

COMPOSING WITH FLEXIBLE PHRASES

The impact of a newly designed digital musical instrument upon composing western popular music for commercials and movie trailers

Dennis Herbert Braunsdorf

A thesis submitted in partial fulfilment of the requirements of
Bournemouth University for the degree of PhD.

2021

Research Supervisors:

Prof. Stephen Deutsch (BU)

Dr. Ambrose Seddon (BU)



This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and due acknowledgement must always be made for the use of any material contained in, or derived from, this thesis.

Abstract

This practice-based research project is an original investigation into the process of producing five compositions employing a self-designed digital musical instrument (DMI) called the 'Flexible Phrase System' (FPS). This research consists of two interconnected design processes: one is the production of a portfolio of music, and the other is the development of the FPS, which supports the production of the music. Composing the music and designing the FPS are developed in an iterative design process. The interconnected processes affect the content of the compositions, and it impacts the development of the FPS. Similarly, the FPS is designed for underscoring western advertising or movie trailers; therefore, the compositions are biased towards western popular music. All of the original works are three minutes long and used western popular instruments. The iterative design process offers insights into the compositional process and the artistic motivation that are intertwined with the possibilities provided by the FPS.

Audio-visual self-study observation methods are utilised to gain knowledge of composing, the impact of the FPS, and the implementation of an iterative design process. These self-study methods focus on the principles of autoethnographic studies, reflection-in-action studies, and studies of the creative process of music composition (CPMC).

This study contributes five western popular compositions for commercials and movie trailers to the repertoire of music, and it also contributes the newly designed DMI to the field of virtual instruments (VIs) and user interface controllers (UICs). It reveals original and critical insights into the iterative design of a DMI. Additionally, other composers, producers, and music technologists can use the developed self-study

observation methods and the iterative design process to analyse their compositional process with the aid of a self-designed DMI.

Acknowledgements

My thanks go to my research supervisors Professor Stephen Deutsch from Bournemouth University (BU), Dr. Ambrose Seddon from BU, and Gerard van Wolferen from the Utrecht School of the Arts (HKU) for their advice and support throughout the process of writing this PhD. My thanks also go to Jur Bruinink who supported me in programming the digital musical instrument (DMI) and to Nico Maas, who played the electric guitar. Furthermore, my gratitude goes to Ildiko Balogh from BU student support services, Eva Lobach, Rens Machielse, André Ettema, and Richard Ram for additional support. My gratitude goes to my parents, sister, family, and friends because without their patience and support I would not have been able to complete this PhD. A very special word of thanks goes to my wife Fleur Braunsdorf for proofreading my thesis and for her unremitting efforts.

TABLE OF CONTENTS

1. INTRODUCTION.....	10
1.1 PERSONAL CONTEXT IN MUSIC COMPOSITION	18
2. THE AIM AND OBJECTIVES OF THE RESEARCH.....	23
3. LITERATURE REVIEW.....	25
3.1 THE DEVELOPMENT OF MUSIC TECHNOLOGY.....	26
3.2 MUSIC AND HUMAN-COMPUTER INTERACTION (M&HCI).....	29
3.3 THE PRINCIPLES OF THE FLEXIBLE PHRASE SYSTEM (FPS).....	38
3.3.1 Defining the requirements for the FPS	41
3.3.2 Defining the synthesised instrument.....	45
3.3.3 Potential impact of the FPS requirements upon composing music	48
4. METHODOLOGY.....	50
4.1 THE PRINCIPLES OF THE ITERATIVE DESIGN PROCESS OF THIS CASE STUDY	50
4.2 DATA COLLECTION THROUGH COMPOSING AND SELF-STUDY TECHNIQUES	55
4.3 MUSICAL DESCRIPTION AND AUDIO-VISUAL ANALYSIS METHODS	59
4.4 POTENTIAL OUTCOMES, BENEFITS, AND RISKS OF CONDUCTING SELF-STUDY RESEARCH WHILE COMPOSING	61
4.5 ON THE ETHICS OF SAMPLING.....	64
5. COMPOSITION AND INSTRUMENT DESIGN	69
5.1 GENERAL COMMENTS ON THE RELATIONSHIP BETWEEN COMPOSING POPULAR MUSIC AND DESIGNING DIGITAL INSTRUMENTS	69
5.2 EVALUATION OF THE MUSIC & HUMAN-COMPUTER INTERACTION (M&HCI).....	72
5.2.1 The Alpha	75
5.2.2 The Beta.....	79
5.2.3 The Gamma	81
5.2.4 The Delta	84
5.2.5 The Omega	87
6. CONCLUSIONS	90

7. REFERENCES..... 93

8. APPENDICES

LIST OF FIGURES

FIGURE 1. THE ITERATIVE DESIGN PROCESS OF ACTION RESEARCH, ACCORDING TO KURT LEWIN.....	51
FIGURE 2. THE FIVE STAGES OF THE ITERATIVE DESIGN PROCESS OF THIS RESEARCH.....	52
FIGURE 3. OVERVIEW OF THE FIVE COMPOSITIONS AND THE FIVE SOFTWARE VERSIONS.	74

LIST OF TABLES

TABLE 1. THE MUSICAL PORTFOLIO AND THE USED FPS.	24
TABLE 2. OVERVIEW OF THE FIVE ITERATIONS.	75
TABLE 3. AN OVERVIEW OF THE FORM OF 'THE ALPHA'.	76
TABLE 4. AN OVERVIEW OF THE FORM OF 'THE BETA'.	79
TABLE 5. AN OVERVIEW OF THE FORM OF 'THE GAMMA'.	81
TABLE 6. AN OVERVIEW OF THE MUSICAL FORM OF 'THE DELTA'.	85
TABLE 7. AN OVERVIEW OF THE FORM OF 'THE OMEGA'.	88

1. Introduction

In 2011, I was working as a composition intern at MassiveMusic¹ in Shanghai. At that time, many companies were interested in underscoring their commercials with western popular music² (hereinafter referred to as popular music). However, in most cases, contractors did not have the time or financial resources to hire instrumentalists. I had to search for low budget alternatives to imitate the sound of an acoustic instrument³. I ended up with virtual instruments⁴ (VIs), which offered me the possibility to impersonate an instrumentalist within digital audio workstations⁵ (DAWs). One example of this process was a commercial for a fast-food company. I had to complete a 45 seconds piece of music within 24 hours - from nothing to a mixed and mastered orchestral piece. After 24 hours of work, I provided a finished version within the given time.

¹ MassiveMusic produces and composes music for the worlds of advertising, broadcast design, digital and film (MassiveMusic 2018).

² In western popular music, it “has been conventional to conceive of three broad categories of music: popular, art (or classical), and folk. Just as “art music” may refer to Gregorian chant, grand opera, string quartets, and minimalist percussion music, “popular music” is an umbrella term that encompasses a wide set of variously distinct though often related genres” (Hamm et al. 2014).

³ See chapter 4.5, including the discussion on the ethics of sampling.

⁴ A virtual instrument (VI) simulates the sound of a musical instrument. A developer of VIs is, for example, the Vienna Symphonic Library. The Vienna Symphonic Library is a research-driven music software and sample library developer based in Vienna, Austria (Vienna Symphonic Library 2002).

⁵ Digital audio workstations (DAW) are software environments used for recording, editing, and producing music and sounds (Kefauver and Patschke 2007).

After my internship, several movie directors and film production studios asked me to compose and produce popular music within the same restrictions as the fast-food commercial: fast, cheap, and high quality. Over the last ten years, having been commissioned by many different companies to compose popular music and musical sound design-based⁶ compositions for advertisement or movie trailers, the following issue arose: whether and how ‘humanising’⁷ the sound of VIs adversely affected my process of composing music.

Humanising VIs costs a lot of time. Within a DAW, I have to impersonate the sound of a VI manually because a Musical Instrument Digital Interface file⁸ (MIDI) that triggers a VI sounds unnatural and too mechanical. In contrast to a computer, an instrumentalist creates constant musical deviations from the score. Musical deviations include expressiveness that helps to clarify the musical character of the composition. Seashore (1938, p. 9) states that “the artistic expression of feeling in music consist in aesthetic deviation from the regular – from pure tone, true pitch, even dynamics,

⁶ Musical sound design-based connotes the use of any acoustic material in a musical way. The material can be, for example, noise, wind, or water sounds; then, the sounds are treated in a musical way.

⁷ In the music industry, the term ‘humanising’ refers to the process of making adjustments to the output of a virtual instrument (VI). These adjustments attempt to make a human-like sound from the instrument (Computer Music 2015).

⁸ The Musical Instrument Digital Interface (MIDI) file format saves recording and playback information. However, the MIDI protocol does not contain actual audio data but instructional information. This instructional information includes, e.g. pitch position, duration, timing, and dynamics. Digital synthesisers, MIDI controllers, or DAW software environments that support MIDI create the instructional information.

metronomic time, rigid rhythms, etc.” The result is the individual expressive interpretation of the musician. The performer’s expressive interpretation is also called a ‘performance gesture’. Within a MIDI environment, the performance gesture of the musician must, in most cases, be inserted manually, which is hugely time-consuming and requires considerable effort and knowledge from the user (Maestre et al. 2009). The process of programming performance gestures manually is based on my understanding of the musical instrument. However, in most cases, I cannot play the instrument, and it is doubtful whether my assessment and understanding of the musical instrument are sufficient to design an authentic performance. For a better understanding of the instrument, I can study Adler (2002) or practice the acoustic version of the VI; however, this would involve years of professional practice. An alternative of humanising is the process of randomising. A randomiser changes musical deviation arbitrarily; however, a randomiser does not take any account of the sound and the expressiveness of the instrument. The other alternative is to hire live musicians or produce hybrid productions by combining live musicians and synthesis. Unfortunately, this is an expensive and time-consuming undertaking.

Secondly, impersonating the performance gestures of an instrumentalist interrupts the compositional process. In contrast to composing, programming musical expressions manually is a more technical task. While writing music, I focus on working with harmonies, melodies, and different arrangements. While humanising, I focus on the sound quality of the instrument and the productional quality of the composition. An adverse effect of humanising is that it involves too much programming detail. As a consequence, I might set the wrong priorities and focus more on programming musical expressions rather than on finishing the composition.

Lindemann (2007), CEO of Synful Orchestra and former director of the IRCAM Signal Processing Workstation Project, has experienced similar issues. He describes how he wants to be able to compose music with the aid of VIs in a convincing expressive manner. For Lindemann, writing music has to be a musical experience rather than a humanising experience. The Reconstructive Phrase Modelling (RPM) is the result of Lindemann's personal musical experience and his search for a musical experience without excessive editing, pushing of individual buttons, or applying special phrasing tools. According to Lindemann, the combination of additive synthesis and phrase-oriented parametric concatenative synthesis is one way to achieve musical expressivity. The RPM system offers a wide range of synthesis parameters (some are related to expression). Unfortunately, such parameters must be tuned manually within the MIDI environment, which is time-consuming, and requires prior user knowledge. The RPM system lacks a representation of musical deviations that an instrumentalist introduces into any particular context (Maestre et al. 2009).

Moreover, humanising with a conventional MIDI keyboard⁹ affects my melodic and harmonic compositional choices adversely. I am a trained piano and keyboard player; therefore, I prefer to compose music with a MIDI keyboard. However, while working with a MIDI keyboard, I tend to write basic melodic or harmonic progressions to keep the process of programming musical expressions simple. I do not switch between different scales or different harmonic voicings because they make the process

⁹ A conventional MIDI keyboard is an external piano-style keyboard that sends and receives MIDI data to the DAW or other musical devices. The MIDI keyboard controls the virtualised instrument.

of humanising more complex. The process of humanising becomes more complex because different scales and harmonic voicings trigger different musical samples that introduce new sound characteristics, and they demand different humanising settings. For example, programming the expressiveness of a 'virtualised instrument' demands careful listening to the sound-properties of the triggered instrument. Every triggered sound demands unique humanising treatments. Composing the musical material within the same musical scale allows me to copy and paste the humanising settings that are programmed for one particular melodic or harmonic progression to a musical variation of another melodic or harmonic progression. After copying the humanising settings, I do minor adjustments to the settings in order to create some expressive variations. Those minor variations do not cost much time, and they fit into my compositional process. Sometimes, I even do not create minor humanising adjustments because the new melodic or harmonic material contains sufficient musical variations. However, if these new melodic or harmonic progressions are performed in another scale, the humanising settings cannot be easily copied and paste from one progression to another progression due to the different characteristics of newly introduced musical samples. Then, the humanising settings need major adjustments. The major adjustments are very time consuming, and they interrupt my production process.

Therefore, I usually prefer to stay in one musical scale and in one harmonic voicing in order to keep the sound quality of different articulations and harmonic progressions similar. Also, the standard MIDI keyboard controller does not provide me with a sufficient overview of the musical possibilities of the VI. It gives an insufficient overview because the physical structures of most non-keyboard instruments differ from

the physical structure of a MIDI keyboard¹⁰. Alternatively, I can use different MIDI controllers such as wind, drum, and percussion controllers, or string instrument controllers (DeRosa and Pejrolo 2011). In my experience, alternative MIDI controllers are less suitable for humanising different VIs because most VIs developers design the user interface based on the layout of a MIDI-keyboard¹¹. Moreover, they require considerable effort and knowledge from the composer as well as the relevant instrumental skill.

In summary, programming musical expressions or performance gestures manually costs a lot of time, interrupts my compositional process, and affects my melodic and harmonic choices. These factors prompted me to design an alternative digital musical instrument¹² (DMI) called the ‘Flexible Phrase System’ (FPS). The compositions which make up the compositional element of this study demonstrate the possibilities of the FPS and illustrate how the FPS affects musical contents,

¹⁰ An exception is the virtual piano instrument. Playing a musical note on an acoustic piano is very comparable to playing a musical note on the conventional MIDI keyboard.

