



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Eelis Lähteenaro

# Realistinen sample-pohjainen soitin- mallinnus

Metropolia Ammattikorkeakoulu

Medianomi

Elokuva- ja televisio

Opinnäytetyö

06.05.2021

Tekijä(t) Otsikko	Eelis Lähteenaro Realistinen sample-pohjainen soitinmallinnus
Sivumäärä Aika	22 sivua 06.05.2021
Tutkinto	Medianomi
Tutkinto-ohjelma	Elokuva- ja televisiotuotanto
Suuntautumisvaihtoehto	Äänisuunnittelu
Ohjaaja	Lehtori Aura Kaarivuo
<p>Virtuaaliset soitinmallinnukset ovat korvaamaton työväline sävellystyössä. Ne tarjoavat kaikille mahdollisuuden minkä tahansa soittimen käyttämiseen sävellyksissään – ilman kalliita soittimia, mikrofoneja, ammattisoittajia ja äänitystiloja. Tietokoneiden laskentatehon kasvun myötä soitinmallinnusten laatu on parantunut, eikä hyvää soitinmallinnusta välttämättä enää edes erota aidosta soittimesta.</p> <p>Nykyään realistiset soitinmallinnukset perustuvat yleensä sampleihin, eli aidosta soittimesta äänitettyihin ääninäytteisiin. Ääninäytteitä manipuloimalla voidaan toteuttaa erilaisia artikulaatioita, jotta mallinnus olisi mahdollisimman realistisen kuuloinen.</p> <p>Tämä opinnäytetyö käsittelee sample-pohjaisen soitinmallinnuksen äänitys- ja ohjelmointiprosessia. Työ koostuu kirjallisesta osasta ja toiminnallisesta osasta. Kirjallisessa osassa käsitellään samplereiden ja soitinmallinnusten historiaa ja toimintaperiaatteita, sekä äänen analysointiin liittyviä tekniikoita ja haasteita. Toiminnallinen osuus koostuu kellopelin realistisen mallinnuksen äänityksestä ja ohjelmoinnista. Kirjallisen osuuden lopussa analysoidaan toteutetun soitinmallinnuksen realistisuutta vertaamalla sitä oikeaan soittimeen.</p> <p>Keskeisimpinä havaintoina onnistuneen soitinmallinnuksen toteutuksen kannalta nousee esiin riittävä valmistelu, hyvä äänitystekniikka, sekä intuitiivinen käyttöliittymä.</p>	
Avainsanat	Samplerit, Soitinmallinnus

Author(s) Title	Eelis Lähteenaro Building a sample-based virtual instrument
Number of Pages Date	22 pages 06 May 2021
Degree	Bachelor of Culture and Arts
Degree Programme	Film and Television
Specialisation option	Sound Design
Instructor(s)	Aura Kaarivuo, Senior Lecturer
<p>Virtual instruments are an irreplaceable tool for both professional and amateur composers. They give anyone the possibility to use the sound of any instrument in their compositions – without the need for expensive instruments, microphones, musicians or a recording studio. These virtual instruments can be made so realistic, that they become practically indistinguishable from real instruments.</p> <p>Nowadays virtual instruments are mainly sample based, meaning that they play back recordings, or samples, of real instruments. In order to create a more realistic sound, different nuances and articulations can be achieved by manipulating these recordings.</p> <p>The purpose of this final project is to study how a sample based virtual instrument is recorded and programmed. For the study, a glockenspiel is sampled, turned into a virtual instrument and analyzed. Additionally, this project covers the history of samplers and instrument modeling, as well as techniques and challenges related to analyzing sound.</p> <p>As a result of the study, the author found that the most important aspects of creating a realistic sample based virtual instrument are the use of a good recording technique, an intuitive GUI and enough time for preparation and planning.</p>	
Keywords	Sampler, Virtual instrument

## Sisällys

1	Johdanto	1
2	Samplaaminen, samplerit ja romplerit	2
2.1	Samplaamisen historia	2
2.2	Ohjelmistosamplerit	3
3	Realistinen soitinmallinnus	4
3.1	Fysiikkapohjainen soitinmallinnus	4
3.2	Sample-pohjainen soitinmallinnus	5
4	Toteutus	5
4.1	Tavoitteet ja suunnittelu	5
4.2	Äänitys ja editointi	6
4.3	Soitinrakennus	8
4.3.1	HISE	8
4.3.2	Toiminnallisuus	9
4.3.3	Käyttöliittymä	10
4.4	Lopputulokset ja kehitysmahdollisuudet	11
5	Realistinen ääni	13
5.1	Ääni fyysisenä ilmiönä	13
5.1.1	Yläsävelsarja ja interferenssi	14
5.2	Äänen tallennus ja toisto	14
5.3	Äänen havaitseminen ja mittaus	16
5.3.1	Mittaaminen	16
5.3.2	Kuulokokemus	17
6	Vertailu	17
6.1	Kuuntelu	20
6.2	Vertailun lopputulokset	20
7	Pohdinta	20

## 1 Johdanto

Johann Sebastian Bachin väitetään sanoneen, että soittimen soittaminen on helppoa — pitää vain painaa oikeaa kosketinta oikeaan aikaan, jolloin soitin soittaa itseään! Bach ei varmasti osannut aavistaa, kuinka osuvasti hän vahingossa ennusti musiikkiteknologian kehityksen suunnan. Nykyään kuka tahansa voi oikeaa kosketinta painamalla helposti soittaa lähes mitä tahansa soitinta, tarkemmin sanoen lähes minkä tahansa soittimen mallinnusta. Nämä mallinnukset ovat jo niin kehittyneitä, että harjaantunutkaan korva ei välttämättä aina erota realismia tavoittelevaa mallinnusta oikeasta live-äänitteestä.

Musiikin säveltäminen mallinnetulla soittimella on mahdollista ilman oikeaa soitinta, soitotaitoa tai musiikin teorian tuntemusta. Madaltunut kynnyks ja musiikin tuottamiseen soveltuvan ohjelmiston saatavuuden helpottuminen viimeisen 20 vuoden aikana on johtanut amatöörituottamisen räjähdysmäiseen kasvuun. Tänä päivänä harvalla kuitenkaan on kattavaa soitinkokoelmaa tai mahdollisuutta äänittää esimerkiksi sinfoniaorkesteria tai pianoa kotonaan — joten siitä, että harrastelijatason musiikin tuottaminen ei rajoitu pelkästään elektroniseen musiikkiin, voimme pitkälti kiittää sampleita ja soitinmallinnuksia.

Myös ammattikentällä sample-soittimia käytetään enenevässä määrin. Varsinkin silloin kun sävellyksessä halutaan käyttää epätavallista instrumenttia, on usein kustannustehokkaampaa hyödyntää mallinnettua soitinta oikean soittimen ja soittajan sijaan. Tästä syystä koen, että varsinkin erikoisempien soitinten mallinnukset tuovat paljon uusia mahdollisuuksia sävellystyöhön, mutta samalla on syytä pohtia, miten realistinen soitinmallinnus ja oikea äänite eroavat. Jos tulevaisuudessa saavutetaan piste, jossa mallinnettua soitinta ja äänitettä ei enää tarkalla mittauksellakaan voida erottaa toisistaan, herää kysymys siitä, tarvitaanko soittajia enää lainkaan musiikin tuottamiseen.

Tässä opinnäytetyössä tarkoitukseni on toteuttaa realistinen mallinnus kellopelistä, verrata lopputulosta saman soittimen oikeaan äänitteeseen sekä tutkia, mitkä seikat vaikuttavat sample-soittimen realistisuuteen. Kirjallisessa osassa paneudun sample-soittimen rakennusprosessiin ja tutkin mikä tekee mallinnetusta äänestä realistista sekä mitä realistinen ääni on ja miten tätä voidaan mitata. Lisäksi käyn läpi sampleiden toimintaperiaatteita ja historiaa.

## 2 Samplaaminen, samplerit ja romplerit

Samplaaminen, suomeksi kirjaimellisesti näytteenotto, tarkoittaa äänen taltioimista ja uudelleenkäyttöä osana toista äänikokonaisuutta, kuten musiikkia. Samplattava ääni voi olla esimerkiksi rummun isku, linnunlaulu tai viulu. Ääninäytteiden sävelkorkeutta, tois- tonopeutta ja taajuusjakamaa muokataan usein jälkikäteen käyttötarkoituksen mukaan.

Samplejen tallentamiseen ja toistoon kykenevää ohjelmistoa tai laitetta kutsutaan sampleriksi. Samplerit voidaan jakaa analogisiin, digitaalisiin, ja ohjelmistosamplereihin, riip- puen ääninäytteiden tallennus- ja toistotavasta. Sample-soittimia, joihin käyttäjä ei voi ladata omia sampleja, kutsutaan myös romplereiksi.

