



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

GRANULÁRNÍ SYNTETIZÉR PRO MOBILNÍ TELEFON S NEOBVYKLÝM OVLÁDÁNÍM

GRANULAR SYNTHESIZER FOR MOBILE PHONE WITH UNUSUAL PARAMETER CONTROL

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Litera

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.

BRNO 2025



Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Tomáš Litera

ID: 247785

Ročník: 3

Akademický rok: 2024/25

NÁZEV TÉMATU:

Granulární syntetizér pro mobilní telefon s neobvyklým ovládáním

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je realizovat plně funkční prototyp nástroje, navrženého v semestrální práci.

DOPORUČENÁ LITERATURA:

[1] PUCKETTE, Miller. The theory and technique of electronic music. Hackensack, NJ: World Scientific Publishing, c2007. ISBN 978-9812700773.

[2] FORRÓ, D., Svět MIDI. Grada, Praha, 1997. 375s. ISBN 80-7169-412-6.

Termín zadání: 10.2.2025

Termín odevzdání: 3.6.2025

Vedoucí práce: doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zaměřuje na vytvoření granulárního syntetizéru, který je ovládán pomocí mobilní aplikace komunikující prostřednictvím protokolu UDP. Granulární syntéza umožňuje tvorbu a úpravu zvuků rozdělením zvukového signálu na drobné části nazývané zrna, která lze různě manipulovat a kombinovat. Tato práce je tvořena v programovacím jazyku Max.

KLÍČOVÁ SLOVA

syntetizér, granulární syntéza, Max/MSP/Jitter, UDP

ABSTRACT

This work focuses on the creation of a granular synthesizer that is controlled using a mobile application communicating via the UDP protocol. Granular synthesis allows the creation and modification of sounds by dividing the audio signal into small parts called grains, which can be manipulated and combined in various ways. This work is created in the Max programming language.

KEYWORDS

synthesizer, granular synthesis, Max/MSP/Jitter, UDP

LITERA, Tomáš. *Experimentální softwarový hudební nástroj nebo zvukový zdroj*. Semestrální práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2030. Vedoucí práce: u doc. Mgr. Ing. MgA Dan Dlouhý, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Tomáš Litera
VUT ID autora: 247785
Typ práce: Semestrální práce
Akademický rok: 2024/25
Téma závěrečné práce: Experimentální softwarový hudební nástroj nebo zvukový zdroj

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Mgr. Ing. MgA. DANU DLOUHÉMU , Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	10
1 Teoretická část	11
1.1 Syntézy a parametry	11
1.1.1 Syntéza zvuku	11
1.2 Granulární syntéza zvuku	16
1.2.1 Historie	16
1.2.2 Struktura zrn	16
1.2.3 Parametry granulární syntézy	17
1.3 Optika a její vlastnosti	18
1.3.1 Světlo	18
1.3.2 Barvy a jejich vnímání	18
1.3.3 Využití	18
1.4 Programovací prostředí <i>MAX/MSP/JITTER</i>	19
1.4.1 Historie	19
1.4.2 Popis programovacího jazyka <i>Max</i>	19
1.4.3 <i>Max</i>	20
1.4.4 <i>MSP</i>	20
1.4.5 <i>Jitter</i>	20
2 Praktická část	21
2.1 Návrh vlastního řešení	21
2.1.1 Granulární syntetizér	21
2.1.2 Vizualizace granulárního syntetizéru	25
2.1.3 Propojení <i>UDP</i> aplikace s <i>Max/MSP</i>	26
2.2 Realizace	28
2.2.1 Granulární syntetizér	28
2.2.2 Problematika <i>UDP</i> protokolu	33
2.2.3 Problematika 3D zobrazení	36
2.3 Hudební a výrazové možnosti	39
2.3.1 Flexibilita a zvukové spektrum při práci s různými vzorky	39
2.3.2 Intuitivnost ovládání zvukových parametrů pohybem mobilního telefonu	40
2.3.3 Potenciální hudební využití a aplikace	41
Závěr	43
Literatura	44

Seznam symbolů a zkratk	46
Obsah elektronické přílohy	47

Seznam obrázků

1.1	Model tvaru zvukového signálu [1]	12
1.2	Způsob tvarování signálu podle aditivní syntézy [1]	13
1.3	Způsob tvarování signálu podle subtraktivní syntézy . Pásmová filtrace se snižující se střední frekvencí.[1]	14
1.4	Způsoby tvarování signálu podle FM syntézy. poměr nosné frekvence k modulační 1:1 (napravo) a poměr nosné frekvence k modulační 2:1 (nalevo) [1]	15
1.5	Způsoby modulace signálu podle tvarovací syntézy [1]	15
2.1	Příklad objektu <i>adsr~</i> v <i>Max/MSP</i>	23
2.2	Objekt <i>metro</i> časově ovládaný pomocí objektu <i>live.slider</i>	23
2.3	Objekt <i>play~</i> ovládaný objektem <i>random</i> nastavený na hodnotu 44100 vzorků	24
2.4	Objekt <i>phasor~</i> (ovládaný hodnotou flonum) a <i>rate~</i>	24
2.5	Objekt <i>line~</i> trvající 50 ms ovládaný objektem <i>dial</i>	25
2.6	Příklad základního nastavení funkcí <i>Jitteru</i> a příkladný výsledek objektu <i>sphere</i>	26
2.7	Příklad aplikace třetí strany podporující UDP protokol	27
2.8	Zapojení subpatche <i>p grain_params</i>	29
2.9	Zapojení parametru délka výběru	31
2.10	Zapojení parametru variace času začátku	31
2.11	Zapojení parametru panorama	31
2.12	Zapojení parametru rychlost spouštění grainů	32
2.13	Zapojení parametru pitch	32
2.14	Zapojení parametru variace pitch	33
2.15	Zapojení patche granulárního syntetizéru	33
2.16	Zapojení externího balíčku CNMAT Externals s výchozími hodnotami	34
2.17	Finální zapojení UDP receiveru s filtrací hodnot a odesláním	35
2.18	Vizualizace 3D zobrazení	38
2.19	První sekce 3D zobrazení syntetizéru: 3D prostředí s jeho parametry	38
2.20	První sekce 3D zobrazení syntetizéru: dynamické řízení a zobrazení granulárních zrn	39

Úvod

Zvuk v našich životech hraje zásadní roli. Od lidské řeči, hudby, až po technické aplikace, jako je například ozvučování, nebo různé zvukové efekty. Od počátku 70. let se jednou z možností stala i granulární syntéza zvuku. Jde o rozdělení signálu na nepatrné segmenty, které se poté různými parametry upravují. Tyto parametry mohou být dále upravovány jak analogově, tak digitálně, pomocí jiných procesů.

Cílem této práce je navrhnout softwarový granulární syntetizér, který by byl ovládan pomocí fyzických senzorů mobilního zařízení. Tímto ovládním by se odlišoval od ostatních nástrojů.

Projekt bude navrhován a zpracováván v prostředí pro vizuální programování Max/MSP.

1 Teoretická část

1.1 Syntézy a parametry

1.1.1 Syntéza zvuku

V této kapitole se budeme nejprve zabývat tím, jak přesně definujeme pojem syntéza, a poté si rozebereme druhy a jejich parametry daných syntéz, a nakonec se více zaměříme na samotnou granulární syntézu, která je součástí praktického vypracování této práce.

Syntéza

Samotný pojem syntéza byl definován v Chambersově slovníku 21. století z roku 2003 jako: *skládání; spojování; vytvoření celku z částí*.

Tento proces znamená tedy spojování a vytváření celku. Není to však jen náhodné seskupování, ale jedná se o koordinovanou kombinaci jednodušších částí. S tímto tématem se můžeme setkat hlavně v prostředí při tvorbě chemických sloučenin a při produkci elektronických zvuků. Existuje však i mnoho dalších forem syntézy.

Syntéza nám může nabídnout velkou škálu různých výstupů podle našich potřeb a specifických zadání. Jejich kontrola, výběr a volba však vyžaduje lidskou zručnost a zásah.

V hudebním oboru je syntéza spojována především s oblastí elektroakustické hudby, nebo jinak řečeno, s elektronickou generací signálů různých charakterů.

Jedna z problematik syntézy tkví v poznávání obecných zákonitostí hudebního signálu, který přes sebedokonalejší analytický přístup je stále velmi náročně postihnutelný ve vztahu k procesu subjektivnímu vnímání.

Typy syntézy

Zde si vyjmenujeme typy syntézy v obecném měřítku, a základně si je definujeme. Všechny syntetizéry jsou ve své podstatě podobné. Hlavní rozdíly spočívají ve formátech jejich výstupu a v metodách, v nichž tento výstup vytvářejí. Zde je souhrn některých ze syntetizérů, které se používají:

Texturální syntetizéry se využívají v grafickém průmyslu, kde je najdeme zejména ve 3D grafice. Poté jsou zde **video syntetizéry**, které jsou určeny pro produkci a zpracování videosignálů. Využívá se například v živých audiovizuálních technikách, nebo při tvorbě speciálních efektů. Princip spočívá ve vytváření obrazové struktury, jako je generace abstraktních grafických norem, nebo vrstvení a

kombinování různých obrazových zdrojů. Jako další typ jsou **barevné syntetizéry**, které jsou používány při představeních typu *son et lumiere* (v překladu světlo a zvuk). Většinou se jedná o přidávání, změnu, nebo generování barev v rámci vizualizace. Pro příklad si zde můžeme zmínit přidávání efektů pro zvýraznění estetických aspektů, nebo světelné instalace kombinací s video syntetizéry. **Hlasové syntetizéry**, které můžeme najít v počítačových a telekomunikačních aplikacích. **Zvukové syntetizéry** jsou typem, který slouží k tvorbě a úpravě zvuků a hudby. Jako poslední zde jsou **textové syntetizéry**, které jsou specifickou kategorií, kterou je třeba chápat v kontextu širšího spektra syntetizačních technik. Na rozdíl od běžně známých syntetizérů zvuku či obrazu, textové syntetizéry pracují primárně s jazykovými daty a textem. Příkladem textového syntetizéru zde může být transformační model *GPT (Generative Pre-trained Transformer)*, který dokáže generovat text na základě vstupních dat.

