



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

EXPERIMENTÁLNÍ SOFTWAREVÝ VST SYNTETIZÉR NA PRINCIPU GRANULÁRNÍ SYNTÉZY, JEHOŽ PARAMETRY LZE ŘÍDIT MJ. I RŮZNÝMI VLASTNOSTMI SVĚTELNÉHO SPEKTRA

BAKALÁŘSKÁ PRÁCE VST SYNTHESIZER BASED ON THE PRINCIPLE OF GRANULAR SYNTHESIS,
THE PARAMETERS OF WHICH CAN BE ALSO CONTROLLED BY VARIOUS PROPERTIES OF THE LIGHT
SPECTRUM

AUTOR PRÁCE

AUTHOR

Vojtěch Kuchař

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MgA. Mgr. Dan Dlouhý, Ph.D.

BRNO 2024

Bakalářská práce

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

Student: Vojtěch Kuchař

ID: 240173

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Experimentální softwarový VST syntetizér na principu granulární syntézy, jehož parametry lze řídit mj. i různými vlastnostmi světelného spektra

POKYNY PRO VYPRACOVÁNÍ:

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

DOPORUČENÁ LITERATURA:

- [1] PUCKETTE, M., Theory and Techniques of Electronic Music, 2006. 337 s. online: <http://msp.ucsd.edu/techniques.htm>
- [2] FORRÓ, D., Svět MIDI. Grada, Praha, 1997. 375s. ISBN 80-7169-412-6.

Termín zadání: 5.2.2024

Termín odevzdání: 28.5.2024

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 věnuje vývoji VST a standalone plug-inu na bázi granulárního sampleru, jehož parametry (délka grainu, jeho AD obálka a míra překrytí) mohou být ovládány pomocí hodnot průměrného odstínu, sytosti a světlosti pixelů snímaných webkamerou. Jednotlivé parametry mohou být řízeny také pomocí ovládacích prvků na panelu nástroje. Plug-in může fungovat také jako jednoduchý sampler.

Klíčová slova

Synthesátor, granulární syntéza, plug-in, VST, JUCE, C++

Abstract

This thesis addresses the development of a VST and standalone plug-in based on a granular sampler, whose parameters (grain length, AD envelope and overlap amount) can be controlled by the average hue, saturation and lightness values of pixels captured by a web camera. The individual parameters can also be adjusted using controls on the tool's panel. Additionally, the plug-in is capable of functioning as a simple sampler.

Keywords

Synthesizer, granular synthesis, plug-in, VST, JUCE, C++

Bibliografická citace

KUCHAŘ, V. *Experimentální softwarový VST syntetizér na principu granulární syntézy, jehož parametry lze řídit mj. i různými vlastnostmi světelného spektra*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. 40 s. Bakalářská práce. Vedoucí práce: doc. Ing. MgA. Mgr. Dan Dlouhý PhD.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Vojtěch Kuchař
VUT ID studenta:	240173
Typ práce:	Bakalářská práce
Akademický rok:	2023/2024
Téma závěrečné práce:	Experimentální softwarový VST syntetizér na principu granulární syntézy, jehož parametry lze řídit mj. i různými vlastnostmi světelného spektra

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 jsem si 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.

V Brně dne: 28. května 2024

podpis autora

Poděkování

Děkuji vedoucímu semestrální práce doc. Ing. MgA. Mgr. Danu Dlouhému PhD. za cenné rady, postřehy, pomoc a zejména trpělivost a pohotovost při konzultacích mé bakalářské práce. Dále bych rád poděkoval Bc. Mojmíru Kyjonkovi za užitečné rady v oblasti programování a algoritmizace.

V Brně dne: 28. května 2024

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. GRANULÁRNÍ SYNTÉZA.....	12
1.1 PRINCIP A METODY GRANULÁRNÍ SYNTÉZY	12
1.1.1 Metoda mřížek (screens)	12
1.1.2 Metoda událostí (events)	14
1.2 HISTORIE.....	15
1.2.1 Iannis Xenakis.....	15
1.2.2 Curtis Roads	15
1.2.3 Barry Truax.....	16
2. SAMPLING	17
2.1 HISTORIE.....	17
2.1.1 Mellotron	17
2.1.2 Digitální samplery	17
2.2 METODY ZMĚNY VÝŠKY TÓNU	18
2.2.1 Analogové nástroje	18
2.2.2 Digitální nástroje.....	18
2.2.3 Multisampling	19
3. SROVNÁNÍ EXISTUJÍCÍCH ŘEŠENÍ.....	20
3.1 RIBS	20
3.2 EMERGENCE	21
3.3 GRAIN STRAIN.....	23
3.4 GRANULAB	24
3.5 ARGOTLUNAR	25
4. NÁVRH VLASTNÍHO ŘEŠENÍ	27
4.1 FRAMEWORK JUCE.....	27
4.2 SNÍMÁNÍ BAREV	27
4.2.1 Model HSV/HSL.....	28
4.3 PRVOTNÍ NÁVRH.....	28
4.3.1 Grafické uživatelské rozhraní	29
4.3.2 Načtení souboru	30
4.3.3 Zobrazení amplitudy v čase.....	31
4.3.4 Výběr úseku načteného souboru.....	31
4.3.5 Granulizace a práce s grainy.....	32
4.3.6 Řazení grainů.....	32
4.3.7 Efektivní sekce.....	33
4.3.8 LFO.....	33
4.3.9 Řešení změny parametrů pomocí barvy	33
5. REALIZACE VLASTNÍHO ŘEŠENÍ	36

5.1	GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ	36
5.2	STRUKTURA NÁSTROJE.....	38
5.3	REŽIM JEDNODUCHÉHO SAMPLERU	38
5.4	REŽIM GRANULÁRNÍHO SAMPLERU	39
5.5	NASTAVITELNÉ PARAMETRY SAMPLERU	40
5.6	NASTAVENÍ PARAMETRŮ POMOCÍ WEBKAMERY	41
6.	TESTOVÁNÍ NÁSTROJE A NÁVRH BUDOUCÍCH ÚPRAV	44
6.1	TESTOVÁNÍ NÁSTROJE.....	44
6.1.1	<i>Práce s parametry.....</i>	45
6.1.2	<i>Nastavení parametrů granulární syntézy pomocí analýzy průměrné barvy.....</i>	45
6.1.3	<i>Práce s dalšími instancemi nástroje</i>	47
6.2	NÁVRH NA BUDOUCÍ ÚPRAVY.....	48
6.2.1	<i>Multisampling</i>	48
6.2.2	<i>Rozptyl a modulace parametrů</i>	48
6.2.3	<i>Aplikace časového roztažení a převzorkování v režimu granulárního sampleru</i>	49
6.2.4	<i>Kontinuální snímání barev a další formy získávání obrazu pro analýzu</i>	49
6.2.5	<i>MIDI Learn</i>	49
7.	ZÁVĚR.....	50
	LITERATURA.....	51
	SEZNAM SYMBOLŮ A ZKRATEK	54

SEZNAM OBRÁZKŮ

1.1	Tabulka rozložení zrn podle jejich frekvence a intenzity. Symbol + označuje přítomnost zrna na této pozici, 0 značí prázdnou buňku tabulky. Zdroj: [3].....	13
1.2	Tabulka pravděpodobnosti přehrání jednotlivých buněk. Výsledná pravděpodobnost pro danou frekvenci je vždy rovna 100 %. Zdroj: [3]	13
1.3	Vizualizace několika oblaků (clouds) v čase. Jednotlivé body reprezentují zrna rozložená ve frekvenční a časové ose. Zdroj: [5]	14
3.1	Grafické rozhraní VST granulárního syntezátoru <i>Ribs</i>	21
3.2	Grafické rozhraní VST granulárního syntezátoru <i>Emergence</i>	22
3.3	Grafické rozhraní VST granulárního syntezátoru <i>Grain Strain</i>	24
3.4	Grafické rozhraní VST granulárního syntezátoru <i>GranuLab</i>	25
3.5	Grafické rozhraní VST granulárního syntezátoru <i>Argotlunar</i>	26
4.1	Srovnání 3D vizualizace modelů HSL a HSV. Zdroj: [37]	28
4.2	Blokové schéma granulárního VST syntezátoru <i>Granny Synth</i>	29
4.3	Návrh GUI granulárního VST plug-inu <i>Granny Synth</i>	30
4.4	Blokové schéma zpracování dat získaných webkamerou	34
5.1	Grafické uživatelské rozhraní granulárního sampleru <i>Granny Synth</i>	36
5.2	Detail displeje při zobrazení nahraného samplu v režimu jednoduchý sampler.....	37
5.3	Detail displeje při zobrazení nahraného samplu v režimu granulární sampler	37
5.4	Diagram znázorňující strukturu granulárního sampleru <i>Granny Synth</i>	38

SEZNAM TABULEK

4.1	Shrnutí parametrů navrhovaného VST plug-inu <i>Granny Synth</i>	35
5.1	Tabulka parametrů granulárního sampleru <i>Granny Synth</i>	41
6.1	Hodnoty zpětně vypočítané z parametrů granulární syntézy určených pomocí průměrné barvy obrázku	46
6.2	Hodnoty referenčních obrázků zpracovaných externím analyzátozem	47

ÚVOD

Syntežátory jsou součástí hudebního světa již desítky let. První z nich byly sestrojeny před více než stoletím a připomínaly dnešní elektrické varhany [1]. Za základ dnešních syntežátorů je považován prototyp Roberta Mooga představený na sdružení *Audio Engineering Society* v roce 1964. Dnešní nástroje jsou schopny pokrýt obrovskou paletu zvuků a jejich barev a existuje jich nepřeberné množství. Technická vyspělost umožňuje přenesení jejich existence z fyzického světa do světa virtuálního a otevírá tím nové možnosti zvukovým tvůrcům. Stejně jako existuje obrovské množství syntežátorů, existuje i množství druhů syntézy, například aditivní, subtraktivní, modulační (AM a FM), granulární, sampling aj.

Granulární syntéza je způsob tvorby zvuku pomocí rozdělení vstupního signálu na spoustu drobných úseků a jejich následné skládání zpět za sebe. V praxi se používají úseky (jinak také granule neboli zrna) o délce přibližně 1–100 ms. Způsoby, kterými jsou jednotlivá zrna za sebe řazena, existují primárně dva – synchronní a asynchronní. Důležitým parametrem pro řízení syntézy je hustota zrn, tedy jejich počet na jednotku času. Dalšími parametry, kterými lze obohatit výsledný zvuk, jsou například délka granulí, změny vzdálenosti mezi jednotlivými granulami, jejich překrývání, rychlost přehrávání, přidávání různých typů obálek a spoustu dalších.

Tato práce se věnuje problematice vývoje softwarového granulárního syntežátoru ve vývojovém prostředí JUCE. Jejím hlavním cílem je vytvoření VST plug-inu schopného vstupní zvuková data jednak rozdělit na granule, zároveň na ně aplikovat některé z výše zmíněných parametrů. Neobvyklost řešení spočívá v možnosti ovládní syntežátoru pomocí vlnové délky světla – respektive jeho barvy snímané webkamerou.

Bakalářská práce je členěna do sedmi kapitol, z nichž první tři slouží jako teoretická část práce. V kapitole 1 je popsán princip a historie granulární syntézy, stručná historie a princip samplerů se nachází v kapitole 2. Srovnání současných volně dostupných softwarových granulárních syntežátorů se věnuje 3. kapitola. Kapitola 4 představuje framework JUCE, shrnuje základní teorii pro snímání barev a věnuje se návrhu praktické části práce – granulárního sampleru *Granny Synth*. Kapitola 5 se věnuje vlastní realizaci a finální struktuře prototypu, kapitola 6 popisuje testování a návrhy na vylepšení a rozšíření funkcionalit. Závěrečná kapitola 7 shrnuje dosažené výsledky a hodnotí projekt jako celek.

1. GRANULÁRNÍ SYNTÉZA

Granulární syntéza je dnes již jednou ze zavedených, ale přesto velmi kreativních forem tvorby umělých zvuků. Její masová využitelnost přišla až s nástupem výkonnějších počítačů, které jsou schopny zpracovat i digitální zvukové soubory s větší bitovou hloubkou, avšak myšlenka tvorby zvuku podobným způsobem byla poprvé realizována s počítači pouze jako pomocnými nástroji pro výpočet některých parametrů, nikoliv jako provádějícími přístroji – hlavním krokem pro realizaci podobných pokusů bylo stříhání a následné spojování magnetofonových pásek. Tato kapitola se věnuje principu tohoto typu syntézy a popisuje některé druhy třídění grainů. Stručné historii granulární syntézy se věnuje kapitola 1.2.

1.1 Princip a metody granulární syntézy

Jak již bylo popsáno v úvodu, koncept granulární syntézy pracuje s krátkými zvukovými úseky v řádech jednotek až desítek milisekund. Tato délka zejména ovlivňuje charakter zvuku z hlediska vnímání tónu. Maďarsko-britský fyzik Dennis Gabor na základě výzkumu stanovil minimální dobu trvání potřebnou pro rozeznání tónu od šumu na 10 ms [2]. Můžeme tedy říct, že délka granulí kratší než tato hodnota vytvoří dojem perkusivního zvuku, zatímco delší granule budou mít spíše tónový charakter.

Důležitým parametrem je také hustota grainů ve výsledném zvuku. S menší hustotou vzniká dojem větší trhanosti, při větší hustotě (či dokonce překryvu zrn) vznikají nové harmonické složky a formantové oblasti v původním signálu neobsažené.

Pro samotný vznik umělého zvuku za pomoci granulární syntézy je nutné zvolit nové pořadí jednotlivých zrn. Kategorie řazení existují v zásadě dvě: synchronní a asynchronní. Synchronní metody spočívají v deterministicky daných mezerách mezi jednotlivými zrn. Může se jednat o konstantní hodnotu, matematický vztah či jiný předpis, podstatná je jeho předvídatelnost. Asynchronní metody takto dané rozestupy mezi granulami nemají. Jejich zobrazení na osách frekvence a času připomíná oblaky – tyto struktury jsou detailněji rozebrány níže v kapitole 1.1.2.

V minulosti bylo velmi zajímavých výsledků dosaženo následujícími metodami, jež lze řadit mezi metody asynchronní:

1.1.1 Metoda mřížek (screens)

Tato metoda byla ve smyslu granulární syntézy – ačkoli ne v dnešním pojetí – použita v 50. letech 20. století Řekem Iannisem Xenakisem. Podrobně ji popisuje ve své knize *Formalized Music: Thought and Mathematics in Composition* [3]. Jednotlivá zrna jsou umístěna do mřížek podle jejich zjištěné frekvence a amplitudy. Každému oknu je následně přiřazena pravděpodobnost jeho přehrání tak, aby každý sloupec (frekvence)

měl dohromady pravděpodobnost přehrání rovnu 100 %. V časové ose je pak zvoleno pořadí frekvencí, na jehož základě jsou podle pravděpodobnosti přehrána jednotlivá okna či pouze určitý počet zrněk v nich. Samo pořadí frekvencí je určeno další pravděpodobnostní tabulkou, případně Gaussovým rozdělením, kdy za střední hodnotu je vybrána libovolná frekvence. Takto získaná data jsou pro Xenakise řídicími instrukcemi, jak pracovat při stříhání a spojování magnetofonového pásu – nosiče vstupních zvukových dat pro granulární syntézu. Zde je patrný rozdíl mezi jeho a dnešním provedením. V 50. letech nebyla dostupná technologie pro digitalizaci zvukového záznamu a jeho další úpravy v této podobě, proto veškerá práce s analogovými nosiči vyžadovala trpělivost, preciznost a s nimi i velkou míru odhodlání. Zároveň i možnosti úprav parametrů jednotlivých zrn či celé syntézy byly oproti dnešnímu digitálnímu provedení velmi ztížené, ba dokonce nemožné.

