

Šablona
BAKALÁŘSKÁ / DIPLOMOVÁ PRÁCE
Verze 2021

Červené texty dále v šabloně přepište odpovídajícím textem nebo smažte.

DESKY

Namísto této stránky vložte stránku **desky** vygenerovanou v IS VUT.

TITULNÍ LIST

Namísto této stránky vložte **titulní list** (s logem) vygenerovaný v IS VUT.

ZADÁNÍ

Namísto této stránky vložte stránku **zadání FEKT** vygenerovanou v IS VUT.

Abstrakt

Tato práce se věnuje vývoji VST plug-inu na bázi granulárního syntezátoru, jehož parametry (jako například mezní frekvence filtru typu horní, dolní nebo pásmové propusti, délka granul, a další) jsou ovládány pomocí vlnové délky světla přicházejícího na webkameru. Plug-in je ovladatelný pomocí jednoho či více MIDI kontrolérů, kdy každý z nich může ovládat další instanci syntezátoru. Je tedy dosaženo naprosté nezávislosti vstupních dat, přijímaných MIDI zpráv i parametrů nastavovaných jednotlivým instancím a tím tedy i zvukové zajímavosti výsledného signálového výstupu VST plug-inu.

Klíčová slova

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

Abstract

This thesis addresses the development of a VST plug-in based on a granular synthesizer, whose parameters (such as the cutoff frequencies of high-pass, low-pass or band-pass filters, grain length, and others) are controlled by the wavelength of light captured by a web camera. The plug-in can be operated using one or more MIDI controllers, with each of them capable of controlling an additional instance of the synthesizer. This achieves complete independence of input data, received MIDI messages, and parameters set for individual instances, thus enhancing the sound richness of the resulting signal output of the VST plug-in.

Keywords

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

Bibliografická citace

KUCHAŘ, V. *Experimentální softwarový hudební nástroj nebo zvukový zdroj*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023. 11 s., 3 s. příloh. Semestrální 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:	Semestrální práce
Akademický rok:	2023/2024
Téma závěrečné práce:	Experimentální softwarový hudební nástroj nebo zvukový zdroj

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

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a 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: 12. prosince 2023

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 mého postupu na semestrální práci.

V Brně dne: 12. prosince 2023

podpis autora

Obsah

SEZNAM OBRÁZKŮ	8
SEZNAM TABULEK.....	9
ÚVOD	10
1. GRANULÁRNÍ SYNTÉZA.....	11
1.1 PRINCIP A METODY GRANULÁRNÍ SYNTÉZY	11
1.1.1 Metoda mřížek (screens)	11
1.1.2 Metoda událostí (events).....	13
1.2 HISTORIE.....	14
2. SROVNÁNÍ EXISTUJÍCÍCH ŘEŠENÍ.....	16
2.1 RIBS	16
2.2 EMERGENCE.....	17
2.3 GRAIN STRAIN.....	19
2.4 GRANULAB	20
2.5 ARGOTLUNAR	21
3. VÝVOJ VLASTNÍHO ŘEŠENÍ	23
3.1 FRAMEWORK JUCE.....	23
3.2 SNÍMÁNÍ BAREV	23
3.2.1 Model RGB	24
3.2.2 Model HSV.....	24
3.3 GRANNY SYNTH.....	25
3.3.1 Grafické uživatelské rozhraní	26
3.3.2 Načtení souboru	27
3.3.3 Zobrazení amplitudy v čase.....	28
3.3.4 Výběr úseku načteného souboru.....	31
3.3.5 Granulizace a práce s grainy.....	31
3.3.6 Řazení grainů	32
3.3.7 Efektová sekce.....	32
3.3.8 LFO.....	32
3.3.9 Řešení změny parametrů pomocí barvy	33
4. ZÁVĚR.....	35
LITERATURA.....	36
SEZNAM SYMBOLŮ A ZKRATEK	38