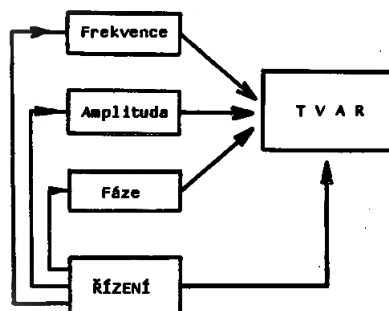
Dále se zaměříme na zvukovou syntézu a její metody. [2]

Zvuková syntéza

Zvuková syntéza je velice specifická v jejím pojetí a analýze. Proto se v oblasti analýzy zabýváme primárně kvantitativní stránkou zvukové syntézy, protože její kvalitativní hodnocení patří do oblasti psychoakustiky a zvukové estetiky.

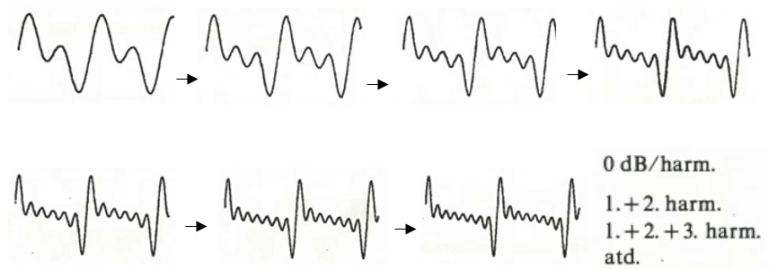
Před tím, než si budeme definovat metody zvukové syntézy je potřeba rozebrat, čím může být definován tvar průběhu hudebního signálu. Vlastní zvuková syntéza totiž vychází z analytického pohledu na jeho barvu. I když samotných metod zvukové syntézy existuje celá řada, tak všechny vycházejí z jednotného modelu působení na podobu tohoto signálu. Toto také můžeme nazvat jako: *tvar jeho časového průběhu prostřednictvím frekvence, amplitudy, fáze či přímého řízení (155)*

Tento model můžeme pozorovat na obrázku č. 1.1



Obr. 1.1: Model tvaru zvukového signálu [1]

Díky tomuto rozdělení můžeme průběh tohoto hudebního signálu popsat těmito parametry:



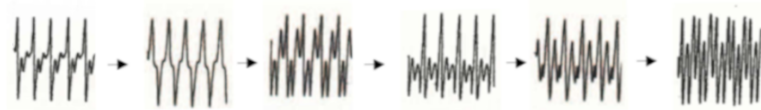
Obr. 1.2: Způsob tvarování signálu podle aditivní syntézy [1]

- frekvenční, amplitudové a fázové uspořádání jednoduchých signálů harmonického i neharmonického charakteru
- změnou frekvenčního, amplitudového a fázového uspořádání složitějšího signálu různých charakterů
- působením na frekvenci, amplitudu a fázi jednoduchého nebo složitějšího signálu různých charakterů
- působením přímo na tvar průběhu jednoduchého nebo složitějšího signálu různých charakterů.

Díky tomuto definování způsobu syntézy si můžeme vyjmenovat a blíže přiblížit tyto základní metody zvukové syntézy.[1]

Aditivní syntéza Aditivní syntéza, nebo jinak součtová metoda, je jedna z nejstarších typů zvukové syntézy. Vychází ze součtu velkého množství signálů v časové i frekvenční oblasti, například sinusových průběhů. K tomuto slouží paralelně zapojené oscilátory, kde každý generuje sinusový signál. V analogovém zapojení je jeden z velkých problémů, které se tu vyskytují, složitost při ovládnutí velkého množství sinusových vln. To usnadňují techniky digitální, které i přestože byly úspěšnější než analogové, tak zůstávají spíše specializovanou oblastí. Tento typ se využívá jako prvek i jinde než v aditivní syntéze. Na teoretické bázi můžeme říct, že součtovou syntézou jsme schopni vytvořit jakýkoliv zvuk, v reálném prostředí je to však těžší. [2]

Subtraktivní syntéza Tato syntéza, jinak nazývaná jako rozdílová, vychází z rozdílu frekvenčních charakteristik komplexního signálu a filtru. Jinak řečeno jsou založeny na řízené filtraci komplexního signálu. Svou jednoduchostí se řadí k nejrozšířenějším způsobům syntézy. Hlavní princip subtraktivní syntézy spočívá v tzv. vyřezávání částí spektra tohoto širokospektrálního signálu pomocí filtrů. Nejprve se vytvoří bohatý zvukový základ, který je generován oscilátorem. Tento oscilátor generuje základní průběhy jako je nejčastěji šum. Poté se z něj pomocí speciálních



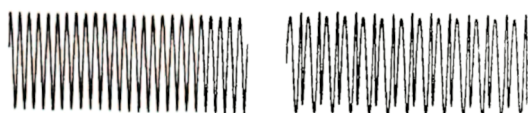
Obr. 1.3: Způsob tvarování signálu podle subtraktivní syntézy . Pásmová filtrace se snižující se střední frekvencí.[1]

filtrů formuje výsledný zvuk, který nabývá tvarů jako je pila, obdélník atd. Pomocí těchto filtrů se odstraňují určité frekvence a zdůrazňují jiné, čímž definují barvu a charakter výsledného zvuku. [2]

Syntéza AM Syntéza AM neboli amplitudová modulace, je druh modulační syntézy, kde nastává změna amplitudy nosného signálu amplitudou modulujícího signálu. V dřívějších dobách tento princip byl používán pro přenos rádiového signálu. Výsledný signál obsahuje kromě výjimek jak původní nosnou frekvenci, tak i součet a rozdíl nosné a modulačních frekvencí. Tyto složky nesou název postranní pásma. Při velmi nízkých modulačních frekvencích (například nižších než 25 Hz), je AM syntéza nazývaná tremolo. Tento jev je definován jako sled rychle se opakujících tónů, přesněji rychlá cyklická změna amplitudy. [2, 1]

Syntéza FM FM syntéza, nebo jinak řečeno frekvenční modulace, byla poprvé popsána Johnem Chowningem v roce 1973.

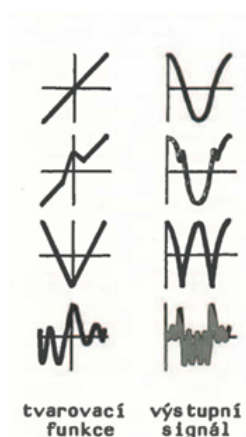
Tato syntéza se běžně využívá k přenosu rozhlasového vysílání. FM syntéza využívá signál vysoké frekvence jako „nosnou“ vlnu pro audio signál. Informace jsou přenášeny modulací frekvence nosné vlny pomocí audio signálu. Změny frekvence se zde označují jako odchylka. Na rozdíl od amplitudové modulace (AM), která typicky generuje jeden pár postranních pásem (nebo i jedno postranní pásmo v případě potlačené nosné), frekvenční modulace (FM) vytváří mnohem bohatší a rozsáhlejší spektrum postranních pásem. To je způsobeno nelineární povahou FM, kde intenzita modulace a poměr frekvencí nosné a modulační vlny (index modulace) ovlivňují počet a amplitudy těchto pásem. Tyto postranní pásma se dají vypočítat. Při nižších frekvencích modulačního signálu vzniká slyšitelný efekt vibrato, což může být definováno jako změna výšky tónu. Je typická pro zvuk smyčcových a dechových nástrojů. Při vyšších frekvencích vytvářen kovový zvuk, který je velice specifický pro tento typ modulace. [2, 1]



Obr. 1.4: Způsoby tvarování signálu podle FM syntézy. poměr nosné frekvence k modulační 1:1 (napravo) a poměr nosné frekvence k modulační 2:1 (nalevo) [1]

Tvarová syntéza Tvarová syntéza, představuje v mnoha ohledech velice odlišný pohled na samotnou tvorbu. Jsou založeny na přímém vytváření signálového průběhu, ať už pomocí nelineárního tvarování nebo přímého grafického zadání, čímž se zdánlivě vzdalují klasickému analytickému pohledu na strukturu signálu. Tyto metody jsou velice úzce spjaty s digitální technologií a zaměřují se na dosažení konkrétní podoby hudebního signálu, ať už tradičního či experimentálního, zcela odlišným způsobem. Tvarová metoda klade důraz na co největší jednoduchost hlavně v oblasti kontroly parametrů. Tyto vlastnosti jsou klíčové hlavně pro komerční využití syntézy v reálném čase, bez aplikace složitých metod součtové syntézy.

Mezi tvarovou syntézou se dá zařadit nelineární tvarování, což je založené na zkreslení sinusového signálu při průchodu obvodem s nelineární charakteristikou (tvarovací funkce). Výsledný signál závisí na tvarovací funkci, amplitudě vstupního signálu a frekvenčních poměrech použitých signálů. Jako další do tohoto tématu patří fázové zkreslení, které přehazuje vzorky v periodě signálu, díky čemuž dostáváme vznik nového průběhu signálu.



Obr. 1.5: Způsoby modulace signálu podle tvarovací syntézy [1]

1.2 Granulární syntéza zvuku

V této kapitole se budeme více do hloubky zabývat granulární syntézou. Řekneme si něco málo o její historii, principech, parametrech a v neposlední řadě o jejích metodách. Granulární syntéza obecně zahrnuje generování tisíců krátkých zvukových "zrn", která se kombinují do větších akustických struktur. Tato technika spadá do kategorie aditivní syntézy, protože výsledný zvuk je tvořen spojením mnoha jednotlivých zrn. Tento proces byl již popsán v kapitole aditivní syntéza.

1.2.1 Historie

Počátky granulární syntézy sahají až do středověkých atomistických filozofií 5. století př. n. l., kdy filozofové jako Leucippus a Demokritos teoretizovali, že hmota se skládá z nedělitelných částic. Tyto koncepty byly později použity v 17. století Descartes a Boyle. Moderní pojetí zvukových částic začalo až s Einsteinovou predikcí fononů v roce 1907, ale teprve Dennis Gibson tuto teorii aplikoval na slyšitelný zvuk v roce 1946 a tím položil základy granulární syntézy. Gabor označil jednotlivá „zrnka“ zvuku jako akustická kvanta a navrhl, že pomocí takového kvantového modelu lze popsat jakýkoliv zvuk. Tato teorie byla matematicky potvrzena až Bastiaansenem v roce 1980.[3] V roce 1959 skladatel Iannis Xenakis prozkoumal Gaborovy teorie prostřednictvím experimentů střiháním magnetofonických pásek. První implementace granulární syntézy na počítači byla až v roce 1974. Tento počítačový vývoj proměnil granulární syntézu v základní techniku v elektronické a počítačové hudbě, která se vyvinula od teoretických diskusí až po praktické aplikace, čímž vytvořila bohaté kompoziční možnosti.