\downarrow	f_1	f_2	f_3	f_4	f_5	f_6	f_7	\dots
g_1	+	0	+	0	0	0	+	
g_2	0	+	0	0	0	+	0	
g_3	0	0	0	+	+	0	0	
\vdots								

Obrázek 1.1 Tabulka rozložení zrn podle jejich frekvence a intenzity. Symbol + označuje přítomnost zrna na této pozici, 0 značí prázdnou buňku tabulky. Zdroj: [3]

\downarrow	f_1	f_2	f_3	f_4	\dots
g_1	0.5	0	0.2	0	\dots
g_2	0	0.3	0.3	1	\dots
g_3	0.5	0.7	0.5	0	\dots

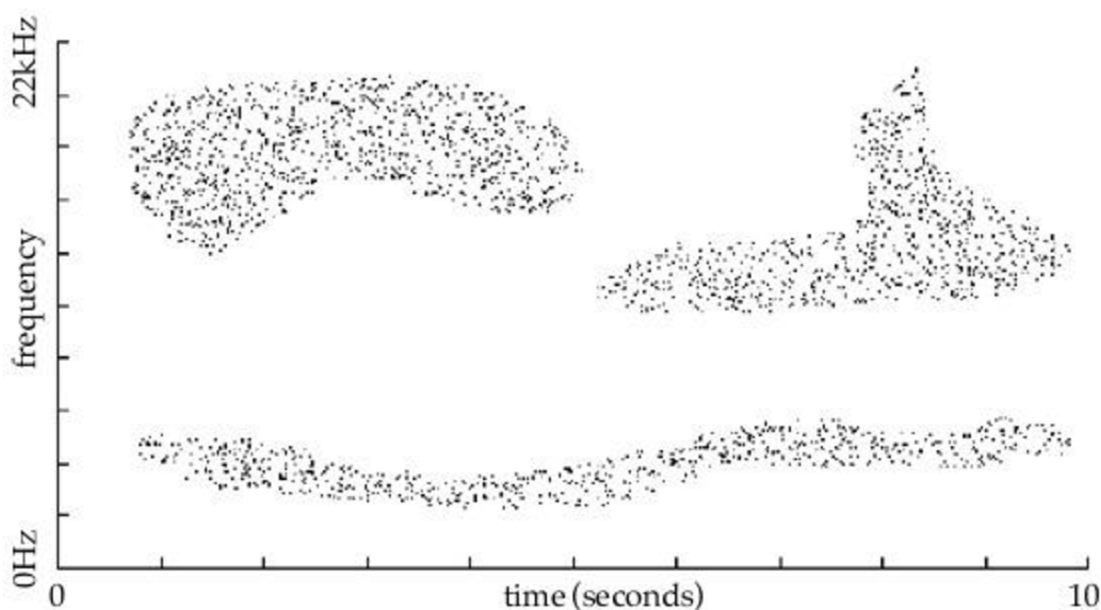
Obrázek 1.2 Tabulka pravděpodobnosti přehrání jednotlivých buněk. Výsledná pravděpodobnost pro danou frekvenci je vždy rovna 100 %. Zdroj: [3]

Celá metoda, ačkoli její výsledek může působit značně nahodile a náhodně, je pevně předepsaná. Pro její konstrukci je využita teorie pravděpodobnosti, z ní vycházející užití Markovových řetězců a dalších stochastických jevů.

1.1.2 Metoda událostí (events)

Jejím autorem je americký hudební skladatel Curtis Roads [4]. Metoda rozděluje granule podle více – až dvanácti – parametrů převážně syntetických zvuků, jako jsou například počáteční tvar vlny, počáteční střední frekvence, šířka pásma a jeho strmost nebo počáteční amplituda. Útvary, do kterých se granule organizují, se nazývají událostmi (events). Výsledné textury dosažené mícháním různých událostí s rozličnými parametry se nazývají mraky (clouds) a přispívají k větší spektrální bohatosti vytvořeného zvuku.

Roads k tomuto tématu dodává, že vítaným rozšířením samotné syntézy by mohlo být vytvoření takzvaných složených granulárních zvuků vzniklých kombinací vln syntetických s vlnami pocházejícími ze samplů živých nástrojů. Dále popisuje princip podobný wavetable syntéze či ROMplerům, systém analýzy/syntézy. Ten ze samplovaného živého nástroje generuje sadu „prováděcích funkcí“, které při následné syntéze ovlivňují tvar vlny k co nejvěrnějšímu napodobení původního zvuku.



Obrázek 1.3 Vizualizace několika oblaků (clouds) v čase. Jednotlivé body reprezentují zrna rozložená ve frekvenční a časové ose. Zdroj: [5]

Granulární syntezátory existují v zásadě ve dvou provedeních. Prvním z nich je tzv. granulizér, tedy svého druhu efekt, který pouze vstupní zvukový soubor rozdělí na jednotlivá zrnka, případně upraví jejich parametry podle dat zadaných uživatelem,

a následně je podle zvoleného algoritmu skládá za sebe a vytváří tím umělý zvuk. Možnost ovládání výšky tónu je zde omezená a dá se provádět například změnou rychlosti přehrávání nebo změnou hustoty zrn. Druhým z nich je plnohodnotný granulární syntezátor, který zrna granulizérem vytvořená a seřazená dokáže přehrávat ve výšce určené uživatelem pomocí klaviatury.

1.2 Historie

Počátky myšlenky tvorby zvuku způsobem podobným granulární syntéze sahají do 17. století. Tehdy byla novověkým nizozemským atomistou Isaacem Beeckmanem do jeho deníku zapsána poznámka – teze o částicové podstatě zvuku. Podle této teze kmitající zdroj zvuku dělí okolní vzduch na zvukové částice, tzv. globuly, šířící se ve všech směrech od zdroje [6]. Jakkoliv tento způsob uvažování o šíření zvuku může dnes působit nepřesně (k žádnému přesunu částic na velkou vzdálenost nedochází, pouze ke kmitání a přenosu energie pomocí vlnění), právě v této poznámce můžeme spatřovat pohled na zvuk ve velmi krátkých časových intervalech, což je ve větším měřítku doménou spíše fyziků minulého století.

1.2.1 Iannis Xenakis

První skutečné pokusy o zvukovou granulární syntézu se objevují v 50. letech 20. století u již zmiňovaného řeckého skladatele a architekta, jenž většinu tvůrčího života prožil ve Francii, Iannise Xenakise (1922–2001). Jeho architektonická práce na pavilonu firmy Phillips pro světovou výstavu v Bruselu se nesla v těsném spojení s hudebním uvažováním a jeho koncepcí *meta-artu*, tedy myšlenky, že každé umění lze pomocí matematických operací převést na jinou formu [3]. Ve své zralé tvůrčí fázi komponuje kromě instrumentálních skladeb komorních i orchestrálních řadu elektroakustických skladeb s využitím ryze elektronických zvuků (včetně granulární syntézy) i upravených zaznamenaných zvuků okolního světa. Granulární syntézu obsahují například jeho skladby *Bohor* (1962) nebo *Persepolis* (1972). Jeho přístup ke granulární syntéze je popsán detailněji v kapitole 1.1.1.

1.2.2 Curtis Roads

Dalším průkopníkem v oblasti granulární syntézy se stal americký skladatel a programátor Curtis Roads (*1951). Jeho skladatelské dílo se zejména specializuje na hudbu tvořenou s pomocí počítačů, granulární a pulsarovou syntézu. Ve své knize *Microsound* (2001) popisuje různé typy obálek aplikované na jednotlivá zrnka pro odstranění artefaktů vzniklých náhlými přechody úrovní hlasitostí mezi jednotlivými grainsy [7]. Z tohoto pohledu rozlišuje Gaussovu, kvazi-Gaussovu, lichoběžníkovou, trojúhelníkovou, sinus cardinalis a exponenciálně rostoucí nebo klesající obálku. Dále se zaměřuje na technologii generování zrn, jejich řazení a organizaci a tvorbu složitějších struktur, tzv. mraků (clouds).

Roadsovi patří i významné prvenství, po setkání s Iannisem Xenakisem počátkem 70. let 20. století zahájil své experimenty s prováděním granulární syntézy s využitím vstupního zvukového signálu v digitální formě [7]. Právě použitím počítače přiblížil granulární syntézu o poznání blíže dnešní podobě. Jeho další přínos spočívá ve zkoumání, popisu, třídění, vytváření pojmů a vývoji dalšího software sloužícího k provádění granulární syntézy. Za důležitou lze také považovat jeho práci v rámci vzdělávání dalších výzkumníků microsoundu – tedy oboru, jenž se zabývá zvuky o délce mezi setinou až desetinou sekundy.

1.2.3 Barry Truax

Významný posun v oboru granulární syntézy provedl také kanadský skladatel Barry Truax (*1947). Jeho přínosem byl vývoj a implementace granulární syntézy v reálném čase (1986) [8], což otevřelo možnosti vývoje směrem k dnešním VST plug-inům, které jsou schopny provádět tento typ syntézy téměř okamžitě, většinou na základě MIDI zpráv dodaných uživatelem. Truaxova implementace spočívala ve vytvoření programu *GSX* s využitím počítače pro digitální zpracování signálu *DMX-1000*. Při vytváření programu využil základ software *PODX* pro Poissonovo rozdělení pravděpodobnosti přehrání zvukových událostí [9].

2. SAMPLING

Sampling je forma tvorby umělého zvuku pomocí změny výšky samplu uloženého v paměti nástroje. Jelikož se tato práce zabývá také samplingem, slouží tato kapitola jako stručné představení jeho historie a některým používaným metodám.

2.1 Historie

Myšlenka použití zvukové nahrávky s měnitelnou výškou přehrávání byla poprvé realizována ve druhé polovině 20. století a velké oblibě a rozsáhlému využití se těší dodnes.

2.1.1 Mellotron

Za jeden z prvních samplerů můžeme označovat již analogový nástroj, který se objevil v 60. letech minulého století – mellotron. Tento elektromechanický klávesový nástroj, poprvé sestrojený v anglickém Birminghamu, používal jako paměťový nosič magnetické pásky s nahranými zvuky v různých výškách. [10] Standardní nástroj měl 35 kláves, pro každý použitý zvuk v nástroji tedy existoval stejný počet magnetických pásek nesoucích informaci o výšce tónu. Mellotron se stal oblíbeným nástrojem hudebníků hledajících v 60. a 70. letech nový zvuk – jeho specifická a těžko zaměnitelná barva se objevuje v mnoha slavných nahrávkách té doby: uvedme například *Strawberry Fields Forever* britské skupiny The Beatles, [11] *Space Oddity* Davida Bowieho, [12] *Heart of the Sunrise* britské skupiny Yes [13] a spoustu dalších.

Původní mellotrony (verze Mk. I a Mk. II) byly vybaveny sadou osmnácti melodických zvuků (např. smyčcové nástroje, žesťové nástroje, flétny, sborové vokály a další) spolu se sadou rytmických doprovodů ovládaných levou částí klaviatury. Takto robustní vybavení vyžadovalo velké množství prostoru i mechaniky pro zajištění uložení a přepínání použitých pásek, proto hmotnost nástroje dosahovala přibližně 150 kg.

Pozdější verze (M300, M400) měly za cíl snížit hmotnost i velikost nástroje, což vyústilo ve snížení počtu okamžitě volitelných pásek a změny v konstrukci rámů – sady pásek bylo nyní možno poměrně rychle vyměnit a dosáhnout tak podobného množství přehrávatelných zvuků nástroje jako u prvních verzí. [14]

2.1.2 Digitální samplery

S příchodem digitálního zpracování signálu se objevila možnost provádět sampling jednodušeji na zvukových nahrávkách zaznamenaných v číslicové podobě. Toto provedení prolomilo limit v počtu použitelných zvuků, neboť do paměti bylo možno nahrát libovolný sample v digitální podobě; jedinou limitaci zde představovala velikost paměti nástroje ovlivňující délku a počet okamžitě použitelných samplů.

Nejvýznamnějšími z prvních samplerů byly nástroje Melodian (Computer Music Inc.), [15] Synclavier (New England Digital), [16] CMI (Fairlight Instruments), [17] Emulator (E-mu Systems), [18] S900 (Akai) [19] nebo S-10 (Roland). [20] Uvedené nástroje byly představeny veřejnosti v rozmezí let 1976–1986 a lišily se způsobem provedení, použitím různých podpůrných počítačů, metodami výběru a možnostmi následných úprav nahraných samplů.

2.2 Metody změny výšky tónu

Základní princip samplingu spočívá ve změně výšky nahraného zvuku. Tento oddíl pojednává o různých principech používaných u analogových i digitálních nástrojů vedoucích k dosažení posunu signálu v kmitočtové oblasti, metodách využívajících pevný i proměnlivý vzorkovací kmitočet i multisampling – použití několika samplů pro rozložení v jednotlivých zónách klaviatury.

2.2.1 Analogové nástroje

Mellotron využíval sady magnetických pásek nesoucích záznam tónu hraného daným nástrojem v dané výšce. Můžeme tak hovořit o multisamplingu, neboť každá klávesa přehrávala pouze svůj přiřazený pásek – počet zón byl tedy roven počtu kláves nástroje. Délka samplu byla rovna přibližně osmi sekundám, poté došlo k ukončení převijení pásku – nebylo tedy možné přehrávat nekonečné smyčky, ani hrát příliš dlouhé držené tóny.

Změna výšky tónu u mellotronu byla prováděna pro všechny pásky současně potenciometrem ovládajícím rychlost převijení. Dalšími ovládacími prvky v závislosti na modelu nástroje byla volba stopy na třístopém magnetickém pásku, ovládání hlasitosti, tónové clony, u starších modelů také volba doprovodu na levém manuálu.

2.2.2 Digitální nástroje

U digitálních nástrojů jsou v praxi využívány dvě metody změny výšky tónů (pitch scaling) nezachovávající původní délku vstupního samplu – změna vzorkovacího kmitočtu a změna počtu přehrávaných vzorků. V obou případech je do sampleru spolu se samplem zanesena i informace o původní výšce tónu, důležitá pro multisampling a dodržení hrané i znějící výšky výstupního signálu.