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].....	12
1.2	Tabulka pravděpodobnosti přehrání jednotlivých buněk. Výsledná pravděpodobnost pro danou frekvenci je vždy rovna 100 %. Zdroj: [3]	12
1.3	Vizualizace několika oblaků (clouds) v čase. Jednotlivé body reprezentují zrna rozložená ve frekvenční a časové ose. Zdroj: [5]	13
2.1	Grafické rozhraní VST granulárního syntezátoru <i>Ribs</i>	17
2.2	Grafické rozhraní VST granulárního syntezátoru <i>Emergence</i>	18
2.3	Grafické rozhraní VST granulárního syntezátoru <i>Grain Strain</i>	19
2.4	Grafické rozhraní VST granulárního syntezátoru <i>GranuLab</i>	20
2.5	Grafické rozhraní VST granulárního syntezátoru <i>Argotlunar</i>	21
3.1	3D vizualizace barevného modelu RGB. Zdroj: [23]	24
3.2	Srovnání 3D vizualizace modelů HSL a HSV. Zdroj: [24]	25
3.3	Blokové schéma granulárního VST syntezátoru <i>Granny Synth</i>	25
3.4	Návrh GUI granulárního VST plug-inu <i>Granny Synth</i>	26
3.5	Blokové schéma zpracování dat získaných webkamerou.....	33

SEZNAM TABULEK

3.1	Shrnutí parametrů navrhovaného VST plug-inu <i>Granny Synth</i>	34
-----	---	----

ÚVOD

Syntezátory 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 syntezátorů 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 syntezátorů, 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, které mohou obohatit výsledný zvuk, jsou například délka granulí, změny vzdálenosti mezi jednotlivými granulemi, jejich překrývání, rychlost přehrávání, přidávání různých typů obálek a spoustu dalších.

Tato práce se věnuje problematice vývoje softwarového granulárního syntezátoru ve vývojovém prostředí JUCE. Jejím hlavním cílem je vytvoření VST plug-inu schopného vstupní zvuková data jednak rozdělit na granule, zároveň na ně aplikovat výše zmíněné parametry. Neobvyklost řešení spočívá v možnosti ovládání syntezátoru pomocí vlnové délky světla – respektive jeho barev snímaných webkamerou.

Semestrální práce je členěna do 4 kapitol, z nichž první dvě slouží jako teoretická část práce. V kapitole 1 je popsán princip a historie granulární syntézy, srovnání dalších veřejně dostupných řešení je obsaženo v kapitole 2. Kapitola 3 představuje framework JUCE a věnuje se praktické části práce – vývoji samotného plug-inu. V kapitole 4 se nachází shrnutí a závěr semestrální práce.

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 pásů. 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 zrněk 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 zrny. 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

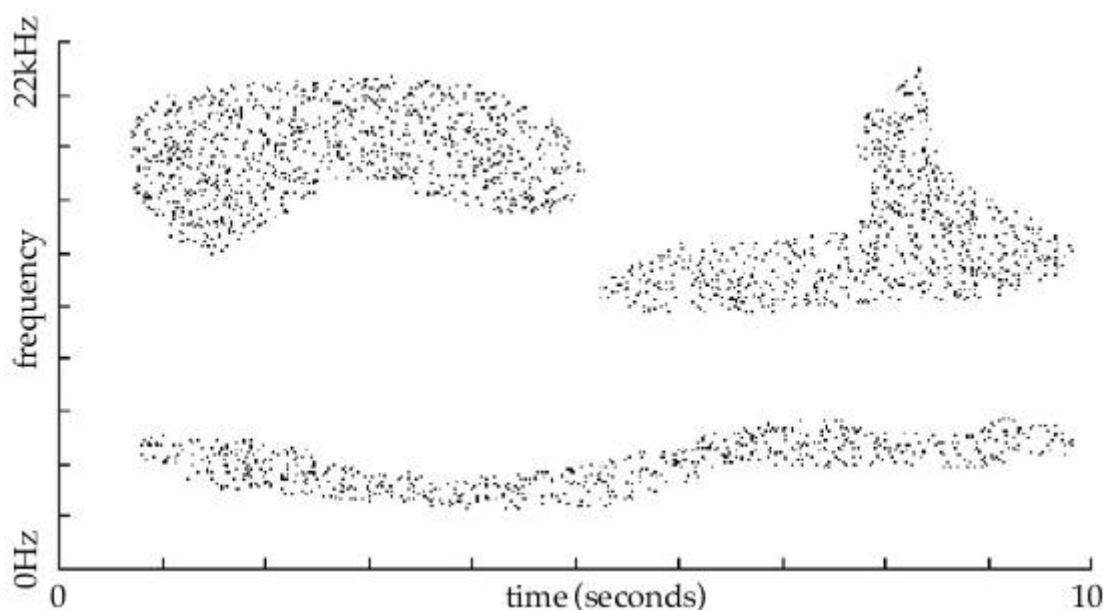
Obrázek 1.2 Tabulka pravděpodobnosti přehrání jednotlivých buněk. Výsledná pravděpodobnost pro danou frekvenci je vždy rovna 100 %. Zdroj: [3]

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 pocházejících ze samplů živých nástrojů. Dále popisuje princip podobný FM syntéze, 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.