¹¹ Common VI environments including Kontakt 6 and Reaktor 6 by Native Instruments, Halion 6 by Steinberg, the BBC Orchestra player by Spitfire Audio, or the SINE player by Orchestral Tools build their VIs engines and sample-mapping strategies based on the layout of a MIDI-keyboard.

¹² A digital musical instrument (DMI) is a musical instrument consists of a virtual instrument (VI) synthesis engine that simulates the sound of a musical instrument, and a user interface controller (UIC), also referred to as a performance controller, hardware interface, or gesture device.

compositional processes, and musical ideas¹³. On the other hand, the HCI of the FPS illustrates how the compositions impacted upon the development of the FPS. The compositions are meant to be used in advertisement or trailer music productions, and they are biased towards popular music; consequently, the human-computer interface of the FPS is biased towards popular musical productions. The FPS aims to offer an interconnected production and composition process, achieved with the assistance of a newly designed ‘Flexible Phrase System Virtual Instrument’ (FPS-VI) and a newly designed ‘Flexible Phrase System User Interface Controller’ (FPS-UIC). Both are independent software entities related to each other by MIDI-mapping strategies. The FPS-VI includes expression-related parameters that are inserted automatically, and it is responsible for the realisation of the performance gestures of an electric guitarist. Maestre et al. (2009, p. 23) state that the synthesis system should insert expression-related parameters automatically “by applying some prior knowledge about the expressive transformations a particular musician introduces when performing a piece.”

The FPS-UIC automatically balances harmonic voicings and inversions within the compositions. It is responsible for the harmonic progression, voicing, and the harmonious arrangement of the original production. According to DeRosa and Pejrolo (2011), an excellent harmonic balanced arrangement for VIs is crucial for the authenticity of the sound. They argue that composing music for digital imitations of

¹³ Once the compositional process has started, thinking about the quality and implications of the musical ideas is central to the art of composing. According to Schoenberg (1934 cited by Belkin 2015), “a musical idea is sheerly musical. It is a relation between tones.” A musical idea is resolved by writing music, listening to music, trying out musical possibilities – in short, through musical thinking.

acoustic instruments is no different from writing for acoustic instruments. The FPS-UIC controls the FPS-VI and other VIs, and the VI forms (together with the UIC) the FPS.

Based on the work of the music information retrieval¹⁴ (MIR) consortium (Serra et al. 2013), this study categorises the interconnected process of composing music and designing software as a creative practice that enhances music production. The MIR consortium emphasises a need for developing creative practices that take user needs and behaviour into account. Much of the work in MIR is technology-driven, rather than being user-, context-, or application-driven. Thus, the challenge is to step into the shoes of users and understand their worldview to produce useful applications (Serra et al. 2013, p. 38). As a music technologist¹⁵, my advantage includes my experience as a composer, music producer, and DMI designer. I am an appropriate researcher to do this practice-based research, which focuses on the interdisciplinarity of composition and organology. The interdisciplinarity contains the production of a portfolio of popular music and the design of the FPS, which supports the production of the music. Baalman (2017), who also studied the interplay between composition and instrument design, says that “with digital media as key element in musical instrument design, the boundaries between music composition and instrument design are blurring” (p. 225). Furthermore, Baalman argues that music technologists do not need to differentiate between composition and organology in their creative process; however, when music

¹⁴ MIR is a “multidisciplinary research endeavour that strives to develop innovative content-based searching schemes, novel interfaces, and evolving networked delivery mechanisms in an effort to make the world’s vast store of music accessible to all” (Burgoyne et al. 2015).

¹⁵ See 3.1. for more information including the definition of a music technologist.

technologists “present their work in the context of established cultural scene, they need to deal with the common division of roles” (Baalman 2017, p.225).

This thesis focuses on the interdisciplinarity of music composition and instrument design and how my professional practice acts in the field between those roles. Composing the music and designing the FPS are developed in an iterative design process. The result is a portfolio of music, which uses the FPS and a description of how the music has impacted the design of the HCIs.

1.1 Personal context in music composition

My professional field of work almost exclusively consists of music for advertising and music for movie trailers. It is worth noting that composing music for these two media is biased towards popular music.

Movie trailer music contractors ask me to compose compositions¹⁶ for action or adventure trailers, and they send me written briefings, existing reference music, or both. These written briefings and existing reference music affect my compositions. The written briefings include timeline descriptions, instrumentations, or music sphere impressions; however, the detail varies hugely. Sometimes, the contractor’s offer is very detailed, and I have to follow a strict plan; at other times, the briefing is very ambiguous, and the contractor allows me room for interpretation. Some contractors use existing compositions as reference tracks to support the written briefing. The reference

¹⁶ Compositions are newly composed compositions that have not been written or performed before. The original works are not music remixes, nor covers, quotes, or imitations of other existing compositions or performances.

music can include musical sound design-based material, orchestral accompaniment, and electric guitar sounds. The contractor asks me to follow the written briefing and to listen carefully to the sound of the reference track without committing plagiarism. Thus, my original work usually includes musical sound design-based material, an orchestra, and electric guitars. However, my compositions are not particularly biased towards moving pictures. A movie trailer music contractor never sends a moving image; consequently, I have to compose the music without a video. The trailer editor edits the moving picture to my music.

My compositions do not usually extend beyond three minutes, because a commercial or trailer movie usually has a duration of one minute to three minutes. Sometimes, contractors require longer musical pieces in order to have the flexibility to edit within the music. Moreover, the form of my compositions is often built from a sequence of clear-cut segments. This segmented form usually consists of four segments: (1) Introduction (A); (2) Exposition (B); (3) Development (C); (4) Climax (C').

The first segment (A) is the introduction of the composition. This segment opens the composition, and it establishes the general mood of the work. The musical material is usually very atmospheric and sound design-based. The material contains very subtle hints and motifs of what is to come. The sound intensity starts very soft and increases slightly throughout the introduction. The dynamic intensification is mostly done by adding more atmospheric or sound design-based elements to the first segment. The second segment (B) is the exposition of the composition, and it presents the primary musical material of the music. This segment establishes melodic, harmonic, or rhythmic material related to the main body of the composition. Orchestral instruments such as strings, brass, or woodwinds, synthesizers, or electric guitars perform the musical material. However, in this segment, the musical material is usually not played by more

than one or two different instrument groups. For example, I usually prefer the string section to perform the initial melodic material, and then the brass section strengthens (variations of) the material in the third segment (C). Sometimes, the exposition starts with an ostinato, performed by synthesisers or electric guitars, which is further worked out in the same segment. The ostinato is used to increase the dramatic tension of the composition. The atmospheric and sound design-based elements that were introduced in the first segment (A) of the composition are still present in the background. They accompany the primary musical material. The sound intensity usually stays soft, and it does not increase significantly.

The third segment (C) develops the primary musical material. This segment is also called the 'Riser', and it prepares the composition for the final part. The primary musical material, that was established in the second segment, is further worked out. Different compositional techniques are used to create a 'rising' effect in this segment. For example, different instruments emphasise the melodic, harmonic, or rhythmic material through doubling or variations. The sound intensity in this segment increases significantly from soft to very loud. This is done by introducing multiple different instruments, or increasing the dynamics, or both. Moreover, I usually introduce in this section percussive instruments such as taikos, timpanis, or cymbals. They aim to create dramatic tension by highlighting the previously introduced rhythmic material, ostinatos, or both.

The fourth and final segment (C') recapitulates the primary musical material from the third segment. However, the final segment presents musical material as intensely as possible. This part is also called the 'Climax', and it is the most critical part of the composition. This climax aims to hold the dynamic intensity throughout the end. The musical material focuses on two, three, or four bars, and they are repeated

consistently. All previously introduced orchestral instruments, synthesisers, electric guitars, and sound-design-based elements come together to make the composition feel big and wide. After the fourth segment, the music stops usually very abruptly, and it does not include an ‘outro’ or a fade out.

My compositional process has the following four steps: (1) Organising the structure of the compositional form based on the contractor’s briefing; (2) Composing harmonic and melodic material; (3) Humanising the sound of virtual instruments (VIs); (4) Incorporating sound design-based material and producing and mixing the musical sounds. I begin the composition by structuring the form. Existing reference music or written briefings affect the musical structure. Second, I compose the harmonic and melodic material. For orchestra-focused works, I usually start with a piano, and the piano plays both the harmonic and melodic material. After finishing the harmony and melody, I orchestrate the piano performance, and I link the harmonic and melodic parts to individual orchestral instruments. The musical instruments include strings, brass, woodwinds, pianos, electric guitars, and percussion. Third, I humanise the sound of the VIs. These are existing VIs from various sample libraries. Fourth, I incorporate sound design-based material, and I produce and mix these with the musical sounds. For this approach, I use existing synthesizers and audio processing plugins that simulate reverbs, compressors, or equalisers. I do not use audio hardware units, but I produce everything within the computer. After finishing the compositional process, I export the composition as a ‘WAV’ file. The contractor then receives a download link including the digital audio file.

Time pressures on the compositional and productional process are considerable, as contractors expect a finished version of the music in one, two, or three days. A typical timeframe for one composition starts on Friday, and it ends on Sunday.

Sometimes, the contractor's briefing and the submission deadline is on the same day. Consequently, I have to work as efficiently as possible without losing production quality and musical originality. However, there is no time or budget to hire musical instrumentalists or other participants; I, therefore, have to impersonate a composer, producer, performer, DMI designer, and sound engineer at the same time. Thanks to my years of experience honing these skills and incorporating the above-mentioned roles, I am exceptionally well-placed to undertake this practice-based research.

2. The aim and objectives of the research

This research project aims to produce five compositions, composed with the aid of my newly designed DMI called the ‘Flexible Phrase System’ (FPS). The FPS includes a virtual instrument (FPS-VI) and a user interface controller (FPS-UIC).

Based on the aim of this research, three research questions and three objectives are defined. The three research questions are:

1. What is the impact of the FPS upon the compositions, the compositional process, and the development of musical ideas?
2. What is the effect of the compositional issues upon the design of the FPS?
3. What is the impact of the iterative design process (of both producing compositions and developing the FPS) upon me as the music technologist?

Based on these three research questions, this research project addresses the following three objectives:

The first objective includes the production of five compositions from different western popular genres. I am the composer of all compositions, and they all employ the use of the FPS-VI, the use of the FPS-UIC, or both (Table 1)¹⁷.

¹⁷ The rationale for these different combinations of the FPS-VI and the FPS-UIC are discussed in chapter 5.2.1 – 5.2.5.

Composition	FPS-VI	FPS-UIC
The Alpha	Used	Not used
The Beta	Used	Used
The Gamma	Not used	Used
The Delta	Used	Used
The Omega	Not used	Used

Table 1. The musical portfolio and the used FPS.

The design of the FPS accomplishes the second objective. The FPS has two design parts: one describes the design process of the FPS-VI, and the other describes the design process of the FPS-UIC, which controls the FPS-VI and other VIs via MIDI-mapping strategies. I am the designer and programmer of both the FPS-VI and FPS-UIC. Jur Bruinink, a student of the Utrecht School of the Arts (HKU), assisted me with programming the FPS-VI.

Objective three elaborates the music and human-computer interaction (M&HCI). It describes the global relationship between composing music and designing software: the FPS impacts upon the music and the compositional issues affect the design of the FPS. It also describes the effect of the iterative design process of both producing compositions and developing the FPS upon me as the music technologist. A description of both the compositional process and the instrument design process forms a significant part of this study. The results of the first, second, and third objectives address the research questions of this thesis.

3. Literature review

The context of this study comprises an iterative design process for producing five popular compositions with the aid of a newly designed digital musical instrument (DMI) called ‘Flexible Phrase System’ (FPS). The five original works are the result of the music and human-computer interaction (M&HCI) between the composer and the digital musical software environment, including the FPS. The literature review identifies the context of this study in more detail, and it contains the following three sections:

Section 3.1 briefly discusses the development of music technology within the field of popular music used in movie trailers and commercials. It discusses the presentation of compositions, the development of the compositional process, and the evolution of the composer’s identity from a composer to a music technologist.

Section 3.2 reviews the context of M&HCI, and it summarises existing knowledge concerning the impact of music technology upon compositional contents, compositional processes, and the development of musical ideas. It also summarises understandings concerning the effects of compositional issues upon the development of music technology and the impact of an iterative design process upon the context of the music technologist.

Section 3.3 discusses the development of DMIs and describes the principles of the FPS. It defines the requirements for the FPS, and it discusses the chosen ‘virtualised instrument’. This section also elaborates on the potential impact of the FPS requirements upon composing music.

The literature review and practical experiences form the framework for the knowledge collected in this thesis. The literature review outlines the development of music technology (3.1), the context of M&HCI (3.2), and the principles of the FPS

(3.3). The framework is needed to understand the practical part of this study, which encompasses the production of five compositions and the design of the FPS within an iterative design process. Section four (4. Methodology) and section five (5. Composition and software design) elaborate on the practical part of this study.

3.1 The development of music technology

The development of music technology¹⁸ has significantly changed the presentation¹⁹ of popular music, the development of the compositional process, and the evolution of the composer's identity from a composer to a technologically informed composer overtime. Nowadays, the presentation popular music used in movie trailers and commercials is in most cases narrowed to a digital audio file, saved on a computer, mobile device, or an external server (cloud) on the internet. Compositions can reach a broad audience because digital audio files can be consumed cheaply and can be shared easily around the world (Albright 2015).

The development of music technology has not only changed the presentation of popular music for commercials and movie trailers; technology as a compositional tool

¹⁸ In this study, music technology is a broad field including any electronic, computer-based, networked, and peripheral equipment (e.g., cables or hardware accessories) that allow performers, composers, artists, designers, and musicians to express new ideas and to create original compositions (Manzo 2016).