Při změně výšky tónu pomocí změny vzorkovacího kmitočtu dochází zároveň ke dvěma jevům: k samotné změně výšky tónu samplu a ke změně doby trvání. Vzorkovací kmitočet je roven násobku počtu vzorků samplu a požadovaného kmitočtu. Problematickými faktory při tomto použití jsou velký rozsah vzorkovacích kmitočtů – pro požadovaný rozsah n oktáv je podíl nejvyššího a nejnižšího kmitočtu roven 2^n , dále se vyskytuje nutnost přelaďování rekonstrukčního filtru D/A převodníku pro zachování barvy zvuku, zároveň také existuje problém s polyfonií – pro každý hlas je třeba mít

vlastní D/A převodník. [21] Rozsah změny vzorkovacího kmitočtu je možno řešit právě multisamplingem, případně použitím jiné metody pro změnu výšky tónu.

Změna výšky tónu v systému s pevným vzorkovacím kmitočtem je prováděna změnou počtu vzorků a následným převzorkováním. Počet vzorků se zde rovná podílu vzorkovacího kmitočtu a požadovaného kmitočtu. Zejména při požadované výšce tónu nižší, než je původní výška tónu samplu, je třeba do signálu vzorky doplnit a interpolovat, což v závislosti na použití metody interpolace může zhoršovat buď odstup signálu od šumu (lineární interpolace) nebo výpočetní náročnost (polynomiální a další typy interpolace). [21]

2.2.3 Multisampling

U některých nástrojů dochází ke změnám ve zvukové barvě při hře v různých polohách a také v různých dynamikách. Použití horizontálních a vertikálních zón (pro výšku a dynamiku tónu) tak napomáhá věrnějšímu zvuku nástrojů při snaze o jejich replikaci v různých samplerech nebo elektronických klávesových nástrojích (lidově označovaných jako „keyboards“). Zároveň lze multisamplingem snížit nároky na rozsah vzorkovacích kmitočtů či nutnost interpolovat velké množství vzorků a tím zhoršovat odstup signálu od šumu nebo zvyšovat výpočetní náročnost.

Některé samplery (zejména ty, které se zabývají replikací reálných nástrojů) také řeší případy, ve kterých je požadovaná délka tónu větší než délka nahraného samplu. V takovém případě dochází k použití smyčky ve fázi „sustain“, nachází-li se u nástroje, a po uvolnění klávesy dochází k napojení části „release“. [22]

3. SROVNÁNÍ EXISTUJÍCÍCH ŘEŠENÍ

Pro vývoj vlastního VST granulárního syntezátoru je vhodné znát některá již funkční a zavedená řešení. V následující kapitole jsou proto představena taková řešení softwarových granulárních syntezátorů na současném trhu, která spadají do podobné kategorie jako vyvíjený prototyp. Jedná se o projekty, které jsou dostupné zdarma a nabízejí podobné možnosti nastavení parametrů granulární syntézy. Ačkoliv volně dostupných plug-inů s podobnými parametry existuje větší množství (např. *Atomic Cloud*, *GGrain*, *Granulator*, *Hadron*, *Lagrange*, *PolyGAS* a další), ve srovnání je detailněji popsáno pouze pět z pohledu autora nejzajímavějších či nejkvalitnějších řešení. Jedná se o VST syntezátory *Ribs*, *Emergence*, *Grain Strain*, *GranuLab* a *Argotlunar*.

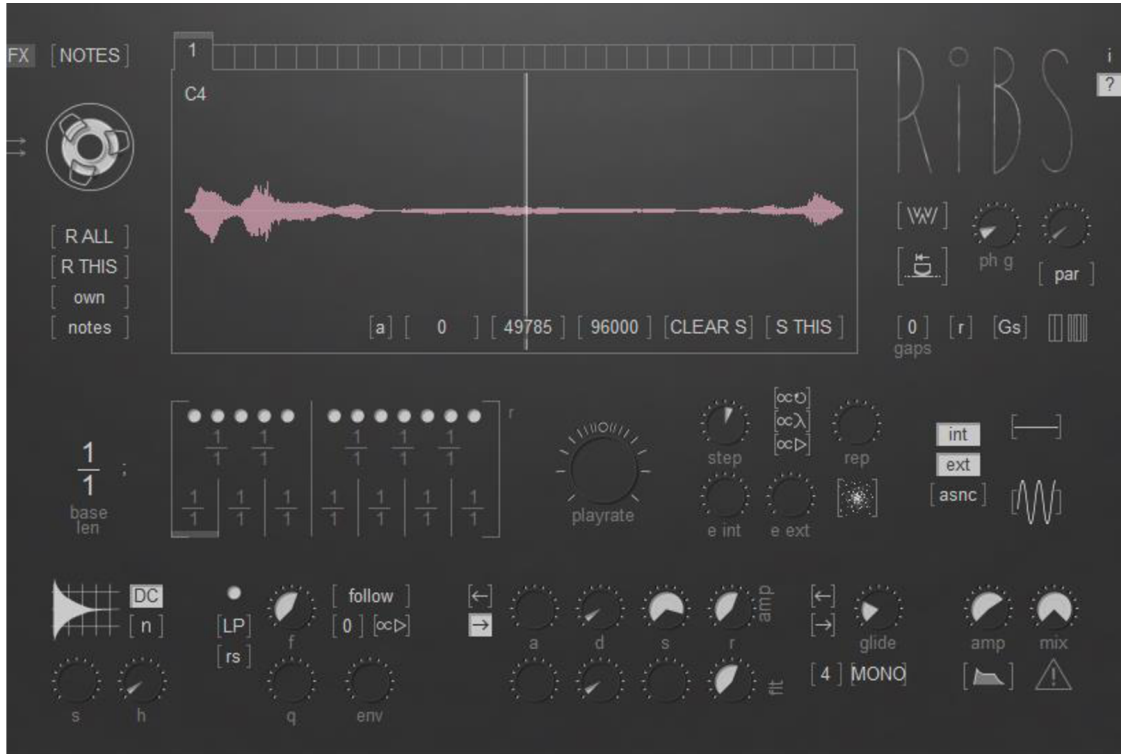
3.1 *Ribs*

Jedná se o VST granulární syntezátor z vývojářské dílny *Hvoya Audio* [23][24]. Nástroj je koncipován pro využití v podobě samostatné aplikace nebo jako DAW plug-in na operačních systémech Windows a MacOS. *Ribs* umožňuje použití až 32 samostatných bufferů o nastavitelné délce 6000–480000 vzorků. Vstupní signál je do plug-inu nahráván pomocí aktivování příkazu *REFILL* a následném přehrání zvukové stopy, ze které má být signál získán.

Syntezátor pracuje ve třech módech: *SIMPLE*, *NOTES* a *BEAT*. V režimu *SIMPLE* je pouze přehráván zachycený signál, nedochází ke granulizaci a nastavitelné jsou jen některé parametry, které ovlivňují celkový výstupní signál. V režimu *NOTES* je možné ovlivnit parametry probíhající granulární syntézy, jak globálně, tak i zvlášť pro jednotlivé MIDI noty. Režim *BEAT* slouží k nastavení délky zrn podle hostitelského DAW, konkrétně dle nastaveného tempa projektu, doladování těchto hodnot lze ale provádět i podobně jako v režimu *NOTES*.

Pro jednotlivá zrna lze nastavit parametry jako je jejich délka, rychlost přehrávání, překryv, typ aplikovaného okna (obálky), nastavení počáteční pozice pro přehrávání zrna o n vzorků, opakování (0–512krát) a nastavení přidaného šumu/chyb v signálu. Pro výstupní signál lze ovlivnit pravděpodobnost výskytu mezer, jejich distribuci (pravidelnou či náhodnou), vyplnění mezer původním signálem, nastavení výchylky rychlosti přehrávání, délky zrna, její tvar a frekvenci. Dále existuje možnost nastavit AGC (automatic gain control), jeho množství, rychlost nástupu, tvrdost, a možnost přidání nebo odebrání stejnosměrné složky signálu. Aplikovat lze také filtr typu dolní, pásmová, nebo horní propust, peak nebo pásmová zádrž s nastavitelnou střední, případně mezní frekvencí, šířkou pásma nebo úrovně přidání obálky. Zmíněnou ADSR obálku lze aplikovat jak na amplitudu výstupního signálu, tak i na filtry, v tomto případě jsou parametry amplitudové i filtrové obálky nezávislé. Poslední sekce

ovládacích prvků se vztahuje k nastavení času pro glide, výběr mezi jednohlasem a vícehlasem s možností ovládat počet hlasů (1–32), nastavení zesílení a poměr mezi čistým a procesovaným signálem ve výstupu [25].



Obrázek 3.1 Grafické rozhraní VST granulárního syntezátoru *Ribs*

Poměrně specifická je možnost nastavit globální délku zrna, a zároveň ovlivnit délku zrna při vyslání MIDI zprávy o konkrétní notě. K tomuto účelu slouží panel s vyobrazenou klaviaturou, na níž jsou délky not vyjádřeny pomocí zlomku. Tato hodnota je následně vynásobena globální délkou zrna a aplikována.

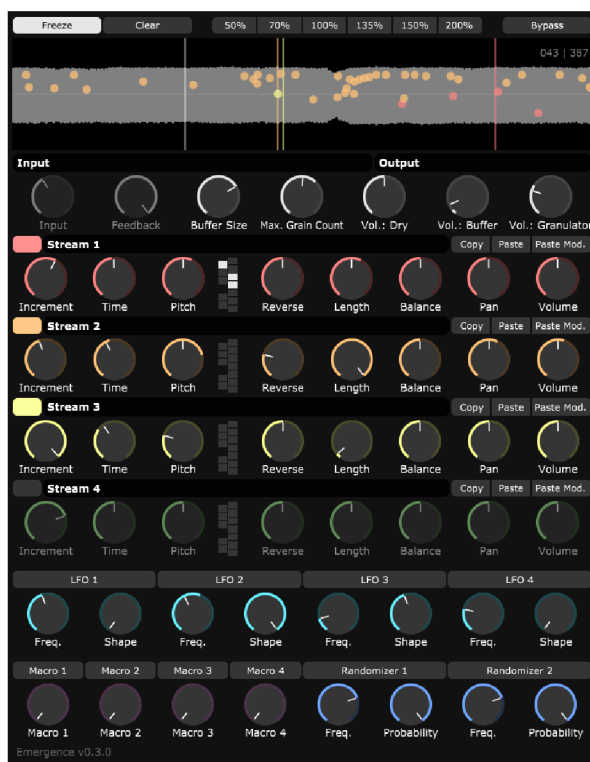
Ribs umožňuje také vybrat konkrétní úsek v bufferu jako zdroj vzorků pro získávání grainů. Provádí se označením takového úseku myší na displeji zobrazovaného tvaru vlny. Obecně lze říci, že jde o plug-in velmi všestranný, nevyžadující dlouhý čas pro pochopení jeho ovládacích prvků a nabízející možnosti pro provádění granulární syntézy s velkým prostorem pro kreativní práci uživatele.

3.2 Emergence

Tento VST plug-in vytvořený vývojářem Danielem Gergelym funguje jako prostý granulizér, dává možnost vytvořit zrna a dále pracovat s jejich parametry [26]. Načtení vstupního signálu probíhá podobně jako u zmíněného projektu *Ribs*, tedy přehráním požadované zvukové stopy a jejím nahráním do bufferu. Lze vytvořit až 600 grainů

naráz a uspořádat je do čtyř proudů (streams).

Co se týče ovladatelných parametrů syntézy, jedná se zejména o následující kategorie: globální nastavení výstupního signálu, parametry jednotlivých proudů (streams), parametry LFO, úroveň makra a parametry dvou nezávislých randomizérů.



Obrázek 3.2 Grafické rozhraní VST granulárního syntezátoru *Emergence*

Způsob vytváření grainů je velmi pěkně graficky znázorněn na displeji zobrazujícím tvar vlny. Zde je možno zobrazit ukazatel pozice v bufferu pro každý proud i pro původní signál zvlášť. Na základě zadaných parametrů granulární syntézy poté vznikají zrna, jež jsou vyznačena opět v časovém průběhu signálu, zároveň je také vidět jejich umístění v panoramatu – v horní části pozorujeme levý kanál, ve spodní části kanál pravý. Parametry, jimiž můžeme ovlivňovat jednotlivé proudy, jsou tyto: *Increment* ovlivňuje rychlost procházení bufferu pro získávání grainů, *Time* slouží k určení periody, po které budou generován nový grain. *Pitch* určuje výšku přehrávaného tónu v půltónech – ke stejnému účelu se dá použít i znázornění klaviatury vedle. Další parametr *Reverse* určuje pravděpodobnost, že budou vzorky z bufferu čteny pozpátku, *Length* upravuje délku granulí, *Balance* a *Pan* určují zdrojový a cílový kanál, ze kterého budou vzorky získávány a do kterého budou zapisovány (0,0 = levý kanál, 1,0 = pravý kanál). *Volume* určuje zesílení proudu.

Nastavení LFO je poměrně prosté: *Frequency* upravuje frekvenci oscilátoru, *Shape* určuje časový průběh oscilace (0,0 = harmonická vlna, 1,0 = šum). LFO lze aplikovat na veškeré parametry včetně oscilátoru samotného. *Macro* slouží k seskupení více ovladačů pod jeden prvek a ovlivnění několika parametrů najednou. Funkce randomizéru spočívá v náhodné změně podřízeného parametru s nastavitelnou frekvencí změn a pravděpodobnosti jejich nastání.

3.3 Grain Strain

Grain Strain je granulární VST syntezátor vyvinutý mezi lety 2021–2022 Björnem Arltem [27]. Umožňuje fungování ve dvou módech: granulizér a granulární syntezátor ovladatelný pomocí MIDI klaviatury. Grafické uživatelské rozhraní je poměrně jednoduché a intuitivní, k ovládání plug-inu je navíc dostupný i manuál [28]. *Grain Strain* provádí okamžitou granulární syntézu na zvukové stopě, které je přiřazen, nedisponuje možností načíst do bufferu zvukový signál a opakovaně k němu přistupovat.

Ovládací prvky syntezátoru jsou rozděleny do čtyř částí: obecná sekce, ve které lze nastavit zesílení a poměr původního/zkresleného signálu, sekce nastavení zrna, kde lze najít přepínač režimů pro určení výšky tónu ze čtyř možností – vstup MIDI, frekvence v rozsahu 16–5000 Hz, synchronizace s tempem projektu v DAW a délka zrn v rozsahu 0,2–2000 milisekund. Pro poslední tři jmenované parametry se v této sekci nachází posuvník pro určení jejich hodnoty. Dále je zde možné nastavit překrytí jednotlivých zrn a aplikovanou obálku.

Následující oddíl se zabývá parametry rámce, ze kterého jsou generována zrna. Mezi ně patří zapojení synchronizace s tempem projektu, jež má přímý vliv na délku rámce; bez jejího synchronizování může tato délka nabývat hodnot mezi 0,2–5000 milisekundami. Ovládání délky v obou případech opět zajišťuje posuvník. Na rámce, podobně jako na zrna, lze aplikovat obálku i nastavit překrytí.