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 mimo jiné s využitím pouze elektronických nástrojů a umělého (=člověkem upravených) zvuků. Granulární syntézu obsahují například jeho skladby *Bohor* (1962) nebo *Persepolis* (1972). Jeho přístup ke granulární syntéze je popsán detailněji v kapitole 1.1.1.

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 úrovně hlasitosti mezi jednotlivými grainy [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.

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. SROVNÁNÍ EXISTUJÍCÍCH ŘEŠENÍ

V následující kapitole jsou představena některá 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 uvedeno pouze pět z nich. Jedná se o VST syntezátory *Ribs*, *Emergence*, *Grain Strain*, *GranuLab* a *Argotlunar*.

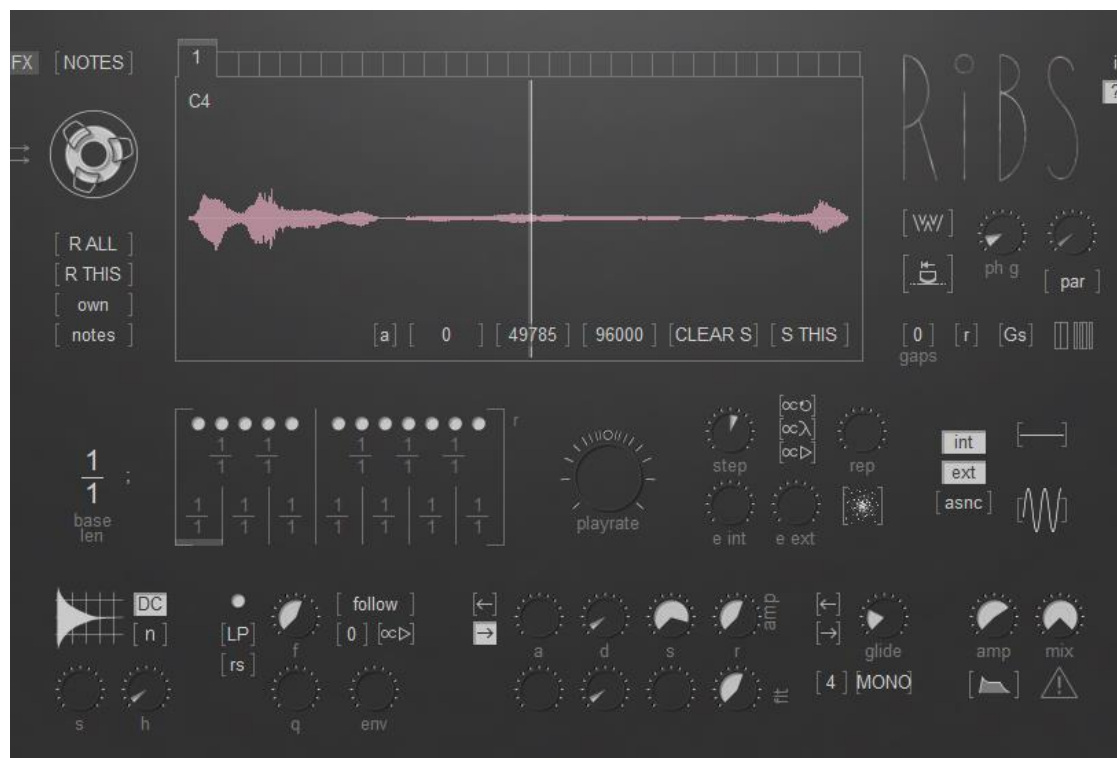
2.1 *Ribs*

Jedná se o VST granulární syntezátor z vývojářské dílny *Hvoya Audio* [10][11]. 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 podle 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 [12].