1.2.2 Struktura zrn

Když jsme se zabývali aditivní syntézou, bylo zjištěno, že složité zvuky mohou být vytvářeny pomocí superpozice (přidávání) jednodušších zvukových průběhů, nejčastěji sinusových vln, které zní "nad sebou" ve frekvenční doméně. Granulární syntéza se oproti tomu neřadí mezi aditivní metody, neboť její princip nespočívá v adici vlnových průběhů ve frekvenční doméně. Místo toho pracuje s tisíci velmi krátkých, diskrétních a potenciálně se překrývajících zvukových impulsů, neboli zrn, které jsou časově řazeny za sebou, a jejichž součet tvoří výsledný zvuk. Tato zrna jsou obvykle kratší než 100 milisekund. [5] V rámci granulárních technik je zvukové zrno definováno jako konečný časový segment libovolné vlnové formy, který je modulován amplitudovou obálkou. Pokud zrno zvuku trvá velmi krátkou dobu, blíží se minimální vnímatelné časové jednotce pro diskriminaci trvání, frekvence a amplitudy. V tomto případě, pokud jednotlivá zrna mají kratší délku než 2 ms, zní jako kliknutí. Toto

odpovídá základním frekvencím pod 500 Hz. I přesto můžeme měnit barvu tónu těchto zrn pomocí změny vlnové formy nebo frekvence zrn. Tyto krátká zrna mohou být vysoce využitelná, protože když stovky krátkodobých zrn vyplní texturu obálky, tyto drobné variace mají klíčový vliv na spektrum celkového signálu. [6] Zrna mají také obálku. Tato obálka je obvykle vytvořena tak, aby začínala a končila na nulové amplitudě. Díky tomu se vyhýbáme náhlým diskontinuitám. Různé ostré změny ve výsledných vlnových formách mohou vytvářet nežádoucí harmonické a výsledek přechodu by zněl jako kliknutí. [2]

Ohledně délky zrna můžeme definovat skutečnost, že čím delší zrno, tím je čitelnost jeho výšky více rozpoznatelná. Můžeme pozorovat rozdíl, pokud je zrno dlouhé 5 ms nebo 25 ms, kdy při delším trvání bude i jasnější dojem z daného zrna.

1.2.3 Parametry granulární syntézy

Velikost zrn

Velikost zrn v granulární syntéze je jeden z klíčových parametrů, který ovlivňuje texturu a charakter vzniklého zvuku. Jak jsme si definovali v kapitole výše struktura zrn, tak zrna jsou definována jako časově konečné segmenty libovolné vlnové formy. Velikost těchto zrn se může lišit a obvykle má přímý vliv na výsledný zvukový výstup.

Menší velikost zrn nám umožňuje podrobnější manipulaci se zvukem, což vede k složitější textuře, protože jednotlivá zrna lze kontrolovat s větší přesností. Naopak větší zrna vytvářejí hladší a soudržnější zvuk, bohužel ztrácejí tím oproti menším velikostem některé detaily. Tyto velikosti se v granulární syntéze pohybují většinou od 1 ms do 100 ms. [4]

Hustota zrn

Hustota zrn je dalším klíčovým kompozičním parametrem granulární syntézy. Definiuje nám poměr zvukových událostí v daném zrnu k tichu ve zvukovém prostředí. V obecném fyzikálním termínu je hustota definována jako poměr hmotnosti k objemu. V hudebních termínech odráží množství zvuku vzhledem k absenci zvuku. Změnou tohoto parametru můžeme dosáhnout různých efektů, jako například koalescence (spojování nebo shlukování zvukových částic) nebo evaporace (rozpad nebo vyblednutí zvuku). [4]

Komplexnost textury zvuku

V kontextu granulární syntézy je tzv. průhlednost úzce spojena s hustotou mikroskopických zrn. Průhlednost nám určuje, jakým způsobem se zvukové vlny vzájemně

ovlivňují z hlediska slyšitelnosti. Vyšší hustota zvukových částic nám zde způsobuje neprůhlednou texturu. V praxi to vypadá, že zvuky mohou blokovat nebo překrývat jiné zvuky ve stejném prostoru z hlediska časového i frekvenčního.

1.3 Optika a její vlastnosti

V tomto teoretickém úvodu se zaměřujeme i na vizualizaci samotné syntézy. Proto je na místě zmínit některé ze základních informací o světle, barvách a jejich vnímání.

1.3.1 Světlo

Světlo je formou elektromagnetického záření, které je vnímáno lidským okem. Viditelné spektrum je pouze úzkou částí širšího elektromagnetického spektra, které zahrnuje také infračervené, ultrafialové a rádiové vlny. Má různé vlnové délky, které určují barvu viditelného spektra, od fialové (krátké vlny, kolem 380 nm) po červenou (dlouhé vlny, kolem 750 nm). Bílé světlo je směsí všech těchto vlnových délek. Při interakci s povrchy může docházet k absorpci, odrazu nebo lomu světla, což ovlivňuje barvu a intenzitu vnímaného obrazu. [9]

1.3.2 Barvy a jejich vnímání

Barvy vnímáme díky třem typům fotoreceptorů v sítnici oka, které se jinak nazývají čípky. Tyto fotoreceptory reagují na červené, zelené a modré světlo. Kombinací těchto tří základních barev lze vytvořit širokou škálu odstínů. V grafických systémech, včetně *Maxu*, je tento model často reprezentován jako RGB (Red, Green, Blue). Intenzitu jednotlivých složek lze nastavit na hodnoty mezi 0 a 255, čímž je možné dosáhnout přesného odstínu. [9]

1.3.3 Využití

Tyto parametry nám dávají možnosti práce s barvami v pojetí této práce. Využití se dá nalézt například ve vizualizaci zvukových parametrů pomocí barev. Intenzita zvuku může být reprezentována jasností barvy. V tomto případě by mohly být hlasitější zvuky jasnější, zatímco tišší tóny tmavší.

Dále mohou být barvy přiřazeny různým frekvenčním pásmům. Zde by mohly být nízké tóny zobrazené jako červené, střední tóny jako zelené a vysoké tóny jako modré. Toto rozdělení odpovídá i spektru světla.

V této práci mohou být světla využita i v realizaci 3D prostoru pro vizualizaci grainů. Například 3D mřížka by mohla zobrazovat pozici grainů a světlo nebo stíny dodají hloubku scény.

1.4 Programovací prostředí *MAX/MSP/JITTER*

1.4.1 Historie

Max (*Max/MSP/Jitter*) je vizuální programovací jazyk, který je zaměřený na tvorbu hudby, ale i na práci s daty, většinou související s vytvářením multimediálních aplikací. S tímto programovacím jazykem poprvé přišel Miller Puckette v roce 1985 na pařížském *IRCAM* (*Výzkumný a koordinační ústav pro akustiku a hudbu*).

Tento program začal jako nástroj nazývaný *Patcher* a nabízel grafické prostředí na Macintoshi pro vytváření interaktivních partitur. Tato verze pracovala pouze s *MIDI*, jelikož neměla možnost generovat zvuk v reálném čase.

Tento program byl poté přejmenován na *Max*. Název *Max* byl zvolen na počest Maxe Mathewse, průkopníka počítačové hudby a autora programů řady *MUSIC*.. Na začátku 90. let společnost *Opcode* uvedla *Max* na trh, na který později jeho vývojář David Zicarelli získal práva a v roce 1997 založil společnost dnes již známou jako *Cycling '74*. První vydání *Max* od společnosti *Cycling '74* zahrnovala *MSP* (*Max Signal Processing*), což přineslo možnost zpracovávat audio signály v reálném čase. [8, 10, ?]

V roce 2003 byla do *Max* přidána funkce *Jitter*, která přinesla podporu pro video a 3D grafiku. Poté v roce 2009 byl *Max* integrován pod Ableton Live a rozšířen o funkci *Gen*, která se používá pro optimalizaci výkonu.[10]

1.4.2 Popis programovacího jazyka *Max*

Základním principem programování v *Maxu* a jeho příbuzných programech je systém toku dat. Program, který je vytvořený v softwaru *Max* se nazývá patch. Tyto patche jsou dále tvořeny základními stavebními objekty, které společně vytvářejí daný program. Objekty fungují jako samostatné moduly. Tyto moduly mají funkci přijímat vstupy, generovat výstupy, nebo obojí. Data se předávají ve formě zpráv, které putují z výstupů jednoho objektu do vstupu jiného objektu, který je s ním propojený.[12, 13]

Tyto data jsou odesílána v určitý okamžik, buď na reakci na akci provedenou uživatelem, což může být kliknutí myši, *MIDI* nota atd., nebo proto, že tato akce měla nastat. Příkladem třeba metro, nebo zpoždění.[12]

Na druhou stranu *MSP* objekty jsou propojeny patchcordy podobně jako standardní objekty v *Maxu*, ale způsob jejich komunikace je odlišný. Místo přenosu zpráv vytváří *MSP* spojení mezi propojenými objekty, které se využívá k výpočtu potřebných zvukových dat v daném okamžiku. Tato struktura *MSP* objektů se označuje jako signálová síť.[12]

1.4.3 Max

Software *Max* je rozdělen na tři části, z nichž každá část je zaměřená na práci s jinými druhy dat. Tyto tři části se nazývají *Max*, *MSP* a *Jitter*.

Max se zaměřuje na komunikaci a ovládání MIDI, tvorbu uživatelských rozhraní a práci s textovými editory, nebo číselnými daty. [12]

V *Maxu* existuje několik typů zpráv, které slouží pro komunikaci mezi objekty. Těmito typy jsou *Bang*, *Int*, *Float*, *Symbol*, *List*, *Set*. Typ *Bang* je nejjednodušší typ zprávy, která znamená "vykonej akci". Obsahuje pouze čísla nula a jedna a většina objektů pak v *Maxu* na tuto zprávu reaguje provedením své primární funkce. Dalším typem je typ *Int*. Tento datový typ podporuje celá čísla a je používán jako vstup nebo argument při výpočtech a dalších operacích. Typ *Float* na rozdíl od typu *Int* podporuje celá ale i desetinná čísla. *Symbol* je libovolná kombinace znaků, které se používají jako textové informace nebo identifikátory. Informace jsou zde ukládána jako *string*. *List* je zpráva, která může obsahovat různé kombinace čísel a symbolů. *Set* nastavuje obsah zprávy bez jejího okamžitého vyslání.[12]

V programovacím prostředí *Max* mají kabely pro přenos *Max* dat šedou barvu.

1.4.4 MSP

MSP (Max Signal Processing) je další část, která *Max* rozšiřuje o schopnost práce se zvukovými signály v reálném čase. Podporuje standardní vzorkovací frekvenci od 44,1 kHz až po vyšší hodnoty, jako je třeba 192 kHz. Objekty v *MSP* pracují s 32bitovými vzorky v režimu *floating point (plovoucí desetinné čárky)*.