Poslední sekce nabízí možnosti pro režim ovládání výšky tónu pomocí MIDI zpráv. Nastavit zde lze až osmihlasou polyfonii a aplikovat ADSR obálku pro výstupní signál granulárního syntezátoru.



Obrázek 3.3 Grafické rozhraní VST granulárního syntezátoru *Grain Strain*

3.4 GranuLab

Toto řešení bylo vyvinuto vývojářskou společností *Ostinato* [29][30]. Nabízí možnost nahrát zvukový soubor ve formátu .wav nebo .aiff jako vstupní signál a dále pracovat s parametry prováděné granulární syntézy. Plug-in je segmentován do pěti skupin – ovládacích bloků. Prvním z nich je *Control/MIDI*, kde je možné nastavit ADSR obálku pro přijímané MIDI noty a nezávisle také globální ADSR pro řízení dalších parametrů. Druhá sekce (*Soundfile*) se zabývá vstupním souborem, jenž je nahráván přetažením do oblasti displeje průběhu amplitudy, nebo kliknutím na něj. Po nahrání souboru lze označit oblast, ze které budou získávány vzorky pro provedení granulární syntézy. Při zobrazení průběhu je na displeji oranžovou čarou znázorněn průběh amplitudy. Ve spodní části displeje jsou uvedeny informace o vstupním souboru – délka, vzorkovací frekvence a počáteční a koncový údaj výběru. Dalším ovlivnitelným parametrem v této sekci je rychlost procházení bufferu s rozsahem nastavitelných hodnot $-25,0$ – $25,0$ a aktuální pozice v bufferu.

V sekci *Grains* je možno nastavit hustotu generování zrn v rozsahu $0,0167$ – 2000 zrn za sekundu, délku zrna mezi $0,1$ – 10000 ms, základní frekvenci s možností nastavení obráceného procházení bufferu a glissando s rozsahem -25 – 25 půltónů pro každý grain aplikované po zmáčknutí noty. Pod těmito otočnými prvky se dále nachází ukazatel hustoty a průběhu grainů v podobě malého grafu se svislou osou frekvence a vodorovnou osou času spolu s AD (*attack–decay*) obálkou pro jednotlivá zrna.



Obrázek 3.4 Grafické rozhraní VST granulárního syntezátoru *GranuLab*

Předposlední sekcí je presetová sekce *Patches*. Do ní je možné ukládat nastavené scény (skupiny parametrů) a následně je vyvolávat. Kapacita aktivních scén pro procházení činí 132. Mezi možnosti ovládání patří nastavení doby trvání přechodu po přepnutí scén v rozsahu 0–600 s, uložení nastavení scény jako soubor .json, přepínače možnosti změny zvukového souboru, přepnutí kontroléru a nastavení ADSR při změně scény.

V poslední sekcí (*Processing*) jsou k nastavení parametry výstupního signálu. Nachází se zde skupina parametrů pro zkreslení zvuku, efekt delay, třípásmový ekvalizér a kombinace zesílení, kompresoru a stereofonního rozložení.

Veškeré zmíněné parametry v sekcích *Soundfile*, *Grains* a *Processing* mohou být řízeny pomocí následujících možností: randomizérem, poměr zesílení ku šumu, LFO, pomocí MIDI vstupů, ADSR obálkou 1 nebo 2, parametry samotné granulární syntézy, parametry efektní sekce a parametry ekvalizačních filtrů. Pro většinu ovládacích prvků existuje možnost vybrat dvě z těchto možností, jimiž budou řízeny. Pro každý tento slot lze nastavit míru ovlivnění ovládaného parametru parametrem ovládacím.

3.5 Argotlunar

Tento plug-in vytvořený Michaelem Ourednikem slouží jako prostý granulizér bez možnosti ovládat výšku tónu pomocí MIDI not [31]. Přesto se jedná o poměrně všestranný nástroj k provádění granulární syntézy, jež nabízí přehledné a intuitivní uživatelské rozhraní. Syntezátor je rozdělen do devíti sekcí parametrů, z nichž většina může být také randomizována. V prvním oddíle je nastaven poměr zkresleného a čistého signálu spolu se zesílením vstupního signálu. Druhá sekce ovlivňuje zesílení výstupního signálu spolu se stereofonním rozložením mezi levým a pravým kanálem.

Pro oba tyto parametry nechybí možnost určení náhodného rozptylu těchto hodnot v zadaném rozmezí. Následující oddíl definuje přidané zpoždění vedlejší větve, hustotu grainů a jejich délku s možností zadat parametry v milisekundách, případně synchronizovat podle tempa projektu v hostitelském DAW. Aplikované hodnoty jsou opět randomizovatelné.

Čtvrtá sekce se věnuje frekvenčním posunům zrn. Lze zde nastavit transpozici v rozsahu dvě oktávy směrem dolů i nahoru a glissando efekt. Parametry zde mohou mít rozptyl k vyšším i k nižším hodnotám.



Obrázek 3.5 Grafické rozhraní VST granulárního syntezátoru *Argotlunar*

V pátém oddíle existuje možnost aplikovat ekvalizační filtry: pásmovou, horní nebo dolní propust, pásmovou zádrž, hřebenový filtr, nebo lze také nastavit náhodný filtr pro každý grain – z parametrů filtru zde lze upravit jejich jakost (šířku pásma). Následující oddíl slouží pro nastavení obálky. Lze vybrat ze tří možností – RCB (zvýšený cosinus), parabolické a trojúhelníkové. Pro obálku typu RCB můžeme nastavit strmost nástupu a její zkosení, pro trojúhelníkovou pouze její zkosení. Sedmá sekce obsahuje korelační matici – zde lze nastavit propojení a míru ovlivnění až pro tři dvojice parametrů. Změnou hodnoty ovládajícího parametru dojde ke změně také u ovládaného parametru v určeném poměru.

Osmý oddíl slouží k nastavení vlivu zpětnovazební větve na vstupní signál. Devátý je poté věnován souboru parametrů: počet vytvářených proudů zrn v rozsahu 0–20, zámek načítání nových dat do bufferu, nastavení typu tónové stupnice a jejího prvního stupně pro přizpůsobení frekvence výstupního signálu. V poslední řadě se v této sekci nachází displej zobrazující aktuálně nastavované hodnoty pro všechny ovládací prvky.

4. NÁVRH VLASTNÍHO ŘEŠENÍ

V rámci semestrální práce byl vytvořen teoretický návrh pojednávající o tom, jak by měl výsledný nástroj vypadat a jakými funkčními bloky bude potenciálně signál zpracováván. Tento návrh ve zkrácené podobě je spolu s popisem použitého frameworku a stručnou teorií snímání a reprezentace barev pomocí matematických modelů obsažen v této kapitole. Podrobnější popis funkce jednotlivých bloků je umístěn v kapitole pojednávající o realizaci nástroje.

4.1 Framework JUCE

JUCE je framework určený primárně pro vývoj audio software v jazyce C++. Jeho hlavním cílem je umožnit vytvoření zdrojového kódu kompilovatelného a spustitelného zároveň na operačních systémech Windows, MacOS, Linux a jejich mobilních derivátech Android a iOS. Název JUCE je zkratkou pro *Jules' Utility Class Extensions* – zmíněným Julesem je Julian Storer, vývojář grafického a zvukového rozhraní pro DAW *Tracktion Waveform*, jenž tento framework vytvořil. První verze JUCE vyšla v roce 2004, po deseti letech byl framework prodán londýnské společnosti specializující se na hudební hardware *ROLI*. V dubnu 2020 byl opět prodán, novým a zároveň současným majitelem se tehdy stala společnost *PACE Anti-Piracy Inc.* specializující se na bezpečnost zvukového software [32].

V rámci JUCE existuje možnost využití různých předdefinovaných grafických ovládacích prvků, mezi něž patří různé posuvníky, tlačítka, potenciometry, ukazatele stavu načítání atd. Při vývoji vlastního řešení tedy odpadá nutnost designování vlastních prvků a zůstává na rozhodnutí vývojáře, zda bude jeho prototyp graficky odlišen od generického vzhledu JUCE ovladačů.

Velikou výhodou frameworku je také jeho dobře psaná a aktualizovaná dokumentace [33] doplněná o ukázky a návody k použití objektů, tříd a jejich metod. Cenným zdrojem informací a rad ohledně řešení problémů spojených s vývojem zvukových aplikací je také JUCE Forum [34], na kterém spolupracují vývojáři z celého světa a sdílejí své nápady, postřehy a další užitečné informace o práci s frameworkem JUCE.

4.2 Snímání barev

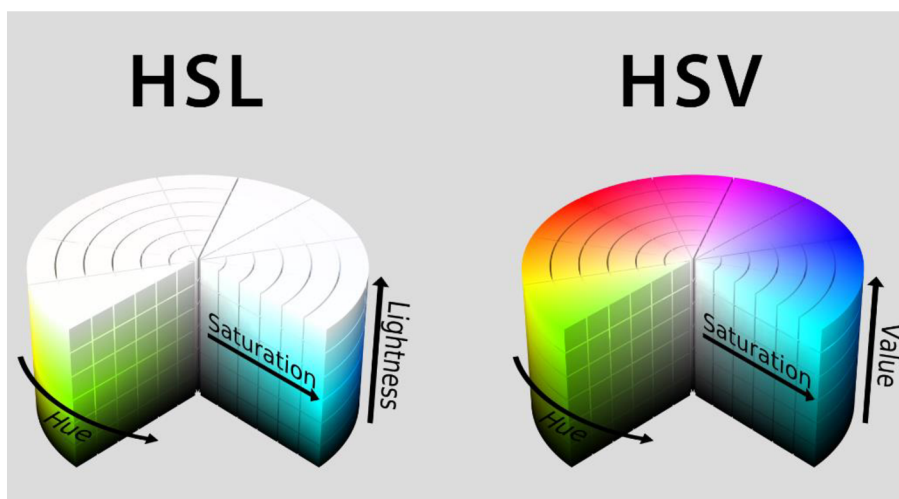
Digitální záznam obrazu byl poprvé umožněn v roce 1970 díky konstrukci fotoaparátu se zabudovaným CCD snímačem [35]. Koncept byl nadále zkoumán a vyvíjen směrem k masovému rozšíření digitální fotografie a videa. Ačkoliv ne všechen obrazový záznam je v současnosti realizován pomocí digitální technologie, pro jeho šíření v elektronické podobě je nutné jej převést do číslicových informací. Pro zobrazení barev existuje několik matematických modelů, jež umožňují nést zprávy o barevných parametrech

světla. Dále je popsán model HSV/HSL využívaný k implementaci ovládání parametrů granulární syntézy vlnovými délkami světla v této práci.

4.2.1 Model HSV/HSL

Tento matematický model nejvíce odpovídá intuitivnímu lidskému vnímání barev. Jeho složky tvoří kombinace barev, nýbrž parametry *Hue* (odstín), měřený jako poloha na barevném kruhu v rozmezí 0° – 360° , *Saturation* (sytost), udává poměr odstínu k šedi na škále 0–100 % a *Value* (hodnota jasu), vyjadřující míru absence černého světla. Velmi podobný je model HSL (zavedený firmou Tektronix), který ovšem využívá parametru *Lightness* (jas), tedy množství bílého světla.

Modely HSV a HSL bývají využívány při editaci fotografií, jelikož jejich parametry umožňují intuitivní práci s barevnými odstíny, sytostí i jasnem snímků. Naopak právě kvůli konstrukčním vlastnostem displejů (RGB segmentace) není nejvhodnějším řešením ukládání matic informací pro jednotlivé pixely.

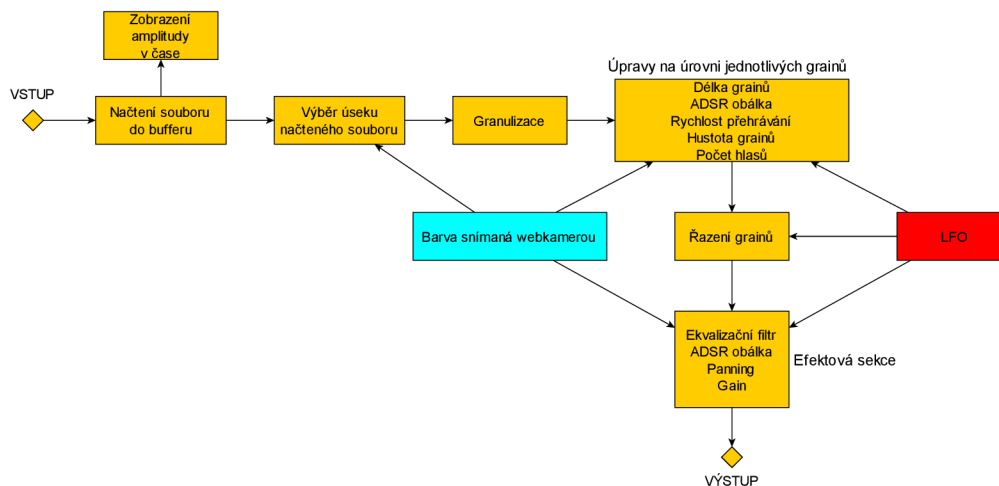


Obrázek 4.1 Srovnání 3D vizualizace modelů HSL a HSV. Zdroj: [37]

Při implementaci granulárního VST plug-inu ovládaného barvami snímanými webkamerou jsou využívány parametry modelu HSL.

4.3 Prvotní návrh

Granny Synth je experimentální granulární VST syntezátor s možností ovládání některých parametrů syntézy pomocí barev snímaných webkamerou. Základ pro jeho vytvoření stojí na třídě *juce::Synthesiser* s využitím podtříd určených pro samplery jako například *SamplerVoice* nebo *SamplerSound*. Základní struktura plug-inu je znázorněna na blokovém diagramu (obr. 4.2). Následující podkapitoly se věnují jednotlivým blokům a jejich zpracování.



Obrázek 4.2 Blokové schéma granulárního VST syntezátoru *Granny Synth*

Výše uvedené blokové schéma návrhu neodpovídá finální podobě syntezátoru, při praktické realizaci byla zvolena poněkud odlišná architektura kódu a některé funkce byly z důvodu časové náročnosti z projektu vypuštěny.

4.3.1 Grafické uživatelské rozhraní

Navrhované grafické uživatelské rozhraní je inspirováno minimalistickým designem s využitím kovových prvků. S odkazem na slovní hříčku v názvu plug-inu – *granny* znamená v angličtině domácké označení pro babičku, zároveň kořenem slova odkazuje na princip syntézy – připomíná návrh GUI zařízení z dob konce minulého století, připomínající ta, která v podobě různých domácích zvukových soustav, CD nebo DVD přehrávačů apod. mohly vlastnit babičky generace lidí narozených kolem roku 2000.