¹⁹ The presentation of compositions describes the way in which the composer expresses his or her music. The composer can, for example, express music in music notation, music performance, or in a recorded performance. Therefore, the music can be presented in different types of media.

and musical instrument in its own right has significantly changed the compositional process²⁰ during the last few decades. The emergence of digitalisation has introduced the compositional process to the digital music environment. Nowadays, the compositional process co-exists within both analogue and digital music environment. In this digital environment, the composer uses technology, including digital musical instruments (DMIs), that consist of a virtual instrument (VI) and a user interface controller (UIC). Both units work within digital audio workstations (DAWs) and are capable of generating compositions. Consequently, scores for TV series and commercials are nowadays almost exclusively created with the help of DAWs and DMIs (Alberge 2013). VIs simulate the sound of acoustic instruments in such authentic ways that composers can produce their final composition without necessarily having to book instrumentalists, expensive ensembles, and recording studios. Carl Davis²¹ and Christopher Gunning²² argue that production companies often prefer digital recreations of acoustic instruments to save money. “Davis is concerned that the sheer proliferation

²⁰ The compositional process focuses on the acts of listening to, thinking about, and writing music. It can be “thought as a feedback loop, in which a human being iteratively modifies and refines some store of musical information, which we call the ‘composition’. Over the course of this process – or many different such processes – the composition gradually takes shape (Evanstein 2019, p. 13).

²¹ Carl Davis (1936) is a leading British film composer known for scores such as *The World at War* (1973-1974) documentary series.

²² Christopher Gunning (1944) is a leading British film composer of concert works as well as music for films and television.

of television channels is taking its toll on budgets for musicianship. Synthesised performances were infinitely cheaper than hiring an orchestra [..]" (Alberge 2013).

Furthermore, working with synthesised instruments enables the composer to create audible revisions of musical compositions quickly and flexibly. The contractor can give detailed feedback on the musical end result, and the composer can then change different segments, musical instruments, or even melodic, harmonic or rhythmic material instantaneously. However, the disadvantage of being flexible is that the compositional process can end up in an almost endless process of musical revisions. In 2017, I worked on the music for a major Hollywood movie trailer. The contractor knew that I was working with synthesised instruments; therefore, the contractor knew that it was technically possible to create different versions of the same music within a few hours. At the end of the collaboration, I created 12 different versions: some of the versions were completely different productions but other versions only included minute musical changes. For example, some of these minute changes focused on shifting single pitch values within an orchestral arrangement, or on changing the dynamics of individual instruments from piano to mezzo-piano. These changes would have been impossible to process on realistic recordings and they would force the composer (and the contractor) to re-hire, -book, and -record the performers for each musical revision over and over again. However, these changes became possible because the music was composed exclusively with synthesised instruments. It also became apparent, that contractors prefer to have access to different versions of the composition in a very short time. From this follows that contractors require composers to produce popular music for movie trailer and commercials almost exclusively with VIs to save budget and to ensure compositional and productional flexibility. As a consequence, a composer is forced to be a composer, producer, performer, DMI designer, and engineer at the same time.

The abovementioned combined professions have dramatically changed the work of a composer during the last decade. In the 21st century, the professional activities of a composer have extended beyond just writing music. When writing for television or moving picture, contractors often expect that a composer can compose, record, mix, and master music (DeRosa and Pejrolo 2011); thus, the identity of a modern composer has shifted towards a music technologist (Walzer 2017).

On the other hand, a music technologist can also be someone who creates new musical instruments, records, or performance music. According to Zattra and Donin (2016), this type of technologist is a so-called computer music designer (CMD). However, Zattra and Donin define the CMD as an assistant/collaborator of the composer and producer rather than the combined profession of DMI designer and composer. In the context of this study, a composer is a music technologist producing both music and developing music software.

3.2 Music and human-computer interaction (M&HCI)

Marshall McLuhan's contemporary John M. Culkin (1967) once said: "We shape our tools and thereafter they shape us." This quote reflects the essence of the human-computer interaction (HCI), including the two-way relationship between humans and computers. HCI is the study of the interaction between humans and computers, and it studies how technologies, that allow humans to interact with computers in novel ways, are designed. It aims to improve the usability of computer interfaces (Grudin 1992). A subfield of HCI is music and human-computer interaction (M&HCI); in some of the literature, it is also called human music interaction (HMI). M&HCI describes the interaction between composers and music technologies that facilitates the creation of music. It investigates the notion that music is an activity that operates at the point of

connection with technologies (Dubber 2015). Both music and technology are designed based on constraints, and constraints “map out a territory of structural possibilities which can then be explored, and perhaps transformed to give another one” (Boden 1990, p. 95). In the context of M&HCI, “constraints are seen as compositional rules that the computer (or the human) must follow” (Magnusson 2010, p. 63). Existing literature proposes different models of constraints. Norman (1999), for example, describes physical constraints²³, logical constraints²⁴, and cultural constraints²⁵, whereas Pearce and Wiggins (2002) describe stylistic constraints²⁶, internal constraints²⁷, and external constraints²⁸. These examples of constraints define the constraints of composers and music technologists in their work.

In the context of M&HCI, constraints are significant to the development of both music and technology, and the development of music technologies significantly impacts

²³ Physical constraints describe the performance possibilities within the given physical environment (Norman 1999).

²⁴ Logical constraints outline how human logic inductively informs us about the environment (Norman 1999).

²⁵ Cultural constraints are shared by cultural groups that form specific conventions (Norman 1999).

²⁶ Stylistic constraints define the limitations of a musical genre or style in which the composer is producing music (Pearce and Wiggins 2002).

²⁷ Internal constraints describe logical possibilities of how a composition can develop according to the given set of rules (Pearce and Wiggins 2002).

²⁸ External constraints include the need that the composition is physically possible to perform (Pearce and Wiggins 2002). External constraints are comparable to Norman’s (1999) concept of physical constraints.

upon compositions, the compositional process, and the development of musical ideas. More than 100 years before the invention of computers, DAWs and VIs, Ludwig van Beethoven composed his Piano Sonata No. 29 in Bb major, Op. 106, also known as the Hammerklaviersonate, composed between 1817 and 1818. The invention of the six-octave range piano, the Broadwood fortepiano by John Broadwood & Sons (Swafford 2003), had a dramatic influence on van Beethoven and consequently on his music. The instrument provided Beethoven with a considerable range of volume, a greater range of pitch and a range of articulation, from a flowing legato to incisive staccatos (Swafford 2003).

Moreover, the invention of the phonograph in 1887 marked the introduction of the music recording industry (Katz 1999) and affected the length of compositions. Symphonies from the 19th and 20th centuries could stretch to an hour; however, the shellac discs could hold only two to three minutes of audio. From this followed that composers started to edit their music down to the size of a disc. A well-known example is Stravinsky's *Serenade in A for Solo Piano* from 1925. Each movement was composed to fit a three-minute side of a disc; two discs, four movements (White 1979). Moreover, Katz (2010) argued that the three-minute pop song is an invention of the phonograph. The phonograph also impacted upon the compositional process of recorded jazz improvisation. At the end of the 19th century, the composer Edgar Stillman Kelley argued that "the invention [of the phonograph] might be so applied as to enable composers to preserve [and improve] their improvisations" (Weeks 2016). However, the limited capacity of shellac discs at the beginning of the 20th century narrowed solo improvisations often to merely eight bars in which to make a musical impression. In contrast, improvisational music performed live could last for "five or six minutes" (Katz 2010, p. 83). From this followed that "the limited playing [...] discouraged [recorded]

improvisations” (Katz 2010, p. 84). As a consequence, early recorded jazz music differed from live performances, and the musicians did not improvise as much as in live circumstances.

In the 1930s, the invention and popularisation of magnetic tape technologies revolutionised “the idea of borrowing sound from recorded sources” (Wallmark 2007, p. 29), and it significantly shaped the process of composing music. The technological (and partly newly introduced) functions such as stop, play, forward, backwards, cut, and paste affected the process of composing. These techniques were indispensable for the presentation of the movement called *musique concrète* which is “constituted from pre-existing elements taken from whatever sound material, be it noise or conventional music, and then composed by working directly with the material” (Pierre Schaeffer in Taylor 2001, p. 45). Contemporaries such as Pierre Schaeffer, Pierre Henry, and (to a lesser extent) Karlheinz Stockhausen and John Cage were able to cut the tape and recombine recordings in novel ways. Decades later, the development of digital environments adapted the functions of play, stop, forwards, backwards, cut, and paste. Most composers still use these functions within their daily work.

The digitalisation of the music industry including the development of digital audio workstations (DAWs) and virtual instruments (VIs) that started in the 1980s has had a significant effect on composing music (Tzanetakis 2016). The visual organisation of DAWs guides composers to create compositions in the style of popular music. Software companies that program music applications within western industrial society powerfully mirror the preoccupations of popular music (Emmerson 2000). For instance, arranging within a DAW affords the arrangement of a composition in the format of popular music. Bold strokes highlight the start-point and end-point of each bar; therefore, the composer localises the first beat of each bar very easily and understands

the bars temporal lengths at a glance. Evanstein noted that composers “produce ‘blocky’ music that is overly influenced by the presence of barlines” (2019, p. 26).

Moreover, the DAW guides composers to set a bar to an evenly-subdivided time grid. The user adjusts the grid to different subdivisions (e.g., $\frac{1}{4}$ note, $\frac{1}{8}$ note, or $\frac{1}{16}$ note). Furthermore, the organisation of pitch deviations on a semitone scale guides a composer to create popular music within western scales. Composers create pitch deviations of their music within the environment of a piano roll that links to the semitone scale of a piano and, as a result, they tend to stay in the semitone scale provided by the DAW. Moreover, the DAW allows compositions to be created in temporally consistent music in 4/4-time: “For instance, the default setting for all DAWs when a new file is created is a 4/4-time signature at 120 beats per minute (bpm). The producer can easily change these parameters; yet when creating a new composition, the most common time signature (and depending on the style, the most common bpm) in popular music is emphasized” (Macchiusi 2017, p. 32-33).

The visually organised DAW also impacts upon the process of composing. Both audible and visual feedback guides the compositional process. The visual organisation of sound has revolutionised the creation of music through the Graphic User Interface (GUI) of the DAW. Most composers use a computer mouse and computer keyboard to control the DAW’s GUI, and these controllers transform general computer skills, including dragging, dropping, copying, and pasting into musical tools in the process of composing music (Macchiusi 2017). Moreover, writing music with the aid of the DAW introduces digital non-linear editing, enabling spontaneous rapid rearrangements and allows for the redoing or undoing of musical actions (Duignan 2008).

Composers using DAWs, VIs, and MIDI keyboards often find themselves both composing and producing using instrumental sound synthesis approaches. They use

computers “as interactive partners to perform operations where the output depends on [the] actual performance” (Vaggione 2001, p. 54). Therefore, the compositional process involves an action/perception feedback loop in which the composer impersonates both the listener and the producer (Vaggione 2001); consequently, “composers are now using production tools, and producers are becoming composers” (Ramshaw 2006, p. 9).

The development of DAWs and VIs also has a significant effect on the development of musical ideas. While composing music using VIs and samples, composers often find themselves writing music based exclusively on the available synthesised sounds and the MIDI keyboard controller or another user interface controller (UIC) performs the sound. It is common for composers to use sounds, loops, and presets as the point of departure and then, use those instruments in the service of melody, harmony, rhythm, and form (DeSantis 2015). Thus, VIs and samples inherently guide the way in which composition takes place: “Unless a composer conceives their music solely in their head and writes all of their ideas down on score paper before embarking on a mock-up, it’s inevitable that the samples will shape the sound of their music” (Schwartz and Asher 2010). Moreover, the productional process, in which the sound or sound creating process becomes the music material, is similar to the process of *musique concrète*.

Another example is the impact of the UIC, such as the MIDI keyboard controller, upon the development of compositional ideas. A MIDI keyboard can control any ‘virtualised instrument’ even those that do not have a physical structure that is similar to a piano. However, the physical construction of a MIDI keyboard controller does not reflect the physical construction and structure of the ‘virtualised instrument’. Therefore, melodic and harmonic material performed on a MIDI keyboard does not match faithfully the actual performance possibilities of the selected VI (Adams 2016).

There are alternative MIDI controllers with different properties and physical shapes, such as electric guitars (Jammy Instruments 2019), drum kits (Roland Corporation 2020), woodwind instruments (inMusic, Inc. 2020), and violins (Cantini Electric Violins 2020). However, many composers (like me) have written and still prefer to write music at a MIDI keyboard (Evanstein 2019) because they MIDI keyboards are easy to interface with existing DAWs. As mentioned in the introduction, I am a trained piano and keyboard player; therefore, I prefer to keep composing popular music with a MIDI keyboard.

The composer can, of course, circumvent most of the DAW's default parameters. Within a DAW, the composer can create different start-points and end-points, micro-rhythms, micro-tunings, tempo changes, and time signature deviations. Moreover, the composer also can use different UICs controllers to control the VIs, disregarding some of the default settings designed by software programmers. To accomplish this, composers have to change these default settings manually, which requires a lot of time and effort. “[Composers] are little aware of the degree to which what they create is shaped by those defaults” (Kitchin and Dodge 2011, p. 123); therefore, many compositions produced with the aid of modern computer interfaces are apparently biased towards by the default settings of DAWs (Macchiusi 2017).

Moreover, recent attention goes to the use of ‘dark patterns’ in the design of user interfaces. The dark pattern theory comes from abusive game design (Wilson and Sicart 2010): “A dark game design pattern is a pattern used intentionally by a game creator to cause negative experiences for players” (Zagal et al. 2013, p. 3). In the context of music technology, the term refers to an instance in which the interface designer intentionally leads the composers towards or away from certain actions, in a way that is detrimental to the user but beneficial to the designer (Evanstein 2019, p. 21).

Compositional issues also affect the development of music technology.