Sample-soittimia käytetään nykyään lähes kaikissa musiikin genreissä, vaikkei sitä aina huomaisikaan. Digitaalinen samplaus ja muut digitaaliset äänen muokkaustekniikat ovat tuoneet uusia mahdollisuuksia säveltämisen lisäksi myös esimerkiksi foley- ja äänisuun- nitteluun. (Albano 2019.)

### 2.1 Samplaamisen historia

Modernin samplaamisen esi-isänä voidaan pitää tekniikkaa, jota hyödynnettiin 1940-lu- vulla musique concrète -genressä. Musique concrète, eli konkreettinen musiikki, oli elektronista musiikkia, joka koostui täysin ympäristöstä äänitetyistä äänitteistä eli samp- leista. Äänenlähteenä saatettiin käyttää esimerkiksi junaa, pianoa tai keittiötarvikkeita, kuten Pierre Schaefferin teoksessa *Étude auz casseroles*. Konkreettista musiikkia tehtiin leikkaamalla ja looppaamalla ääninauhoja manuaalisesti, joten prosessiin ei tarvittu var- sinaista sampleria. (Howell 2005.)

Soittimia mallintavia samplereita alettiin rakentaa 1950-luvulla. Nämä elektronmekaan- iset laitteet toistivat soitinääniä analogisilta nauhoilta, mutta lopputulos ei lyhyiden ääni- näytteiden ja heikon äänenlaadun takia ollut kovinkaan realistinen – joskaan se ei vält- tämättä ollut tavoitteenakaan.

Ensimmäinen oikea askel kohti realistista soitinmallinnusta tapahtui 1969, kun BBC Ra- diophonic Workshop kehitti ensimmäisen digitaalisen samplerin, EMS musyksen. Digi- taaliset samplerit ovat tietokoneita, joiden muistiin on tallennettu ääninäytteitä. Näitä ää- ninäytteitä pystytään toistamaan ohjaimella, kuten koskettimistolla. (Davies 1996.)



Kuvio 1. Farlight CMI, ensimmäinen digitaalinen sampleri (Kuva: Wikipedia 2008)

Termiä sampling alettiin käyttämään digitaalisten samplereiden myötä. Digitaalinen tallennus mahdollisti aiempaa paremman äänenlaadun, mutta laitteiden pieni työmuisti ei vielä riittänyt kovinkaan aidolta kuulostavaan soitinmallinnukseen, sillä muistiin ei voitu tallentaa riittävän pitkiä ääninäytteitä, jotta kaikkia ääniä olisi voitu toistaa kokonaisuudessaan. Tästä huolimatta digitaaliset samplerit olivat todella suosittuja lähes kolmen vuosikymmenen ajan. (Davies 1996; Howell 2005.)

## 2.2 Ohjelmistosamplerit

Tietokoneiden laskentatehon kasvaessa 1990-luvulla alkoi markkinoille ilmestyä ohjelmistosamplereita. Ohjelmistosampleriksi kutsutaan ohjelmistoa, jota voidaan käyttää digitaalisen tai elektromekaanisen samplerin toiminnan emuloimiseen tietokoneella. Jotkin ohjelmistosamplerit toistavat vain yhden valmiiksi äänitetyn soittimen ääniä, kun taas toisiin voi ladata haluamiaan ääninäytteitä. Nykyään ohjelmistosampleri on todennäköisesti eniten käytetty samplauksen väline ohjelmistojen helposta saatavuudesta johtuen. (Howell 2005.)

Tämän opinnäytetyön teososana rakennan sample-pohjaisen soitinmallinnuksen, jota voidaan pitää yhdenlaisena ohjelmistosamplerina tai sample-soittimena. Ohjelmistosamplereihin liittyvä terminologia on häilyvää, mutta tässä opinnäytetyössä viitataan kaikkiin sampleja toistaviin ohjelmistoihin termillä sample-soitin. Realistisesti soitinta emuloi-viin, sampleja toistaviin ohjelmistoihin viitataan termillä sample-pohjainen soitinmallinnus.

### 3 Realistinen soitinmallinnus

Soittimen äänen mallintaminen toisella soittimella ei ole uusi konsepti. Jo muinaisessa Roomassa kehitetyn pilliurun pillit nimettiin sen mukaan, mitä soitinta niiden tuottama ääni lähiten muistutti. 1800-luvulla mekaanisilla uruilla pystyttiin jo tarkoituksellisesti imi-toimaan sinfoniaorkesterin soittimia. Nykyään realistista soitinmallinnusta tehdään yleensä ohjelmistollisesti, joko sample- tai fysiikkapohjaisesti. (Davies 1996.)

#### 3.1 Fysiikkapohjainen soitinmallinnus

Fysiikkapohjainen soitinmallinnus perustuu algoritmeihin, joilla kuvaillaan soittimen fyysistä äänentuottomekanismia. Esimerkiksi kitaraa mallintaessa nämä algoritmit kuvaisivat kielen värähtelyä, äänen fyysistä siirtymistä tallan kautta runkoon ja edelleen ilman läpi korviimme. Fysiikkapohjaisella soitinmallinnuksella voidaan teoriassa luoda erottamaton kopio oikean soittimen äänestä. (Applied Acoustics 2021.)

Soitinten äänentuottoon liittyy paljon muuttujia. Esimerkiksi pianon vasara tuottaa erilaisen äänen osuessaan jo värähtelevään kieleen riippuen siitä, missä vaiheessa oskillaatiota kieli on. Kaikkia tämänkaltaisia nyansseja on mahdotonta mallintaa täydellisesti sample-pohjaisella mallinnuksella, sillä se vaatisi loputtoman määrän äänitteitä. Käytännössä soittimen äänentuottoon liittyy kuitenkin yleensä niin paljon edellä mainitun kaltaisia muuttujia, että näitä kaikkia ei voida huomioida fysiikkapohjaisessakaan mallinnuksessa ilman, että vaadittava laskentateho tai mallinnuksen toteuttamiseen käytetty aika kasvaisi kohtuuttomaksi. (Applied Acoustics 2021.)

Lisäksi fysiikkapohjainen soitinmallinnus on suuren laskumäärän takia todella prosessori-intensiivistä jopa moderneilla tietokoneilla, joten sample-pohjainen mallinnus on rajoitteistaan huolimatta edelleen näistä kahdesta yleisempää. (Applied Acoustics 2021)

### 3.2 Sample-pohjainen soitinmallinnus

Sample-pohjainen soitinmallinnus perustuu mallinnettavasta soittimesta äänitettyjen ääninäytteiden toistamiseen samplerilla. Kaikki äänet, joita soittimella on mahdollista tuottaa, pyritään äänittämään, jotta lopullinen sample-soitin kattaisi kaikki mahdolliset artikulaatiot ja soittotavat. Näiden artikulaatioiden määrä ja laatu riippuvat soittimesta ja määrittelevät pitkälti sen, kuinka vaikeaa soitin on mallintaa. (Brown 2019; Howell 2005.)

Prosessin yksinkertaistamiseksi tehdään yleensä tiettyjä kompromisseja. Esimerkiksi pianoa samplattaessa jokaista ääntä ja soittotapaa ei välttämättä äänitetä erikseen. Sen sijaan äänitteiden sävelkorkeutta ja karakteristiikkaa voidaan muuttaa jälkikäteen, jotta lopullinen virtuaalisoitin kattaisi kaikki äänet sekä suuren määrän artikulaatioita – ilman kohtuuttoman suurta määrää sampleja. Tyypillinen esimerkki tällaisesta jälkikäteen ohjelmallisesti toteutetusta artikulaatiosta on pitkät sointiajat, jotka voidaan toteuttaa looppaamalla lyhyitä sampleja. Näin vältetään tilanne, jossa soittimen jokaisesta äänestä jouduttaisiin erikseen äänittämään eripituisia versioita. (Brown 2019; Howell 2005.)

## 4 Toteutus

Tässä luvussa tutkin, miten sample-pohjaista soitinmallinnusta tehdään HISE-ohjelmistokehystä hyödyntäen. Mallinnettavaksi soittimeksi valitsin kelloperin, koska ajattelin sen olevan harjoituksen kannalta sopivan yksinkertainen soitin mallinnettavaksi ja siitä voisi olla hyötyä sävellystyössä.

Kelloperin on perkussiosoitin, joka koostuu useasta eri vireisestä metallilevystä. Levyt tuottavat heleän kellomaisen äänen, kun niitä lyödään vasaralla. (Encyclopaedia Britannica 2021a.)