Nejedná se o první případ, kdy je v názvu objektu vztahujícího se ke granulární syntéze využito zmíněné slovní hříčky. Například jedno ze zmíněných dalších řešení, VST syntezátor *GranuLab*, využívá tohoto názvu k pojmenování ukládaných presetů – scén určených k uložení hodnot parametrů a jejich následnému vyvolání.



Obrázek 4.3 Návrh GUI granulárního VST plug-inu *Granny Synth*

Minimalistický design pro tento plug-in je do jisté míry popírán velkým množstvím různých ovládacích prvků, jež dohromady působí mírně neuspořádaným dojmem. Takovéto zpracování bylo momentální nutností pro velké množství parametrů granulární syntézy, které mělo být uživateli dáno k nastavení pro pevnou kontrolu nad průběhem výstupního signálu VST plug-inu. Během další práce byly některé funkce z projektu vyloučeny (různé typy filtrů, jejich parametry, nízkofrekvenční oscilátor, možnost výběru a další), tudíž došlo ke zjednodušení celého GUI.

Grafické uživatelské rozhraní je v JUCE implementováno pomocí předdefinovaných tříd, jejichž vzhled může být upravován tak, aby odpovídal navrhovanému vzhledu. V následujících podkapitolách věnovaných jednotlivým blokům jsou jmenovány ovládací prvky vztahující se ke konkrétním funkcím, je zde popsán jejich účel, případně také rozsah nabývaných hodnot.

4.3.2 Načtení souboru

Načtení zvukového souboru lze provést dvěma způsoby. Prvním z nich je stisknutí tlačítka *SOUBOR*, jež zajistí otevření dialogového okna umožňujícího uživateli vybrat zvukový soubor z počítače. Druhým způsobem je metoda drag and drop, tedy přetažení požadovaného souboru myši přímo do prostoru plug-inu. V obou případech je ze souboru přečten název souboru a zvuková data, jež jsou odeslána k dalšímu zpracování následujícími bloky.

Po zmáčknutí tlačítka *SOUBOR* tedy dojde k odstranění aktuálního nahraného zvuku ve třídě *SamplerSound*, dále je otevřeno dialogové okno, ve kterém uživatel

může vybrat nový zvukový soubor k načtení. V případě zvolení některého zvukového souboru je zjištěn jeho název, který je dále předán pro zobrazení na displeji plug-inu a soubor je přečten a případně dále zpracován.

V případě načtení souboru přetažením myši je vhodné se ujistit, že se jedná o zvukový soubor. Při přetažení souboru do prostoru plug-inu tedy dojde ke kontrole přípony, zda vyhovuje přípustným formátům pro zpracování a nastavení booleovské hodnoty na *true* nebo *false* podle zjištěného výsledku. Při kladné návratové hodnotě je z přetaženého souboru získán jeho název a opět předán displeji plug-inu pro zobrazení, následně je pak signál odeslán k dalšímu zpracování.

4.3.3 Zobrazení amplitudy v čase

Průběh vstupního signálu lze po načtení sledovat na displeji plug-inu. Realizace tohoto zobrazení je prováděna pomocí načtených dat o amplitudách jednotlivých vzorků a vynášena v podobě bodů do zobrazeného grafu, jehož vodorovná osa znázorňuje čas, svislá pak amplitudu. Tyto body jsou následně spojeny křivkou a prostor mezi nimi a nulovou úrovní amplitudy vyplněn spojitou barvou. Aktuálně přehrávaný vzorek je pro orientaci uživatele v průběhu signálu vyznačen ukazatelem pozice. Na displeji je také v pravém horním rohu zobrazen název aktuálního vstupního zvukového souboru.

Implementace zobrazení průběhu amplitudy v čase je prakticky prováděna pomocí získání informací o zvukové vlně nahraného souboru pomocí metody *readWaveform*. V případě, že se v načteném bufferu nachází alespoň jeden vzorek, je vytvořena nová vykreslovací cesta, předchozí body jsou smazány. Následně je pro čtení získán tento buffer s uloženými hodnotami a z jeho počtu vzorků určen poměr tak, aby výsledná vlna přesně vyplnila místo na displeji. Následně je zahájeno čtení.

Hodnoty amplitud načtených vzorků z bufferu jsou uloženy jako vektor a je vytvořena nová cesta s počátkem na displeji vlevo uprostřed. V rozsahu jeho celé výšky pak jsou po jednotlivých vzorcích umístěny body odpovídající vzorkovým amplitudám. Závěrem jsou všechny body spojeny čarou a prostor mezi nimi a nulovou hodnotou amplitudy vyplněn.

Je nutné, aby ukazatel pozice v načteném souboru procházel soubor rychlostí odpovídající frekvenci, kterou je přehráván. K tomuto účelu slouží jeho nastavení v metodě *processBlock*, jež je součástí třídy *PluginProcessor*.

4.3.4 Výběr úseku načteného souboru

Pro provádění granulózní syntézy lze vybrat také konkrétní úsek vstupního signálu. Toto rozmezí je nastavováno dvojicí otočných ovládacích prvků *VÝBĚR MIN.* a *VÝBĚR MAX.* Jejich potencionální rozsah sahá od prvního až k poslednímu vzorku původního souboru, přičemž je vhodné ošetřit, aby nebylo možné nastavit parametr začátku úseku na vyšší vzorek, než na kterém výběr končí, a naopak. Vybraný úsek je zvýrazněn pomocí dvou indikátorů a potměním nevybraných oblastí.

4.3.5 Granulizace a práce s grainy

Pro provedení granulární syntézy je klíčovým krokem granulizace – rozdělení vstupního signálu na časové úseky s délkou přibližně 10–100 ms. Tuto možnost dává ovládací prvek *DĚLKA* s rozsahem 2–200 ms umístěný v sekci *GRAIN* vlevo. Současná podoba návrhu je zamýšlena pouze pro fixní délku zrn, v další práci by mohlo být pracováno s odchylkou doby trvání v nastavitelném rozsahu. Implementace v kódu může být zajištěna, s ohledem na různé možnosti vzorkovací frekvence vstupních zvukových dat, dvěma způsoby. Prvním z nich je převzorkování vstupního signálu na jednotnou hodnotu, např. 48 kHz, což ovšem může způsobovat relativně malé, ovšem stále rozpoznatelné změny barvy zvuku při podvzorkování vstupních zvukových souborů s vyšší vzorkovací frekvencí, např. 96 nebo 192 kHz. Tento problém není uspokojivě řešitelný ani nadvzorkováním na vyšší frekvenci, než je běžný standard pro zvukovou produkci (48 kHz), jelikož tímto způsobem není možné zajistit opětovné získání informace o podobě zejména vysokých frekvencí. Naopak, tento proces může být zdrojem nežádoucích artefaktů vzniklých interpolací již existujících hodnot pro vznik přidávaných vzorků.

Druhou možností určení délky zrn v sekundách na základě známé vzorkovací frekvence je přiřazení hodnoty počtu vzorků na sekundu a její následné uložení do proměnné. Hodnota počtu vzorků je rovna převrácené hodnotě vzorkovací frekvence. Tento postup ošetřuje jednak standardně použitelné vzorkovací frekvence (CD standard – frekvence 44,1 kHz a její násobky mocninou čísla 2 i standard pro zvukovou produkci – frekvence 48 kHz a její násobky mocninou čísla 2), zároveň také i jakoukoliv možnou použitou vzorkovací frekvenci vstupní nahrávky.

Na takto vytvořená zrna jsou následně aplikovány další parametry: ADSR obálka, rychlost přehrávání a překryv (hustota) zrn. Nenulové hodnoty zejména parametrů *Attack* a *Release* jsou klíčové pro odstranění přechodových jevů mezi vzorky s různými hodnotami počáteční a koncové amplitudy, jež se projevují jako slyšitelné klikání nebo praskání. Rychlost přehrávání ovlivňuje základní frekvenci granulizovaného signálu a její iniciální hodnota je 1,0. Rozsahem tento parametr pokrývá škálu 0,1–10,0. Překrytí/hustota zrn je v momentálním návrhu zpracováno způsobem synchronní granulární syntézy, nastavitelná je fixní hodnota aplikovaná na všechna zrna. Podobně jako u délky grainů je ovšem možné i zde při budoucí práci přidat parametr náhodné odchylky a přinést tak výstupu syntézy určitou asynchronnost. Rozsah parametru překrytí je iniciálně nastaven na 0 % a může nabývat hodnot mezi –500–75 %. Provedeno může být opět dvěma způsoby v závislosti na zvoleném postupu pro určení délky grainů (převzorkování nebo využití proměnné udávající počet vzorků na sekundu.

4.3.6 Řazení grainů

Nastavení řazení grainů je prováděno na třech tlačítkách v sekci grain s označením *R*, *A* a *F*. Tato písmena jsou zkrácenou podobou způsobů řazení. *R* označuje *RANDOM*,

tedy náhodné řazení, *A* je zkratkou pro *AMPLITUDE* a má za následek seřazení zrn vzestupně podle zjištěné průměrné amplitudy. Zkratka *F* slouží pro *FREQUENCY*, v tomto režimu jsou grainy řazeny vzestupně podle jejich zjištěné průměrné frekvence.

4.3.7 Efektivní sekce

Součástí tohoto oddílu je ADSR obálka aplikovaná na celý výstupní signál, ekvalizační filtry (horní a dolní propust, *peak*) s nastavitelnou mezní nebo střední frekvencí a jakostí (šířkou pásma). Přepínání mezi typy filtru se provádí stisknutím tlačítka *FILTR*. Existují zde režimy *HP* (horní propust), *DP* (dolní propust), *P* (*peak*) a *OFF* (nezapojení filtru do průběhu signálu). V efektové sekci se dále nachází nastavení panoramy (stereofonního rozložení zvuku) a zesílení výsledného signálu.

4.3.8 LFO

Poslední sekcí ovládacích prvků klasického typu jsou parametry nízkofrekvenčního oscilátoru. Rozsah frekvence tohoto zařízení se nachází mezi 0,1–20 Hz. LFO je aplikovatelné na všechny parametry řízené otočnými ovládacími prvky s výjimkou rozsahu úseku pro zdroj grainů.

Míra ovlivnění jednotlivých parametrů oscilátorem je nastavována stisknutím tlačítka LFO. Tímto je spuštěn režim indikace vlivu LFO – rozsah všech otočných ovládacích prvků je změněn na 0–100 % a nastavená hodnota udává hloubku aplikování modulační vlny.

4.3.9 Řešení změny parametrů pomocí barvy

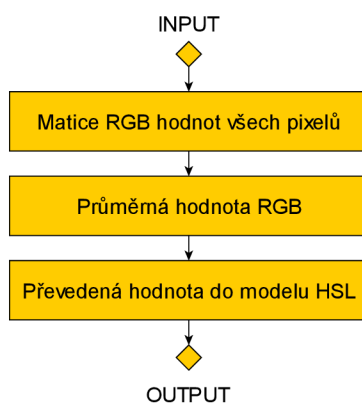
Jedná se o experimentální složku celé práce. Aktivace této funkce je prováděna stisknutím tlačítka s ikonou RGB soustavy. V rolovacím menu lze následně zvolit zařízení, které bude použito jako snímač barev a po spuštění je možné zvolit kliknutím na příslušný ovládací prvek parametry ovládané barvou. Některé bloky, jako například ADSR nebo délka, jsou ovládané jasovou složkou (parametrem *Lightness* barevného modelu HSL), jiné, kupříkladu střední nebo mezní frekvence ekvalizačního filtru jsou ovládané hodnotou *Hue* ve stejném barevném modelu. Problémem použití parametru *Hue* je jeho cyklická povaha – je udáván jako stupňová míra na standardním barevném kruhu. V praxi to znamená extrémní změnu hodnoty u dvou velmi podobných odstínů červené, z nichž každý je na opačné straně odstínového spektra (0° a 359°). Využití této barvy by mohlo vytvářet nežádoucí skoky mezi minimální a maximální hodnotou ovládaného parametru. Možným řešením tohoto problému je výpočet hodnoty pro nastavení parametrů dle vztahu (4.1):

$$H = |\cos(Hue)|, \quad (4.1)$$

kde H je novou vypočtenou hodnotou využitou k ovlivnění parametrů granulární syntézy a Hue je hodnota odstínu barevného modelu HSL. Tento přepočet by měl za následek dva odstíny nastavující maximální (0° a 180°) i minimální (90° a 270°) hodnotu. Změna mezi těmito extrémy by pak nebyla lineární, nýbrž by probíhala podle goniometrického předpisu. Toto řešení vyžaduje další zkoumání, zda odpovídá intuitivnímu vnímání změny barev v závislosti na změně parametrů například zmiňovaného ekvalizačního filtru.

Problém způsobený kombinací parametrů Hue a $Lightness$ vyvstává i v případě čtení velmi světlých nebo naopak velmi tmavých barev. Při vysoké nebo nízké hodnotě parametru $Lightness$ se stává hodnota Hue velmi nestabilní a například při čtení hodnot modelu HSL snímaného bílého povrchu existuje riziko jejich nežádoucích skokových změn v závislosti na změnách okolního světla odráženého tímto povrchem.

Barevný model RGB je využíván jako přijímaný formát barev získávaných z webkamery a jejich prvotním zpracováním, následně je ovšem převáděn do hodnot modelu HSL. Získání jedné hodnoty z matice hodnot všech zaznamenaných pixelů je prováděno průměrováním hodnot RGB. Průběh tohoto provedení znázorňuje následující diagram (obr. 4.4).



Obrázek 4.4 Blokové schéma zpracování dat získaných webkamerou

Veškeré parametry spolu s jejich minimální i maximální hodnotou a ovlivnitelností pomocí LFO a barvy popsané v této podkapitole jsou pro shrnutí přehledně uvedeny v následující tabulce (tab. 4.1).

Tabulka 4.1 Shrnutí parametrů navrhovaného VST plug-inu *Granny Synth*

Parametr	Minimální hodnota	Maximální hodnota	Ovlivnitelnost LFO	Ovlivnitelnost barvou
VÝBĚR MIN.	První vzorek načtený v bufferu	Poslední vzorek načtený v bufferu	NE	NE
VÝBĚR MAX.	První vzorek načtený v bufferu	Poslední vzorek načtený v bufferu	NE	NE
DĚLKA	2,0 ms	200,0 ms	ANO	ANO
ATTACK	0,0 ms	10,0 s	ANO	ANO
DECAY	0,0 ms	10,0 s	ANO	ANO
SUSTAIN	0,0	1,0	ANO	ANO
RELEASE	0,0 ms	10,0 s	ANO	ANO
RYCHLOST PŘEHRÁVÁNÍ	0,1	10,0	ANO	ANO
PŘEKRYV	-500 %	75 %	ANO	ANO
FREKVENCE	20,0 Hz	20,0 kHz	ANO	ANO
ČINITEL JAKOSTI	0,26	10,0	ANO	ANO
LFO FREKVENCE	0,1 Hz	20,0 Hz	NE	ANO
PANORAMA	-100 %	100 %	ANO	ANO
ZESÍLENÍ	$-\infty$	+6 dB	NE	ANO

5. REALIZACE VLASTNÍHO ŘEŠENÍ

Jedním z výsledků této bakalářské práce je funkční prototyp nástroje *Granny Synth*. Jedná se o experimentální VST a standalone plug-in fungující v režimech jednoduchého a synchronně-granulárního samplu. V této kapitole je popsána realizace nástroje, popis jeho funkčních bloků a experimentální části nástroje – nastavení parametrů granulární syntézy pomocí barevné složky světla.