Organologists have scrutinised possible motivations and concerns for the creation of new musical instruments and music technologies, which include economic, social, and political factors alongside creative issues, artistic issues, or both (Bijsterveld and Peters 2010). Music technologists are mostly driven to design new digital musical instruments (DMIs) because they are seeking for a more embodied experience when performing and producing compositions. According to Emerson and Egermann (2018), the lack of musicality and expressiveness within the compositional process using existing UICs and VIs is one of the main issues discovered by music technologists when designing their DMIs. In the case of this study, however, the DMI actively encourages a less embodied experience, and the system automatically inserts expressive-related parameters without relying on the expressive interpretation of the music technologists²⁹.

Another example of the effect of compositional determinants upon music technology was the development of the signal processing environment called ‘Max’³⁰ at the beginning of 1986. ‘Max’ was programmed at the ‘Institut de Recherche et Coordination Acoustique/Musique’ (IRCAM) in Paris by Miller Puckette. The researcher’s initial aim was to control the 4X synthesiser (Winkler 2001). However, at that time, some composers were more interested in “transforming the sound of a live instrument in some way” (Puckette 1991, p. 66). Puckette and his team then changed the

²⁹ See chapter 1. for more information concerning the issue of whether and how ‘humanising’ the sound of VIs adversely affects compositional processes.

³⁰ Max later became the real-time audio processing graphical programming environment called Max/MSP (Maximal Signal Processing).

focus of ‘Max’ from a controlling environment for the ‘4x synthesiser’ to a ‘signal processing’ environment that was capable of synchronising a live player and the computer (Puckette 1991). Due to Max/MSP composers can manipulate signals without dedicated DSP hardware. The literature on this topic provides several other examples of how compositional issues impacted on the development of musical instruments and music technologies.

Furthermore, literature provides insights into how composing music and designing software, within an iterative design process, affect the context of the music technologist. One example is Baalman (2017) who researched the interplay between composition, instrument design, and performance. Baalman argues that her choices in the composition are informed by how she performs, and her compositional needs inform how she designs the instrument. Then, the instrument design again affects how Baalman has to redo the composition and the performance (Baalman 2017, p. 229). Another example is Chadabe (1997), who describes the iterative design process of music composition and organology as *interactive composing*. Drummond (2009) discusses Chadabe’s term and suggests that:

[..] the musical outcome from these interactive composing instruments was a result of the shared control of both the performer and the instruments programming, the interaction between the two creating the final musical response. [...] In interactive music systems the performer can influence, affect and alter the underlying compositional structures, the instrument can take on performer-like qualities, and the evolution of the instrument itself may form the basis of a composition (Drummond 2009, p. 125).

Another example is Trevor Wishart, who composes and designs software concurrently. Wishart usually starts with a basic software design and then improves on

it as he continues to use it. He describes a number of ways of approaching the design of new software instruments:

In some cases, I become aware while composing that a particular [software] instrument could be developed further, or that it has limitations that are not necessary, or that I am very often using particular processes in combination, or in slow succession. [...]. In other cases, I speculate that a certain process might be interesting and just go for it and try it out. I might even develop an initial instrument into a family of related processes (like the "distort" family of waveset distortion instruments) (Vassilandonakis 2009, p. 20).

Wishart argues, that he got into writing software because he “simply couldn’t afford the endless train of updated black boxes that university and commercial studios could buy out of their equipment budgets” (Vassilandonakis 2009, p. 19). Wishart has written copious essays and books about music composition and instrument design.

3.3 The principles of the Flexible Phrase System (FPS)

One of the most versatile musical tools of the last decades is the digital audio workstation (DAW) (Bell 2015; Clauhs 2020; Ashbourn 2021) including virtual instruments (VIs) and user interface controllers (UICs). VIs use synthesis methods to emulate the sound of musical instruments³¹. UICs are performance controllers, hardware interfaces, or gesture devices that control both the DAW and different VIs. The composer uses UICs, for example, a computer keyboard, computer mouse, MIDI keyboard, or touchscreen-based tablet (among many other types of controllers) to

³¹ See appendix 8.2.1 for more information concerning different types of digital synthesis techniques.

interact with computers³². The computer then responds with visual information, audio information, or both, based on the interactions of the composer. Some VIs and UICs together form a digital musical instrument (DMI) (Miranda and Wanderley 2006). With traditional acoustic instruments, there is no distinction between the instrument controller and the sound of the instrument itself. However, with DMIs that are solely computer-based, both the VI and the UIC are independent units only related to each other by MIDI-mapping strategies; consequently, the UIC can control any timbre, or it can produce any sound regardless of the form or audio material³³. Additionally, UICs can control most of the VIs.

The newly designed FPS is a DMI for both digital music production and composition for popular music for commercials and movie trailers. The FPS-VI simulates the sound of an electric guitar³⁴ with the help of the principles of the corpus-based concatenative synthesis technique (CBCS) developed for this purpose. CBCS uses sampling as the synthesis method. Cook (2002) regards sampling as the simplest and most popular form of audio synthesis method. This synthesis method includes

³² See appendix 8.2.3 for more information concerning different UIC environments.

³³ There are also DMIs that blur the boundaries between the sound generator and the physical controller. One example is the halldorophone (Úlfarsson 2018). “It is an electroacoustic string instrument loosely based on the cello, which makes use of feedback for sound generation, enabling individual gain control for each string and a sympathetic set of, two to four, strings below the main strings” (Úlfarsson 2019, p. 1). Inspired by the halldrophone, Eldridge and Kiefer designed the so-called Feedback Cello. “The signals from pickups under each string are sent to a speaker built into the back of the instrument, and a vibrational transducer fixed on the front (Eldridge & Kiefer 2017, p.25).

³⁴ See 3.3.2 for more information concerning the synthesised musical instrument.

processing samples and transforming them from an analogue to a digital representation. The digital representation comprises a series of numbers. These numbers describe “the value of the audio signal at specific points in time and these are called samples” (Russ 2004, p. 48). This method dominates the commercial music industry and is very widespread in both commercial and non-commercial hardware and software (Percival 2013). Sampling is the most popular synthesis method that is used to reconstruct the sound of traditional musical instruments due to the availability of technology for high-fidelity sound synthesis (Schwarz 2006). The FPS-VI contains an audio database of recorded electric guitar performance gestures, and the audio database contains segmented audio samples³⁵.

The software and audio database are programmed within the Reaktor 6 digital signal processing (DSP) environment. Reaktor 6 is a modular dataflow visual programming language (DFVPL) including a GUI environment. Reaktor 6 can be run alone or as a virtual instrument ensemble. A Reaktor instrument ensemble is a software format in which programmers use sound generators and effects to design the VIs. Furthermore, the Reaktor 6 environment is available as virtual studio technology (VST) plug-in, and it is easy to interface with existing DAWs. The interface ensures that the composer can integrate the FPS within the existing compositional and productional process. The GUI is programmed and linked automatically to the Reaktor 6 processing environment, and the composer controls the GUI with a computer mouse, computer keyboard, MIDI keyboard, and the customised external user interface controller (FPS-

³⁵ See appendix 8.2.1 for more details concerning different synthesis methods.

UIC). Within the DAW engine, the Reaktor 6 instrument ensemble receives the target as a MIDI file. The selector selects sound units in real-time that fit best to the given MIDI file. The audio units are then concatenated and synthesised. The synthesised phrase is play-backed in real-time within the DAW engine³⁶.

The FPS-UIC is programmed within the Lemur environment, including the programming language C. Lemur is a suitable environment to develop the FPS-UIC because it is well-suited for use with Reaktor 6. Additionally, Lemur is available as an application for iOS and Android. The Lemur UI includes a graphic user interface environment and is easy to interface via MIDI with both the DAW and the FPS-VI³⁷. The design process of the DMI consists of the following steps: defining the requirements for the FPS (see 3.3.1); defining the synthesised instrument (see 3.3.2). Appendices 8.2.1, 8.2.2, and 8.2.3 elaborate on the technical details of the selected synthesis method (CBCS), the selected processing environment (Reaktor 6), and the selected UIC (Lemur). Moreover, in section 3.3.3, the potential impact of the FPS requirements and the research restrictions upon composing music is described.

3.3.1 Defining the requirements for the FPS

The FPS aims to offer an interconnected process of composing and producing music, and it tunes expression-related parameters and harmonic voicings automatically. Therefore, the design process of the FPS has to fulfil the following requirements.

³⁶ See appendix 8.2.2 for more details concerning different processing environments.

³⁷ See appendix 8.2.3 for more information on interface controller environments.

The design process of the FPS has to fulfil usability requirements. While composing, revisable software environments ensure the design of the ‘Flexible Phrase System Virtual Instrument’ (FPS-VI) and ‘Flexible Phrase System User Interface Controller’ (FPS-UIC). A revisable software environment enables small or significant changes to the FPS, and it allows software revisions that are carried out within an iterative design process. A dataflow visual programming language (DFVPL) is a revisable software environment, and this is the type of language used to build the FPS. A DFVPL allows programmers to develop software by connecting modules with arcs or wires (Johnston et al. 2004). In the context of this study, the music technologist is a novice programmer and DFVPLs enables novice programmers “to focus on building programs rather than learning the syntax” (Aglíć Čuvić 2018, p. 35)

In contrast to the DFVPL programming paradigm, a modular Object-Orientated Processing (OOP) language environment is often used to design (music) software. The OOP programming paradigm enables modularity, extensibility, and reusability. New attributes and behaviours extend the objects, and the programming software reuses these objects across the application. The reusability of objects allows for faster software design (Shoop 2013). In contrast to DFVPLs, OOP languages “tend to result in higher-quality software” (Shoop 2013, p. 1). However, OOP requires a steep learning curve, and the programmer needs proper programming and design skill to create software. Moreover, the FPS uses real-time audio playback. The real-time audio playback functionality allows composers to manipulate musical parameters of the VI in real-time. For example, a composer changes the pitch value of a musical note with the aid of a UIC; then, the synthesis engine of the VI changes in real-time the pitch of the played note. The FPS produces the sound of a musical instrument, i.e. the sound of an electric guitar. The authenticity of the sound is indispensable for the usability of the VI and the

performer of the electric guitar confirmed the authenticity after listening to the compositions.

The design process of the FPS also has to fulfil compatibility requirements. The DFVPL is easy to interface within a DAW. The FPS has to work within a DAW because it is the primary digital environment used to produce the five compositions. Moreover, the FPS is compatible with high-level consumer computers that are accessible to the public. It is an essential requirement that the user of the FPS is not forced to hire super-computers³⁸ or forced to have access to expensive computer servers. Moreover, the sound of the FPS-VI is compatible with different external reverberations. The FPS-VI excludes synthesising the reverberation of the room in which the instrumentalist performs the instrument. Synthesising the reverberation would force this research to focus on technical and acoustical challenges rather than on compositional.

The design process of the FPS also has to fulfil music expressive requirements. The FPS-VI simulates the sound of an electric guitar. It is not the intention to use the FPS to re-create existing instrument performances; instead, it is built to create new compositions. Much research into audio synthesis investigates the digital recreation of the timbres of musical instruments. Other research investigates expression-related parameters; deviations of pitch, dynamics, and rhythm, that an instrumentalist always creates while performing music (Maestre et al. 2009). The simulation of automatically tuned expression-related parameters is an essential requirement of the FPS and a

³⁸ A supercomputer is a computer that includes high-level performances.

necessary restriction of this research. Common VIs use MIDI information to enable expressiveness within digital compositions; however, they do not add the musical expression of the instrumentalist's performances automatically to the composition. Therefore, the requirements of the FPS include the facilitation of the instrumentalist's interpretation. This requirement is met by the synthesis method which embeds the interpretation. Digital music notation and MIDI files are not responsible for the musical expression of the instrumentalist's performance. The expressiveness lies within the stochastic material of the recorded musical phrases combined with single note recordings³⁹. Moreover, the FPS-UIC is not responsible for the realisation of the expressiveness of the instrumentalist's performance. It merely activates the synthesis engine of the VI, and it is responsible for the compositional and productional structuring of the composition.

The restrictions of this research also frame the design process of the FPS. The practical research activities of designing a new digital musical instrument (DMI) must be realisable within the duration of this research. The duration of this research gives the framework for the complexity and the functionality of the FPS. Also, the resources available for this research limit the design of the FPS. The resources include a researcher who designs and programs both the FPS-VI and FPS-UIC and Jur Bruinink, a student at the Utrecht School of the Arts (HKU), who supports programming the FPS-VI within the Reaktor 6 environment.

³⁹ A more detailed discussion of the expressiveness of performances can be found in appendix 8.2.

3.3.2 Defining the synthesised instrument

Within the restrictions of this study and the requirements for the FPS, the electric guitar seemed an appropriate musical instrument to synthesise. The electric guitar is an excitation-instantaneous instrument, and it is a suitable instrument to digitalise because of its expressive parameters. In comparison to excitation-continuous musical instruments, an instrumentalist excites excitation-instantaneous musical instruments such as percussions, mallets, or pianos mostly by instantaneous actions in the shape of impulsive hits or plucks (Maestre Gómez 2009). Moreover, these excitation-instantaneous instruments have a lesser degree of expressiveness and a smaller amount of musical deviation variations after excitation. For example, a piano cannot, in comparison to a violin, change pitch, dynamics, or articulation in a single note. However, in contrast to a piano or percussion, the electric guitar player has total control of the plucking event (plucking angle, force, and width⁴⁰). The guitar player can, for example, bend the string to alter the pitch (Henri and Välimäki 2011), or the instrumentalist can mute the performed tone with the ball of the thumb.