### 4.1 Tavoitteet ja suunnittelu

Sample-pohjaista soitinmallinnusta voidaan tehdä usein eri tavoin. Mallinnettavasta soittimesta äänitetyt samplet voidaan esimerkiksi ladata valmiiseen sampleriin tai rompleriin, tai äänitteistä voidaan kasata itsenäinen virtuaalisoitin. Jälkimmäinen vaihtoehto on edeltävää työläämpää ja sisältää yleensä jonkin verran ohjelmointia, mutta lopputulos on paremmin kustomoitavissa. Lisäksi itsenäinen virtuaalisoitin toimii monella eri käyttöjärjestelmällä.

Soitinrakennusprosessia voidaan helpottaa käyttämällä ääniohjelmointiin soveltuvaa ohjelmistokehystä eli ohjelmaa, joka jo valmiiksi kattaa osan lopullisen virtuaalisoitimen ominaisuuksista. Tässä opinnäytetyössä päätin toteuttaa soitinmallinnuksen itsenäisenä virtuaalisoitimena HISE ohjelmistokehystä hyödyntäen. Päätös perustui ensisijaisesti haluun tehdä itsenäinen soitin, joka ei olisi riippuvainen muusta ohjelmistosta. HISE:stä ja sen ominaisuuksista kerron lisää luvussa 4.3.1.

Toteutustavasta riippumatta tulee sample-pohjaista soitinmallinnusta tehdessä kiinnittää huomiota myös mallinnettavan soittimen äänentuottomekanismiin, soittajaan, äänitystapaan ja äänitystilaan. Näitä muuttujia on rajattomasti, joten on tärkeää jo valmisteluvaiheessa päättää, mitkä asiat halutaan äänittää – ja miten kattavasti.

Omassa virtuaalisoitimessani tavoittelin realistisuutta, mutta halusin pitää ääninäytteiden määrän vähäisenä, jotta lopullinen soitin olisi prosessorille kevyt ja helpompi ohjelmoida. Siksi päätin äänittää soittimesta vain yhden artikulaation. Lisäksi mallintamani kellopelin äänen karakteristiikka ei äänenvoimakkuuden lisäksi muutu merkittävästi riippuen siitä, kuinka kovaa soitinta lyödään. Päätin, että riittävän realistiseen lopputulokseen riittäisi, että virtuaalisoitimen toistaman ääninäytteen äänenvoimakkuus ja nousu-aika olisi sidonnainen ohjausdatan sisältämään velocity-arvoon eli käytännössä siihen, miten kovaa soitinta ohjaavan MIDI-ohjaimen kosketinta painetaan. Käsittelen velocity-arvoa ja siihen liittyvää toiminnallisuutta tarkemmin luvussa 4.3.

## 4.2 Äänitys ja editointi

Soitinmallinnukseen soveltuvia sampleja äänitettäessä on äänenlähde taltioitava mahdollisimman realistisen kuuloisena - joskin realismi tässä yhteydessä on häilyvä käsite. Miltä soitin kuulostaa riippuu kuuntelijan fysiologisista, koulutuksellisista ja psykologisista ominaisuuksista, soittimesta, soittajasta, soittotavasta, huoneesta, kuunteluetaisyydestä ja kontekstista, jossa soitinta kuunnellaan. Paneudun tähän tarkemmin luvussa 5. Tässä vaiheessa oleellista on taltioida mallinnettava soitin siten, että se kuulostaa tallennettuna mahdollisimman samanlaiselta kuin äänityshuoneessa kuultuna.

Päätin äänittää kellopelin Finnvox Studiot OY:n B-studion soittohuoneessa, yhdellä COLES 4038 -mikrofonilla, noin metrin etäisyydellä soittimesta Avidin Pro Tools Ultimate -ohjelmistoon. Mikrofonivalinta perustui vertailuun muutaman eri mikrofonin välillä. Mie-

lestäni COLES taltioi kokeilemistani mikrofoneista parhaiten sen, miltä soitin kuulosti korvaani samalta etäisyydeltä. Mikrofonivalinnan olisi myös voinut tehdä mittaamalla kaikkien vaihtoehtojen taajuusvasteet ja käyttämällä tasaisimman taajuusvasteen omaavaa mikrofonia, mutta päätin luottaa omaan aistikokemukseeni, sillä tasaisin mahdollinen taajuusvaste ei välttämättä takaa miellyttävimmän kuuloista ääntä. Lisäksi äänityshuone korostaa ja vaimentaa tiettyjä taajuuksia, joten tasaisen taajuusvasteen omaava mikrofoni ei välttämättä takaa tasaista äänitettä.



Kuvio 2. Äänitystilanne (Kuva: Eelis Lähteenaro)

Soittimen asetin pöydälle keskelle huonetta. Tämän jälkeen äänitin jokaisen kielen erikseen antaen äänen soida, kunnes se vaimeni kuulumattomiin. Äänitettyäni kelloperlin jokaisen kielen kuuntelin äänittämäni materiaalin ja äänitin uudestaan ne äänet, joiden karakteristiikka oli poikkeava tai joihin olin tyytymätön. Osoittautui haastavaksi lyödä kaikkia kieliä täsmälleen samalla tavalla, joten jouduin äänittämään useita ottoja ja valitsemaan jälkikäteen niistä sopivimmat. Tämän olisi voinut välttää käyttämällä ammattisoittajaa, kuten varmasti jatkossa pyrinkin tekemään.

Kaikki mikrofonit, kaapelit ja etuasteet lisäävät äänitteisiin jonkin verran häiriöääntä, joka tyypillisimmin ilmenee kohinana. Varsinkin kelloperlin kaltaisissa hiljaisissa äänissä tämä

kohina nousee helposti kuultavalle tasolle, kun äänitteiden äänenvoimakkuutta joudutaan nostamaan jälkikäteen. Poistaakseni äänitteestäni näitä häiriöääniä päädyin käyttämään Izotopen RX8 denoise -kohinanpoisto plug-iniä. Kohinanpoisto voi vaikuttaa äänen taajuusjakaumaan, mutta totesin tämän olevan niin vähäistä, että se ei vaikuttaisi merkittävästi siihen, miten realistiselta lopullinen soitinmallinnus kuulostaisi. Siivottuani äänitteen kohinasta leikkasin kaikki iskut erillisiksi ääninäytteiksi, jotka exportoin .wav-tiedostoina.

### 4.3 Soitinrakennus

Jotta ääninäytteistä voitaisiin rakentaa virtuaalisoitin, on soittimelle suunniteltava ja toteutettava käyttöliittymä sekä logiikka, joka mahdollistaa ääninäytteiden toistamisen halutulla tavalla. Virtuaalisoitimen käyttöliittymän rakentaminen ei sopivalla ohjelmistokehyksellä välttämättä vaadi juuri yhtään ohjelmointia. Ohjelmoimalla voidaan kuitenkin esimerkiksi muokata soittimen toiminnallisuutta, tai lisätä ominaisuuksia, joita ohjelmistokehys ei tarjoa valmiina.

#### 4.3.1 HISE

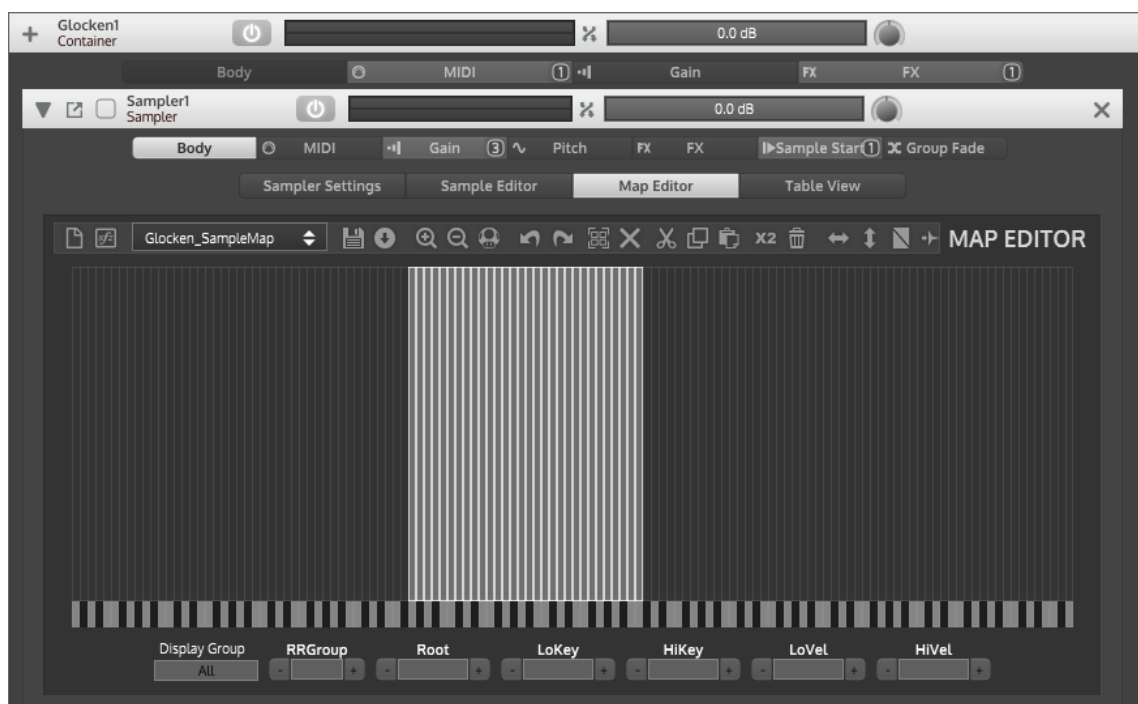
Ohjelmointia helpottaakseni päätin käyttää HISE-nimistä ohjelmistokehystä, jonka ominaisuudet toimivat valmiina runkona lopulliselle virtuaalisoitimelleni. HISE on avoimen lähdekoodin kehys, joka on kehitetty nimenomaan sample-pohjaisia soittimia varten. HISEn tarjoamat, tässä opinnäytetyössä kehittämäni soittimen kannalta oleelliset ominaisuudet ovat äänen suoratoistoa varten optimoitu sampler-moottori, MIDI-yhteensopivuus sekä käyttöliittymäeditori.