Stručný popis jednotlivých bloků je obsažen v příloženém uživatelském manuálu, jenž je součástí přílohy, tato kapitola se věnuje techničtějšímu popisu jednotlivých bloků a parametrů. Nástroj je složen ze dvou hlavních částí – zobrazovacího displeje a sady tlačítek a otočných ovládacích prvků sloužících k řízení parametrů granulární syntézy.



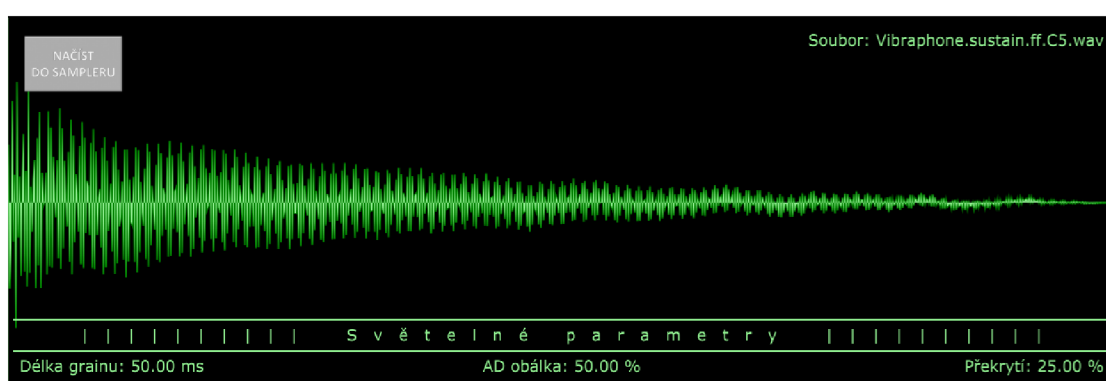
Obrázek 5.1 Grafické uživatelské rozhraní granulárního samplu *Granny Synth*

5.1 Grafické uživatelské rozhraní

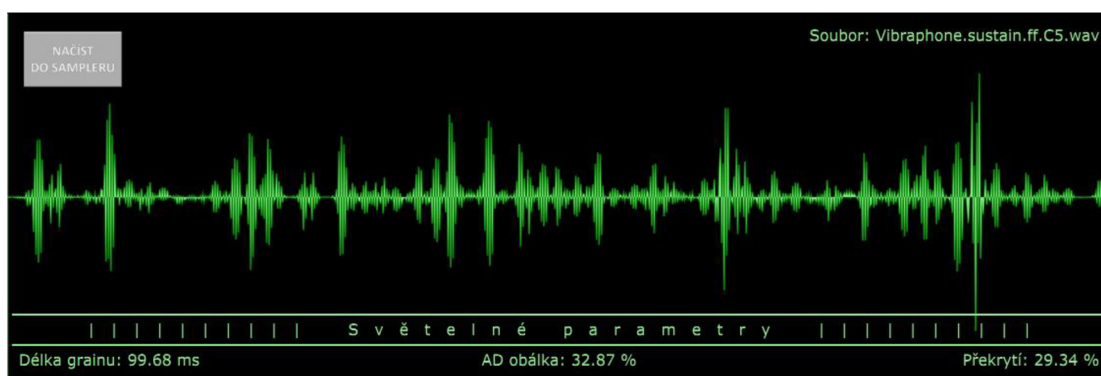
GUI nástroje dává uživateli k dispozici sadu prvků ovladatelných myší, rozdělených do skupin tlačítek ovlivňujících režimy, ve kterých nástroj pracuje, kruhových posuvníků pro MIDI Root Note a ohraničení zóny samplu, lineárního posuvníku pro nastavení zesílení výstupního signálu, sady pro nastavení parametrů granulizace, panoramy a ADSR obálky pro výstupní signál. Granulizace samplu od uživatele přijímá následující

parametry: délku grainu v milisekundách, lineární AD obálku jednotlivých grainů a překrytí určující hustotu generování grainů v procentech délky grainu parametry ADSR obálky sampleru a MIDI Root Note určující, na které notě se bude přehrávat sample v původní výšce. Tlačítka slouží k aplikaci parametrů syntézy čtených z ovládacích prvků nebo z webkamery, případně spuštění režimu jednoduchého sampleru.

Po načtení signálu dojde na displeji ke zobrazení přehrávaného samplu, případně již granulizovaného výstupu. V pravém horním rohu je vypsán název zvukového souboru, na dolní liště jsou zobrazeny poslední zaznamenané parametry určené analýzou barvy. Orientaci pozice v samplu napomáhá čárový ukazatel pohybující se rychlostí poslední stlačené klávesy.



Obrázek 5.2 Detail displeje při zobrazení nahraného samplu v režimu jednoduchý sampler



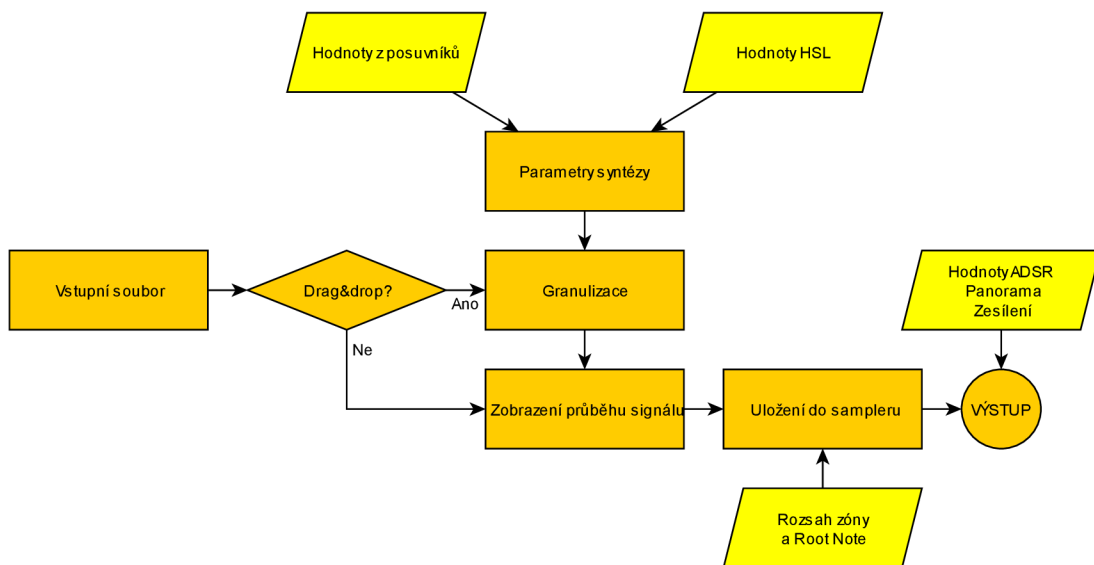
Obrázek 5.3 Detail displeje při zobrazení nahraného samplu v režimu granulární sampler

Vzhled grafického uživatelského rozhraní nepůsobí výrazně přelácaně, sestává z prvků v odstínech černé, šedé a zelené barvy. Displej zařízení svým striktním použitím kombinace černé a výrazné zelené barvy připomíná starší CRT monochromatické obrazovky s využitím zeleného fosforu.

5.2 Struktura nástroje

Finální struktura prototypu granulárního sampleru *Granny Synth* je odlišná od původně navrhované struktury v rámci semestrální práce. Došlo k některým změnám, které přirozeně vyplynuly z práce na projektu v podobě nápadů na rozlišení režimů, ve kterých bude nástroj pracovat, k omezení funkcionality v podobě nerealizace nízkofrekvenčních oscilátorů, filtrů a náhodného rozptylu kolem nastavených hodnot parametrů.

Současná struktura odpovídá blokovému diagramu uvedenému na obrázku 5.4.



Obrázek 5.4 Diagram znázorňující strukturu granulárního sampleru *Granny Synth*

Schéma ukazuje cestu vstupního signálu a jeho ovlivnění hodnotami určenými vstupem uživatele (žlutě vyznačené bloky). Je patrné, že nahráním souboru přetažením ze zdrojové složky dojde automaticky ke granulizaci podle momentálně nastavených parametrů granulární syntézy, naopak soubor vybraný pomocí dialogového okna nebude granulizací zasažen a aplikovány na něj budou pouze hodnoty obálky ADSR.

5.3 Režim jednoduchého sampleru

Nástroj pracuje v režimu jednoduchý sampler při nahrání souboru pomocí tlačítka umístěného v prostoru displeje nebo po stisknutí tlačítka „De-granulizace“. V prvním případě dojde ke spuštění dialogového okna s možností výběru zvukových souborů v povolených formátech – WAV, MP3, FLAC, AIFF, WMA a OGG. Po zvolení souboru proběhne jeho přečtení a uložení do bufferu *originalBuffer* a jeho načtení do sampleru, jak je ukázáno v následujícím příkladu kódu.

```

*****
void loadFileViaButton()
{
    FileChooser chooser{ "Vyberte soubor", ... };
    if (chooser.browseFileForOpen())
    {
        auto file = chooser.getResult();
        myFormatReader = myFormatManager.createReaderFor(file);
    }
    else
    {
        return;
    }
    originalBuffer.setSize(myFormatReader->numChannels, ...);
    myFormatReader->read(&originalBuffer, ... );

    degranulize();
}
*****

```

Buffer *originalBuffer* je v projektu využíván jako úložiště vzorků původního souboru. Jedná se o veřejný objekt ve třídě *PluginProcessor* a jako takový je teoreticky dostupný všem ostatním objektům a třídám z této třídy odvozeným. Metoda *degranulize()* je společnou metodou zajištění uložení obsahu objektu *originalBuffer* do třídy *Juce::Sampler*.

```

*****
void degranulize()
{
    myGrannySynth.clearSounds();
    if (originalBuffer.getNumSamples() > 0)
    {
        readWaveform(originalBuffer);
        BigInteger range;
        range.setRange(0, 128, true);
        myGrannySynth.addSound(new SamplerSound("Sample", ...);
        updateADSR();
    }
}
*****

```

V tomto okamžiku lze nástroj používat jako jednoduchý jednozónový sampler s proměnlivým vzorkovacím kmitočtem a možností opakování nekonečné smyčky.

5.4 Režim granulárního sampleru

Režim granulárního sampleru je spuštěn, pokud je soubor do sampleru nahrán přetažením ze zdrojové složky, případně aplikací parametrů granulární syntézy na sample v jednoduchém sampleru. V obou případech je použit obsah bufferu *originalBuffer*, jenž je dále zpracován metodou *granulisation()*. Dle nastavených

parametrů v milisekundách a procentech délky grainu je s ohledem na vzorkovací kmitočet vypočtena hodnota těchto parametrů ve zvukových vzorcích, s nimiž pracuje třída *AudioBuffer<float>*. Způsob, jakým jsou počty vzorků určeny, je nastíněn ve zkráceném výpisu zdrojového kódu níže.

```
*****  
void granulation()  
{  
    float grainLengthInMs = myValTrSt.getRawParameterValue( ... );  
    int grainSize = static_cast<int>(grainLengthInMs*sampleRate/1000);  
  
    ...  
  
    float overlapInPercent = myValTrSt.getRawParameterValue( ... );  
    int overlapInSamples = static_cast<int>(grainSize*overlapPercent/100);  
  
    ...  
  
    float attackInPercent = myValTrSt.getRawParameterValue( ... );  
    int attackInSamples = ...;  
  
    ...  
}  
*****
```

Podobným způsobem jsou vypočítány všechny počty zvukových vzorků. Komunikace ovládacích prvků nastavení a parametrů granulární syntézy je zajištěna pomocí proměnné *myValTrSt*, instancí objektu *ValueTreeState*, jenž sleduje, uchovává a nastavuje hodnoty a rozsahy posuvníků. Dle těchto parametrů pak probíhá kopírování vzorků z původního bufferu a ukládání na náhodně určené indexy dočasného bufferu. Tímto postupem je zaručeno rozdělení vstupního souboru na jednotlivé grainy a jejich náhodné přeskládání spolu s aplikací lineární AD obálky.

Po naplnění dočasného bufferu všemi grainy získanými ze vstupní nahrávky je jeho obsah kopírován do finálního bufferu s aplikací překrytí grainů. Vzorky, jež jsou tímto překrytím zasaženy, jsou průměrovány s překrývanými vzorky začátku dalšího bufferu (což je v kontrastu s klasickým překrytím u jiných granulárních syntezátorů, kde jsou překrývané vzorky sčítány). Tímto průměrováním je dosaženo vzniku velmi krátkých průběhů (maximálně 50 ms) v původním signálu vůbec neobsažených a tím i ke vzniku nových zvukových barev.

5.5 Nastavitelné parametry sampleru

Tato podkapitola je věnována uživatelem nastavitelným hodnotám granulární syntézy vstupního samplu a jejich rozsahům pro lepší orientaci v dosažitelných hodnotách používaných nástrojem. Pro přehlednost jsou tyto parametry a jejich teoretické rozsahy uvedeny v tabulce 5.1.

Tabulka 5.1 Tabulka parametrů granulárního sampleru *Granny Synth*

Parametr	Rozsah v původních jednotkách	Rozsah v milisekundách	Jednoduchý sampler / granulární sampler
MIDI Root nota	0–127 (–)	–	Jednoduchý i granulární sampler
Délka grainu	0,1–100 ms	0,1–100	Granulární sampler
Attack grainu	0–100 %	0–100	Granulární sampler
Decay grainu	0–100 %	0–100	Granulární sampler
Překrytí	0–50 %	0–50	Granulární sampler
Panorama	–100–100 %	–	Granulární sampler
Attack	0–10 s	0–10 000	Jednoduchý i granulární sampler
Decay	0–10 s	0–10 000	Jednoduchý i granulární sampler
Sustain	0–1 (–)	–	Jednoduchý i granulární sampler
Release	0–10 s	0–10 000	Jednoduchý i granulární sampler

Hodnota délky grainu platí pouze pro notu odpovídající MIDI Root Note – tedy té notě, na které sampler přehrává nahrávku v původní výšce. Při změně výšky tónu je změněna délka nahrávky, a tedy i délka grainu podle následujícího vztahu (5.1):

$$t_1 = t_0 \cdot 2^{\frac{\text{MIDI Note} - \text{MIDI Root Note}}{12}}, \quad (5.1)$$

kde MIDI Note je číslo stisknuté noty a MIDI Root Note je číslo MIDI noty zvolené pro přehrávání samplu v původní výšce. Dané řešení není ideální a je spíše následkem nedostatku programátorských zkušeností a kvalifikace než záměrem, nicméně po umělecké stránce nástroji nikterak zásadně kvalitu neubírá.