The electric guitar is also an appropriate instrument to synthesise because of the possibility to record the instrument using the direct input (DI) technique. The DI technique connects an unbalanced output signal to a balanced input (Huff 2018) without using microphones to capture the output. A DI unit connected to the computer converts an unbalanced signal to a balanced signal. Furthermore, the DI technique captures the

⁴⁰ It is worth mentioning that volume settings and pickup settings of the electric guitar also impact the output of the plucking event. An aggressive pluck at a sharp angle from a wide position may produce only a weak sound if the volume settings and pickup settings are not set appropriately.

direct signal of the electric guitar, but it does not take reverberation into account. Recording the reverberation⁴¹ of the electric guitar requires the connection of the instrument to the amplifier. Then, practical microphone recording techniques capture the audio signal. Alternatively, the producer adds artificial reverberation to the recorded DI signal. Artificial reverberation is an electronically produced echo effect, and it allows the producer to add a sense of any acoustic space to any DI signal.

In contrast to microphone techniques, recording an electric guitar with the DI technique simplifies the recording process; moreover, the compositional process becomes more flexible by adding reverberation to the audio signal digitally. The practical advantage of DI recordings and artificial reverberation is especially useful for VIs. It allows the composer to integrate the VI into different acoustic spaces and instrumental combinations. Furthermore, the composer can also wait until mixdown before deciding which acoustic spaces best suit individual parts. A mixdown describes the process of merging multiple pre-recorded tracks into a combined output master.

Initially, this research began with an acoustic violin and took Saint Saëns' *Dance Macabre* as a source of inspiration for designing the synthesis model of the violin. The violin, among other excitation-continuous musical instruments such as brass or woodwinds, is a highly challenging acoustic instrument to synthesise because of its high degree of expressiveness and continuous modulation of physical actions. Moreover, the recording process of a violin captures both the sound of the instrument itself and the

⁴¹ The reverberation defines what one perceives as timbre, volume, and sound colouration, and it largely determines one's perceptions of the location of the instrument, whether it is near or far (Doyle 2005, p. 38).

reverberation. Therefore, the recording process includes both close⁴² microphone placement and distant⁴³ microphone placement. The combination of close microphone placement and distant microphone placement is used to create a balance between the direct signal of the instrument and its acoustic environment.

After experimenting with different sets of recorded violin databases⁴⁴, the reverberation became a significant problem for synthesising the acoustic violin. The issue of reverberation would bias this research towards technical challenges rather than composition. It is not the focus of this thesis to obtain quality recordings using only natural reverberation. It requires a great deal of experience, as well as extreme care in selecting and setting up recording environments. The circumstances needed to record a violin do not satisfy the requirements of the FPS; consequently, they do not fit within the restrictions of this research. Therefore, this study switched to the electric guitar as the synthesised instrument. The electric guitar excludes the recording of the acoustic

⁴² Close microphone placement locates “the mic at a distance of about one inch to three feet away from the sound source” (Huber and Williams 1998, p. 24). This technique excludes the recording of the acoustic environment.

⁴³ Distant microphone placement “refers to the placement of one or more microphones about three or more feet from a sound source” (Huber and Williams 1998, p. 27). This technique is used to record the acoustic environment. The recorded signal of the distant microphone placement is indispensable for the sound of an acoustic instrument because “size and room reflections are always part of the sound of an acoustic instrument” (Huber and Williams 1998).

⁴⁴ An acoustic violin was recorded within an anechoic chamber and a recording studio at the Utrecht School of the Arts (HKU).

environment, and it allows to focus on the integration of the FPS within the process of producing compositions.

3.3.3 Potential impact of the FPS requirements upon composing music

Previous research⁴⁵ on music and human-computer interaction (H&MCI) elaborates on the effects of music technology, in particular digital audio workstations (DAWs) and virtual instruments (VIs), upon compositions, the compositional process, and the development of musical ideas. The requirements for the ‘Flexible Phrase System’ (FPS) and restrictions of this research potentially creates a bias towards one type/style of composition or another.

The FPS works within existing DAWs, and the limitations of DAWs amplify the restrictions of the FPS. For example, the FPS works with musical phrases and a limited set of time signatures. The musical phrases determine the development of musical material, and the limited set of time signatures frames the use of the metrical grid. Also, the corpus-based concatenative sound synthesis (CBCS) method limits the use of different tempo settings. The FPS database uses sample synthesis, and the recorded samples embed the recorded tempo; consequently, the FPS-VI is not able to perform every possible musical tempo.

The requirements and restrictions also bias the process of composing music. The musical phrases of the FPS force the composer to follow a particular compositional process. First, the composer programs a musical phrase within the GUI environment of

⁴⁵ See 3.2 Music and human-computer interaction (M&HCI) for more information.

the FPS-VI and then controls the phrase with MIDI controllers. It is not possible to control a musical phrase with a MIDI controller without firstly programming a phrase within the GUI of the FPS-VI. Moreover, the FPS-VI uses a real-time playback engine, which is essential to the process. The real-time playback engine cycles through the programmed musical phrase, in which each note has a specified note length. The composer then determines the pitch of each note by playing different pitch values on the MIDI controller⁴⁶.

The requirements and restrictions affect the development of musical ideas. A composer who uses the FPS-VI writes music based exclusively on the synthesised sound of the electric guitar. The musical phrases of the FPS frame the development of a musical idea and guide a composer subconsciously to which note and musical deviations to choose. The potential effect of the newly designed FPS-UIC is another example of the potential restrictions upon the development of musical ideas. The controller tunes harmonic voicings automatically and it potentially guides the composer to choose voicings generated by the controller.

⁴⁶ See appendix 8.9.3 for more information.

4. Methodology

4.1 The principles of the iterative design process of this case study

This practice-based research adopts an action research epistemology and a case study methodology. A music technologist, who often combines the professions of both composer and developer of human-computer interfaces, commonly uses iterative design processes (Nielsen 1993) to structure the process of composing music and designing musical instruments. An iterative design process meets the requirements of the process-based case study methodology because it aims to make improvements based on findings from previous iterations. Nielsen also argues that developers should build the usability engineering lifecycle around the concept of iteration. Alternative processes for the production of compositions and the design of human-computer interfaces are the so-called parallel design process and competitive testing. A parallel design process emphasises the parallelisation of different tasks. It is a “systematic approach to the integrated, concurrent design of products and their related processes [...]” (Winner 1988). Competitive testing is comparable to the parallel design approach, except that the original design alternatives are existing artefacts from different people or companies. However, this research focuses solely on the iterative design process approach. Both parallel design process and competitive testing work best in a teamwork setting in which two or more people work on the same task simultaneously.

The iterative design process of this study adopts the fundamental theory of action research by Kurt Lewin (1946). Lewin used action research for the first time as a way to apply various psychological concepts within practice. Hult and Lennung (1980) define action research as follows:

Action research simultaneously assists in practical problem-solving and expands scientific knowledge, as well as enhances the competencies of the respective actors, being performed [...] in an immediate situation using data feedback in a cyclical process aiming at an increased understanding of a given social situation, primarily applicable for the understanding of change processes in social systems and undertaken within a mutually acceptable ethical framework (Hult and Lennung 1980, p. 241).

Lewin describes action research as an iterative design process. The iterative design process includes preparing, taking, and evaluating action and research (Figure 1).

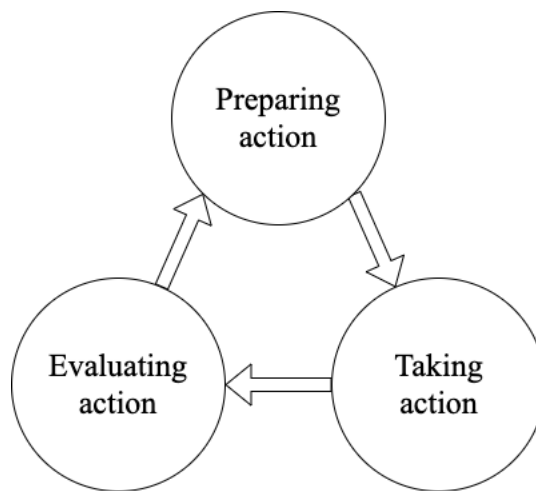


Figure 1. The iterative design process of action research, according to Kurt Lewin.

Since the initial usage, Kurt Lewin's action research has altered, and many general fields of research have applied his methodology. One of these research fields is the study of human-computer interaction (HCI) (Zimmerman 2003), in which action research is used to evaluate the impact of things such as engineering techniques or user experiences (Kock 2014). Moreover, researchers have developed different variations of the iterative design process. Examples are the software development life cycle (SDLC) (Elliott 2004) or the design thinking process (Brown 2008) to name just two. Inspired by Lewin, the iterative design process of this practice-based case study research consists of the following five stages (Figure 2): (1) Define the problem; (2) Design FPS; (3) Compose and observe; (4) Analyse; (5) Evaluate.

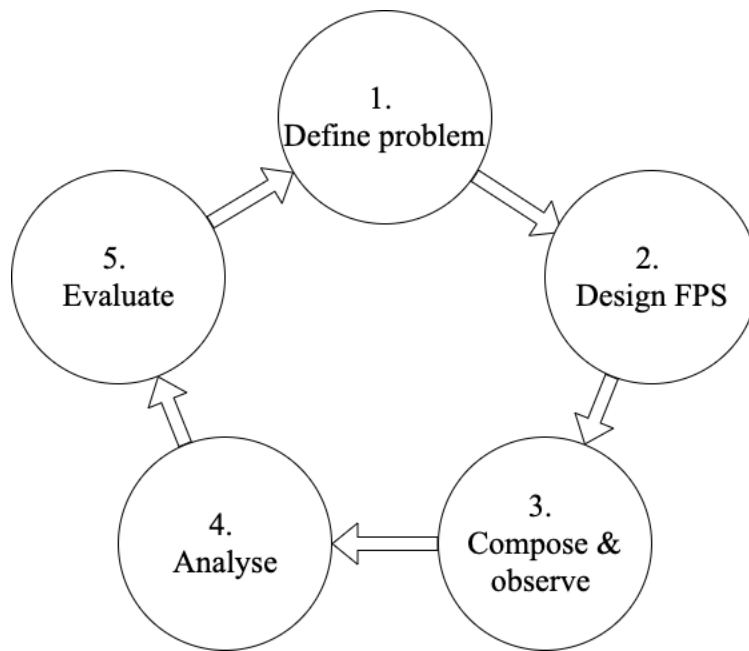


Figure 2. The five stages of the iterative design process of this research.

The first stage of the iterative design process defines the problem. The identification of the problem guides the design choices of both the ‘Flexible Phrase System’ (FPS) and the composition. The first iteration starts with the initially defined problem of this research, which describes the adverse effect of existing digital musical instruments (DMIs) upon compositions, compositional processes, and musical ideas⁴⁷. The following problems of the second, third, fourth, and fifth iterative design process are defined based on the evaluation (see 5. Evaluation) of the previous iterations which are linked to the initial defined problem of this research. For example, the described problem of the second iterative design process is defined based on the evaluation of the first iterative design process, and the described problem of the third iterative design process is identified based on the evaluation of the second iterative design process.

⁴⁷ See 1. Introduction and 1.1 My context of composing original music for more information.

The second stage of the iterative design process constitutes the design of the FPS, and it sets out its functional design specifications, which focus on the actions of new or revised (software) programs. It identifies the requirements and constraints of the FPS that are outlined in the first stage (1. Define the problem) of the iterative design process. The process of designing the FPS focuses on potential revisions of the issues identified, and it focuses on the implementation within the DMI. The FPS I designed works within my existing process of composing music so it can be tested without difficulty. Then, I use the FPS in each following stage (3. Compose and observe) of each iterative design process.

The third stage of the iterative design process describes the collected audio-visual data of the compositional process. The generated audio-visual data is collected through composing music and self-study techniques. This research uses self-study techniques to observe the process of producing compositions with the aid of the FPS. Writing music and observing the musical arrangements together form a single stage because the iterative design process executes both processes concurrently. Section 4.2 elaborates in more detail on the chosen observation and self-study techniques of this research.

The fourth stage of the iterative design process briefly describes the compositions and analyses the audio-visual data. The audio-visual data is analysed based on an inductive analysis approach, consisting of three individual analysis steps: (1) Open coding; (2) Axial coding; (3) Selective Coding⁴⁸. The three steps organise the

⁴⁸ Appendix 8.3. illustrates the open coding, axial coding, and selective coding.

collected data into significant concepts which are related to the research questions of this thesis. Moreover, section 4.3 elaborates in more detail on the description of the compositions and describes the chosen audio-visual analysis methods that are inspired by grounded theory and inductive analysis approaches.

The fifth and final stage of the iterative design process evaluates the composition. The general description of the composition and audio-visual data are both evaluated based on the research questions of this study; in particular, the first and second research question. This stage evaluates the general relationship between composing music and designing software: (1) the impact of the newly designed FPS upon the composition content, the compositional process, and the musical ideas; (2) how compositional issues impact upon the revision of the FPS in the following iterative design process. These descriptions are used to evaluate the impact of the FPS upon the musical content.

Furthermore, this research uses analysed self-study recordings to evaluate the impact of the FPS upon the compositional process and the development of musical ideas. The outcomes of the evaluation describe the compositional issues that occur while composing with the aid of the FPS. Moreover, the fifth stage also outlines possible FPS revisions, and the subsequent first stage (1. define the problem) in the following iteration recapitulates the previous fifth stage of the second, third, fourth, and fifth iterative design process. This research does not recapitulate the fifth stage of the fifth and last iterative design process because it does not include a sixth iterative design process. Appendices 8.4, 8.5, 8.6, 8.7, and 8.8 elaborate in detail the five stages of each iteration separately.

4.2 Data collection through composing and self-study techniques

Action research uses participant observation as essential data collection approach (The Interaction Design Foundation 2014). While composing, this action research applies self-study observation techniques to collect audio-visual data. The collected data includes audio-visual observations of the compositional processes and the compositions; however, as this study aims to produce a portfolio of compositions, this research is not collecting audio-visual data of the process of designing, programming, and revising the ‘Flexible phrase System’ (FPS).