Jotta HISEn avulla voisi exportoida rakentamiaan ohjelmistoja, on HISE ensin käännettävä lähdekoodista itsenäiseksi ohjelmaksi. Jotta kääntäminen olisi mahdollista, tarvitaan lähdekoodin lisäksi myös Visual Studio 2017 -kääntäjä, Steinbergin ASIO SDK- ja vst SDK -ohjelmistokehityspaketit sekä Intel Performance Primitives -ohjelmistokirjasto. Lähdekoodi, tarkemmat ohjeet kääntämiseen sekä linkit kolmannen osapuolen ohjelmistoihin löytyy GitHubista, osoitteesta <https://github.com/christophhart/HISE>.

### 4.3.2 Toiminnallisuus

Tässä luvussa kuvailen virtuaalisoitimen rakennusprosessia, mutta en aio paneutua tarkemmin HISE:n sisäiseen logiikkaan ja kirjoittamiini skripteihin. Tässä opinnäytetyössä tarkkoja koodiesimerkkejä oleellisempaa on, mikä haluttu toiminnallisuus on ja miten se voidaan saavuttaa.

Soittimen rakentaminen alkoi tuomalla äänittämäni ääninäytteet HISEen, ja määrittelemällä näiden sävelkorkeus ja velocity-arvo HISEn sample map -editorilla. Velocity-arvo on luku 0 ja 127 välillä, joka voidaan välittää osana MIDI-tiedonsiirtojärjestelmää. MIDIä käytetään usein tiedonvälittämiseen MIDI-kontrollerin (kuten MIDI-koskettimiston) ja virtuaalisoitimen välillä, jolloin velocity-arvo voi välittää ohjelmistolle esimerkiksi tiedon siitä, miten kovaa ohjelmistoa kontrolloivan kontrollerin kosketinta on painettu. Velocity-arvon lisäksi MIDI:n kautta välittyy esimerkiksi sävelkorkeusarvo, jonka myös määrittelin jokaiselle ääninäytteelle erikseen. (Musician's HQ 2021.)



Kuvio 3. HISEn sample map -editori. Jokainen valkoinen pystyviiva vastaa yhtä ääninäytettä. Ääninäytteet asetetaan editorilla siten, että niiden sävelkorkeudet vastaavat alla olevan koskettimiston koskettimia.

Velocity-arvoa voi hyödyntää virtuaalisoitimessa määrittelemällä sen mukaan esimerkiksi toistettavan ääninäytteen äänenvoimakkuus. Näin tein myös omassa soittimessani

– mitä kovempaa kontrollerin kosketinta painaa, sitä kovempaa virtuaalisoitin toistaa ääninäytteen. Näin voidaan saavuttaa realistisempi soittotuntuma MIDI-koskettimistolla - jos oikeaa kellopeliä lyö kovaa, on luonnollisesti myös soittimen tuottamaa ääni kovempi. HISE sisältää jo valmiiksi MIDI-datan vastaanottamiseen ja hyödyntämiseen tarvittavat ominaisuudet, joten pystyin helposti hyödyntämään MIDI-ominaisuuksia omassa virtuaalisoitimessani ilman ohjelmointia.

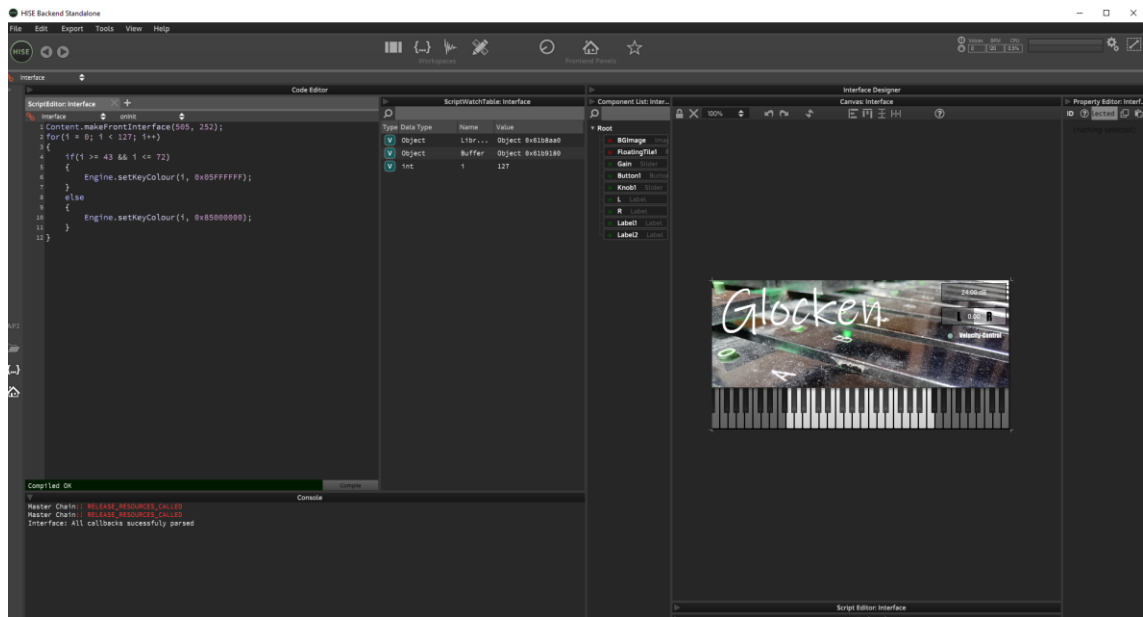
Oikean kellopelin transientti, eli äänen syttymisen ensimmäinen vaihe, muuttuu riippuen siitä, kuinka kovaa kieltä lyödään. Mitä hiljempaa kieltä lyödään, sen pehmeämmäksi transientti muuttuu. Mallintaakseni tätä päätin käyttää velocity-arvoa paitsi äänenvoimakkuuden ohjaamiseen, myös toistettavan ääninäytteen aloitusajan siirtämiseen. Siirtämällä aloitusaikaa velocity-arvon mukaan niin, että matalilla velocity-arvoilla alkuperäisen ääninäytteen transientti siirtyi kuulumattomiin, saavutin pehmeämmän äänen matalilla velocity-arvoilla. Omassa soittimessani halusin pitää ääninäytteiden määrän vähäisenä, joten päädyin toteuttamaan erilaiset artikulaatiot ohjelmallisesti. Samankaltaisen ja todennäköisesti realistisemmän lopputuloksen olisi saavuttanut äänittämällä jokaisesta kielestä erikseen hiljempaa lyödyn version, ja toistamalla matalilla velocity-arvoilla tämän äänitteen.

#### 4.3.3 Käyttöliittymä

Kun kaikki ääninäytteet oli määritelty vastaamaan tiettyä nuottiarvoa ja kun soittimessa oli kaikki haluamani velocity-arvoon liittyvää toiminnallisuutta, suunnittelin käyttöliittymään. HISE tarjoaa tähän valmiin käyttöliittymäeditorin, jota hyödyntäen lisäsin soittimeen säätimet äänenvoimakkuudelle, panoroinnille sekä kytkimen sille, reagoiko soitin velocity-arvoihin vai ei. Asetin soittimen taustakuvaksi äänitysvaiheessa ottamani valokuvan. Tässä vaiheessa päätin myös nimetä rakentamani soittimen Glockeniksi kellopelin saksankielisen nimen Glockenspielin mukaan. Lisäsin nimen soittimen käyttöliittymään ja sommittelin säätimet sopivan kokoisiksi ja sopiville paikoille käyttöliittymäeditorilla. HISE kirjoittaa itsenäisesti jokaiselle käyttöliittymäeditorilla lisätylle säätimelle koodia, jota editoimalla voi määritellä säätimen toiminnallisuuden. Pelkäsin tämän olevan yksi koko projektin haastavimmista osuuksista, mutta HISEn työkalut osoittautuivat erittäin kattaviksi, eikä toiminnallisuuden toteuttamiseen liittynyt juuri yhtään ongelmia.