Objekty v *MSP* jsou označovány symbolem ~ (např. "cycle~" je objekt pro generování sinusového signálu). Funkce *MSP* probíhají paralelně s ostatními funkcemi *Maxu*, přičemž audio je řízeno pomocí interního scheduleru (centrální komponenta v prostředí *MAX* a *MSP*, která zajišťuje časovou správu všech procesů probíhajících v programu). Propojovací kabely *MSP* dat jsou přerušovaně šedé. [12]

1.4.5 Jitter

Jitter je část, která je zaměřená na zpracování videa, grafiky a 3D objektů. Umožňuje manipulovat s obrazem, animacemi a audiovizuálními zobrazeními v reálném čase. Objekty, které *Jitter* používají, mají předponu *jit*. *Jitter* pracuje s maticemi, pro které využívá *jit.matrix*, což je objekt pro ukládání a manipulaci s maticemi dat. Propojovací kabely v *Jitteru* jsou přerušovaně tmavě zeleně šedé. Pro renderování složitých dat je zde integrován OpenGL, což umožňuje tvorbu komplexních 3D scén a efektů. [12]

2 Praktická část

2.1 Návrh vlastního řešení

V rámci semestrální práce bude vytvořen teoretický návrh, který bude demonstrovat výsledný nástroj se všemi parametry. Tento návrh ve zkrácené podobě je spolu s popisem jednotlivých částí, ze kterých budou nástroje postaveny, obsažen v této kapitole. Podrobnější popis funkce jednotlivých bloků bude upřesněn v dalších kapitolách.

2.1.1 Granulární syntetizér

Pro tvorbu nástroje granulární syntézy v *Maxu/MSP* je klíčové vytvořit modulární a flexibilní patch, který umožňuje dynamické ovládání grainů, jejich parametrizaci a vizualizaci. Tento proces lze rozdělit do několika kroků. Využijeme zde jak základní objekty, tak externí nástroje dostupné v komunitě *Max*. Pro tento nástroj budou sloužit objekty jako *buffer~*, *poly~*, *metro*, *line~* nebo *adsr~*, *groove~*, *phasor~*, *rate~* a další, které budou rozebrány v dalších kapitolách.

Objekt *buffer~*

Základním stavebním prvkem je *buffer*, který ukládá zvukový vzorek a přehrává data. K tomu slouží objekt *buffer~*. Funguje jako paměťové místo, kam lze nahrát nebo uložit zvukové soubory, které mohou být následně přehrávány, modifikovány a analyzovány. Do tohoto objektu se načte vzorek o konkrétní délce. Pro přehrávání vzorku lze poté použít objekt *groove~*, který umožňuje přesné řízení přehrávání v čase. Alternativně lze použít *sfplay~* pro přehrávání externích souborů nebo *record~* pro nahrávání živého vstupu.[12]

Pro načtení vlastního zvukového záznamu stačí přidat příkaz *read*, který vyvolá dialogové okno pro výběr souboru. Případně je možné určit cestu přímo.[12]

Objekt *poly~*

Tento objekt umožňuje polyfonní přístup ke zvuku. V kontextu granulární syntézy spravuje více „hlasů“. Spravování více hlasů je ideální pro simultánní přehrávání více grainů. Objekt *poly* může být propojen s objektem *buffer* a ovládán dynamicky pomocí zpráv z hlavního patche. Polyfonie, jinak označována jako mnohozvučnost, zajišťuje, že každý grain je zpracováván nezávisle, což dodává zvuku komplexitu.[12]

Objekt *metro*

Objekt *metro* generuje pravidelné časové impulzy, které mohou být použity k řízení hustoty grainů. Často se kombinuje s objektem *bang*, aby spouštěl další operace. Příkladem může být aktivace *play* pro přehrávání jednotlivých grainů. [12]

Objekt *line~* / *adsr~*

Oba objekty vytvářejí obálky, které ovládají amplitudu grainů. Objekt *line~* produkuje plynulé lineární přechody mezi hodnotami a je užitečný pro jednoduché obálky. Objekt *adsr~* umožňuje detailnější kontrolu pomocí parametrů *attack*, *decay*, *sustain* a *release*, což je ideální pro granulární syntézu s proměnlivou dynamikou, která bude použita v tomto projektu. [12]

Objekt *groove~*

Tento objekt přehrává obsah *buffer~*. Obsahuje pokročilejší funkce jako smyčkování a časovou interpolaci. Tyto funkce umožňují plynulejší manipulaci se zvukem při granulární syntéze. V rámci této seminární práce bude využit další objekt *random*, který dynamicky náhodně určuje začínající pozici grainů. [12]

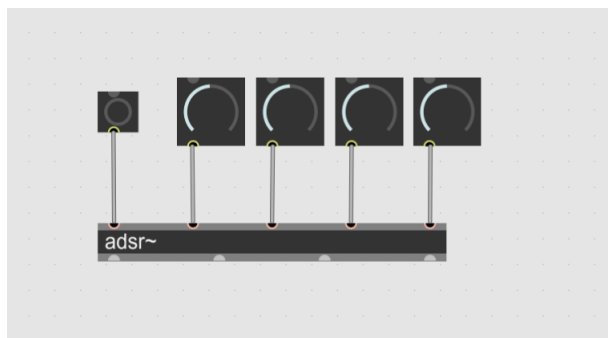
Objekty *phasor~* a *rate~*

Objekt *phasor~* generuje periodický signál, který může řídit pohyb čtecí hlavy v objektu *buffer~*. Objekt *rate~* se používá k úpravě rychlosti přehrávání. Tato změna parametru vede k efektům, jako je pitch-shifting, nebo časová komprese. Samotný pitch-shifting bude v tomto projektu využíván. [12]

Parametry granulárního syntetizéru

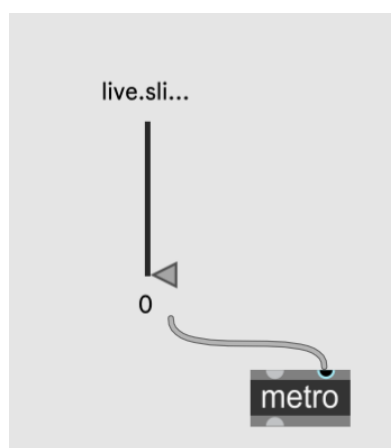
Tvar obálky grainů Tvar obálky určuje, jak hladce grain začíná a končí. Pro toto zapojení bude použit objekt *adsr~*. V tomto objektu jsou zahrnuty čtyři fáze obálky řídící průběh amplitudy. Těmito fázemi jsou *attack*, *decay*, *sustain*, *release*. *Attack* je doba, během které signál narůstá z nuly na maximální hodnotu. Nastavení hodnoty je typicky 1–50 ms (pro granulární syntézu velmi krátká hodnota pro rychlý náběh). *Decay* je čas potřebný k poklesu signálu z maximální hodnoty na úroveň *sustain*. Typické hodnoty jsou mezi 1–100 ms (delší *decay* zjemňuje zvuk grainu). *Sustain* je hladina, na které signál zůstane po dobu trvání grainu. Hodnota se nachází v rozmezí 0–1, kdy 0 je úplný pokles, a 1 je udržení maximální amplitudy. *Release* je doba, během které signál poklesne z úrovně *sustain* na nulu po ukončení grainu. Hodnota se nachází obvykle mezi 10–50 ms pro přirozené zakončení.

Objekt `adsr` musí být spuštěn pomocí objektu jako `metro` nebo jiných triggerů. Demonstrující obrázek ukazuje objekt `adsr~` spouštěný objektem `button` a ovládaný potenciometry `dial`. [12]



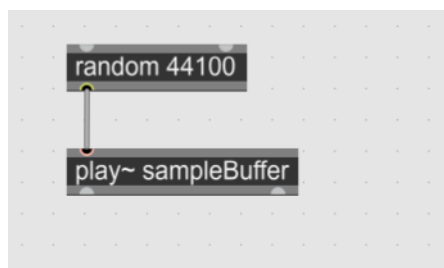
Obr. 2.1: Příklad objektu `adsr~` v `Max/MSP`

Hustota grainů Hustota určuje počet grainů přehrávaných za sekundu. Nižší hustota generuje oddělené zvuky, zatímco vyšší hustota vytváří plynulou zvukovou masu. Pro tento parametr bude využit objekt `metro`, který generuje pravidelné spouštěcí impulsy pro vytváření grainů. Například příkaz `metro 50` znamená, že jeden grain se bude generovat každých 50 ms. Tato hodnota je možná upravovat i dynamicky pomocí různých objektů, například `live.slider`. V případě této práce bude tento parametr ovládán hodnotami získávanými z mobilního zařízení pomocí UDP protokolu. [12]



Obr. 2.2: Objekt `metro` časově ovládaný pomocí objektu `live.slider`

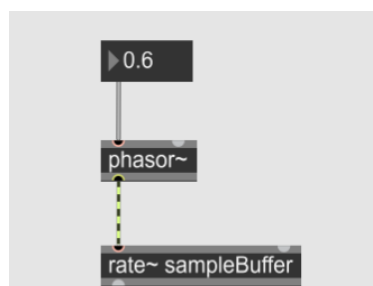
Počáteční pozice v bufferu Počáteční pozice v bufferu určuje část zvukového vzorku, která bude použita pro daný grain. Pozice může být statická nebo náhodně



Obr. 2.3: Objekt *play~* ovládaný objektem *random* nastavený na hodnotu 44100 vzorků

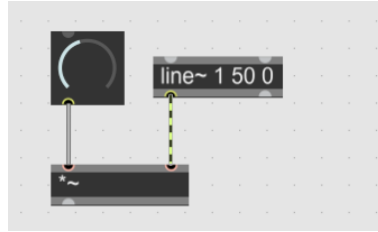
generovaná. Pro tento parametr bude použit objekt *play~*, který bude ovládán objektem *random*. Tento randomizátor vybírá hodnoty odpovídající délce vzorku uloženého v bufferu. [12]

Frekvence grainů Frekvence ovlivňuje rychlost přehrávání grainu, což má přímý vliv na jeho výšku. Toto vytváří efekt nazývaný pitch-shifting. K tomuto parametru bude využit objekt *phasor~* a *rate~*. Hodnota objektu *phasor* určuje, zda bude přehrávání zrychlené (při hodnotách větších než 1), zpomalené (při hodnotách od 0 - 1), nebo obrácené (hodnoty menší než 0). Výchozí hodnota je zde 1, kdy při této hodnotě je výška přehrávání stejná jako v originále a nijak se nemění. [12]



Obr. 2.4: Objekt *phasor~* (ovládaný hodnotou flonum) a *rate~*

Délka grainů Délka grainu určuje časový rozsah jednoho grainu. Typické hodnoty se pohybují mezi 10–100 ms. Krátké grainy mají za následek textury podobné šumu, zatímco delší grainy připomínají originální zvuk. Pro tento parametr bude využit objekt *line*, který se používá k vytvoření obálky, která řídí průběh amplitudy grainu od jeho začátku po jeho konec. Tato hodnota se dá ovládat dynamicky. Pro demonstraci je použit objekt *dial*. [12]



Obr. 2.5: Objekt *line~* trvající 50 ms ovládaný objektem *dial*

2.1.2 Vizualizace granulárního syntetizéru

Pro vizualizaci grainů z granulárního syntetizéru v *Max/MSP* využíváme nástroje z *Jitter*, které umožňují zpracovávat a zobrazovat data v grafické podobě. V následujících kapitolách budou rozebrány některé objekty, které jsou pro toto zobrazení potřebné.