Obrázek 2.1 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.

2.2 Emergence

Tento VST plug-in vytvořený vývojářem Daniele Gergelym funguje jako prostý granulizér, dává možnost vytvořit zrna a dále pracovat s jejich parametry [13]. 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 2.2 Grafické rozhraní VST granulárního syntezátoru *Emergence*

Způsob vytváření grainů je velmi dobře 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.

2.3 Grain Strain

Grain Strain je granulární VST syntezátor vyvinutý mezi lety 2021–2022 Björnem Artem [14]. 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 [15]. *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 2.3 Grafické rozhraní VST granulárního syntezátoru *Grain Strain*

2.4 GranuLab

Toto řešení bylo vyvinuto vývojářskou společností *Ostinato* [16][17]. 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 2.4 Grafické rozhraní VST granulárního syntezátoru *GranuLab*

Předposlední sekci je presetová sekce *Patches*. Do ní je možné ukládat nastavené scény (skupiny parametrů) a následně je vyvolávat. Kapacita aktivních scén pro procházení je

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 změně ADSR při změně scény.

V poslední sekci (*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 efektové 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.

2.5 Argotlunar

Tento plug-in vytvořený Michaelem Ourednikem slouží jako prostý granulizér bez možnosti ovládat výšku tónu pomocí MIDI not [18]. 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 2.5 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 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.

3. VÝVOJ VLASTNÍHO ŘEŠENÍ

V rámci vývoje vlastního VST plug-inu pro granulární syntézu byl využit JUCE – framework jazyka C++ pro vývoj zvukového software. Jako vývojové prostředí bylo zvoleno *Visual Studio 2019*. Následující kapitoly se věnují popisu zvoleného frameworku, segmentaci jednotlivých součástí vyvíjeného plug-inu a popisu navrhovaného řešení jeho experimentální složky – ovládání parametrů granulární syntézy pomocí barev.

3.1 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 [19].

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 prvků.

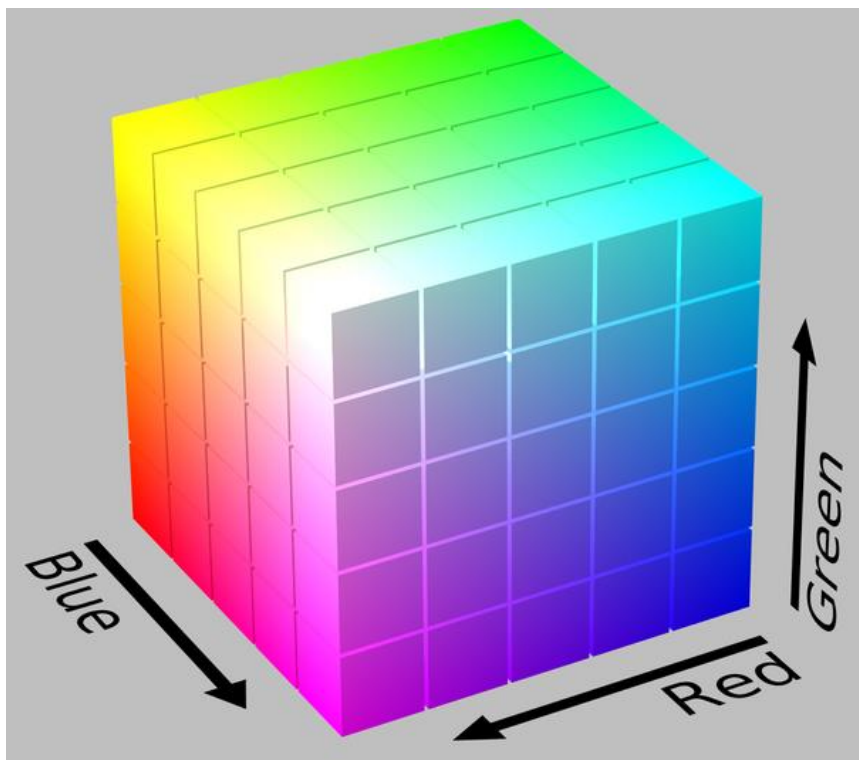
Velikou výhodou frameworku je také jeho dobře psaná a aktualizovaná dokumentace [20] doplněná o ukázky a návody k použití objektů, tříd a jejich metod. Cenným zdrojem informací a rad ohledně řešení problémů spojených s vývojem zvukových aplikací je také JUCE Forum [21], na kterém spolupracují vývojáři z celého světa a sdílejí své nápady, postřehy a další užitečné informace o práci s frameworkem JUCE.