Käyttöliittymän visuaalinen suunnittelu ja sommittelu osoittautui sen sijaan odotettua haastavammaksi. Olin etukäteen piirtänyt suunnitelman käyttöliittymästä, mutta HISEn

käyttöliittymäeditorin rajallisten toimintojen sekä ajanpuutteen vuoksi jouduin tekemään jonkin verran kompromisseja. Seuraavassa soittimessani aion toteuttaa käyttöliittymän kokonaan HISEstä erillisenä kuvana, jonka päälle lisään HISEssä toiminnallisuuden. Kytkeinten ja säädinten ulkonäön muokkaaminen vaatii myös osaamista vektorigrafii-kasta, jota minulla ei ole. Siksi päädyin käyttämään HISEn valmiina tarjoamia default-säätimiä, jotka eivät mielestäni ulkonäöllisesti sopeeneet käyttöliittymään.



Kuvio 4. Valmis käyttöliittymä avattuna HISEn käyttöliittymäeditorissa. Kuvassa vasemmalla näkyvä koodi määrittelee käyttöliittymän koskettimiston koon ja värin.

#### 4.4 Lopputulos ja kehitysmahdollisuudet

Lopullinen soitinmallinnus on itsenäinen ohjelma, joka lukee MIDI-ohjausdataa ja toistaa ääntä. Soitinta voi soittaa esimerkiksi MIDI-koskettimistolla, ja soittotuntuma on mielestäni hyvä. Äänenvoimakkuus muuttuu odotetunlaisesti riippuen siitä, miten kovaa koskettinta painaa, ja soittimella voi soittaa riittävästi ääniä samanaikaisesti.

Realistisuutta tavoittelevassa soitinmallinukessa kehitysmahdollisuudet ovat kuitenkin rajattomat. Glockenin tapauksessa varsinkin käyttöliittymä olisi kaivannut enemmän hioamista. Kaikki ohjaimet toimivat, mutta esimerkiksi äänenvoimakkuuden asteikko -100–24 dB tuntuu umpimähkäiseltä ja epäintuitiiviselta. Lisäksi käyttöliittymässä on muutamia bugeja, joita en ehtinyt korjata – kuten se, että tekstejä voi maalata tuplaklikkaamalla ja se, että ikkuna ei skaalaudu oikein, kun ohjelmaa suurennetaan vetämällä.

Tyypillisesti soitinmallinuksissa on myös enemmän soittimen ääneen liittyviä toimintoja ja mahdollisuuksia kuin Glockenissa. Monesti käyttäjä voi esimerkiksi valita usean eri mikrofoniin ja mikrofoni tekniikan välillä. Lisäksi soitinmallinuksissa hyödynnetään usein round-robiniksi kutsuttua tekniikkaa. Kun ihminen soittaa oikealla soittimella saman äänen monta kertaa peräkkäin, kuulostaa ääni joka kerta hieman erilaiselta. Tätä variaatiota voidaan emuloida round-robinilla, eli toistettavan ääninäytteen satunnaisella valinnalla. Jotta tämä olisi mahdollista pitää jokaisesta mallinnettavan soittimen koskettimesta äänittää useampi äänite.

Näiden toimintojen poisjättäminen oli tietoinen valinta, sillä tiesin että ensikertalaisena soitinrakentajana niiden toteuttaminen veisi liikaa aikaa. Tuleviin soitinmallinuksiin aion äänittää enemmän ääninäytteitä, jotta voin hyödyntää round-robinia. Lisäksi haluan tulevaisuudessa käyttää ammattisoittajaa minimoidakseni äänitteiden väliset erot ja nopeuttaakseni äänitysprosessia.

Glocken on testattu toimivaksi Windows 7 -käyttöjärjestelmällä itsenäisenä ohjelmana sekä VST-pluginina Image Linen FL-Studio 20 -ohjelmistossa. En kuitenkaan ehtinyt rakentaa Glockenille asennusohjelmaa, joten soittimen asentaminen ei ole kovinkaan käyttäjäystävällistä. Asennuksen yhteydessä käyttäjän pitää itse määritellä, mihin Glockenin käyttämä, erikseen ladattava samplemap-tiedosto on tallennettu. Tuleville soitinmallinuksille aion rakentaa myös asennusohjelman, joka hoitaa linkittämisen käyttäjän puolesta. Lisäksi haluaisin tehdä myös macOS:lla itsenäisesti toimivan version, sekä AAX-pluginin, jota voi käyttää esimerkiksi Avidin Pro Tools -ohjelmistossa.



Kuvio 5. Lopullinen soitin ja käyttöliittymä itsenäisenä ohjelmana

## 5 Realistinen ääni

Kun aloin rakentaa Glockenia, tavoitteeni oli toteuttaa mahdollisimman realistinen mallinnus oikeasta soittimesta. Jotta voisin tutkia, miten realistiselta rakentamani virtuaalisoitin kuulostaa, on ensin pohdittava, mitä realismi tässä yhteydessä tarkoittaa, miten sitä voidaan mitata sekä mikä siihen vaikuttaa.

Realismi määritellään sanakirjassa todellisuuden todenmukaiseksi kuvaamiseksi. (Wikisanakirja 2021) Äänestä puhuttaessa tämä todellisuus on kuitenkin monen subjektiivisen ja objektiivisen tekijän summa. Esimerkiksi äänen tallennus- ja toistometodi, korvan anatomia, kuuloaivokuori sekä erilaiset psykologiset tekijät, kuten kuuntelutottumukset, vaikuttavat käsitykseen äänestä. Tässä luvussa käyn läpi näitä muuttujia sekä pohdin millä tavoin ääntä voidaan analysoida.

### 5.1 Ääni fyysisenä ilmiönä

Ääni on värähtelyn aiheuttama paineenmuutos, joka aiheuttaa ihmisessä kuulokokemuksen. Äänenlähteenä voi toimia mikä tahansa esine tai asia, joka värähtelee noin 20–20000 kertaa sekunnissa. Hidas värähtely kuuluu matalana äänenä, kun taas nopea värähtely kuuluu korkeana äänenä. Tämän taajuusalueen ulkopuolisia ääniä kutsutaan

infra- ja ultraääniksi, eikä ihminen voi kuulla niitä paljaalla korvalla. (Laaksonen 2013, 4-5.)

Jotta ääni voisi muodostua, tarvitaan värähtelijän lisäksi välittäjäaine, jonka lävitse värähtelijän aiheuttamat paineenmuutokset eli ääniaallot, voivat saavuttaa kuuntelijan. Kuulokokemusten yhteydessä tämä välittäjäaine on yleensä ilma, mutta ääni voi kulkea missä tahansa riittävän tiheässä välittäjäaineessa. (Laaksonen 2013, 4-5.)

## 5.2 Yläsävelsarja ja interferenssi

Sample-pohjaisessa soitinmallinnuksessa realismin kannalta tärkeäksi ilmiöksi nousee samanaikaisesti soivien äänten yläsävelsarjat ja interferenssi. Ääni, kuten muutkin säännölliset aaltoliikkeet, koostuvat useista yksittäisillä taajuuksilla värähtelevistä osataajuuksista. Äänestä puhuttaessa näitä osataajuuksia sanotaan yläsäveliksi tai harmonisiksi osavärähtelyiksi. Kun esimerkiksi pianolla soitetaan tietty ääni, määrittelee kaikkein matalin osavärähtely kyseisen äänen sointikorkeuden. Tätä alinta osasäveltä kutsutaan perustaajuudeksi, ja sen ylle rakentuvat osasävelet muodostavat yläsävelsarjan. Kun kaksi ääntä soi samanaikaisesti, kertautuvat äänten perustaajuudet ja yläsävelsarjat aaltoliikkeiden aallonharjojen kohdatessa säännöllisin väliajoin. Tätä ilmiötä sanotaan interferenssiksi, ja se vaikuttaa molempien äänten yläsävelsarjoihin – ja täten myös sointiin. (Laaksonen 2013, 8-9.)