Objekt *jit.pwindow~*

Tento objekt je základním prvkem pro zobrazení maticových dat v *Maxu*. Tento objekt je klíčový pro jednoduchou vizualizaci a monitorování datových proudů. Tento objekt však vykreslí námi zadané hodnoty pouze ve 2D. [12]

Objekt *jit.gl.mesh*

Objekt *jit.gl.mesh* je specializovaný na vykreslování maticových dat ve formě 3D geometrie. Využití v granulárním syntetizéru je, že umožňuje vizualizovat grainy jako body, čáry nebo povrchy. Matici s daty grainů lze použít k určení pozice, barvy nebo velikosti těchto prvků. [12]

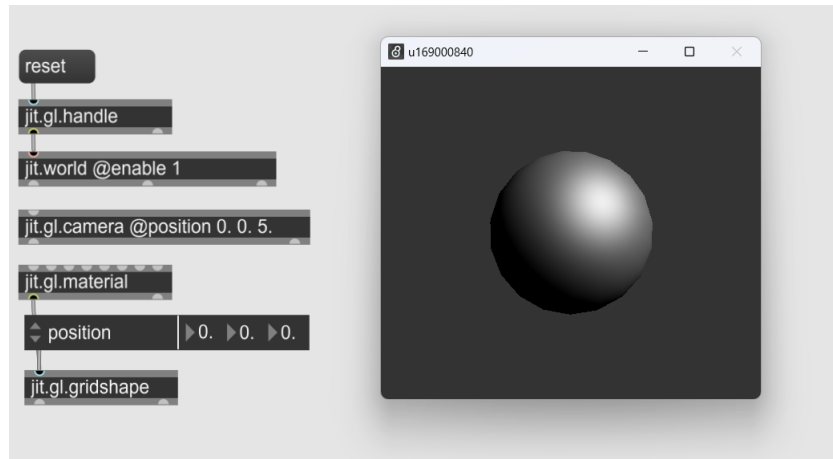
Objekt *jit.world*

Tento objekt kombinuje několik klíčových funkcí, které usnadňují práci s OpenGL grafikou a vizualizacemi. Poskytuje kontext pro vykreslování (rendering) a spravuje zobrazení veškerých grafických výstupů na obrazovku, což zahrnuje 3D modely, textury a další grafické prvky.

Realizace vizualizace

Data z granulárního syntetizéru je potřeba převést na vizualizovaný formát. Každý parametr granulárního syntetizéru se mapuje na datové struktury v *Jitteru* pomocí objektu *jit.matrix*. Dále pak tyto parametry mohou být dynamicky propojeny s barvami nebo pohybem objektů ve 3D prostoru.

Hlavní grafické okno pro vykreslování 3D prostředí se vyvolá pomocí objektu *jit.world*. Pro vykreslení komplexních objektů, které budou znázorňovat jednotlivé grainy, se použije objekt *jit.gl.mesh*. Dále je potřeba, aby scéna byla dostatečně osvětlená. K tomu slouží objekt *jit.gl.light*. Pozice zobrazení se nastavuje pomocí *jit.gl.camera*.



Obr. 2.6: Příklad základního nastavení funkcí *Jitteru* a příkladný výsledek objektu *sphere*

2.1.3 Propojení *UDP* aplikace s *Max/MSP*

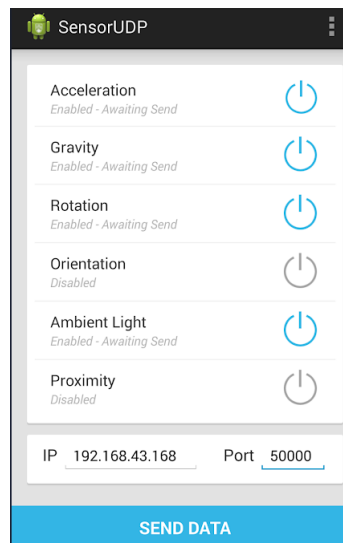
Protokol *UDP* (*User Datagram Protocol*) je komunikační protokol používaný pro přenos dat přes síť v podobě paketů. Tento protokol je bezspojový, což znamená, že před odesláním dat nevyžaduje navázání spojení. Výhody *UDP* jsou nízká latence, jednoduchost a vhodnost pro real-time aplikace, protože zde neprobíhá žádná fáze potvrzení. Kvůli rychlému zpracovávání dat v reálném čase je zde částečná ztráta paketů. Mezi další výhody *UDP* protokolu patří podpora multicastu. Tato podpora umožňuje odesílání dat více příjemcům zároveň. Využití této podpory lze uplatnit například pro služby přenosu času nebo vysílání dat pro širokou veřejnost. [14]

Každý *UDP* datagram má 8 bajtů dlouhou hlavičku. Dohromady 64 bitů, které jsou rozděleny na několik částí. Každá z těchto částí má velikost 16 bitů. První část zprávy je Source Port, což je port odesílatele. Druhá část je Destination Port, což je cílový port. Třetí část je celková délka datagramu, která se skládá z hlavičky a samotných dat. Poslední část je Checksum, což zahrnuje kontrolu integrity dat. Ta je volitelná v *IPv4*, nebo povinná v *IPv6*.

Za hlavičkou následují samotná uživatelská data, jejichž délku určuje pole *Length*.

Integrace UDP pro vzdálené připojení

Pro integraci propojení UDP slouží objekt *udpreceive*, který umožňuje přijímat parametry z externí aplikace, například z mobilního zařízení přes síť. Tyto hodnoty mohou již dále ovládat námi zadané parametry. V případě této práce jsou těmito parametry hustota grainů, délka grainů, randomizace a rozložení v bufferu. Na portu objektu *udpreceive* bude nastaveno stejné číslo, které je nastaveno ve vysílací aplikaci. Data z tohoto portu budou následně přijímána do daného objektu. Tato data lze přiřadit pomocí objektů jako jsou *route* nebo *unpack* pro předem specifikované parametry granulární syntézy. [12]



Obr. 2.7: Příklad aplikace třetí strany podporující UDP protokol

2.2 Realizace

Cílem této kapitoly je podrobný popis vytvořeného hudebního nástroje, jeho částí, popis jeho funkcí, parametrů a jeho ovládacích prvků. Části, kterým se bude tato kapitola zabývat, budou granulární syntetizér, 3D zobrazení a propojení protokolu UDP.

2.2.1 Granulární syntetizér

Primárním zdrojem zvukového materiálu pro proces granulace je objekt *buffer~*. Ten slouží k načtení a uchování celého zvukového souboru v paměti. Pro načtení souboru do bufferu se stará subpatch *p buffer*. Velikost a délka bufferu jsou pak nastaveny podle délky samotného zvukového souboru.

Generování jednotlivých zvukových zrn je řízeno dynamickými parametry, které definují jejich akustické charakteristiky a umístění v čase vzhledem ke zdrojovému materiálu. Impulzy pro spouštění nových zvukových zrn vytváří objekt *metro*, který funguje jako časovací mechanismus, generující periodické triggery. Frekvence objektu *metro* přímo koreluje s hustotou zrn v čase.

Subpatch *p grain_params* Generování dynamických parametrů pro každou jednotlivou instanci zvukového zrna probíhá v rámci subpatche *p grain_params*. Tento subpatch je klíčovým modulem, který na základě vstupních řídicích hodnot a vnitřních algoritmů zahrnujících randomizaci určuje specifické charakteristiky každého nově spuštěného zrna. Konkrétně jeho startovní pozici v rámci zdrojového audio bufferu, faktor výškové transpozice a dobu trvání.

Subpatch je spouštěn externím triggerem z objektu *metro*, jak bylo popsáno dříve, který signalizuje požadavek na spuštění nového zrna. Objekt *t b b* rozděluje tento spouštěcí signál do dvou paralelních větví zpracování, zodpovědných za výpočet startovní pozice grainů v levé části a výškové transpozice a doby trvání zrna v pravé části.

Levá větev zpracování se zaměřuje na určení počátečního bodu čtení v hlavním audio bufferu pro dané zrno začíná generováním náhodného čísla objektem *random 1000*, které vytvoří celočíselnou hodnotu v rozsahu 0 až 999. Tato hodnota je následně škálována. Sekvence objektů *+ 1. a * 0.001* ji konvertuje na hodnotu v rozsahu 0.001 až 1.0. Objekt *scale 0. 1. -1. 1.* dále transformuje tento rozsah 0 až 1 na symetrický rozsah -1.0 až 1.0. Tato hodnota v rozsahu -1 až 1 reprezentuje náhodný faktor odchylky od centrální pozice.

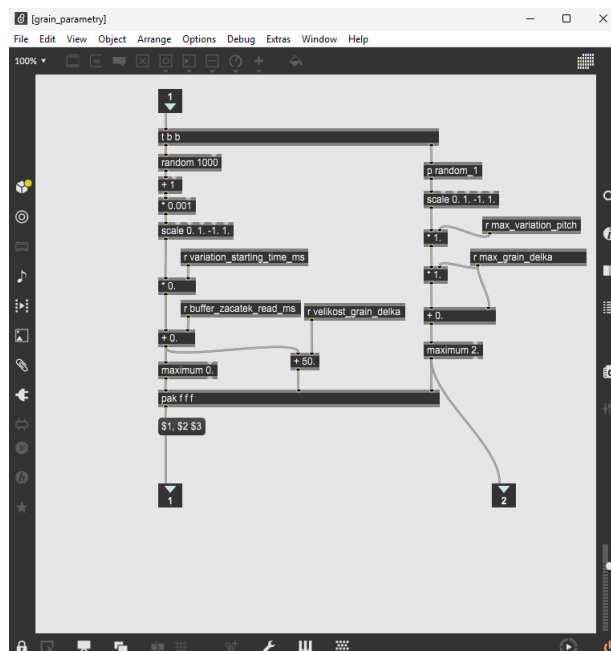
Tento náhodný faktor je poté vynásoben hodnotou přijatou z externího zdroje přes objekt *r variation_starting_time_ms*. Tato hodnota definuje maximální povolenou variaci startovní pozice v milisekundách. Výsledkem násobení je náhodný

offset startovní pozice v rozsahu.