3.2 Snímání barev

Digitální záznam obrazu byl poprvé umožněn v roce 1970 díky konstrukci fotoaparátu se zabudovaným CCD snímačem [22]. Ačkoliv ne všechny obrazové záznamy je v současnosti realizován pomocí digitální technologie, pro jeho šíření v elektronické podobě je nutné jej převést do číselných informací. Pro zobrazení barev existuje několik matematických modelů, jež umožňují nést zprávy o barevných parametrech světla. Dále jsou popsány modely RGB a HSV využívané při implementaci v této práci.

3.2.1 Model RGB

Jedná se o model založený na barevné citlivosti lidského oka. Výsledná barva vzniká skládáním tří složek (R – červená, G – zelená, B – modrá), řadíme jej tedy mezi modely aditivní. Model RGB je využíván pro zobrazení barev na displejích, kde každý pixel je složen ze tří barevných složek o různých intenzitách, což při komplexním pohledu vytváří dojem barevného obrazu.

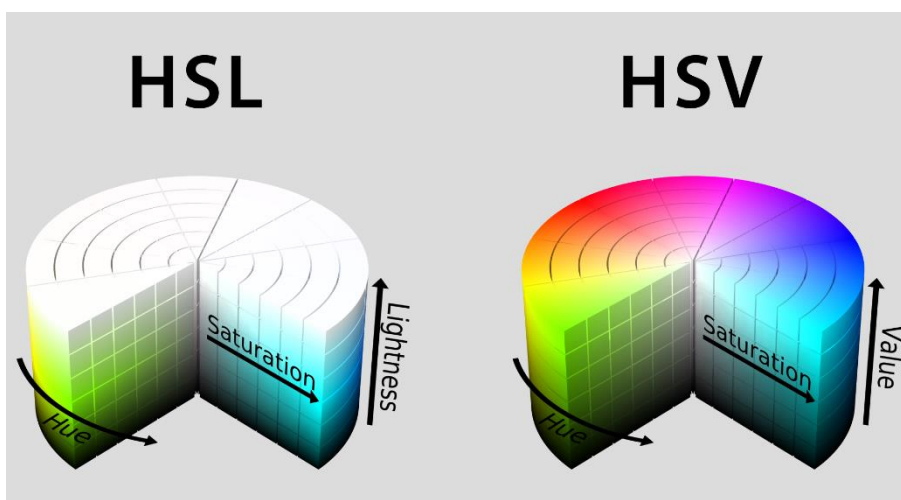


Obrázek 3.1 3D vizualizace barevného modelu RGB. Zdroj: [23]

3.2.2 Model HSV

Tento matematický model nejvíce odpovídá intuitivnímu lidskému vnímání barev. Jeho složky netvoří kombinace barev, nýbrž parametry *Hue* (odstín), měřený jako poloha na barevném kruhu v rozmezí 0° – 360° , *Saturation* (sytnost), 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.

Model HSV bývá využíván při editaci fotografií, jelikož jeho parametry umožňují intuitivní práci s barevnými odstíny, sytností i jasnem snímků. Naopak právě kvůli konstrukčním vlastnostem displejů (RGB segmentace) není nejvhodnějším řešením ukládání matic informací pro jednotlivé pixely.