Sample-pohjaisessa soitinmallinnuksessa näitä samanaikaisesti soivien äänten muodostamia interferenssejä on todella vaikea mallintaa, sillä tämä vaatisi jokaisen mahdollisen äänikombinaation erillisen samplaamisen, joka esimerkiksi pianon tapauksessa tarkoittaisi satoja miljardeja yhdistelmiä. On siis odotettavaa, että samanaikaisesti soitettut äänet eivät Glockenissakaan kuulosta yhtä realistisilta kuin yksittäin soitettut äänet.

## 5.3 Äänen tallennus ja toisto

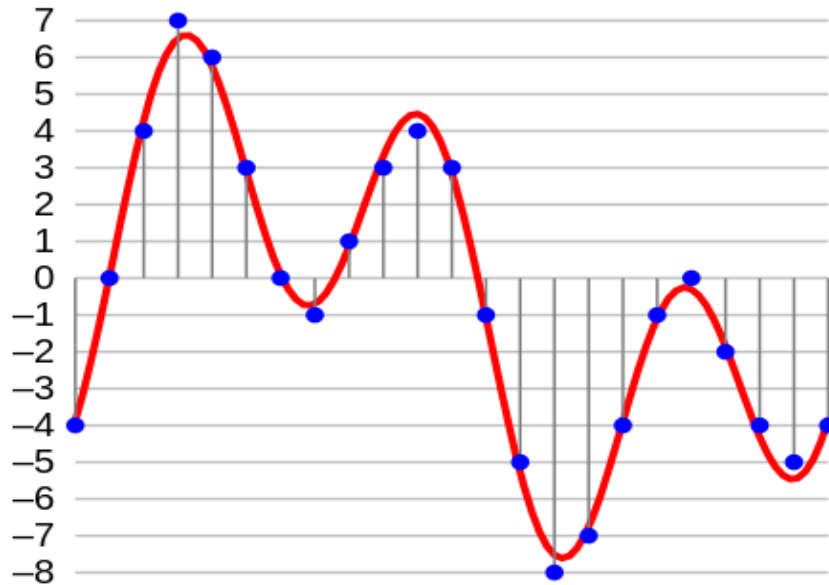
Ensimmäinen ja kenties tärkein muuttuja realismin kannalta samplepohjaisessa soitinmallinnuksessa on alkuperäisen äänen tallentamisen metodi. Äänen tallentaminen käsittää ääniaaltojen mekaanisen, optisen tai digitaalisen tallentamisen myöhemmin toistettavaan muotoon, kuten vahalevyyn tai .wav-tiedostoon.

Tallennukseen käytetään yleensä mikrofonia, joka poimii akustisen äänen aiheuttamat paineenmuutokset ilmasta, muuntaa ne sähköksi ja lähettää ne edelleen tallennettavaksi. (Laaksonen 2013, 230.)

Mainitsin luvussa 4.2, että äänitin teososassa käyttämäni samplet 48 kHz/24 bit .wav -formaattissa. Tämä tarkoittaa, että sample-soittimeni toistamiin .wav-tiedostoihin tallennettu bittivirta sisältää 48000 24-bittistä samplea per sekunti. Tässä yhteydessä sanalla sample ei viitata ääninäytteisiin, joita samplerit toistavat, vaan numeroon, joka kuvaa tallennetun ääniaallon muotoa tietyssä ajallisessa pisteessä. Näiden pisteiden määrää sekunnissa sanotaan näytteenottotaajuudeksi. Suurempi näytteenottotaajuus johtaa tarkempaan representaatioon akustisesta äänestä, mutta luonnollisesti myös suurempaan tiedostokokoon. Jo 44,1 kHz näytteenottotaajuus mahdollistaa koko ihmisen kuuloalueen tallentamisen, ja siksi se on yleisin käytetty näytteenottotaajuus, vaikka taajuus voisi periaatteessa olla paljon suurempikin. (Brown 2019) Päätin äänittää Glockenin ääninäytteet 48 kHz näytteenottotaajuudella, sillä arvelin tätä suuremman näytteenottotaajuuden vaikeuttavan usean samplen reaaliaikaista toistamista virtuaalisoitimella - varsinkin hitaammilla tietokoneilla.

Toinen äänenlaatuun vaikuttava muuttuja digitaalisesti tallennetussa äänessä näytteenottotaajuuden lisäksi on bittisyvyys. Bittisyvyys määrittelee, kuinka suurta asteikkoa ääniaallon amplitudin kuvaamiseen voidaan käyttää. Suurempi bittisyvyys johtaa parempaan mittaustarkkuuteen sekä tarkempaan representaatioon akustisesta äänestä, mutta myös suurempaan tiedostokokoon. Esimerkiksi 24 bittiin mahtuu 16777215 eri arvoa ja yksi 24-bittinen sample on tallennetun ääniaallon amplitudia kuvaava arvo asteikolla -388608–8388607. (Home DJ Studio 2021.)

Tavoitteeni oli tallentaa kellopeli realistisesti, eli niin kuin minä kuulin sen huoneessa. Teososani realistisuuden analysoimisen kannalta on oleellista pohtia, miten käyttämäni digitaalinen tallennusprosessi vaikuttaa ääneen, sillä tallennettu ääni on aina vain tietyn tarkkuuden omaava representaatio oikeasta akustisesta äänestä. Täten täydellinen realismi ei ole digitaalisessa samplepohjaisessa soitinmallinnuksessa saavutettavissa, edes silloin kun sampleja on rajaton määrä. Ääntä voidaan tallentaa myös analogisesti, jolloin realismin kannalta ongelmalliseksi nousevat epätarkkuuksien sijasta analogiseen tallennukseen vaadittavien sähkökomponenttien aiheuttamat häiriöt, mutta tässä opinnäytetyössä keskityn digitaaliseen tallennukseen.



Kuvio 6. Visuaalinen representaatio digitaalisesti tallennetun äänen bittisyvyydestä ja näytteenottotaajuudesta. Punainen linja kuvastaa ääniaaltoa, ja vasemmalla on bittisyvyyden määrittelemä ääniaallon amplitudia kuvaava asteikko. Siniset pisteet ovat näytteitä, joiden avulla ääniaallon muoto voidaan approksimoida ja tallentaa digitaalisesti. (Kuva: Wikipedia 2013)

#### 5.4 Äänen havaitseminen ja mittaus

Ääneen liittyy paljon mitattavissa olevia muuttujia, kuten taajuus ja äänenvoimakkuus. Näitä mittaamalla voidaan muodostaa tarkka, objektiivinen kuva äänestä. Ihmisen pääasiallinen tapa havaita ääntä on kuitenkin kuuloaisti. Siksi on järkevää hyödyntää mittauksen lisäksi myös aistikokemusta sample-soittimen realistisuuden arvioimiseen, vaikka aistikokemus onkin aina subjektiivinen.

##### 5.4.1 Mittaaminen

Äänen eri ominaisuuksia voidaan mitata erilaisin keinoin. Tavallisimmin mitattu ominaisuus on todennäköisesti äänenvoimakkuus, joka ilmaistaan desibeleinä (dB) ja mitataan desibelimittarilla. Toinen analysoinnin kannalta merkittävä mitattava määre on äänen spektri eli taajuusjakauma. Taajuusjakauma voidaan mitata spektrianalysointorilla, joka tarjoaa graafisen representaation äänen sisältämästä taajuusinformaatiosta.

Ääniä analysoidessa on perusteltua tutkia ennen kaikkea näitä fyysisiä ominaisuuksia, sillä nämä ovat helposti verrattavissa keskenään, eivätkä ole riippuvaisia kuuntelijasta.

### 5.4.2 Kuulokokemus

Äänen matka välittäjäaineessa liikkuvasta ääniaallosta kuulokokemukseksi alkaa ulkokorvasta, jonka kautta ääniaallot matkaavat korvakäytävää pitkin välikorvaan. Välikorvassa vasara-, alasin- ja jalustinluu siirtävät äänen edelleen simpukkaan, jossa ääni muuttuu sähköimpulsseiksi, joiden avulla aivot muodostavat kuulokokemuksen. (Lotto & Holt 2010.)

