 fps a animace by tedy měla být dostatečně plynulá.

4.6.2 Testy II, III a IV: veřejné vystoupení

Na 9. května bylo naplánováno veřejné vystoupení. Za předkapelu byli zvoleni Human Ketchup hrající živý jungle. Jedná se o bubeníka a dva hudebníky s elektronickými nástroji, přičemž všichni tři částečně improvizují v rámci nepřestávající dramaturgie. Stylově se tak více než hodí.

Před živým vystoupením proběhla v sále ještě kompletní zkouška první skladby, která se obešla zcela bez problémů. Ke třem neočekávaným událostem ale mělo dojít po skončení předkapely, během které se sál zaplnil a atmosféra byla ideální. Jedna z neočekávaných událostí byla pozitivní: zvukový technik se nabídl, že pro zvukový záznam využije vlastní zařízení připojené přímo do mixážního pultu. Nebyl tak nakonec využit rekordér Zoom H6, který by zaznamenal pouze zvuk v sále, místo toho byl k dispozici jak záznam zvuku

Tabulka 4.1: Test zpoždění odezvy. Počet snímků udává časový rozdíl mezi aktivací padu (tlačítka) na nástroji a vizuální odezvou na monitoru.

MIDI nástroj	počet snímků										průměr
<i>Alesis SR-16</i>	7	6	7	4	5	4	3	5	6	5	5,2
<i>Akai Professional MPX16</i>	4	3	4	3	3	4	2	3	3	4	3,3

v sále, tak záznam zvuku přímo z mixážního pultu. Tyto dva záznamy byly pro účely záznamu přiloženého k této práci smíchány, aby adekvátně reprezentovaly zvukovou atmosféru v sále.

Další dvě události byly negativní. Přes předchozí testování se nepodařilo nakonfigurovat kameru, která měla být u zvukového technika, a nebyl z ní pořízen žádný záznam (záznam na přiloženém médiu je tedy pouze z kamery z pódia). Druhou komplikací bylo poškození adaptéru bicího automatu, bez kterého by nebylo možné vystoupení realizovat. První variantou řešení bylo adaptér nahradit jiným. Byly prohledány veškeré technické rezervy, jednak vlastní, jednak předkapely, jednak klubu, celkem bylo ozkoušeno přes dvacet adaptérů, ale žádný nevyhovoval starému standardu 12 V střídavého proudu (v podstatě veškeré dnešní adaptéry používané pro hudební zařízení jsou stejnosměrného proudu). Nakonec byl tedy adaptér přímo na pódiu rozebrán a poškozený (nejspíše natržený a následně přehořelý) kontakt nouzově opraven zvukovým technikem, který tím celé testování v podstatě zachránil.

Další komplikace během vystoupení nenastaly. Zpětná vazba byla až na výjimky silně pozitivní (zpětná vazba po koncertě by se ale neměla přeceňovat, jak bylo uvedeno v sekci 2.3), návštěvníci chválili čitelný vztah mezi obrazem a zvukem.

4.7 Shrnutí

Přes některé komplikace byly provedeny všechny testy s příznivými výsledky. Zpoždění odezvy je dost nízké, aby za běžných podmínek bylo jen těžko patrné, a během vystoupení systém bezchybně fungoval. Chybou muzikantů byl systém zapnut během druhé zvukové zkoušky těsně před vystoupením a objekty ve scéně, které měly přibývat postupně, byly vidět již od začátku. V chaotické situaci s potenciálně daleko horšími výsledky bylo ale zapomenuto skladbu resetovat. Další drobnou komplikací byl požadavek systému na manuální přepínání skladby – nebylo sice nikdy zapomenuto, ale bylo činěno vždy na poslední chvíli. Lepším řešením by možná bylo změnu skladby navázat na MIDI zprávu změny programu.