The following three studies are used for the observation techniques; the collecting, logging, and describing of self-study data: (1) autoethnography research studies; (2) reflection-in-action studies; (3) studies of the creative process of music composition (CPMC).

The first, autoethnographic research studies, inspired the observation techniques of this thesis. Autoethnographic research aims to mediate transparency in self-studies. According to Ellis (2010), the process of doing autoethnography is accomplished by retrospectively and selectively writing about epiphanies. These “epiphanies stem from, or are made possible by, being part of a culture, or by possessing a particular cultural identity” (Ellis et al. 2010, p. 2). Ellis also argues that autoethnographers use both their methodological tools as well as research literature to analyse experiences, and they use personal experience to illustrate facets of cultural experience. In doing so, autoethnographers make the characteristics of a culture familiar to insiders and outsiders (Ellis et al. 2010).

The second, reflection-in-action studies, also inspired the observation techniques of this research. Reflection-in-action is a self-study observation technique conceptualised by Donald Schön. It is a technique of researching within a practical

context, linking the “art of practice in uncertainty and uniqueness to the scientist’s art of research” (Schön 2003). Professional designers use research-based theories and techniques consciously; however, professional designers depend on ‘tacit knowledge’ in their daily practice. Polanyi (1966) describes tacit knowledge as knowing-in-action, and he defines this type of knowledge as having the following four properties:

- (1) judgements are made spontaneously;
- (2) the designer is often unaware of learning this knowledge;
- (3) sometimes, once they are aware, they subsequently internalised it;
- (4) they are usually unable to describe the knowledge.

Reflection-in-action studies aim to gain insights into this tacit knowledge.

The third, studies of the creative process of composition (CPMC), also inspired the observation methods of this research. Roels (2014) examines the study of, and reflection on, the main activities and processes involved in the act of composition. He describes methodological observation tools based on real-time data collection-techniques. Real-time data collection techniques, obtained at different moments in the creative process, help the composer (and researcher) to reconstruct a rich picture of the dynamic process of composing. This real-time data includes spoken diaries, audio recordings, and video recordings (Roels 2013). The collection of real-time data supports the identification and later, recollection of old and new elements of the creative process; moreover, it is necessary to avoid a solely retrospective analysis.

Based on the three observation studies, the designed research instrument of this study contains the following five data collection techniques:

- (1) Video recordings of the composer while producing music;
- (2) Audio recordings of the composer while composing music;
- (3) Screen recordings of the computer while producing music;

(4) Audio recordings of the audible output of the DAW;

(5) A short diary, written after producing music for retrospective analysis.

The five different data collection techniques enable a thoroughly designed research instrument for the observation of compositional processes.

The first and second data collection techniques, the recording of video and audio of the composer (researcher) while producing music, includes the use of the verbal protocol. The verbal protocol is a think-aloud method that aims to elicit cognitive and physical process descriptions from the user (Collins 2005), and it aims to gain cognitive insight into complex behaviours (Stanton et al. 2013). The reason for using verbal protocols is to elicit the emotions, thoughts, and musical ideas during the compositional process. The following three questions form the basis of the verbal protocol: (1) What were my thoughts while composing music? (2) What decisions do I take with the help of the FPS? (3) Why do I take these decisions with the help of the FPS?

Besides these three questions, I communicate everything that comes into my mind while composing. The communications give observers insight into the cognitive processes (rather than only the final product), to make thought processes as explicit as possible during the compositional process. Moreover, the audio recordings and video recordings start at the beginning of each compositional process, and they stop at the end of each compositional process. A 4k webcam records the video data, and an external condenser microphone records the verbal protocol, both installed on the music studio desk.

The third and fourth data collection techniques include screen recordings of the computer and audio recordings of the musical output of the DAW while producing music. This third observation technique is a lesser-known method of collecting data in the studies of the creative process of composition, reflection-in-action, and

autoethnography. However, it is crucial to counterbalance the audio and video recordings, including a verbal protocol, with real-time traces of the computational activities. The screen recording displays how I use the FPS within a DAW while composing. The screen recording elicits real-time insights into the effect of the human-computer interface of the FPS upon the compositional content, compositional process, and musical ideas. The software application called ScreenFlow⁴⁹ records both the computer screen activity and audio output of the DAW. It is important that the third and fourth observation techniques do not technically inhibit the process of composing.

The fifth data collection technique is a short diary for retrospective analysis, which complements the four real-time data collection techniques. Reflective writing is a well-known method in self-studies. According to (Moon 2003), “reflective writing is helpful in personalising and deepening the quality of learning.” Diaries may be highly structured or ‘free’ (Moon 2003). Diary studies contextualise the understanding of user behaviours and experiences over time. They are useful for understanding long-term behaviours and are often structured to focus on the understanding of how people complete specific activities or to gather information about user behaviour. According to Flaherty (2016), “diaries [...] give contextual insights about real-time user behaviours and needs, helping define user experience feature requirements.” First, the date and time of the diary entry are defined. Next, the diary objectively describes how the music is composed with the aid of the FPS and how long it took.

⁴⁹ ScreenFlow from Telestream, Inc. is a screencasting and video editing software for the macOS operating system. It can capture the audio and video from the computer, edit the captured video and add highlights or annotation (Telestream Inc. 2019).

4.3 Musical description and audio-visual analysis methods

This section outlines the general description of the compositions and the approach used to analyse the audio-visual data.

The description of the compositions is based on the musical form and the chosen musical instruments. The study of the form contains both the description of the duration and the description of the segments. Types of segments include, for example, the introduction, refrain, or bridge. Capital letters (A, B, C, or D) describe the segments, and a table outlines the form. Moreover, it also includes the description of the instruments used in the composition, and it describes which function each instrument fulfils within the composition. For example, some musical instruments play the foreground material, and others play the background material.

Moreover, the musical description also outlines the metaphors that are part of my musical motivation. A metaphor is used to describe a composition including its musical ideas, and a metaphor also “reifies their features and relations in a particularly pungent and insightful way: it makes sense of them in ways not formerly possible” (Guck 1994, p. 71). Moreover, the compositions are described based on my musical inspiration. Music is initially inspired, and without an initial spark, there is no music (Graf 1947).

The analysis method of the audio-visual data focuses on how the FPS and the compositional content interact. The analysis approach follows the principles of grounded theory (Glaser and Strauss 1967). Grounded theory is a qualitative inductive research analysis approach that implies an investigative process for systematically collecting and analysing relevant data (Charmaz 2006a). According to Pace (2012), the researcher allows the theory to emerge from the collected data. Grounded theory is formulated from data using a constant comparative method of analysis; however, the

analysis method of this research does not compare the evaluations of each iterative design process to each other. The evaluation of the previous iteration guides each following iteration. For example, the evaluation of the second iteration guides the third evaluation; however, a comparison between the evaluation of the first and second iterative design process does not guide the third iteration.

Moreover, the method divides the analysis of the audio-visual observations into three stages: (1) Open coding; (2) Axial coding; (3) Selective coding. Open coding is the first stage of the analysis method, and it involves breaking down the collected observation data into significant concepts. Significant concepts are general descriptions of the compositional process. Examples are the initial preparation of the composition, the use of the FPS, or non-related compositional tasks such as software bugs or distractions. However, these significant concepts depend on the properties of each iterative design process, which means that significant concepts of the first composition ‘The Alpha’ differ from the significant concepts of second composition ‘The Beta’⁵⁰. Next, axial coding is the second stage of the analysis method. It involves reassembling the significant concepts with propositions concerning their relationships with each other⁵¹. Selective coding is the third and last stage of the analysis method. In this stage, only those concepts and relationships related and relevant to the core explanatory concept determine the focus of the analysis. The core explanatory concept outlines the impact of the FPS upon the compositional process.⁵² The collected audio-visual

⁵⁰ See Appendix 8.3.1 for an example of the open coding process.

⁵¹ See Appendix 8.3.2 for an example of the axial coding process.

⁵² See Appendix 8.3.3 for an example of the selective coding process.

observation data is coded in MAXQDA⁵³, and it is subsequently analysed and exported as a datasheet.

Unfortunately, less information is available in the literature on how researchers analyse audio-visual observations from self-study approaches. Collins (2005) scrutinises data analysis methods to analyse compositional processes. These analysis methods include ‘real-time mapping’, ‘thematic mapping’, and ‘structural mapping’ (Collins 2005). However, researchers use Collins’ method as an approach for the analysis of self-studies of composers where the researcher was not one of the composers. To me, Collins’ mapping is too complicated for an autoethnographic study. In this study, I am the researcher who takes on the role of both the composer and self-study researcher; consequently, I have more knowledge of my compositions and composition process than a researcher who is not part of the process of composing music. Thus, I prefer to use a more straightforward analytical method derived from grounded theory.

4.4 Potential outcomes, benefits, and risks of conducting self-study research while composing

This section describes potential outcomes, benefits, and risks of practice-based self-study research while composing and developing a digital musical instrument (DMI)

⁵³ MAXQDA is a software program designed for computer-assisted qualitative and mixed methods data, text, and multimedia analysis in academic, scientific, and business institutions (VERBI Software GmbH 2019).

within an iterative design process. The potential of this practice-based research includes the unique combination of three roles that I perform in this self-study. The three roles are (1) Researcher; (2) Composer; (3) Software developer.

Existing literature little discusses the amalgamation of the three roles in which the researcher is composer and self-study researcher of the compositional process as well as the software developer of a DMI. In 2005, Collins undertook a three-year case study of a composer to track the composer's compositional process in real-time. However, Collins' role was solely that of a researcher and not the composer. Additionally, Collins' study did not include the design of a DMI. Newman (2008) and Roels (2014) both performed a self-study of their compositional processes; however, neither researcher included the design of a newly designed DMI within their self-study, and they did not scrutinise the impact of music technology upon the production of music. Lalama (1998), Sloboda (1986), Sudnow (1993), and Curch (1996) are other examples of early self-study researchers in music. However, they also do not focus on the iterative process of both composing music and designing software.

Another potential outcome of conducting a practice-based self-study is the improvement of the theoretical understandings of my creative practice. Methodologically speaking, self-study research is researcher-friendly. The self-study method "allows researchers easy access to the primary data source from the beginning because the source is the researchers themselves" (Chang 2008, p. 52). Therefore, self-study is the creation of a deeper understanding of the general workings of my creative process. It also concerns learning specific compositional and productional methods of my creative workflow with the aid of the FPS. Moreover, the way in which I create awareness of my working practices (the so-called tacit knowledge) could be streamlined and improved. Roels (2013) explains that he repeatedly noticed how quickly he had

forgotten previous steps of the creative process while he was going through parts of the data of his self-study. I expect that with the help of the self-study data, I can store compositional ideas in my memory for longer.

Another potential benefit of conducting self-study research while composing is that the audio-visual observation techniques do not interfere with the compositional processes. The data collection technique relies significantly on the existing computer hardware and background software, so the risk of interference is minimal. An example from literature is a three-year study by Collins (2005). For his research, Collins purposively selected a professional composer, and the composer's preferred mode of writing music was via a home computer-based system. Therefore, the data collection methods with the aid of a computer were neither artificial nor unnatural to the composer. Based on the literature, this thesis presupposes that audio-visual observation techniques do not interfere with the process of composing.

However, a potential risk of self-study research is to put too much weight on the data collection method. The focus on the collected observations forces the creative process into a tight procedure and can undermine the notion of real-world study. According to Roels (2013), it takes a lot of time and energy to do both self-study research and to produce music. Newman (2008) notices an ironic twist; namely, he had to stop composing and performing in order to complete his self-study research. In my case, I have my way of working and my tacit knowledge base. However, I have never previously made audio-visual recordings of myself while composing music, and I have never recorded verbal accounts while producing content. Therefore, it is reasonable to assume that it could interfere with my usual compositional process and that I might place too much emphasis on the collection of audio-visual observations. I am willing to resolve these tensions by focussing on the compositional process. The audio-visual

observation supports the compositional process. Moreover, this research method also includes an excessive focus on myself, in isolation from others, which is another potential pitfall (Chang 2008). The aim of the self-study observations is to describe my compositional process in as much detail as possible; however, self-study may focus too much on technical details instead of the analysis and interpretation of the music and human-computer interaction (M&HCI). Describing self-study observations also forces the researcher to break the conventions of academic writing and use first person pronouns to report their results.

4.5 On the ethics of sampling

The purpose of the newly designed ‘Flexible Phrase System’ (FPS) is to spur on the compositional process. The FPS, like most sampled instruments, uses recorded sound, which often precludes the use of live instrumentalists. At this point, an issue arises: whether the automation in music production replaces real musicians because virtual instruments (VIs) are less expensive and freely accessible for composers. Some researchers argue that technology (and the FPS) does raise new concerns. They argue that technology adversely affects many occupations. Brynjolfsson⁵⁴ and McAfee⁵⁵ (2011 cited by Rotman 2013) believe that advanced technology destroys jobs faster than it creates them because modern technology reinforces productivity without spurring on concurrent occupation growth. However, in the field of music technology, comparable

⁵⁴ Erik Brynjolfsson is a professor of management science at the MIT Sloan School of Management.

⁵⁵ Andrew McAfee is the associate director of the centre of digital business at the MIT Sloan School of Management.

digital musical instruments (DMIs) already exist, and productions that are supported with sufficient financial resources tend to hire instrumentalists, whereas productions with limited financial resources tend to use DMIs.