K tomuto vypočtenému náhodnému offsetu je přičtena základní startovní pozice zrna přijatá přes objekt *r buffer_zacatek_read_ms*. Výsledkem sčítání je finální vypočtená startovní pozice pro dané zrno. Objekt *maximum 0*. na konci této větve zajišťuje, že vypočtená startovní pozice nebude nikdy záporná, čímž efektivně zabraňuje pokusu o čtení dat před začátkem bufferu.

Pravá větev zpracování se zabývá určením pitch shift (výškové transpozice) a doby trvání pro zrno. Tato větev sdílí stejný náhodný faktor odchylky, který je využit v levé větvi pro oba parametry.

Výpočet výškové transpozice se vypočítává za pomoci náhodného faktoru v rozsahu -1 až 1 a je vynásoben hodnotou přijatou přes objekt *r max_variation_pitch*. Tato hodnota určuje maximální míru povolené výškové odchylky. Výsledek násobení představuje náhodnou výškovou odchylku pro dané zrno. Objekt *maximum 2*. aplikuje omezení na tuto vypočtenou hodnotu, zajišťující, že výšková odchylka nepřekročí hodnotu 2 (v jednotkách, které jsou následně interpretovány objektem *mc.play* pro řízení rychlosti přehrávání). Výpočet doby trvání zrna využívá ten samý náhodný faktor v rozsahu -1 až 1 a je také vynásoben hodnotou přijatou přes objekt *r max_grain_delka*, která definuje maximální povolenou variaci v délce zrna. Výsledkem je náhodná odchylka doby trvání. Tato odchylka je poté přičtena k základní délce zrna, která je přijata přes objekt *r velikost_grain_delka*. Suma základní délky a náhodné odchylky představuje finální dobu trvání pro aktuální zrno.



Obr. 2.8: Zapojení subpatche *p grain_params*

Multichannel objekty Vzhledem k polyfonní struktuře, kde může současně znít mnoho zrn, je klíčové směřování správných parametrů ke správné instanci zrna. Objekt *mc.voices @voices 100* spravuje polyfonii, alokující a řídící až 100 nezávislých kanálů. Objekt *mc.target* umožňuje adresovat a posílat zprávy s parametry konkrétnímu hlasu. Objekt *mc.line~* se pak často používá pro generování plynulých přechodů v řídicích parametrech (např. pro plynulý náběh amplitudy nebo změnu výšky tónu) po dobu trvání zrna. Tyto řídicí signály jsou posílány do objektu *mc.play~*, který je interpretuje.

Každý grain je po objektu *mc.target* dynamicky tvarován v čase. V kontextu tohoto projektu je využita standardní ADSR obálka objektu *mc.adsr~*. Tato obálka se skládá ze čtyř fází, kterými jsou attack, decay, sustain a release. Uživatelské nastavení parametrů ADSR obálky je vytvořeno prostřednictvím dedikovaného vizuálního prvku s názvem ADSR obálka. Tento prvek sdružuje čtyři rotační ovladače objektu *dial*, které umožňují nezávisle nastavit trvání všech čtyř fází. Hodnoty z objektu *dial* jsou generovány ve výchozím rozsahu 0 až 127. Tyto hodnoty jsou převáděny za pomoci objektů *scale* připojených k výstupu ovládacích prvků na rozsahy relevantní pro časové a úroňové parametry ADSR obálky následovně:

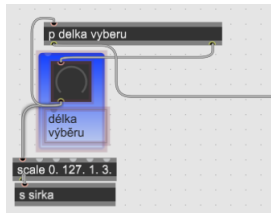
- hodnoty pro parametr attack jsou převedeny na rozsah 0 až 20, kde tyto hodnoty představují dobu trvání fáze v milisekundách,
- hodnoty parametru decay jsou škálovány na rozsah 0 až 10, kde stejně jsou hodnoty udávány v milisekundách,
- hodnoty parametru sustain jsou škálovány na rozsah 0 až 1, kde tato hodnota definuje úroveň fáze jako faktor amplitudy (0 = ticho a 1 = plná amplituda),
- hodnoty parametru release jsou škálovány na rozsah 0 až 20 v milisekundách.

Objekt *mc.adsr~* vyžaduje spouštěcí signál pro zahájení generování obálky. V tomto projektu je tímto zastoupen objekt *mc.target*, který při každém přidělení specifického hlasu vyšle signál, což automaticky spouští obálku ADSR pro příslušný grain.

Problematika nastala, kdy při aktivaci ADSR obálky byl v audio výstupu slyšitelný hluk na nízkých kmitočtech. Tento problém byl vyřešen za pomoci highpass filtru bloku *onepole~* zapojeném za ADSR obálkou.

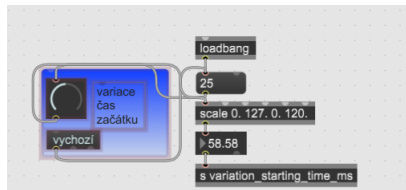
Parametry syntetizéru Každý z parametrů syntetizéru má specifické nastavení hodnot.

Délka výběru v samplu nastavuje základní dobu trvání jednotlivých grainů. Krátké grainy znějí více perkusivně, zato delší grainy znějí jako krátké tóny nebo segmenty původního zvuku. Maximální hodnota délky je nastavena pomocí bloku *r audio_delka_ms*, což je samotná délka samplu v bufferu.



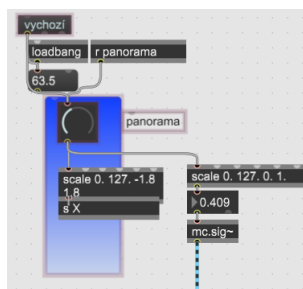
Obr. 2.9: Zapojení parametru délka výběru

Parametr variace času začátku řídí míru randomizace aplikované na startovní pozici, ze které je grain čten v hlavním audio bufferu. Jak bylo popsáno v subpatchi *p grains_params*, kde k základní startovní pozici je přičten náhodný offset. Tento parametr určuje maximální rozsah tohoto náhodného offsetu v milisekundách. Vyšší hodnota variace vede k většímu rozptylu startovních pozic zrn v rámci zdrojového zvuku, což může vytvořit chaotičtější nebo rozostřenější zvukovou texturu.



Obr. 2.10: Zapojení parametru variace času začátku

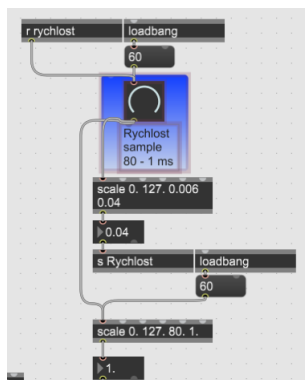
Parametr panorama kontroluje umístění jednotlivých zrn ve stereofonním zvukovém poli. Typicky se hodnoty pohybují od -1, což znamená zcela vlevo, přes 0, znamenající střed, do +1, což udává hodnoty zcela vpravo. V tomto projektu je prvek škálován z rozsahu 0 až 1227 na hodnoty 0 až 1, což představuje posuv od levého kanálu k pravému.



Obr. 2.11: Zapojení parametru panorama

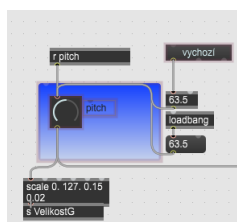
Rychlost spouštění grainů určuje hustotu nových zvukových samplů v čase. Rychlejší spouštění zrn vede k hustšímu, někdy až souvislému zvuku, kdežto pomalejší

spouštění k řidšímu, zrnitějšímu zvuku. Dynamický prvek *dial* je zde škálován z hodnot 0 až 127 na hodnoty 80 až 1. Hodnoty jsou záměrně v opačném pořadí než běžné rozsahy, z důvodu nastavení hustoty nejprve na řidší a až poté na hustší. Tyto hodnoty jsou představovány v milisekundách.



Obr. 2.12: Zapojení parametru rychlost spouštění grainů

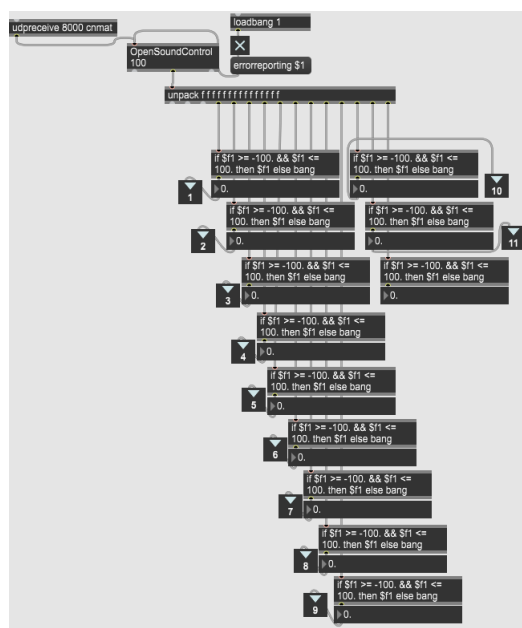
Pitch parametr řídí základní výšku tónu, se kterou jsou grainsy přehrávány. V granulární syntéze je změna výšky tónu typicky realizována změnou rychlosti čtení audio segmentu bufferu, ze kterého je zrno generováno. Vyšší rychlost čtení vede k vyššímu tónu a zkrácení zrna, nižší rychlost čtení k nižšímu tónu a prodloužení zrna. Rozsah je zde škálován na hodnoty 0.1 až 2. Tento rozsah představuje faktor rychlosti přehrávání, kde 1.0 je původní rychlost, hodnoty pod 1.0 snižují výšku a nad 1.0 ji zvyšují.