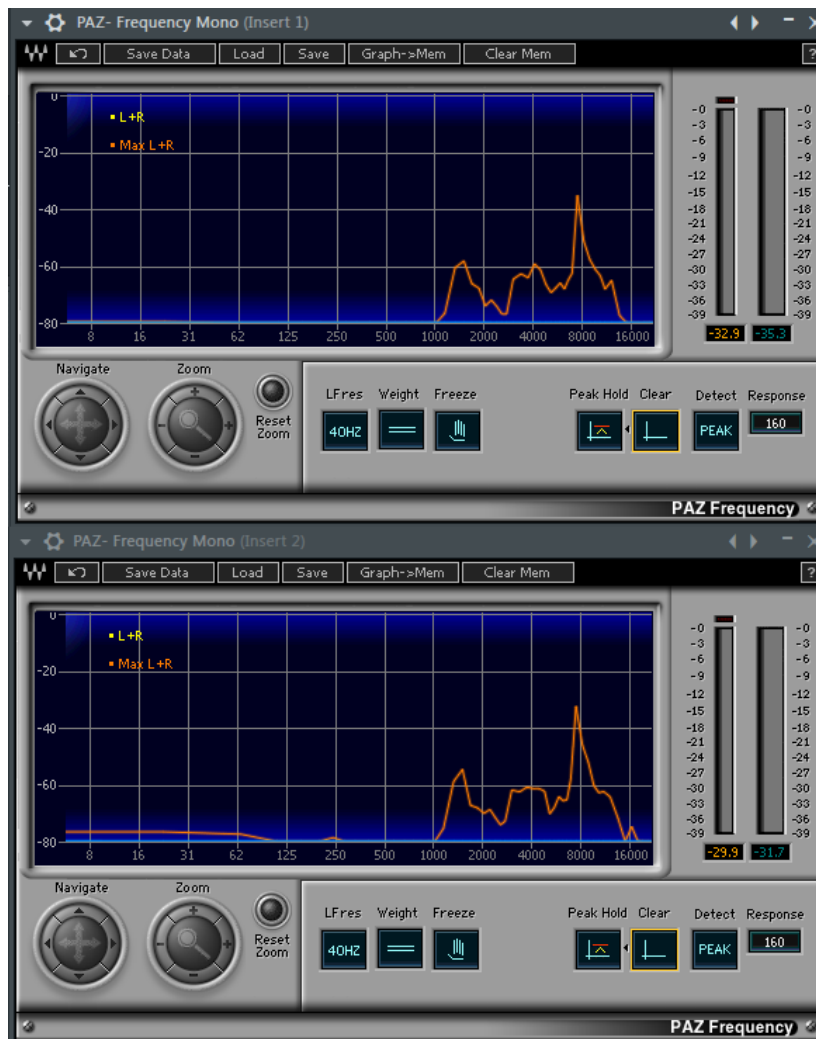
Fysiologian lisäksi on selvää, että kuulokokemukseen vaikuttavat psykologiset tekijät. Läheskään kaikkia näitä tekijöitä ei tunneta, mutta on helppo kuvitella, että ainakin aikaisemmat kuuntelukokemukset sekä assosiaatiot vaikuttavat tapaamme tulkita kuulemaamme ääntä. (Lotto & Holt 2010.) Tässä opinnäytetyössä tekemäni kuunteluun perustuvan arvioinnin yhteydessä päätin sivuuttaa näiden psykologisten tekijöiden vaikutuksen, sillä niiden mittaaminen ja minimointi on mahdotonta, ja uskon vaikutuksen olevan vähäpätöinen.

## 6 Vertailu

Tutkiakseni, miten realistiselta Glocken kuulostaa, vertasin sen ääntä alkuperäisen soitinajan äänitykseen. Tässä luvussa käyn läpi vertailuprosessia, sekä vertailun lopputuloksia. Vertailuun käytin spektrianalyysiä ja kuuloaistiani.

### 6.1 Spektrianalyysi

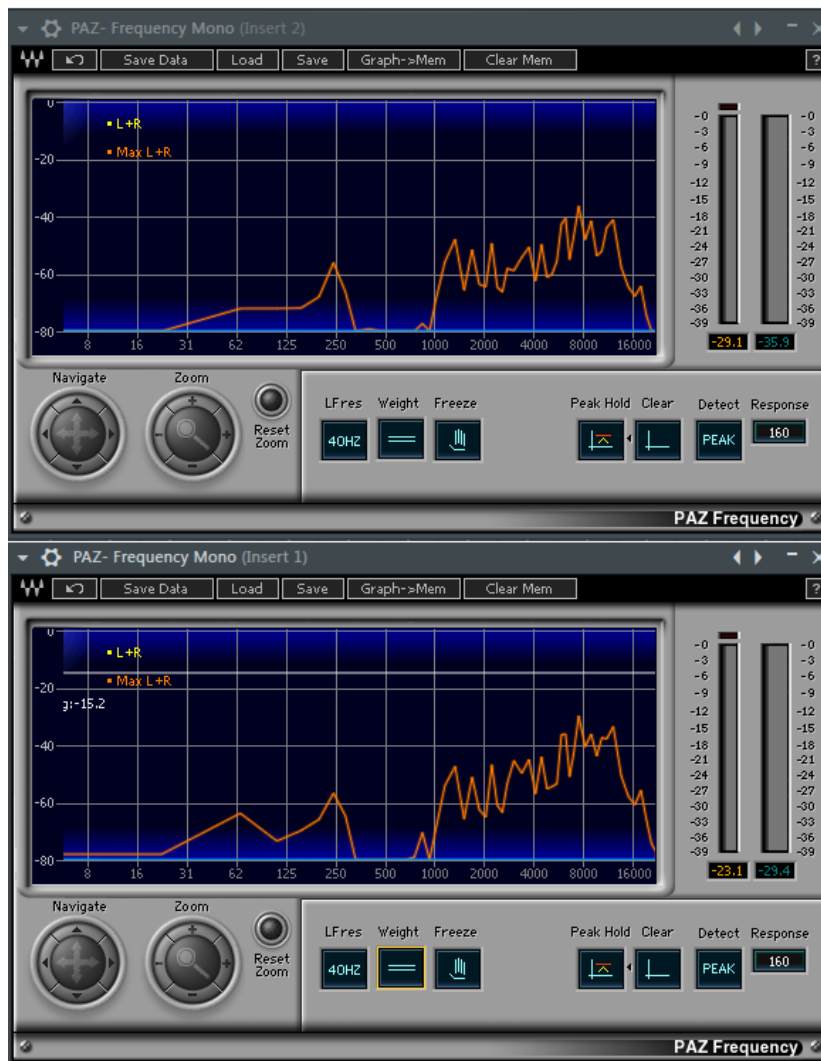
Aloitin vertailun äänittämällä lyhyen melodian oikealla kellopelillä, samalla mikrofonilla ja mahdollisimman samanlaisella tekniikalla kuin mallinnusta varten äänittämäni ääninäytteet. Tämän jälkeen soitin ja äänitin saman melodian soitinmallinnuksellani. Toin molemmat äänitteet Image Linen FL-studio -äänenkäsittelyohjelmistoon, reititin ne omille mikserikanavilleen ja asetin äänitteiden äänenvoimakkuudet samalle tasolle. Aloitin spektrianalyysillä, mittaamalla molemmista äänitteistä ensin yhden yksin soivan nuotin taajuusjakauman Wavesin PAZ frequency-analyzer -spektrianalysointilla.



Kuvio 7. F-nuotin taajuusvaste Glockenilla (ylempi kuva) ja kellopelillä (alempi kuva) soitettuna. Y-akseli kertoo äänenvoimakkuuden ja X-akseli kertoo taajuuden. Oranssi linja on representaatio koko taajuusvasteesta.

Kuten kuviosta 7 huomaa, ovat yksittäisten nuottien taajuusvasteet hyvin samankaltaiset. Selvin ero on 100 hertsin alapuolella kellopelin taajuusvasteessa näkyvä kohina, joka on siivottu pois Glockenin käyttämisestä sampleista. Kohina on niin hiljaista, ettei sitä kuule paljaalla korvalla, joten tämä ei ole realistisuuden kannalta ongelma. Myös sointi-alueella on kuitenkin havaittavissa pieniä eroavaisuuksia, selvimpänä kellopelin äänitteessä näkyvä piikki 16 kHz kohdalla. Jää epäselväksi, mistä tämä poikkeama johtuu, mutta epäilen myös tämän johtuvan kohinanpoistosta, sillä tämä taajuusalue puuttuu Glockenilla soitetusta äänestä kokonaan.

Seuraavaksi päätin verrata useamman samanaikaisesti soitetun nuotin taajuusvasteita. Oletukseni oli, että interferenssin puute Glockenilla soitetuista äänistä näkyisi taajuusvasteessa.



Kuvio 8. Usean samanaikaisesti soivan nuotin taajuusvasteet mallinnuksella (ylempi kuva) ja kellopelillä (alempi kuva) soitettuna. Y-akseli kertoo äänenvoimakkuuden ja X-akseli kertoo taajuuden. Oranssi linja on representaatio koko taajuusvasteesta.