However, the design and use of the FPS does raise some additional concerns that go beyond the ethics of sampling generally, considering that the impact of recording sampling and other automated systems of producing music and its impact on professional lives of musicians are not new. It raises some additional concerns because the FPS facilitates musical performance gestures of an instrumentalist. According to Hatten (2004), a musical performance gesture is “biologically and culturally grounded in communicative human movement”. The synthesis engine of the FPS allows composers to play and to manipulate musical performance gestures in a flexible way. In contrast to the FPS, most sample libraries include single note recordings rather than performance gestures; consequently, these sample libraries do not include the fingerprint of the instrumentalist, but they aim to capture the sound of the recorded instrument and to design a user-friendly interface. However, there are also VIs and sample libraries such as the ‘Session Guitarist – Electric Sunburst’ instrument (Native Instruments 2021) that offer recorded loops and phrases, and these recordings incorporate the musical gestures of performers. The disadvantage of recorded loops and phrases is that they are not as playable as the user interface of the FPS because they are ‘locked’ to a fixed recorded set of phrases. For example, the ‘Session Guitarist – Electric Sunburst’ instrument contains 154 fixed rhythmic and melodic phrases. Composing beyond these 154 phrases forces the composer to follow several digital signal processing steps (e.g. exporting the phrase to an audio-file, time-stretching or pitch-shifting notes within the phrase).

Moreover, the FPS raises some additional questions because the output of the FPS is not distinguishable from an original recorded performance played by the instrumentalist. For example, the newly designed musical guitar riffs used in the second composition ‘The Beta’ and the fourth composition ‘The Delta’ sounded, according to one of the instrumentalists (Nico Maas), incredibly realistic. Maas argued that the performances sounded as if he heard himself playing. However, Maas had not performed these musical guitar riffs before. This research spawned a virtual instrument (VI) that did not only recreate the sound of an electric guitar; it impersonated performance gestures of an electric guitarist. The compositions include the impersonation of the electric guitarist, and it raises an additional question on the ethics on sampling: to what extent does Maas own ‘related rights’⁵⁶ of the compositions (and future works written with the aid of the FPS) without actively participating in the process of performing the original music? One option could be to register Maas as an instrumentalist for every composition that is realised with the aid of the FPS. However, this would imply that other composers working with the FPS also need to register Maas for every compositional work. This is not feasible and introduces additional administration work for every composer. Another option includes a “Buy-Out”⁵⁷ of

⁵⁶ “Related rights [...] protect the legal interests of certain persons [e.g. musical performers] that contribute to making works available to the public or that produce subject matter which [...] contains sufficient creativity [...] to justify recognition of a copyright-like property right” (World Intellectual Property Organization 2016, p. 26).

⁵⁷ In the context of this study, Maas gave me exclusive rights to use his performance gestures for this PhD. See appendix 8.10 including a copy of the Transfer of Rights Agreement document.

Maas' performances and recordings. In exchange, I offer Maas a revenue share for every sold FPS. This constellation reduces the administrative work for every composer, who wants to work with the FPS, and it simultaneously creates an additional revenue stream for the instrumentalist.

However, the abovementioned additional concerns take place in a broader context of existing concerns, and the FPS does not change these existing concerns. One example of existing concerns on sampling is Harry Chamberlin's Mellotron developed at the end of the 1940s in California; the first instrument using tape recordings (tape replay technology) controlled by a keyboard. It was the earliest form of what we know today as "sampling"⁵⁸. According to Robb (2015), "Chamberlin had run into some trouble with the American Federation of Musicians (AFM) for creating, what the union felt, was an instrument that had the potential to replace multiple working musicians with just one keyboard." Dick Moore, a spokesman for AFM, which banned the device from all venues, argued that "our [AFM] problem is job loss due to automation in canned music. We recognise the use of tape instruments as long as they are not utilised to displace another musician" (1976 cited by Eepand 1976, p. 27). Using a Mellotron constituted a contract violation and the outcome for studios that used it was being shut down; consequently, the Mellotron goes uncredited on most albums. However, instrumentalists were not as threatened as much as the AFM expected. Chamberlin argues that instrumentalists are, musically, much more versatile than the Mellotron. Therefore, the Mellotron is used as an 'addition' to the instrumentalist, band, or

⁵⁸ See appendix 8.2.1 for more information concerning sample synthesis techniques.

orchestra rather than as a replacement for musicians, “and what they used [the Mellotron for] [...] is to sweeten up the music” (1976 cited by Eepand 1976, p. 28).

Instead of causing less employment for musicians, music technologists and especially musical instrumentalists, VIs can create different types of work for musical performers or alter existing situations. However, these new opportunities do not entirely replace the work that is lost through sampling. The ‘Flexible Phrase System Virtual Instrument’ (FPS-VI) uses sample synthesis techniques, which include studio recordings of instrumentalists. Therefore, hiring an instrumentalist with studio experience is indispensable for the creation of sample-based VIs. Music technology also reshapes the occupation of a studio instrumentalist. Sometimes, the work of a studio instrumentalist “shift[s] from playing composed parts in the sixties and seventies to improvisation in modern times” (Herbst and Albrecht 2018, p. 30). Therefore, a modern studio performer can sometimes become a co-creator of musical ideas rather than an instrumentalist who follows the score and plays the music. In these times, when music is developed with computers, a studio instrumentalist adds personal variations and ideas to the compositions (Campelo 2015).

The newly designed FPS does raise some additional concerns that go beyond the ethics of sampling generally; however, the FPS does not impact significantly upon the overall pattern of work shifting away from live performances.

5. Composition and instrument design

This practice-based study focused on the production of five compositions, realised with the ‘Flexible Phrase System’ (FPS). Section 5.1 describes the relationship between composing and designing software generally, and section 5.2 discusses each of the five iterations of composing music and designing software individually. It describes how certain characteristics of the FPS affected the composition process and the compositions themselves and how the evaluation of these, in turn, shaped the next version of the FPS⁵⁹.

5.1 General comments on the relationship between composing popular music and designing digital instruments

As mentioned in the literature review⁶⁰, the cultural constraints, stylistic constraints, and physical constraints of both digital and analogue musical instruments always impacted upon compositional processes and musical content. This was also true for the software; it affected the compositional process and the musical content. In contrast to those other factors, however, the software was adjusted from one composition to the next. With each iteration, the adjustments to the software anticipated new compositional possibilities. Because of this, designing and adjusting the software became an act of

⁵⁹ Appendices 8.4, 8.5, 8.6, 8.7, and 8.8 elaborate on software design and the analysis of the audio-visual material of each iteration in more detail.

⁶⁰ See 3.2 Music and human-computer interaction (M&HCI) and 3.3.3 Potential impact of the FPS for more details.

composing and, because the evaluation of each composition suggested new adjustments to the software, composing became a part of software design.

In designing the software, it was necessary to determine the details of the interface. Each choice resulted in different compositional possibilities and constraints. It turned out that compositional ideas helped to shape the interface. Designing the details of the interface thus became part of the compositional process. This is a significant difference to the use of third-party ready-made music production software applications. For example, a developer of sampled musical instruments determines how a composer can switch between different pitch values, rhythmic values, dynamic values, and articulations. A developer of synthesizers determines how the composer can manipulate different parameters such as envelopes, oscillators, and LFO's, to name just a few. Therefore, the compositional process, with the aid of ready-made music production software applications, starts by controlling the virtual instrument (VI) or creating new sounds within the synthesizer, rather than by designing the human-computer interface of the software, like in this project.

Furthermore, adjusting the software gave rise to a new perspective on composing music and generated new compositional ideas, which were then designed into the software. For example, for the fourth composition 'The Delta', several ways of modulating between different scales were programmed into the interface. The inspiration for programming this functionality was a result of a technical obstacle to implementing the cycle of fifth: for programming the software it was necessary to

calculate different modulations. This deepened the understanding of musical rules of modulating between different scales, which were then applied in new compositions⁶¹.

Unfortunately, exploring the usefulness and limitations of the programming possibilities proved to be quite a distraction at times. Therefore, during composing, the focus often shifted to trying out and ‘testing’ the possibilities of the software rather than to composing. In software design, getting ‘stuck’ in an endless process of software improvement is a well-known phenomenon. In order to avoid the sheer endless process of software design, some digital musical instrument (DMI) designers decide to ‘freeze’ the state of the DMI before composing, in order to gain better mastery of it (Waisvisz 1999). However, an important characteristic of this research was the iterative design process, which allowed the software to evolve; therefore, it was not possible to ‘freeze’ the DMI⁶². Each iteration revised the human-computer interface, so for every new composition there were new compositional functionalities available to try out. Consequently, the process of software design became intertwined with the compositional process.

In contrast to earlier compositions, this iterative process created the feeling of being just as much the author of the compositions as the author of the software. Through knowing the software inside out, it was possible to put its tailor-made productional possibilities to full use.

⁶¹ See, appendices 8.4.2, 8.5.2, 8.6.2, 8.7.2, and 8.8.2 for more detailed information on the deepening of musical rules within the newly designed software.

⁶² At the end of the last iteration the software was finished, and it reduced the time expended in exploiting productional possibilities.

5.2 Evaluation of the music & human-computer interaction (M&HCI)

As mentioned in the introduction, the compositions made for this thesis are meant for western trailers and commercials. Therefore, the Flexible Phrase System (FPS) focused primarily on rendering realistic sounds of western musical instruments. The FPS introduced musical possibilities to the compositional process and thus enabled novel musical solutions that existing third-party software did not provide. As with any interface, however, the FPS interface also had to have a number of constraints that limited these novel musical solutions. These limitations encouraged compositional creativity because they forced me to produce music with new possibilities. Some of these limitations were arbitrary⁶³, and some of these limitations were planned⁶⁴ and necessary to set the boundaries within which the FPS could create more realistic sounding instrumental parts. The iterative design process was a quest to find a balance between implementing the desired compositional constraints and preserving the realistic sound for which the FPS was designed. Each iteration went through a number of stages⁶⁵.

The first composition called ‘The Alpha’ was produced with the first version of the FPS_{v1}. Afterwards, the impact of the software constraints on the composition was evaluated using M&HCI. This evaluation identified five types of software constraints:

⁶³ Arbitrary possibilities are software functionalities that were not thought out in advance and had unexpected impacts upon the composition.

⁶⁴ Planned possibilities are software functionalities that were thought out in advance.

⁶⁵ See chapter 4.1 ‘The principles of the iterative design process of this case study’ for a detailed description of each stage.

1. planned constraints with expected impacts;
2. planned constraints with unexpected positive impacts;
3. planned constraints with unexpected negative impacts;
4. arbitrary constraints with positive impacts;
5. arbitrary constraints with negative impacts.

The evaluation of the impact of the constraints on the composition and the compositional process determined the type of modifications to the software in the next iteration. Planned compositional constraints that met expectations were in most cases not modified further. Planned compositional constraints that did not meet expectations were either adjusted or removed.

Arbitrary constraints that had a positive effect on the compositions would sometimes be maintained without any adjustment, or else be further refined. Arbitrary constraints that had a negative impact were either revised or removed. Subsequently, a new (second) iteration started and the revised software, the FPS_{v2}, was then used to compose 'The Beta'. This procedure was then repeated until the final composition. In total there were five iterations and each of the five iterations was built around a composition.

The following five sections discuss each iterative design process chronologically (see Figure 3), from the first composition (The Alpha) in the first section to the fifth composition (The Omega) in the last section. Each section describes the general compositional characteristics and discusses the M&HCI. A couple of selected examples illustrate the impact of the software upon the compositions (Figure 3, green arrows) and

the impact of the compositional issues upon software revisions (Figure 3, red arrows).

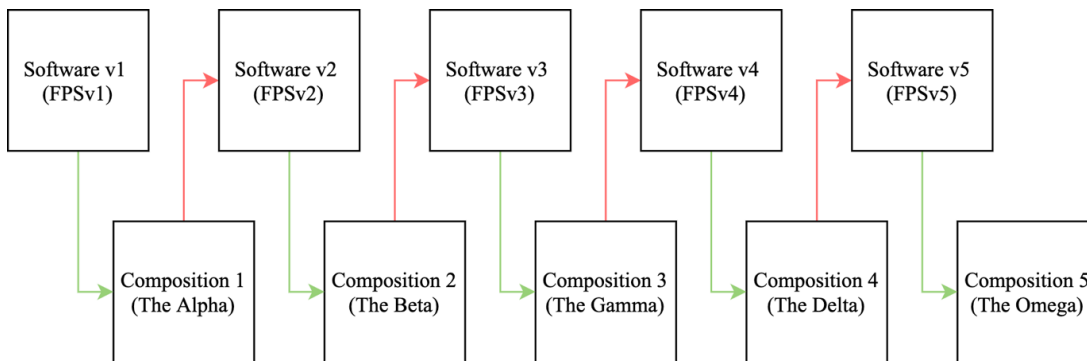


Figure 3. Overview of the five compositions and the five software versions.

In general, all of the compositions were created with the aid of the FPS and existing virtual instruments (VIs) produced subsidiary sounds. The musical form of all compositions is similar to a structure that is frequently used for trailer music productions. The five compositions have a minimum length of three minutes, composed with the help of a Mac computer and a digital audio workstation (DAW) Cubase 9.5. Both computer mouse and computer keyboard were used to compose all compositions⁶⁶. A Neumann KH310 stereo speaker set and AKG K-812 headphone were used to monitor all compositions and an Apogee AD/DA converter connected them to the Mac computer. The musical process produced 50 hours of observation video material. Table 2 provides an overview of the five iterations.

⁶⁶ Some compositions were also composed with the aid of a conventional Native Instruments Komplete Kontrol MIDI keyboard, a Roli Lightpad controller, and a touchscreen tablet including the software Lemur which functions as the FPS-UIC.

Iteration	1	2	3	4	5
Composition	The Alpha	The Beta	The Gamma	The Delta	The Omega
Length of composition	3:12 minutes	3:24 minutes	3:05 minutes	3:11 minutes	3:21 minutes
Start date	05.10.2018	06.12.2018	26.02.2019	24.04.2019	29.04.2019
End date	11.10.2018	14.12.2018	12.03.2019	25.04.2019	30.04.2019
Self-study videos	±8 hours	±11 hours	±20 hours	±9 hours	±2 hours
FPS-VI ⁶⁷	v1	v2	v3 (not used)	v4	v5 (not used)
FPS-UIC ⁶⁸	Not designed	v1	v2	v3	v4

Table 2. Overview of the five iterations.