Závěr

Splnění cílů práce

1. **Provést rešerši existujících softwarových nástrojů pro vizualizaci hudby** V kapitole 1 byla provedena rešerše v oblasti hudební a technické. Byly zkoumány hudební proudy a s nimi spojené vizualizace v oblasti světél, VJingu, projekcí a dalších. Dále byly zkoumány technické možnosti realizace vizualizací, zejména rozdíl mezi použitím spektrální analýzy hudebního signálu a použitím standardu MIDI. Také byly prozkoumány předchozí příklady použití standardu MIDI pro účely hudebních vizualizací.
2. **Navrhnout technicky netriviální princip vizualizace** V kapitole 2 bylo navrženo použití standardu MIDI pro komunikaci hudební informace. Dále byl zvolen programovací jazyk C++ a knihovna OpenGL verze 4.0. Byla navržena objektová struktura kódu snadno rozšiřitelná o další prvky ve scéně, efekty a vizualizované skladby.
3. **Navrhnout esteticky čitelný vztah zvuku a obrazu** V kapitole 2 byl navržen princip sestávající ze sady objektů ve scéně a efektů na ně aplikovaných. MIDI zprávy aktivují zobrazení objektů a aplikaci efektů podle typu zprávy, vizualizované skladby a pozice v dramaturgii. Objekty a efekty se ve scéně překrývají a kombinují a odpovídají tak komplexnímu hudebnímu projevu.
4. **Navrhnout systém vhodný pro využití během živého vystoupení** V kapitole 2 byl navržen systém čerpající vstup téměř výhradně ve formátu MIDI, aby během vystoupení nevyžadoval pozornost účinkujících a ti se mohli soustředit na generování hudby a odpovídajících MIDI zpráv. Návrh systému počítá s konfigurovatelností formy vstupu a výstupu – systém tak bude možné nasadit v různých podmínkách a s použitím různých zařízení pro MIDI vstup a obrazový výstup.

5. **Realizovat systém generující vizuální odezvu na vstupní proud dat ve formátu MIDI** V kapitole 3 jsou popsány postupy použité při implementaci systému. Byly implementovány vizualizace pro tři testovací skladby.
6. **Otestovat systém při živém veřejném vystoupení** Jak popisuje kapitola 4, systém byl veřejně otestován 9. 5. 2016 v Kulturním centru Klubovna v Praze. Záznam z vystoupení byl o týden později zveřejněn on-line [8] a je také přiložen na datovém médiu.

Budoucnost

Hlavním cílem do budoucna je připravit další skladby a vizualizace pro ně a sestavit plnohodnotné vystoupení prezentovatelné na dalších akcích a festivalech. Navazovat bude vydání díla na odpovídajícím nosiči (bude nutné provést patričnou rešerši, jaké médium je vlastně vhodné pro kombinaci zvuku a obrazu a zároveň je dostatečně dostupné).

S přípravou dalších skladeb a vizualizací bude rozšířen nástrojový park. V plánu je analogový basový syntezátor s podporou MIDI (například Arturia MiniBrute) a použití herní konzole Nintendo GameBoy Color jako syntezátoru – takový úkol dokáže splnit například aplikace Nanoloop [9]. Otázkou zůstává, jakou variantu synchronizace GameBoye zvolit – existují převodníky GameBoy/USB, GameBoy/MIDI na platformě Arduino [10] a další.

Dalším plánem je přidání podpory protokolu DMX pro ovládání pódiových světel. S navrženou strukturou kódu by to nemělo přinést větší potíže, pravděpodobně bude přidána třída zajišťující DMX komunikaci podobně jako třída `CMIDIController`. Náročnější bude pořízení světel samotných, alespoň pro testování – ne že by nebyla dostupná, ale jsou finančně nákladná.

Posledním plánem je větší inkluze hudebních nástrojů bez implementace MIDI. Představa zahrnuje mechanické spínače na kytarových efektech posílající informace do malého počítače (například na platformě Arduino), který by generoval MIDI zprávy. Dalším nápadem je zahrnutí bicích snímačů a doplnění bicího automatu živými bicími – alespoň pro přechody a dramatické části (ostatně je nedávným trendem na alternativní hudební scéně, že zpěvák má k dispozici jeden či dva bubny a podle potřeby doplňuje rytmickou sekci).