Ero on selvempi kuin yksin soivia nuotteja verrattaessa, mutta erot taajuusvasteissa osoittautuivat silti odottamaani pienemmäksi (ks. kuvio 8). Odotin interferenssin puutteen näkyvän selvästi varsinkin mallinnuksen korkeissa taajuuksissa, mutta vaikka äänitteiden välillä on pieniä eroja, on vaikea sanoa johtuvatko ne yläsävelsarjoista, kohinanpoistosta, eroista äänitystavoissa vai jostain muusta.

Oikean kellopelin äänityksessä on edelleen nähtävissä matalien taajuuksien kohina, mutta merkittävimmäksi eroksi nousee 60 hertsin yläpuolella näkyvä piikki, joka puuttuu mallinnuksesta kokonaan. En keksi ilmiölle mitään selitystä. Oletan tämänkin johtuvan jostakin Glockenin ääninäytteille tehdystä prosessoinnista tai pienistä eroista äänitystavoissa.

## 6.1 Kuuntelu

Kuunteluvertailun tein kuuntelemalla kellopelin äänitettä ja soitinmallinnuksella soitettua äänitettä vuoron perään. Kuuntelemiseen käytin Audio Technican ath-m50x -kuulokkeita. Mallinnus ja oikea äänite kuulostivat mielestäni lähes erottamattomilta. Ne pienet erot, joita mallinnuksen ja äänitteen välillä oli kuuntelemalla havaittavissa, johtuvat todennäköisesti pienistä eroista soittotavoissa ja mikrofonin sijainneissa. Odotin usean samanaikaisen ja peräkkäin soitettun äänen kuulostavan mallinnuksella soitettuna keinoitekoisemmalta, mutta tämäkään ei omaan korvaani kuulostanut mitenkään selvästi erilaiselta kuin oikeassa akustisessa tilassa samanaikaisesti soivat äänet.

## 6.2 Vertailun lopputulokset

Kuuntelun lopputulos ei ole yllättävä – onhan Glockenin toistamat äänet täsmälleen samalla (tai ainakin hyvin samankaltaisella) tavalla äänitetty kuin vertailuun käyttämäni äänitteet. Olen tyytyväinen lopputuloksiin, sillä tarkoitukseni oli rakentaa realistiselta kuulostava sample-pohjainen soitinmallinnus ja mielestäni onnistuin tavoitteessani. Toisaalta olisin toivonut vertailun tarjoavan mielenkiintoisempia lopputuloksia, varsinkin spektrianalyysin suhteen. Käyttämäni metodi jätti äänitteiden välisten erojen syyt lähinnä arvailun varaan, joten spektrianalyysi ei toiminut kovinkaan onnistuneesti realismin mittarina.

## 7 Pohdinta

Kun aloitin Glockenin rakentamisen, minulla oli vain hämärä käsitys kaikista samplepohjaiseen soitinmallinnukseen ja virtuaalisoitinten rakentamiseen liittyvistä työvaiheista. Halusin kuitenkin oppia lisää paitsi soitinrakennuksesta myös ääniohjelmoinnista ja ohjelmistoarkkitehtuurista yleisesti, ja ajattelin itsenäisenä ohjelmana toimivan soitinmallinnuksen rakentamisen opettavan minulle näitä molempia. Valmista ja hyvin dokumentoitua alustaa kuten Native Instrumentsin Kontaktia käyttämällä olisi todennäköisesti säästänyt aikaa ja vaivaa, mutta valmis soitin olisi silloin myös ollut täysin riippuvainen Kontaktista toimiakseen, joten olen tyytyväinen päätökseeni toteuttaa Glocken itsenäisenä ohjelmana.

Käyttämäni ohjelmistorajapinta HISE yllätti minut myös positiivisesti valmiiksi tarjoamiensa toimintojen helppokäyttöisyydellä ja määrällä – joita oli niinkin paljon, että olisin

välillä jopa toivonut joutuvani harjoituksen vuoksi ohjelmoimaan enemmän toimintoja itse. Toisaalta myös koko projektin haastavin osuus oli todennäköisesti HISEn kääntäminen lähdekoodista toimivaksi ohjelmaksi. Kääntämiseen liittyi kymmeniä yhteensopivuusongelmia, HISEn heikohkon dokumentoinnin aiheuttamia harmaita hiuksia sekä lähdekoodin manuaalista editointia – kunnes HISE lopulta toimi tietokoneellani. Tämä oli minunlaiselleni amatööriohjelmoijalle valtava haaste, mutta tässä onnistuminen oli jo itsessään lähes yhtä palkitsevaa ja opettavaista kuin Glockenin rakentaminen.

Myös päätökseni mallintaa helposti mallinnettava soitin kuten kellopele osoittautui oikeaksi. Tiesin että käytettävissä oleva aika, resurssit ja ennen kaikkea omat taitoni olivat rajalliset, enkä usko, että esimerkiksi jousi- tai puhallinsoittimen sample-pohjainen mallintaminen olisi onnistunut läheskään yhtä vaivattomasti.

Tiivistetysti voisi sanoa, että olen tyytyväinen lopputulokseen. Tarkoitukseni oli tutkia ja oppia, miten realistista soitinmallinnusta tehdään, ja koen että onnistuin siinä hyvin. Glocken on toimiva työkalu, ja vaikka en tässä projektissa hyödyntänytäkään kaikkia soitinmallinnukseen liittyviä kehittyneempiä tekniikoita kuten round-robinia, koen että ymmärrykseni sekä soitinmallinnuksesta että ohjelmoinnista on huomattavasti vahvempi kuin ennen Glockenin rakentamista.

Mitä tulee soitinmallinnusten tulevaisuuteen, uskon sen olevan erittäin valoisa. Sekä sample- että fysiikkapohjaiset soitinmallinnukset kehittyvät jatkuvasti, ja olen varma, että tulevaisuudessa mikä tahansa soitin voidaan mallintaa täysin erottamattomaksi oikeasta soittimesta.

## Lähteet

Albano, Joe 2019. Understanding The Architecture Of Sample-Based Virtual Instruments <https://ask.audio/articles/understanding-the-architecture-of-sample-based-virtual-instruments> (luettu 13.03.21)

Applied Acoustics. Physical Modeling Tech Talk. <https://www.applied-acoustics.com/techtalk/physicalmodeling/> (luettu 19.03.21)

Brown, Griffin 2019. Digital Audio Basics: Sample Rate and Bit Depth. <https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html> (luettu 12.04.21)

Davies, Hugh 1996. A History of Sampling. Organised Sound 1. [http://www.ocopy.net/wp-content/uploads/2017/10/davies-hugh\\_a-history-of-sampling.pdf](http://www.ocopy.net/wp-content/uploads/2017/10/davies-hugh_a-history-of-sampling.pdf) (luettu 15.03.21)

Encyclopaedia Britannica 2021a <https://www.britannica.com/art/glockenspiel>

Home DJ Studio. Ultimate Guide To Audio Bitrate & Audio Formats <https://homedjstudio.com/audio-bitrates-formats/> (luettu 21.04.21)

Howell, Steve 2005. The Lost Art of Sampling part 1, 2 & 3. Sound on Sound Magazine. <https://web.archive.org/web/20181209123730/https://www.soundonsound.com/techniques/lost-art-sampling-part-1> (luettu 15.03.21)  
<https://www.soundonsound.com/techniques/lost-art-sampling-part-2> (luettu 15.03.21)  
<https://www.soundonsound.com/techniques/lost-art-sampling-part-3> (luettu 15.03.21)

Laaksonen, Jukka 2013. Äänityön Kivijalka. 2. uudistettu painos. Keuruu: Otavan Kirjapaino Oy.

Lotto, Andrew & Holt, Lori 2010. Psychology of auditory perception. [https://www.cmu.edu/dietrich/psychology/holtlab/PDF/lori.%20MY%20Papers/Lotto-Holt\\_WIRES.pdf](https://www.cmu.edu/dietrich/psychology/holtlab/PDF/lori.%20MY%20Papers/Lotto-Holt_WIRES.pdf) (luettu 23.04.21)

Musician's HQ 2021. A Beginner's Guide To MIDI: What Is It? How Does It Work? <https://musicianshq.com/a-beginners-guide-to-midi/> (luettu 6.5.2021)

Wikipedia 2008 [https://fi.wikipedia.org/wiki/Fairlight\\_CMI#/media/Tiedosto:Fairlight.JPG](https://fi.wikipedia.org/wiki/Fairlight_CMI#/media/Tiedosto:Fairlight.JPG)

Wikipedia 2013. [https://en.wikipedia.org/wiki/Audio\\_bit\\_depth#/media/File:4-bit-linear-PCM.svg](https://en.wikipedia.org/wiki/Audio_bit_depth#/media/File:4-bit-linear-PCM.svg)

Wikisanakirja 2021. Realismi. <https://fi.wiktionary.org/wiki/realismi> (luettu 6.5.2021)