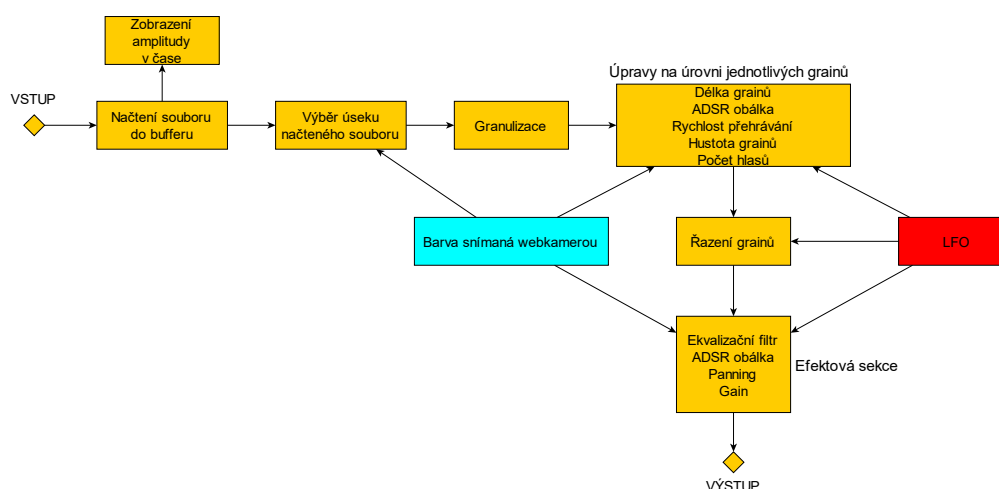


Obrázek 3.2 Srovnání 3D vizualizace modelů HSL a HSV. Zdroj: [24]

Během implementace granulárního VST plug-inu ovládaného barvami snímanými webkamerou jsou využívány oba zmíněné barevné modely podle vhodnosti využití.

3.3 Granny Synth

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. 3.3). Následující podkapitoly se věnují jednotlivým blokům a jejich zpracování.



Obrázek 3.3 Blokové schéma granulárního VST syntezátoru *Granny Synth*

3.3.1 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 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 3.4 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 je ale momentální nutností pro velké množství parametrů granulární syntézy, které je uživateli dáno k nastavení pro pevnou kontrolu nad průběhem výstupního signálu VST plug-inu. Není v žádném případě vyloučeno, že během další práce na projektu bude navržené rozhraní měněno a zjednodušováno při zachování současných parametrů, či dokonce jejich rozšíření.

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.

3.3.2 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 myší 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.

Pro načtení souboru existují v kódu projektu dvě metody: *loadFileViaButton* a *loadFileViaDragNDrop*.

```
/******  
void loadFileViaButton()  
{  
    myGrannySynth.clearSounds();  
  
    FileChooser chooser{ "Vyberte soubor", {}, "*.wav;*.flac;*.aiff;  
                                                *.wma;*.ogg"};  
  
    if (chooser.browseForFileToOpen())  
    {  
        auto file = chooser.getResult();  
  
        auto myFile = std::make_unique<File>(file);  
        myWaveThumbnail.myFileName = myFile->getFileName();  
  
        myFormatReader = myFormatManager.createReaderFor(file);  
    }  
    else  
    {  
        return;  
    }  
  
    ...  
  
    myGrannySynth.addSound(new SamplerSound( ... ));  
  
    ...  
}
```

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 (momentálně lze nahrávat pouze formáty .wav, .flac, .aiff, .wma a .ogg, neboť pro další rozšířené formáty – např. formát .mp3 – se zatím

nepodařilo ošetřit chyby, které brání ve správném fungování plug-inu). V případě, že byl vybrán některý zvukový soubor, je zjištěn jeho název, který je dále předán pro zobrazení na displeji plug-inu a soubor je přečten pomocí instance třídy *FormatReader*. Zvuková data jsou poté uložena do nové instance třídy *SamplerSound* s odpovídajícími parametry.

Metoda *loadFileViaDragNDrop* je volána po předchozím ujištění, že přetahovaný soubor je ve vhodném formátu. Toto zajišťuje dvojice metod *isInterestedInFileDrag* a *filesDropped*.

```
/******  
bool isInterestedInFileDrag(const StringArray& files)  
{  
    for (auto file : files)  
    {  
        if (file.contains(".wav") || file.contains(".aiff") ... )  
        {  
            return true;  
        }  
    }  
    return false;  
}  
  
void filesDropped(const StringArray&files, int x, int y)  
{  
    for (auto file : files)  
    {  
        if (isInterestedInFileDrag(file))  
        {  
            auto myFile = std::make_unique<File>(file);  
            myWaveThumbnail.myFileName = myFile->getFileName();  
  
            audioProcessor.loadFileViaDragNDrop(file);  
        }  
    }  
}
```

Při přetažení souboru nad prostor 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 pak je zavolána samotná metoda *loadFileViaDragNDrop*, jejíž funkce i podoba je téměř totožná s metodou *loadFileViaButton*.