Obr. 2.13: Zapojení parametru pitch

Parameter variace pitch podobně jako u startovní pozice, tento parametr kontrolyje míru randomizace aplikované na výšku tónu každého zrna. Určuje maximální rozsah náhodné odchylky od základní výšky tónu nastavené parametrem *pitch*. Vyšší hodnota vede k většímu rozptylu výšek tónů mezi jednotlivými zrny, což může vytvořit sborový efekt, metalický zvuk nebo šumovou texturu, v závislosti na velikosti variace.

Celkový návrh realizace patche granulárního syntetizéru je vidět na obrázku 2.15.



Obr. 2.17: Finální zapojení UDP receiveru s filtrací hodnot a odesláním

2.2.3 Problematika 3D zobrazení

Vizualizace granulárního syntetizéru ve 3D prostoru je důležitou součástí projektu a je implementována v samostatném subpatchi s názvem *gran_3d*. Tento subpatch slouží jako dedikované prostředí pro generování a vykreslování grafické reprezentace jednotlivých zrn (grains) granulární syntézy v reálném čase. Architekturu subpatche lze koncepčně rozdělit do dvou hlavních sekcí: vytvoření a správa samotného 3D grafického prostředí a dynamické řízení a zobrazení jednotlivých zrn, reprezentovaných jako 3D objekty.

Sekce vykreslení 3D scény

Tato část subpatche *gran_3d* se zaměřuje na nastavení základních parametrů 3D scény. Zahrnuje konfiguraci vykreslovacího enginu Jitter pomocí objektu *jit.world*, který je zodpovědný za správu celého vykreslovacího procesu.

Součástí scény je také virtuální kamera (*jit.gl.camera*), definující pohled pozorovatele do 3D prostoru, a osvětlení (*jit.gl.light*), které ovlivňuje vzhled a stínování zobrazených objektů. Tyto prvky tvoří základní rámec, ve kterém jsou zrna následně vykreslována. Nastavení těchto objektů (například pozice kamery, směr a intenzita světla) určuje vizuální dojem z celé scény. Vykreslovaný obraz je typicky zobrazován v samostatném okně nebo panelu, který automaticky generuje samotný objekt *jit.world*.

Tyto nastavení jsou vidět na obrázku 2.10.

Sekce dynamické řízení a zobrazení granulárních zrn

Druhá, a dynamicky složitější část, se věnuje samotným zrnům. Každé zrno je vizuálně reprezentováno jako 3D koule, vytvořená pomocí objektu *jit.gl.gridshape*. Pro efektivní správu a vykreslování velkého počtu zrn je využíván objekt *jit.gl.multiple*. Tento objekt umožňuje vykreslit mnoho instancí jednoho geometrického tvaru, v našem případě koule, na základě dat uložených v argumentu tohoto objektu. Tento objekt slouží jako centrální datový sklad pro informace o každém zrně, jako je jeho 3D pozice (x, y, z), velikost, barva a případně další vizuální atributy. Samotná 3D pozice a velikost jsou dále upravovány objekty popsány níže.

Řízení pohybu zrn

Pohyb jednotlivých kuliček v 3D prostoru je řízen externě přijímanými daty a vnitřním generováním komplexních trajektorií. Hlavním zdrojem externího řízení jsou hodnoty přijímané bezdrátově, například z mobilního telefonu. Tyto hodnoty jsou mapovány na různé parametry ovlivňující pohyb a vizuální vlastnosti zrn. Kromě

toho je integrováno i vizuální ovládání přímo v prostředí *Max/MSP* pomocí objektů *live.object* (např. slidery, číselná pole), které umožňují fyzickou manipulaci s řídicími hodnotami v rámci patche.

Generování organického a nepředvídatelného pohybu je realizováno pomocí tří instancí objektu *jit.mo.perlin*. Objekt *jit.mo.perlin* je součástí sady Jitter Motion a je navržen pro generování *Perlinova šumu*. Každá ze tří instancí *jit.mo.perlin* je dedikována pro řízení pohybu zrn podél jedné z hlavních os x, y, z 3D prostoru .

Klíčovým aspektem pro dosažení individuálního pohybu každého zrna je využití atributu *@phase* u objektů *jit.mo.perlin*. Atribut *@phase* efektivně posouvá čtení Perlinova šumu v čase nebo prostoru pro každé zrno zvlášť. Tím se zajistí, že i když jsou všechna zrna řízena stejnými objekty *jit.mo.perlin*, fáze šumu, která ovlivňuje jejich konkrétní pozici, se pro každé zrno liší. Výsledkem je, že se kuličky pohybují nepravidelně a nezávisle na sobě, čímž se vizualizace stává dynamičtější a méně uniformní. Hodnoty generované objekty *jit.mo.perlin* jsou následně aplikovány na pozice zrn uložené v objektu *jit.gl.multiple*.

Toto zapojení je vidět na obrázku 2.11.

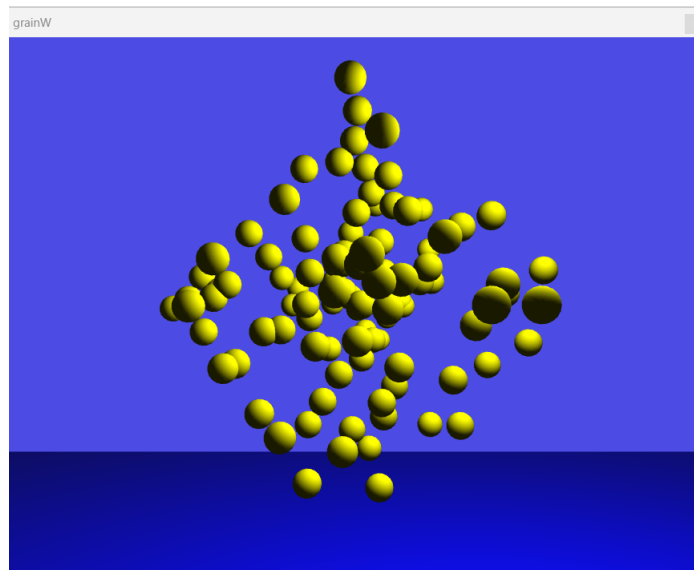
Barevné nastavení vizualizace Grafická reprezentace byla navržena s ohledem na jasnou vizuální diferenciaci mezi hlavními prvky scény.

Samotná zvuková zrna, která jsou v 3D prostoru reprezentována jako objekty koule, jsou vykreslena v jasném, výrazném žlutém odstínu. Tato volba barvy byla vybrána z důvodu zvýšení jasů a pozornosti, což vhodně symbolizuje aktivní a dynamické zvukové elementy. Technicky je tato barva definována v objektu *jit.gl.gridshape* atributem *@color 1 1 0 1*, což odpovídá plné intenzitě červené (1) a zelené (1) složky s nulovou modrou (0) a plnou průhledností (1), čímž vzniká sytá žlutá. Díky hladkému stínování (*@smooth_shading 1*) a zapnutému osvětlení (*@lighting_enable 1*) získávají koule hloubku a plastičnost, což zvyšuje vnímání jejich prostorového rozmístění a pohybu.

Kontrastně k žlutým zrnům je 3D prostředí, ve kterém se zrna pohybují, ztvárněno v modrých odstínech. Konkrétně horní část pozadí je vykreslena v sytější modré, zatímco spodní část je tmavší modrá, s náznakem horizontální linky, která naznačuje dělicí rovinu. Modrá barva pozadí poskytuje stabilní vizuální základ, který efektivně kontrastuje s dynamikou žlutých zrn. Tato volba modrého pozadí pomáhá "vysunout" žluté objekty do popředí díky výběru komplementárních barev.

Propojení dat a vizualizace

Data z externích zdrojů jsou před aplikací na zrna zpracována. To zahrnuje škálování a filtrování hodnot pomocí standardních *Max* objektů (např. *scale*, *filter*). Tyto

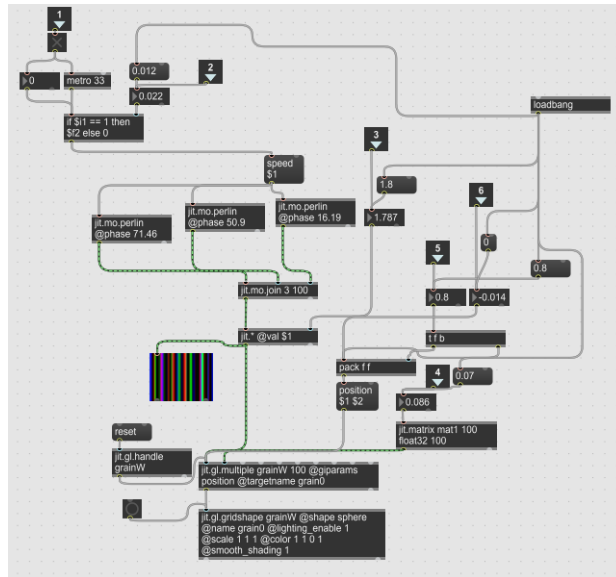


Obr. 2.18: Vizualizace 3D zobrazení

zpracované hodnoty jsou pak formátovány a ukládány pro každé jednotlivé zrno v objektu *jit.gl.multiple*.



Obr. 2.19: První sekce 3D zobrazení syntetizéru: 3D prostředí s jeho parametry



Obr. 2.20: První sekce 3D zobrazení syntetizéru: dynamické řízení a zobrazení granulárních zrn

2.3 Hudební a výrazové možnosti

Tato kapitola se věnuje hloubkové analýze hudebních a výrazových možností experimentálního granulárního syntetizéru. Cílem je demonstrovat nejen technickou funkčnost implementace, ale především obhájit smysl a přínos vytvořeného nástroje pro současnou hudební tvorbu a experimentální přístupy k zvuku.

2.3.1 Flexibilita a zvukové spektrum při práci s různými vzorky

Díky schopnosti granulární syntézy rozkládat a rekombinovat zvukový materiál na mikroskopické úrovni dává nástroji obrovskou flexibilitu. Vložení různých typů zvukových vzorků (samplů) do objektu *buffer* odhaluje široké spektrum zvukových možností.

Perkusivní vzorky

Vložením krátkých, úderných zvuků, např. bicích, kliknutí, šumů, a následnou manipulací s parametry grainů, zejména délka zrna, hustota spouštění a variace pozice, lze generovat nové, komplexní rytmické textury, glitch efekty nebo šumové plochy s jemnými vnitřními pohyby. Krátká, hustě se překrývající zrna z perkusivních vzorků mohou vytvářet bzučivé tóny nebo "drone" zvuky, zatímco řidší uspořádání vede k mikro rytmům.