5.6 Nastavení parametrů pomocí webkamery

Experimentální složkou této práce je možnost nastavení parametrů granulární syntézy pomocí barev zaznamenaných webkamerou. Původním záměrem byla možnost kontinuálního snímání a průběžné změny parametrů syntézy, toto řešení se však ukázalo

jako příliš výpočetně náročné vzhledem k algoritmu, jakým jsou grainy tvořeny a ukládány, znamenalo by neustálý proces přepisování obsahu sampleru a pro jeho implementaci by bylo nutné od základu změnit způsob, jakým jsou grainy tvořeny. Proto bylo nakonec zvoleno řešení podstatně více odpovídající již existující struktuře sampleru, tedy vyhodnocení průměrné hodnoty HSL pixelů vytvořené fotografie.

Současný proces ovládání parametrů granulární syntézy tedy vypadá takto:

```
*****
void colourModifier()
{
    myCameraDevice->takeStillPicture([this] const juce::Image&){
        this->averagePixel(image);
    });
}
*****
```

Nejprve je tlačítkem spuštěn proces nastavení parametrů pomocí fotografie. Ta je vytvořena voláním metody *takeStillPicture*, jež z interního či připojeného zařízení sejme aktuálně zaznamenaný obraz, jenž slouží jako obrázek k analýze předaný dále metodě *averagePixel*.

```
*****
void averagePixel(const Image& img)
{
    std::vector<Colour> pixelColours;
    std::vector<float> pixelHue;
    std::vector<float> pixelSaturation;
    std::vector<float> pixelLightness;

    for (int row = 0; row < img.getHeight(); row++)
    {
        for (int pixel = 0; pixel < img.getWidth(); pixel++)
        {
            pixelColours.push_back(img.getPixelAt(pixel, row));
        }
    }

    for (int pixel = 0; pixel < pixelColours.size(); pixel++)
    {
        pixelHue.push_back(pixelColours[pixel].getHue());
        pixelSaturation.push_back(pixelColours[pixel].getSaturation());
        pixelLightness.push_back(pixelColours[pixel].getLightness());
    }

    ...

}
*****
```

V této metodě je nejprve vytvořen vektor objektů *Colour*, do něhož jsou postupně načteny barvy všech pixelů vytvořené fotografie. Následně je tento vektor čten a do tří

dalších vektorů pro jednotlivé parametry modelu HSL jsou ukládány jejich hodnoty. V další části této metody je vytvořen průměr jednotlivých hodnot odstínu, sytosti a světlosti, jež nabývají hodnot 0,0 až 1,0. Pro aplikaci vzorce (4.1) na průměrnou hodnotu odstínu je nutné její rozsah zvýšit na 0 až 360 úhlových stupňů. Dále pak vzhledem k rozsahům hodnot přiřazených parametrů syntézy (odstín ovládá délku grainu, sytost ovládá symetrickou AD obálku a světlost ovládá překrytí) je třeba získané průměry přizpůsobit rozsahům parametrů, jak lze vidět na následující ukázce kódu.

```
*****  
void averagePixel(const Image& img)  
{  
    ...  
  
    this->averageHueRanged = 100 * abs(cos(365 * averageHue));  
    this->averageSaturationRanged = 100 * averageSaturation;  
    this->averageLightnessRanged = 50 * averageLightness;  
  
    ...  
  
}  
*****
```

V poslední části této metody je volána metoda *granulisationByColour*, jež je přizpůsobenou variantou metody *granulisation* a jako parametry přijímá právě hodnoty průměrného odstínu, sytosti a barvy rovnajících se příslušným parametrům granulární syntézy. Původním záměrem pro zachování jednoduchosti kódu bylo načítat hodnoty průměrných proměnných do objektu *ValueTreeState*, funkční komunikaci těchto dvou částí kódu se nicméně nepodařilo včas provést.

6. TESTOVÁNÍ NÁSTROJE A NÁVRH BUDOUCÍCH ÚPRAV

V této kapitole jsou popsány výsledky testování standalone a VST3 verze nástroje. V rámci pokusného použití v DAW Reaper vzniklo několik krátkých nahrávek, jež jsou přílohou této práce. Tyto zvukové výstupy přibližně nastiňují potenciál, jenž prototyp nástroje nabízí pro zvukovou tvorbu. Obsahem kapitoly je také posouzení silných stránek a slabin nástroje. Dále jsou popsány návrhy na možné úpravy projektu směrem k jeho zkvalitnění a rozšíření funkcionalit, které může nástroj nabídnout.

6.1 Testování nástroje

Pro účely ilustrování kreativních možností nástroje byla vytvořena sada šesti zvukových souborů živě nahrávaných improvizací s využitím různých samplů – jednalo se o samplu vibrafonu, klavíru, celesty, violoncella, syntezátoru Moog a zkreslených digitálních varhan z banky vytvořené společností Akai. Podstata nástroje uživatele inspiruje spíše k použití za účelem vytváření nálad a komplexnějších harmonických struktur, ačkoliv špatných výsledků nástroj nedosahuje ani při využití k tvorbě melodických linek.

Největším problémem pro použití nástroje během živé produkce je nepochybně chybějící možnost mapování virtuálních ovládacích prvků na reálné ovladače kontroléru a také nemožnost multisamplingu, jenž by pomohl větší bohatosti okamžitě použitelných zvuků, ačkoliv tato možnost je technicky nahraditelná spuštěním více instancí VST3 formy nástroje v rámci stop DAW. Toto ale rozhodně není ideální řešení, jelikož představuje zvýšení náročnosti přípravy na produkci.

Naopak poměrně zajímavou je možnost nastavovat délku grainů na hodnoty nižší, než je běžný rozsah granulárních syntezátorů – mezi 0,1–10 ms. Takto nízké nastavení nachází uplatnění při šumových kompozicích, jelikož při zmíněném rozsahu délek grainů již není možné rozeznat výšku tónu, tudíž se zvukovým výstupem nástroje stává šum. Kombinace nízké délky grainu a různá nastavení AD obálky vytvářejí zajímavou paletu netónových zvuků a ruchů, jež mohou dotvářet ambientní atmosféru pro vrstvení dalších tónových i netónových složek, jak lze slyšet na nahrávce s použitím granulizované celesty (na začátku a na konci syntetizované v šum a různé ruchy).

Kombinace krátké fáze attack a delší fáze decay vytváří ostře a poněkud perkusivně znějící grainy připomínající nástroje, jejichž průběhy znějí podobně – zvukový výstup v tomto případě evokuje hru drnkáním, jehož četnost se odvíjí od parametrů délky grainu a překrytí. Takový příklad nabízí nahrávka granulizovaného violoncella.

Nástroj je vhodný k vytváření ambientních struktur, zvukomaleb a tvorbě s využitím šumů a ruchů. Použití samplů neobsahujících hudební tóny nebo naopak části zvukových nahrávek, ve kterých se již harmonická struktura nachází, přináší možnost

tuto hudbu znovu skládat do nových struktur mimo harmonii a experimentovat s napodobováním zvuků z oblasti přírody nebo různých prostředí.

6.1.1 Práce s parametry

Uživateli je dána možnost nastavit relativně omezenou nabídku parametrů granulární syntézy a sampleru. Vzhledem ke způsobu, jakým je nástroj řešen, se problémem jeví absence schopnosti měnit parametry u již stisknuté noty, čehož umožnění by jistě zvýšilo tvůrčí potenciál nástroje, nicméně existuje možnost kontinuálního snímání parametrů při každé změně polohy jakéhokoliv posuvného ovládacího prvku, jež nabízí alespoň schopnost nastavit hodnoty parametrů mezi stisknutím jednotlivých tónů a tím dosáhnout jednodušší proměny výstupního signálu alespoň v rámci dvou a více hraných not. Vzhledem k současné náročnosti a nutnosti ovládání myší, touchpadem nebo jinou periferií připojenou k počítači byla u nástroje zachována i možnost měnit parametry jednorázově se změnou aplikovanou po stisknutí tlačítka nastavujícího hodnoty pro granulární syntézu.


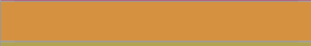





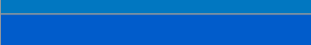
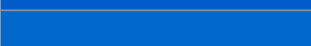
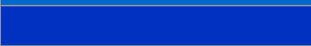









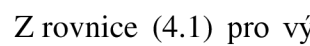
6.1.2 Nastavení parametrů granulární syntézy pomocí analýzy průměrné barvy

Testováním byla ověřena také funkce nastavení parametrů granulizace pomocí průměrování hodnot modelu HSL. Z hlediska navrhovaného kontinuálního snímání došlo u projektu ke zjednodušení v podobě snímání pouze statického obrazu a jeho analýzy pro jednorázové nastavení parametrů, nikoli kontinuálního snímání a analýzy řady v krátké době zaznamenaných obrazů. Toto řešení bylo zvoleno kvůli výraznému zpomalení běhu programu a zhoršení kvality zpracování zvukových dat při pokusném použití opakovaného volání granulizace prakticky vylučujícím použití pro živou produkci s využitím softwarového nástroje. Kompromisním řešením mezi navrhovaným i současným stavem by mohlo být snížení frekvence, s jakou jsou obrazy zpracovány, jež by zachovalo možnost průběžného získávání obrazů a vytvořilo jakousi analogii možnosti nastavovat parametry kontinuálně pomocí ovládacích prvků, nicméně do prototypu nástroje popisovaného v této práci již nebylo toto řešení z časových důvodů zahrnuto.

V rámci ověření funkce průměrování barev byla vytvořena testovací série dvaceti obrázků pomocí kódu umožňujícího uložení zachycené fotografie do souboru ve formátu PNG. Předmětem fotografie byl v 16 případech běžný pokoj laděný do červené, oranžové a hnědé barvy s osvětlením pomocí přepínatelného LED pásku. Tímto světelným zdrojem bylo zajištěno ovlivnění barvy zaznamenané webkamerou. Zbývající čtyři fotografie vznikly pod bílým osvětlením a zachycovaly naprostou tmou zajištěnou překrytím webkamery, lidský obličej zblízka (přibližně 10 cm), ruku ze střední vzdálenosti (přibližně 30 cm) a ruku zblízka (přibližně 10 cm). Parametry HSL byly vypočítány zpětným přepočtem z nastavených parametrů délky grainu (dle vztahu (4.1) modifikovaného pro proměnnou *averageHueRanged* a vyjádřením parametru odstínu ve

stupních), AD obálky (jejíž hodnota by se teoreticky měla rovnat sytosti barvy v procentech) a překrytí (výsledek zpětného výpočtu parametru světlosti byl zjištěn zdvojnásobením hodnoty). Referenční hodnota průměrného HSL byla zjištěna pomocí webové stránky *Get Average Color of Image*. [38] Zjištěné výsledky jsou uvedeny v tabulce 6.1 a 6.2.

Tabulka 6.1 Hodnoty zpětně vypočítané z parametrů granulární syntézy určených pomocí průměrné barvy obrázku

Barva zaznamenaná <i>Synth Granny</i>	Vypočítaná hodnota odstínu (°)	Vypočítaná hodnota sytosti (%)	Vypočítaná hodnota světlosti (%)
	338	91	42
	33	64	54
	51	38	51
	42	36	54
	29	37	54
	180	96	31
	198	98	38
	203	99	38
	213	99	40
	209	99	40
	225	99	38
	226	98	42
	238	93	47
	254	88	49
	269	88	49
	294	38	50
	325	98	1
	48	38	42
	196	39	54
	94	17	44

Z rovnice (4.1) pro výpočet délky grainu z průměrného odstínu obrázku vyplývá, že totožná délka může být nastavena použitím nejméně dvou a nejvíce čtyř odstínů. Při zpětném propočtu tedy byly řešením rovnice 2–4 různé výsledky. Vybrán byl vždy výsledek číselně nejbližší referenční hodnotě, případně výsledek nejbližší k odstínu, který byl posouzením fotografie určen jako pravděpodobně nejdominantnější.

Tabulka 6.2 Hodnoty referenčních obrázků zpracovaných externím analyzátořem

Barva určená externím analyzátořem	Referenční hodnota odstínu (°)	Referenční hodnota sytosti (%)	Referenční hodnota světlosti (%)
	357	72	42
	2	38	54
	7	16	51
	26	6	54
	33	4	53
	177	91	30
	192	92	38
	196	93	38
	206	94	39
	203	94	39
	218	95	38
	219	85	43
	230	60	49
	247	51	52
	264	53	49
	276	6	49
	330	100	0
	33	9	42
	193	18	53
	98	5	45

Zřetelný rozdíl mezi hodnotami určenými zpětným výpočtem z výstupu nástroje a externího analyzátořem může být způsoben rozdílnými formáty obrázků – třída *Image* používaná syntezátorem zpracovává obraz ve formě BMP – tedy formátu bez komprese, zaznamenané obrázky sloužící jako vstup externího analyzátořem byly uloženy ve formátu PNG – bezztrátového, ale kompresního formátu. Komprese u formátu PNG probíhá tzv. prediktivním kódováním, při kterém jsou hodnoty pixelů kódovány na základě předchozích kódovaných pixelů.

6.1.3 Práce s dalšími instancemi nástroje

Oproti původnímu návrhu ze semestrální práce není v současném prototypu možnost vytvořit více souběžně fungujících instancí nástroje ovladatelných více klaviaturami naráz v rámci jednoho spuštění programu, nicméně tato možnost se stále dá zajistit vícenásobným spuštěním programu a přiřazováním MIDI kontrolérů jednotlivým instancím. Podobný způsob je možné zvolit také při spuštění VST3 verze nástroje v rámci DAW. V rámci jednoho i více kanálů lze zvolit zásuvný modul nástroje a ovládat každý kanál samostatným kontroléřem, případně i různými MIDI kanály, pokud toto řešení kontrolér umožňuje.

Stejným způsobem je možné nahradit i chybějící multisampling – pomocí spuštění více instancí nástroje s nastavením různých rozsahů horizontálních zón přehrávaných MIDI not lze ovládat jednotlivé instance nástroje různými zónami MIDI klaviatury. Toto řešení bylo experimentálně vyzkoušeno, a přestože je daný způsob pouze nutným východiskem a implementace multisamplingu je z hlediska potřebného času pro přípravu produkce i obecné praktičnosti výhodnější volbou, stále se jedná o funkční řešení.

6.2 Návrh na budoucí úpravy

Nástroj v současné podobě není zdaleka ideální. Uživateli není dána absolutní možnost kontroly nad parametry granulární syntézy, samplingem a některé uživatelem využitelné zajímavé funkce v současnosti chybí. Není zároveň zcela využit potenciál experimentální složky celého nástroje – ovládání paramterů syntézy pomocí parametrů světelného spektra. Většina těchto současných nedostatků je způsobena momentální nedostatečnou kvalifikací autora v oblasti programování a omezeným časem na realizaci nápadů, které by mohly být v budoucnu k funkcionalitám nástroje přidány. V následujících podkapitolách jsou rozebrány některé návrhy, jejichž realizací by byla zvýšena umělecká využitelnost nástroje.