3.3.3 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*.

```
/******  
void readWaveform()  
{  
    auto sampleLength = static_cast<int>(myFormatReader->  
                                           lengthInSamples);  
    myWave.setSize(1, sampleLength);  
    myFormatReader->read(&myWave, 0, sampleLength, 0, true, true);  
}
```

Argumenty metody *read* jsou buffer, do kterého budou data uložena, pozice prvního vzorku v tomto bufferu, počet čtených vzorků a první čtený vzorek. Posledními dvěma argumenty jsou booleovské hodnoty určující čtení pro levý a pravý kanál vstupního signálu.

Vykreslení požadovaného tvaru vlny je provedeno jako součást metody *paint* ve třídě *WaveThumbnail*.

```
/******  
void paint(juce::Graphics& g)  
{  
    ...  
    auto waveform = audioProcessor.getWave();  
    ...  
    if(waveform.getNumSamples() > 0)  
    {  
        Path p;  
        myAudioPoints.clear();  
        ...  
        auto wave = audioProcessor.getWave();  
        auto ratio = wave.getNumSamples() / getWidth();  
        auto buffer = wave.getReadPointer(0);  
    }  
    *  
}
```

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.

```

*
    for(int sample = 0;
        sample < wave.getNumSamples();
        sample += ratio)
    {
        myAudioPoints.push_back(buffer[sample]);
    }

    p.startNewSubPath(0, getHeight() / 2);

    for (int sample=0; sample < myAudioPoints.size(); ++sample)
    {
        auto point = jmap<float>(myAudioPoints[sample],
                                -1.0f, 1.0f, getHeight(). 0);
        p.lineTo(sample, point);
    }

    ...

}
}

```

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*.

```

/*****
void processBlock(AudioBuffer<float>& buffer,MidiBuffer& midiMessages)
{
    ...

    MidiMessage m;
    MidiBuffer::Iterator it(midiMessages);
    int sample;
    static double midiNoteInHertz;

    while (it.getNextEvent(m, sample))
    {
        if (m.isNoteOn())
        {
            midiNoteInHertz =
                m.getMidiNoteInHertz(m.getNoteNumber());
            myIsNotePlayed = true;
        }
        else if (m.isNoteOff())
        {
            myIsNotePlayed = false;
        }
    }
}

```



```

float pitchedSamples = buffer.getNumSamples() * (midiNoteInHertz
    / 523.25113);
mySampleCount = myIsNotePlayed ? mySampleCount += pitchedSamples
    : 0;
...
}

```

V této části metody je nastavena rychlost pohybu ukazatele pozice podle stisknuté MIDI noty. Jako referenční hodnota je zvolena frekvence 523,25113 Hz, jež odpovídá tónu c^2 (MIDI nota 72), při kterém je přehráván nahraný signál v původní nezměněné výšce.

3.3.4 Výběr úseku načteného souboru

Pro provádění granulární 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 potměněním nevybraných oblastí.

3.3.5 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 přidaných 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.

3.3.6 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.

3.3.7 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.

3.3.8 LFO

Poslední sekci 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.

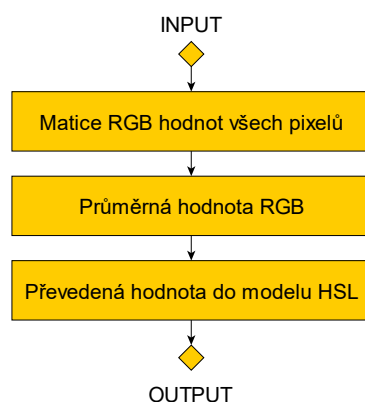
Míra ovlivnění jednotlivých parametrů oscilátorem je nastavována stisknutím tlačítka LFO. Tímto je spuštěn režim indikace vlivu LFO – rozsah všech otočných ovládacích prvků je změněn na 0–100 % a nastavená hodnota udává hloubku aplikování modulační vlny.