Melodické a harmonické vzorky

Použití delších tónů, vokálů nebo nástrojových frází jako zdrojového materiálu umožňuje transformovat původní zvuk do éterických, rozmazaných ploch, "frozen" akordů nebo plovoucích atmosfér. Manipulací s parametry *pitch* a *Variace pitch* lze vytvářet mikrotonální posuny, harmonické akordy z jediného tónu nebo bohaté sborové efekty. Změna rychlosti spouštění zrn zde hraje klíčovou roli v plynulosti a transparentnosti výsledné textury.

Atmosférické a dronové vzorky

Vložení ambientních nahrávek, šumů nebo dlouhých dronů jako zdrojového materiálu rozšiřuje potenciál pro vytváření hlubokých, texturálních zvukových krajin. Granulární proces může tyto vzorky zahušťovat, rozšiřovat, nebo je rozkládat na jemné, třpytivé detaily, které v původní podobě nebyly slyšitelné. Experimentování s parametrem *Variace času začátku* zde umožňuje prozkoumávat různé části původního vzorku s každým zrnem, což vede k živějším texturám.

2.3.2 Intuitivnost ovládání zvukových parametrů pohybem mobilního telefonu

Jedním z nejdůležitějších experimentálních aspektů tohoto nástroje je jeho netradiční řízení prostřednictvím sensorických dat z mobilního telefonu, za pomoci akcelerometru nebo gyroskopu, odesílaných pomocí protokolu UDP. Tato metoda ovládání přináší do procesu zvukové syntézy bezprecedentní úroveň intuitivnosti.

Přímá fyzická interakce

Namísto manipulace s virtuálními slidery uživatel přímo interaguje se zvukem skrze pohyb svého těla a telefonu v prostoru. To vytváří hlubší a přirozenější propojení mezi gestem a zvukovým výsledkem.

Naklánění telefonu plynule mění výšku tónu nebo jeho rotace ovlivňuje hustotu zrn. Tato nastavení jsou rozebrána detailně níže.

Vícerozměrná modulace

Senzory mobilního telefonu poskytují data ve třech osách, což umožňuje současnou modulaci více parametrů syntetizéru jedním senzorem. Osy orientace byly mapovány na parametry rychlost (hustotu) sample, délku výběru v bufferu a variaci pitch. Rychlost sample byla přidělena na *roll*, což je otáčení telefonu doprava a doleva. Délka výběru byla přidělena funkci *pitch*, což znamená rotaci telefonu dopředu a

dozadu. Parametr variace pitch byl přidělen k funkci *azimuth*, což je otáčení telefonu dokola.

Osy rotace byly mapovány na parametry výběr v bufferu, pitch a panorama. Parametr výběr v bufferu byl přidělen k rotaci osy X, parametr pitch byl přidělen k ose Y a parametr panorama byl přidělen k ose Z.

Tato vícerozměrná kontrola umožňuje složité a organické zvukové transformace, které by byly s tradičními MIDI kontrolery obtížně dosažitelné.

Dynamické a organické přechody

Plynulost dat ze sensorů by měla vést k velmi hladkým a organickým zvukovým přechodům, i přesto některé hodnoty ze sensorů byly přijímány ve skokových hodnotách a vyhlazovány v postprocessingu. Tyto skokové změny jsou typické pro digitální rozhraní. To propůjčuje zvuku živost a výraznou expresivitu, která je vlastní akustickým nástrojům.

Performativní potenciál

Ovládání pohybem transformuje syntetizér z pouhého nástroje na generování zvuku v performativní médium. Pohyb se stává součástí uměleckého vyjádření, což otevírá nové možnosti pro improvizaci, živá vystoupení a interaktivní instalace.

2.3.3 Potenciální hudební využití a aplikace

Vzhledem k výše popsaným charakteristikám má tento experimentální granulární syntetizér široké spektrum potenciálního hudebního využití napříč různými žánry a uměleckými formami.

V experimentální a ambientní hudbě je zde schopnost vytvářet rozsáhlé zvukové plochy, drony, textury evolvující zvukové krajiny, které činí nástroj ideálním pro ambientní, drone a experimentální žánry.

Dynamické ovládání pohybem umožňuje vytvářet neustále se proměňující a nepředvídatelné zvukové události.

Ve filmové hudbě a zvukovém designu je pro tvorbu atmosférických zvukových efektů, abstraktních znělek, podkresů pro scény s napětím nebo neobvyklých zvukových krajin nástroj velmi cenný. Možnost transformovat jakýkoli vzorek do nečekaných zvukových forem je pro sound designéry vysoce atraktivní.

V živé elektronické hudbě může syntetizér sloužit jako dynamický prvek pro přidávání komplexních textur, rytmických modulací nebo abstraktních sólo partů, které reagují na gesto performerů. V rámci improvizace nabízí rychlou a intuitivní odezvu.

Tento nástroj by mohl po dalším vývoji být spárováný s dalšími zařízeními, kde pohyb publika pomocí těchto spárovaných zařízení může ovlivňovat zvukovou krajinu.

Pro edukaci a výzkum může díky transparentní implementaci v *Max/MSP* a možnosti vizuálního zobrazení zrn sloužit nástroj jako vynikající platforma pro edukaci o principech granulární syntézy a interakci se zvukem.

Závěr

Cílem práce bylo vytvoření experimentálního softwarového nástroje granulární syntézy s neobvyklým způsobem ovládání parametrů. Tyto parametry budou ovládány pomocí gyroskopu a akcelerometru mobilního zařízení.

Práce obsahuje teoretický úvod, kde jsou popsány druhy syntéz a jejich parametry. Hlavní část je zaměřena na syntézu zvuku a její druhy. Dále je zde rozebráno programovací prostředí Max/MSP a jeho možnosti. V praktické části je popsán návrh řešení této práce a její teoretické vypracování s ukázkami dílčích programů.

Oproti semestrální práci byla práce bakalářská rozšířena o zásadní části rozebírající praktickou část zadání. Syntetizér navržený v semestrální práci byl zrealizován a upraven do požadované podoby s parametry, které granulární syntéza vyžaduje. Bylo vytvořeno vizuální prostředí znázorňující fungování syntetizéru a vytvořen komunikační systém pro mobilní telefon na bázi *UDP* protokolu.

Zjištěné hudební možnosti a využití za pomoci pohybu mobilního telefonu potvrzují experimentální charakter nástroje, jeho jedinečnost a smysl celé práce.

Cíl práce byl tedy splněn. Byl vytvořen granulární syntetizér se schopností ovládání parametrů za pomoci mobilního zařízení, díky čemuž se liší od klasických syntetizérů.

I po ukončení této práce by bylo možné pokračovat ve vývoji vizuálního prostředí a vylepšení komunikace mezi mobilním telefonem a programovacím prostředím *MAX*, který je pro potřeby této práce dostačující.

Literatura

- [1] SYROVÝ, Václav. *Hudební signál a jeho syntéza*. Živá hudba 1986, [online] Dostupné z: <https://ziva-hudba.info/hudebni-signal-a-jeho-synteza/> [cit. 19. 11. 2024].
- [2] RUSS, Martin. *Sound Synthesis and Sampling*. 3rd ed. Oxford: Focal Press, 2009. ISBN 978-0-240-52105-3.
- [3] ROADS, Curtis. *Introduction to Granular Synthesis*. In: *Computer Music Journal*, vol. 12, no. 2, 1988, pp. 11–13. JSTOR, [online] Dostupné z: https://www.jstor.org/stable/3679937?searchText=&searchUri=&ab_segments=&searchKey=&refreqid=fastly-default%3Af6d19fc1f1115f13dd691142927c198a&seq=1 [cit. 27. 11. 2024].
- [4] ROADS, Curtis. *The evolution of granular synthesis: an overview of current research*, University of California, [online] Santa Barbara CA 93106-6065 USA, Dostupné z: https://static1.squarespace.com/static/5ad03308fcf7fd547b82eaf7/t/5b75a255352f53388d8ef793/1534435933359/EvolutionGranSynth_9Jun06.pdf [cit. 27. 11. 2024].
- [5] BURK, Phil et al. *Music and Computers*. In: *Music and Computers: A Theoretical and Historical Approach*. [online] 2011. Dostupné z: https://musicandcomputersbook.com/chapter4/04_08.ph
- [6] ROADS, Curtis. *Microsound*. PAP/CDR edition. The MIT Press, 20 August 2004. ISBN 0-262-18215-7.
- [7] OPIE, Timothy. *Sound in a nutshell: Granular Synthesis* [online]. La Trobe University, 1999. Dostupné z: <http://www.granularsynthesis.com/hthesis/hthesis.html> [cit. 27. 11. 2024].
- [8] PUCKETTE, Miller. *The Patcher*. ICMC Proceeding 1988 Dostupné z: <https://msp.ucsd.edu/Publications/icmc88.pdf> [cit. 28. 11. 2024].
- [9] NASA Science. *Visible Light*. [online]. NASA Dostupné z: https://science.nasa.gov/ems/09_visiblelight [cit. 9.12. 2024].
- [10] Cycling74. Max/MSP History - *Where did Max/MSP come from?*. Archived from the original on 2009-06-09. [online] Dostupné z: <https://msp.ucsd.edu/Publications/icmc88.pdf> [cit. 28. 11. 2024].

- [11] *MayMSPHistory*. LilliHart. Archived on 25 Apr 2008 - 22:52 [online] Dostupné z: <https://web.archive.org/web/20090609205550/http://www.cycling74.com/twiki/bin/view/FAQs/MaxMSPHistory#expand> [cit. 29. 11. 2024].
- [12] Max 8 Documentation [online]. San Francisco: Cycling '74, 2019. Dostupné z: <https://docs.cycling74.com/legacy/max8> [cit. 29. 11. 2024]
- [13] Building Max with JUCE. JUCE [online]. London: JUCE, 2024. Dostupné z: <https://docs.cycling74.com/legacy/max8> [cit. 29. 11. 2024]
- [14] PICKLE, Brian. *UDP Definition*. [online]. TechTerms.com. Sharpened Productions, 05 October 2022. Dostupné z: <https://techterms.com/definition/udp> [cit. 6.12. 2024]

Seznam symbolů a zkratek

son et lumiere v překladu světlo a zvuk

Perlinův šum gradientní šum, který se často využívá pro simulaci přírodních jevů jako je kouř, oheň nebo komplexní pohyb

ADSR obálka druh úpravy dynamiky v čase

Obsah elektronické přílohy

Elektronická příloha obsahuje ZIP složku `granularni_syntetizer` obsahující patcher granulárního syntetizéru, spustitelný v programu MAX, a PDF soubor Manuál k použití v podsložce *other*. Syntetizér byl testován ve verzi Max 8.6.4.

Příloha: `granularni_syntetizer.zip`