6.2.1 Multisampling

Možnost multisamplingu by mohla výrazně pomoci při řešení problému odchylky zadané a reálné délky grainu u not vzdálených od *MIDI Root Note*. Použití více samplů by sice zvýšilo výpočetní náročnost při granulizaci, nicméně toto zvýšení by bylo zanedbatelné při omezení maximální délky nahraného samplu na jednotky sekund (současné omezení je z důvodu testování nastaveno na 40 minut). Zároveň by tato funkce dopomohla lepšímu využití nástroje jako plnohodnotného sampleru.

Pro vytvoření této funkce by bylo potřeba přidat části kódu přiřazující zvukový soubor uživatelem nastavenému rozsahu a vytvořit nové přívětivé GUI pro implementaci multisamplingu.

6.2.2 Rozptyl a modulace parametrů

Dalším způsobem rozšíření funkcionality nástroje může být původně zamýšlený náhodný rozptyl parametrů kolem určené hodnoty, modulace těchto parametrů pomocí nízkofrekvenčních oscilátorů a zpracování výstupního signálu soustavou různých filtrů a efektů. Prvek náhodného rozptylu například hodnoty panoramy by obohatil prostorové vnímání zvuku nástroje, možnost náhodné délky grainů či překryvu by způsobila vznik jisté asynchronicity syntézy, a naopak možnost modulace těchto i dalších parametrů pomocí nízkofrekvenčního oscilátoru by přinesla jistou kontrolu nad drobnými změnami parametrů, jež v současné době u nástroje chybí.

6.2.3 Aplikace časového roztážení a převzorkování v režimu granulárního sampleru

Současný problém odchylky délky grainu u not vzdálených od *MIDI Root Note* by mohl být také řešen aplikací *time-stretchingu* (časového roztážení) výstupu sampleru. S touto myšlenkou bylo při vývoji nástroje již pracováno, nakonec ale nebyla kvůli své náročnosti do tohoto prototypu zahrnuta. Její aplikace v praxi by mohla vypadat takovým způsobem, že by byla vypočítána odchylka reálné délky grainu od zadaného parametru, která by poté implementací time-stretchingové knihovny pracující v reálném čase mohla být výrazně až úplně zredukována. Otázkou zůstává, jak výpočetně náročný by tento proces byl, jaký by existoval vliv na kvalitu a barvu zvuku, či zda by nebyl dostupný jiný způsob, který by zmíněnou funkci zastával při jednodušší implementaci.

6.2.4 Kontinuální snímání barev a další formy získávání obrazu pro analýzu

Původně zamýšlený koncept ovlivňování parametrů granulární syntézy spočívající v neustálém snímání obrazu webkamerou, analýzou průměrných hodnot HSL a jejich aplikací na parametry syntézy byl zjednodušen z důvodu vysoké výpočetní náročnosti při implementaci do stávajícího algoritmu nástroje. Pro jeho funkční a méně náročné začlenění do systému by byly nutné významné změny ve fungování celého nástroje a tvorba jednotlivých grainů by pravděpodobně probíhala postupně, nástroj by tedy nefungoval jako sampler, ale jako klasický granulární syntezátor.

Dalším zdrojem získávání průměrných barev by mohly být uživatelem dodaná média – fotografie a videa, jejichž analýzou vzniklé parametry by byly schopny dodat jistou opakovatelnost nastavení například při živé produkci, u které může docházet k častým změnám světelných podmínek. Dále by tato možnost nabídla variantu přímé transformace obrazového umění do umění zvukového.

6.2.5 MIDI Learn

Při použití kontinuálního snímání hodnot parametrů na posuvných ovládacích prvcích by jejich nastavování výrazně usnadnila možnost mapování ovládacích prvků na fyzické ovladače kontroléru, kterým je nástroj řízen. Výrazně by se tak snížila vzdálenost, kterou musí uživatel překonat pohybem rukou pro dosažení změny parametru, což by prospělo plynulosti změn hodnot a obecně jednoduššímu používání nástroje.

7. ZÁVĚR

V rámci této bakalářské práce byla shrnuta stručná historie a principy granulární syntézy a samplingu, dále proveden průzkum a srovnání volně dostupných softwarových granulárních syntezátorů. Na základě tohoto výzkumu pak byl navržen, vytvořen a otestován vlastní prototyp granulárního sampleru *Granny Synth* s experimentálním ovládním parametrů granulizace pomocí analýzy barev registrovaných webkamerou, popsány jeho parametry, slabiny a navrženy kroky pro zlepšení uživatelské přívětivosti a zvýšení kreativního potenciálu nástroje. Mezi ně patří například mapování ovládacích prvků nástroje na fyzické ovladače MIDI kontroléru, multisampling, looping sustain fáze samplu nebo podrobnější možnost nastavení parametrů granulizace vstupního zvukového signálu.

Pro vývoj experimentálního prototypu byl využit JUCE, framework programovacího jazyka C++ pro vývoj zvukového a obrazového software. Cíle práce byly z velké části splněny, byl vyvinut funkční prototyp experimentálního nástroje, ve srovnání s prozkoumanými volně dostupnými alternativami poměrně ojedinělý, což není ve všech aspektech výhodou. Zejména z časových důvodů nebyly beze zbytku splněny všechny cíle práce, některé původně zamýšlené funkce součástí prototypu nejsou, případně jsou zpracovány ve zjednodušené podobě.

Tato bakalářská práce a v rámci ní vyvíjený prototyp jsou zejména vhodné pro zvukové tvůrce, kteří teprve granulární syntézu objevují a experimentují s jejími nejrůznějšími výstupy. V současné době sice nenabízí plnou kontrolu nad všemi myslitelnými parametry, nicméně množství ovládacích prvků poskytuje dostatečnou možnost a míru uživatelského vstupu pro syntetizování zajímavých zvukových krajin, harmonických i melodických struktur a rozmanitých tónových a netónových kompozic. Možnost řízení parametrů granulizace pomocí analýzy barev zachycených interní či externí kamerou zároveň může nástroj činit zajímavým pro multimediální tvůrce či experimentátory, ačkoliv je tato funkce zatím použitelná pouze v základní jednoduché podobě.

LITERATURA

- [1] *Syntezátor*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z: <https://cs.wikipedia.org/wiki/Syntez%C3%A1tor>. [cit. 2023-11-13].
- [2] GABOR, Dennis. Acoustical Quanta and the Theory of Hearing. *Nature*. 03 May 1947, roč. 1947, č. 159, s. 593.
- [3] XENAKIS, Iannis. *Formalized Music: Thoughts and Mathematics in Music*. Second, revised English edition, with additional material translated by Sharon Kanach. Pendragon Press, 2001. ISBN 1-57647-079-2.
- [4] ROADS, Curtis. Introduction to Granular Synthesis. *Computer Music Journal*. Summer, 1988, roč. 1988, č. 12, s. 11–13. ISSN 01489267.
- [5] BURK, Phil et al. *Music and Computers*. Online. In: Music and Computers: A Theoretical and Historical Approach. 2011. Dostupné z: https://musicandcomputersbook.com/chapter4/04_08.php. [cit. 2023-12-02].
- [6] ROADS, Curtis. *The Computer Music Tutorial*. The MIT Press, 27 February 1996. ISBN 978-0262680820.
- [7] ROADS, Curtis. *Microsound*. PAP/CDR edition. The MIT Press, 20 August 2004. ISBN 0-262-18215-7.
- [8] TRUAX, Barry. Real-Time Granular Synthesis with a Digital Signal Processor. *Computer Music Journal*. Summer 1988, roč. 12, č. 2, s. 14–26. ISSN 0148-9267.
- [9] TRUAX, Barry. Composing with Real-Time Granular Sound. *Perspectives of New Music*. Summer 1990, roč. 28, č. 2, s. 120–134. ISSN 0031-6016.
- [10] AWDE, Nick. *Mellotron: The Machine and the Musicians that Revolutionised Rock*. Desert Hearts, 2008, s. 17. ISBN 978-1-898948-02-5.
- [11] EVERETT, Walter. *The Beatles As Musicians: Revolver through the Anthology. 1st Edition*. Oxford University Press, 1999, s. 146. ISBN 978-0-19-802960-1.
- [12] WAKEMAN, Rick. The day I played the Mellotron for David Bowie. *The Guardian*. 8th January 2017. ISSN 1756-3224.
- [13] WELCH, Chris. *Close to the Edge: The Story of Yes. 3rd edition*. Omnibus Press, 2003. ISBN 978-0711995093.
- [14] VAIL, Mark. *Vintage Synthesizers: Pioneering Designers, Groundbreaking instruments, Collecting Tips, Mutants of Technology. Second edition*. Backbeat, 2000. ISBN 978-0879306038.
- [15] CHINEN, Nate. Synthesizing Music and Science. Online. *The Pennsylvania Gazette*. Sept–Oct 2013. Dostupné z: <https://thepenngazette.com/synthesizing-music-and-science/>. [cit. 2024-05-11].
- [16] *Synclavier Early History*. Online. Synclavier. 2016. Dostupné z: <http://www.500sound.com/synclavierhistory.html>. [cit. 2024-05-11].
- [17] VOGEL, Peter. *The Fairlight Story*. Online. 1 June 2008. Dostupné z: <https://www.enerd.com/fairlight/index.htm>. [cit. 2024-05-11].

- [18] KEEBLE, Rob. *30 Years Of Emu*. Online. *Sound on sound*. September 2002. Dostupné z: <https://www.soundonsound.com/music-business/30-years-emu>. [cit. 2024-05-11].
- [19] WIFFEN, Paul. Akai S900 Sampler. Online. *Electronics & Music Maker*. 1986, roč. 1986. Dostupné z: <https://www.muzines.co.uk/articles/akai-s900-sampler/1931>. [cit. 2024-05-11].
- [20] COLBECK, Julian. Roland S-10 Sampling Keyboard. Online. *Sound on sound*. November 1986. Dostupné z: <https://www.muzines.co.uk/articles/roland-s-10-sampling-keyboard/1426>. [cit. 2024-05-11].
- [21] SCHIMMEL, Jiří. *Digitální syntézy zvuku*. Online prezentace, s. 20–25. [cit. 2024-05-11].
- [22] SCHIMMEL, Jiří. *Digitální hudební nástroje*. Online prezentace, s. 51–54. [cit. 2024-05-11].
- [23] *Ribs*. Online. KVR Audio. 2017. Dostupné z: <https://kvraudio.com/forum/viewtopic.php?f=1&t=486995>. [cit. 2023-12-06].
- [24] *Free Granular VST Plugins*. Online. In: *Bedroom Producers Blog: Freeware For Musicians*. 2022. Dostupné z: <https://bedroomproducersblog.com/2012/03/30/bpb-freeware-studio-best-free-granular-vstau-plugins/>. [cit. 2023-12-06].
- [25] VASAPOLLI, Giuseppe [@Giuseppe Vasapolli]. *Ribs Manual v1.2.3 PDF*. Online. In: Scribd. Dostupné z: <https://www.scribd.com/document/399694113/Ribs-manual-v1-2-3-pdf>. [cit. 2023-12-06].
- [26] GERGELY, Daniel [@daniel gergely]. *Emergence*. Online. In: *Itch.io*. Dostupné z: <https://daniel-gergely.itch.io/emergence>. [cit. 2023-12-06].
- [27] *Grain Strain: v1.1.1*. Online. *Full bucket music*. Dostupné z: <https://www.fullbucket.de/music/grainstrain.html>. [cit. 2023-12-12].
- [28] ARLT, Björn. *Grain Strain: Grain Straining Effect Plug-In*. Online. 2022. Dostupné z: https://www.fullbucket.de/music/dl/grainstrain_manual_1_1.pdf. [cit. 2023-12-12].
- [29] *Granulab VST*. Online. *Free VST Plugins Multilingual*. Dostupné z: https://filebay.altervista.org/granulab-vst/?doing_wp_cron=1702285621.3285369873046875000000. [cit. 2023-12-12].
- [30] *Granulab Granny*. Online. November 25, 2018. Dostupné z: https://www.abc.se/~re/GranuLab/files/GranuLab.VST_2018-11-25.pdf. [cit. 2023-12-12].
- [31] *Argotlunar: Granulator VST / AudioUnit*. Online. *Github.io*. Dostupné z: <https://mourednik.github.io/argotlunar/>. [cit. 2023-12-12].
- [32] *Juce: Encyclopedia, Science News & Research Reviews*. Online. *Academic Accelerator*. Dostupné z: <https://academic-accelerator.com/encyclopedia/juce>. [cit. 2023-12-12].

- [33] *Documentation*. Online. *JUCE*. Dostupné z: <https://juce.com/learn/documentation/>. [cit. 2023-12-12].
- [34] *JUCE Forum*. Online. *JUCE*. Dostupné z: <https://forum.juce.com/>. [cit. 2023-12-12].
- [35] SMITH, George E. The invention and early history of the CCD. *Journal of Applied Physics*. 31 May 2011, roč. 109, č. 1.
- [36] *File:RGB Cube Show lowgamma cutout a.png* [@SharkD]. Online. In: *Wikipedia: the free encyclopedia*. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z: https://en.m.wikipedia.org/wiki/File:RGB_Cube_Show_lowgamma_cutout_a.png. [cit. 2023-12-12].
- [37] *File:Hsl-hsv models b.svg* [@SharkD]. Online. In: *Wikipedia: the free encyclopedia*. San Francisco (CA): Wikimedia Foundation, 2001. Dostupné z: https://it.m.wikipedia.org/wiki/File:Hsl-hsv_models_b.svg [cit. 2023-12-12].
- [38] *Get Average Color of Image*. Online. Dostupné z: <https://matkl.github.io/average-color/>. [cit. 2024-05-13].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

AD	Attack–Decay
ADSR	Attack–Decay–Sustain–Release
AGC	Automatic gain control
AIFF	Audio Interchange File Format
BMP	Windows Bitmap
CCD	Charge coupled device
CD	Compact Disc
DAW	Digital Audio Workstation
DMX	Digital Multiplex
FEKT	Fakulta elektrotechniky a komunikačních technologií
FLAC	Free Lossless Audio Codec
GSX	Granular Synthesis
GUI	Graphic User Interface
HSL	Hue–Saturation–Lightness
HSV	Hue–Saturation–Value
iOS	iPhone Operating System
JUCE	Jules' Utility Class Extensions
LFO	Low frequency oscillation
MacOS	Macintosh Operating System
MIDI	Musical Instrument Digital Interface
MP3	MPEG Audio Layer III
PNG	Portable Network Graphics
PODX	Poisson Distribution
RCB	Raised cosine bell
VST	Virtual Studio Technology
VUT	Vysoké učení technické v Brně
WAV	Waveform audio file format
WMA	Windows Media Audio