3.3.9 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 ovládány jasovou složkou (parametrem *Lightness* barevného modelu HSL), jiné, kupříkladu střední nebo mezní frekvence ekvalizačního filtru jsou ovládány 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 použití této hodnoty jako argumentu absolutní hodnoty funkce kosinus, což by mělo 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 jejích 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 zaznamenávaných pixelů je prováděno průměrováním hodnot RGB. Průběh tohoto provedení znázorňuje následující diagram (obr. 3.5).



Obrázek 3.5 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. 3.1).

Tabulka 3.1 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

4. ZÁVĚR

V rámci této práce byl shrnut princip i vývoj granulární syntézy jakožto experimentální metody zvukové syntézy, došlo k otestování a srovnání momentálně existujících řešení a jejich parametrů, a v neposlední řadě bylo dosaženo návrhu experimentálního VST granulárního syntezátoru *Granny Synth*, jenž může sloužit jako předloha pro práci na navazujícím bakalářském projektu s cílem jeho kompletní realizace. Během jeho vytváření byla promyšlena některá úskalí současného konceptu, bylo poukázáno na některé zjevné problémy řešení experimentální složky projektu a popsáno několik bodů návrhu, které budou vyžadovat důsledné a promyšlené řešení pro použitelnost tohoto plug-inu. Konkrétně se jedná o přepočty vzorků na časové úseky v rámci granulizace vstupního signálu, využití složek barevného modelu HSL pro ovládání parametrů granulární syntézy i o grafické uživatelské rozhraní a jeho přístupu ke koncepci připomínat vzhledem i způsobem ovládání reálný přístroj.

V navazující bakalářské práci je plánována postupná realizace návrhu, řešení aktuálně známých i zatím skrytých problémů týkajících se provedení granulárního syntezátoru a jejím předpokládaným cílem je vytvoření kompletního prototypu experimentálního VST plug-inu s poměrně ojedinělou možností ovládání parametrů pomocí vlnové délky (barvy) světla snímaného webkamerou.

LITERATURA

- [1] *Syntežátor*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: <https://cs.wikipedia.org/wiki/Syntež%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] *Ribs*. Online. KVR Audio. 2017. Dostupné z: <https://kvraudio.com/forum/viewtopic.php?f=1&t=486995>. [cit. 2023-12-06].
- [11] *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].
- [12] 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].
- [13] GERGELY, Daniel [@daniel gergely]. *Emergence*. Online. In: Itch.io. Dostupné z: <https://daniel-gergely.itch.io/emergence>. [cit. 2023-12-06].
- [14] *Grain Strain: v1.1.1*. Online. Full bucket music. Dostupné z: <https://www.fullbucket.de/music/grainstrain.html>. [cit. 2023-12-12].
- [15] 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].

- [16] *Granulab VST*. Online. Free VST Plugins Multilingual. Dostupné z: https://filebay.altervista.org/granulab-vst/?doing_wp_cron=1702285621.3285369873046875000000. [cit. 2023-12-12].
- [17] *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].
- [18] *Argotlunar: Granulator VST / AudioUnit*. Online. Github.io. Dostupné z: <https://mourednik.github.io/argotlunar/>. [cit. 2023-12-12].
- [19] *Juce: Encyclopedia, Science News & Research Reviews*. Online. Academic Accelerator. Dostupné z: <https://academic-accelerator.com/encyclopedia/juce>. [cit. 2023-12-12].
- [20] *Documentation*. Online. JUCE. Dostupné z: <https://juce.com/learn/documentation/>. [cit. 2023-12-12].
- [21] *JUCE Forum*. Online. JUCE. Dostupné z: <https://forum.juce.com/>. [cit. 2023-12-12].
- [22] SMITH, George E. The invention and early history of the CCD. *Journal of Applied Physics*. 31 May 2011, roč. 109, č. 1.
- [23] *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].
- [24] *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].

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

ADSR	Attack–Decay–Sustain–Release
AGC	Automatic gain control
CD	Compact Disc
DAW	Digital Audio Workstation
DMX	Digital Multiplex
FEKT	Fakulta elektrotechniky a komunikačních technologií
GSX	Granular Synthesis
GUI	Graphic User Interface
JUCE	Jules' Utility Class Extensions
LFO	Low frequency oscillation
MIDI	Musical Instrument Digital Interface
PODX	Poisson Distribution
RCB	Raised cosine bell
VST	Virtual Studio Technology
VUT	Vysoké učení technické v Brně