Použité zdroje

- [1] Pendulum: *Hold Your Color* [CD]. England: Breakbeat Kaos, 2005.
- [2] Pendulum: *Live at Brixton Academy* [CD]. London: Warner Music UK Ltd., 2009.
- [3] *The Glitch Mob Behind The Blade* [film][online]. Režie Matthew F. Smith, 2015, [cit. 4. 5. 2016]. Dostupné z: <https://youtu.be/xUu8jxWDSOU>
- [4] *Changing Faces – Liquid Flights* [videozáznam][online]. DnBPortal.com, 2015, [cit. 5. 5. 2016]. Dostupné z: <https://youtu.be/7gRHuXqFmu0>
- [5] Etherwood: *Blue Leaves* [Vinyl]. London: Med School, 2015.
- [6] Toundra: *II* [Vinyl]. Schönkirchen: Narshardaa Records, 2014.
- [7] *OpenTZZT Documentation* [online]. Sourceforge, 2010, [cit. 7. 5. 2016]. Dostupné z: <http://opentzt.sourceforge.net>
- [8] Petrov, V.: *Rejfpark live @ Klubovna 9. 5. 2016* [videozáznam][online]. 2016, [cit. 16. 5. 2016]. Dostupné z: <https://youtu.be/HzFjP2zBFHc>
- [9] Wittchow, O.: *Nanoloop 1.7 manual* [online]. Hamburg: University of Fine Arts of Hamburg, 2014, [cit. 4. 5. 2016]. Dostupné z: <http://www.nanoloop.com/one/nanoloop17.html>
- [10] Lamb, T.: *Arduinoboy* [online]. GitHub, 2015, [cit. 16. 5. 2016]. Dostupné z: <https://github.com/trash80/Arduinoboy>
- [11] Shreiner, D.; Sellers, G.; Kessenich, J.; aj.: *OpenGL programming guide*. Upper Saddle River, NJ: Addison-Wesley, 8 vydání, 2013, ISBN 03-217-7303-9.
- [12] Gerald, P.; Zapotocky, S.: *Realtime Music Visualization with MIDI and OpenGL*. Vienna: Vienna University of Technology, 2009.

- [13] Biggiogero, D.: *Gameboy MIDI Chrome visuals* [online]. GitHub, 2016, [cit. 7. 5. 2016]. Dostupné z: <https://github.com/zeroerrequattro/chrome-visuals>
- [14] Avery, R.: *MIDI 2 OpenGL Visualization* [online]. 2004, [cit. 7. 5. 2016]. Dostupné z: <http://chancesend.com/2004/12/midi-2-opengl-visualization>
- [15] *The complete MIDI 1.0 detailed specification*. Los Angeles: MIDI Manufacturers Association, druhé vydání, 2001, ISBN 09-728-8310-X.
- [16] Wright, M.: *The Open Sound Control 1.0 Specification* [online]. 2002, [cit. 11. 5. 2016]. Dostupné z: http://opensoundcontrol.org/spec-1_0
- [17] Woods, D.: *Developer's Image Library Manual* [online]. 2002, [cit. 11. 5. 2016]. Dostupné z: <http://openil.sourceforge.net/docs/DevIL%20Manual.pdf>
- [18] *MIDI Visual Control Specification*. Los Angeles: MIDI Manufacturers Association, 2011.
- [19] *DMX 101: A DMX 512 HANDBOOK*. Los Angeles: Elation Professional®, 2008.

Seznam použitých zkratek

MIDI	Musical Instrument Digital Interface
OSC	Open Sound Control
DMX	Digital Multiplex, také DMX-512
DJ	Disc Jockey
VJ	Video Jockey
PAR	parabolické pódiové světlo
FFT	Fast Fourier Transformation, Rychlá Fourierova transformace
CV/Gate	Control Voltage / Gate
VST	Virtual Studio Technology
DAW	Digital Audio Workstation
VIP	Variation In Production, autorský remix
API	Application Programming Interface
OpenGL	Open Graphics Library
GPU	Graphic Processing Unit
GLEW	Graphics Library Extension Wrangler
GLFW	Graphics Library Framework
GLM	OpenGL Mathematics
GLSL	Graphics Library Shading Language
DevIL	Developer's Image Library

A. SEZNAM POUŽITÝCH ZKRATEK

Assimp	Open Asset Import Library
WinMM	Windows Multimed
PVM	Project Model View (matice)
RGB	Red Green Blue, barevný model používaný pro obrazovky
RGBA	RGB s kanálem průhlednosti alfa
DTP	Desktop Publishing
RAM	Random Access Memory
BPM	Beats Per Minute
VSync	Vertikální Synchronizace
PNG	Portable Network Graphics
Lo-Fi	Low Fidelity, záměrně nízká kvalita záznamu
MD5	Message-Digest algorithm

Obsah přiložené SD karty

readme.txt	stručný popis obsahu karty
exe	adresář se spustitelnou formou implementace
src		
impl	implementace
src	zdrojové kódy
visualstudio2015	projekt pro Visual Studio 2015
thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
BP_Petrov_Vadim_2016.pdf	text práce ve formátu PDF
rec	záznam z testování