5.2.1 The Alpha

Compositional characteristics

The first composition called ‘The Alpha’ is a piece of rock music. Typical band instruments such as electric guitars, electric bass and drum kit perform the music along with digital sound effects such as impacts, risers, and boom. The two electric guitars are responsible for the melodic performances and the electric bass, the drum kit, and additional sound effects have subsidiary roles. The first composition has a total length

⁶⁷ Flexible Phrase System – Virtual Instrument (FPS-VI).

⁶⁸ Flexible Phrase System – User Interface Controller (FPS-UIC).

of 03:12 minutes and is composed in 120 bpm. The musical form contains seven musical segments (see Table 3).

Number	From	To	Segment
1	00:00	00:34	A
2	00:34	01:05	B
3	01:05	01:36	C
4	01:36	01:51	D
5	01:51	02:06	D'
6	02:06	02:37	B
7	02:37	03:12	D''

Table 3. An overview of the form of 'The Alpha'.

Impacts of the FPS-VI_{v1} on the 'The Alpha'

Many planned software constraints had their expected impacts on 'The Alpha'. One example is the static two-bar long sequencer window with a 4/4-time signature. It was to be expected that the majority of composed melodies repeated after two bars and that it would be difficult to circumvent this constraint. Moreover, the FPS-VI_{v1} could only perform one dynamic value and it met expectations that all melodies continually remained at the same dynamic level. A planned constraint that had an unexpected positive impact was the 16th rhythmic grid functionality of the FPS-VI_{v1} sequencer window. In general, the functionality met the expectations, but it was unexpected that it could create authentic and sophisticated guitar solos. It turned out to be very practical; consequently, the functionality was extensively used in 'The Alpha'. An example of a planned constraint with an unexpected negative impact was the FPS-VI_{v1} programming functionality of musical phrases. The functionality was insufficiently developed and did

not work as anticipated⁶⁹; consequently, new phrases were not composed, and so the composition was bound to a very small set of pre-designed phrases.

However, there were also arbitrary software constraints that affected ‘The Alpha’ negatively. For example, most of the segmented audio units of the FPS-VI_{v1} had unexpected audio glitches, and they adversely affected the authenticity of the sound. The sound impurities had either to be accepted or to be circumvented. The latter was chosen by using a heavy amplifier simulation effect to mask the impurities. Lighter amp simulations were unfortunately not able to mask the impurities. As a consequence, ‘The Alpha’ overused the sound of a heavily distorted electric guitar which was initially not the intention. An unexpected arbitrary constraint was the functionality to program modes and root notes manually within the digital audio workstation (DAW). It turned out that FPS did not receive every programmed mode because the automation curve that controlled the mode selection had to be accurately timed on each bar. Unfortunately, the DAW did not allow me to quantise the automation curve of modes, which would be a normal process to ensure timing accuracy in a DAW. Consequently, the process was inaccurate and did not work well with the FPS and other VIs. In summary, it can be said that the FPS-VI_{v1} had some promising functionalities, but the static 4/4-time signature, the use of heavily distorted amp simulations, and the limited use of only a small set of pre-defined musical phrases made ‘The Alpha’ too predictable and a bit boring.

⁶⁹ It did not work as anticipated because the graphic representation of new programmed phrases did not match with the sound of the end result. This confused the compositional process; and therefore, new musical phrases were not used.

Impacts of the evaluation on the software revision

Based on the evaluation of ‘The Alpha’, the FPS-VI_{v1} was revised (called FPS-VI_{v2}). For example, the revised FPS-VI_{v2} incorporated a time signature functionality, which now allowed the user to switch between different time signature values such as $\frac{3}{4}$, $\frac{4}{4}$, or $\frac{10}{4}$. Also, different musical phrases were programmed beforehand. This enabled access to more rhythmic variations and articulations. Moreover, sound impurities of the segmented audio units were removed, thus allowing a choice of different amp simulations. All of these changes were meant to create a more diverse, more interesting, and more dynamic sound with the FPS-VI_{v2} in the next composition.

In addition, a new user interface controller (UIC) called the FPS-UIC_{v1} was designed in order to control modes and root notes that were performed by different virtual instruments (including the FPS-VI). In ‘The Alpha’, modes and root notes were programmed manually within the digital audio workstation (DAW); however, this process was inaccurate and time-consuming. After trial and error, it appeared that the Lemur⁷⁰ software environment allowed the design of a UIC that performed different modes and root notes easily and quickly. This motivated me to introduce a customised FPS-UIC_{v1} installed on an external iPad. The FPS-UIC_{v1} is connected to the FPS-VI_{v2}, and it is integrated within the DAW information stream.

⁷⁰ Lemur is a software application developed by the company Liine. See appendix 8.2.3 for more information.

5.2.2 The Beta

Compositional characteristics

The second composition called ‘The Beta’ is a piece of jazz music. The electric bass and drum kit accompany two electric guitars and a piano. The electric guitars form the lead instruments. The piano is responsible for the harmony and the electric bass supports the tonic chords of the piano. ‘The Beta’ contains seven segments (Table 4) totalling 3:22 minutes in 120 bpm without deviations.

Number	From	To	Segment
1	00:00	00:33	A
2	00:33	01:08	B
3	01:08	01:20	C
4	01:20	01:40	B’
5	01:40	02:08	A
6	02:08	02:40	A’
7	02:40	03:22	B’

Table 4. An overview of the form of ‘The Beta’.

Impacts of the software on the ‘The Beta’

In ‘The Beta’, having learned from ‘The Alpha’, the musical content changed as a result of the new possibilities and constraints of the FPS-VI_{v2} and the FPS-UIC_{v1}. The planned constraint of changing between different time signatures had its expected impact. The functionality was quickly adapted and had additional value to the composition. Beside the $\frac{4}{4}$ -time signature, the musical content of ‘The Beta’ also included time signature values such as $\frac{10}{4}$, $\frac{2}{4}$, and $\frac{3}{4}$. The revised audio database was another planned constraint that met expectations. In contrast to ‘The Alpha’, the revised database allowed cleaner amp simulations to be chosen because there was no need to mask

impurities anymore. Showcasing the pure sound of the FPS-VI_{v2} was a significant motivation; and therefore, less or even no post-processing effects were used. However, programming different phrases at the very beginning of the compositional process was a planned constraint but it had an unexpected negative impact. After programming the pre-designed musical phrases, there were no new phrases or variations designed. The FPS-VI_{v2} provided still an awkward way of choosing and programming different rhythmic phrases; therefore, the unacceptable phrase switch restrained the composition to design or to use new phrases. As a consequence, all melodies and harmonies are composed with the rhythmical limitations of the two pre-designed phrases.

The FPS-UIC_{v1} had some arbitrary constraints with positive impacts upon ‘The Beta’. Initially, the compositional process started by determining the musical form. ‘The Beta’ was an exception: while finding out the productional boundaries of the FPS-UIC_{v1}, it became apparent that the software could also be used as a tool to determine the musical form. Consequently, the form of ‘The Beta’ was inherently connected to switching between different modes. Moreover, this process also simplified the use of different modes. But on the other hand, this new functionality overwhelmed the process of composing and had an unexpected negative impact: it was very challenging to maintain an overview of the available harmonic possibilities. As a consequence, the harmonic progressions remained straightforward and focused solely on the tonic chord of each mode. Moreover, ‘The Beta’ exclusively used the root position of each chord, and it ignored the use of first, second, and third chord inversions. In some cases, it was desirable to use a chord inversion rather than the root position; however, it became too complicated to design chord inversions with the FPS-UIC_{v1}.

Impacts of evaluation on the software revision

Based on the result of the evaluation of ‘The Beta’, both the FPS-VI_{v2} and the FPS-UIC_{v1} were revised to the FPS-VI_{v3} and the FPS-UIC_{v2}. The FPS-VI_{v3} included a revised phrase sequencer window, which should allow the composer to change and program more intuitively. Moreover, the FPS-UIC_{v1} had some promising techniques, but the interface was completely revised. The FPS-UIC_{v2} did not change between different modes anymore but between different scales. Working with musical scales was closer to my existing compositional process and should therefore provide a better compositional overview. Moreover, the FPS-UIC_{v2} also included a chord inversion functionality which was intended to provide the compositional process with more compositional flexibility.

5.2.3 The Gamma

Compositional characteristics

The third composition called ‘The Gamma’ is a piece of piano music. The instrumentation focuses on a piano, and it excludes the use of other musical instruments. The piano plays both the melody and the harmonic progression. The form contains four segments (Table 5), which are together 3:05 minutes long, produced in 120 bpm without any tempo changes.

Number	From	To	Segment
1	00:00	01:13	A
2	01:13	01:50	B
3	01:50	02:22	C
4	02:22	03:05	A’

Table 5. An overview of the form of ‘The Gamma’.

Impacts of the software on the ‘The Gamma’

In ‘The Gamma’, having learned from ‘The Beta’, the musical content changed as a result of the FPS-UIC_{v2}. Only a few planned constraints had their expected impacts. One example was working in scales instead of in modes. It was expected that it would simplify the process of choosing between different chords and scales and this was achieved. However, many of the planned software constraints didn’t have their expected impacts and were disappointments. One example was the limited harmonic palette of the FPS-UIC_{v2}. It was planned that the interface limited the selection of different harmonic chords; however, it was not expected that the interface would be biased towards constantly switching between different scales. The reason for this was an unbalanced design of the user interface. The interface was well designed for the performance of the first chord. The first chord had the option of eleven different root notes, five different types of harmonies, and four-chord inversions. However, less time was invested in the production of the harmonic chords that followed after the first chord. As a consequence, the musical content was constrained to the harmonic progression that included only a M^7_{b5} chord (second chord), and a min6 or maj6 chord (third chord) followed by the second chord. After the third performed chord, the ‘The Gamma’ continued the harmonic progression by either using a harmonic evolved M^7_{b5} chord or by interrupting the chord progression by returning to the initial first chord. Unfortunately, the FPS-UIC_{v2} did not provide any other harmonic options that occur naturally within the selected scale; consequently, the composition was forced either to stay solely on one harmonic chord or to switch always between different scales. The latter was chosen and, as a result, the harmonic progression sounded arbitrary.

Another planned constraint that did not meet my expectations was the development of melodies with the aid of the MIDI keyboard. It was planned that the

software would transpose musical notes automatically, which would ensure that only notes that occurred naturally within the selected mode could be played. However, it was not expected that the recorded MIDI notes did not interact with the modulation parameters that the DAW transmitted to the FPS-UIC_{v2}. Once the MIDI notes were played and recorded, the playback engine of the DAW played the actual recorded MIDI notes within the piano roll. As a consequence, the FPS-UIC_{v2} forced me to perform all melodies live rather than to edit the MIDI notes in the piano roll. It was also not possible to copy and paste the melody to another bar or musical segment.

An arbitrary constraint that did not meet my expectations was a short time delay of transmitted control change information from the DAW to the FPS-UIC_{v2} on the first beat of each bar. This technical delay forced melodies to be timed perfectly on the first beat of a bar. However, real-time improvisational performances on the MIDI Keyboard controller were rhythmically inaccurate; thus, it was almost impossible to perform a musical note perfectly on the first beat of a bar. As a consequence, this technical issue was circumvented by ignoring the first beat of each bar and by starting all improvisational performances on the second, third, or fourth beat of each bar. This constraint also had a decisive effect on the absence of the FPS-VI_{v3}. The compositional process started with the use of the FPS-UIC_{v2}, which controlled a digital recreation of a piano sound. After the real-time piano performance, I strived to use the FPS-VI_{v3} and to continue with an ‘improvisation’ of the electric guitar on the melodic and harmonic progression of the piano. However, the functionality of the FPS-UIC_{v2} was much more advanced in comparison to the FPS-VI_{v3}. It was not possible to incorporate the FPS-VI_{v3} within the compositional process because the software was not able to perform different time signature deviations in real-time, and it was also not able to perform triple notes. Triple notes formed the rhythmical motive of the composition. ‘The Gamma’

went beyond the limits of the FPS-VI_{v3}; therefore, it was decided that the FPS-VI_{v3} would be excluded.

Impacts of evaluation on the software revision

Based on the result of ‘The Gamma’, both the FPS-VI_{v3} (called FPS-VI_{v4}) and the FPS-UIC_{v2} (called FPS-UIC_{v3}) were significantly revised because of the many planned and arbitrary constraints that mostly did not meet expectations. The imperfect realisation of the time signature function is one of the main reasons why ‘The Gamma’ did not include the FPS-VI_{v3}. Therefore, the time signature functionality, the tempo deviation functionality and the sequencer window of FPS-VI_{v4} were all revised significantly with the hope that they would meet my expectations in the fourth composition. Moreover, the FPS-VI_{v4} database was increased with stochastic performance recordings for a more sophisticated sounding instrument.

In addition, the FPS-UIC_{v3} was significantly improved. The human interface controller was completely revised: scales, chords, and inversions were significantly changed and expanded. In the fourth composition, it would be possible to have a greater palette of harmonic choices and to modulate to different scales in various ways.

5.2.4 The Delta

Compositional characteristics

The instrumentation of the fourth compositions called ‘The Delta’ contains a piano and two electric guitars. The electric guitars are the lead instruments. The form has five musical segments (Table 6). ‘The Delta’ is a 3:11 minutes long original musical work, produced in 90 bpm without any tempo changes.

Number	From	To	Segment
1	00:00	01:15	A
2	01:15	01:55	A'
3	01:55	02:17	B
4	02:17	02:37	A''
5	02:37	end	C

Table 6. An overview of the musical form of 'The Delta'.

Impacts of the software on the 'The Delta'

In 'The Delta', having learned from 'The Gamma', the musical content changed significantly as a result of both the FPS-VI_{v4} and the FPS-UIC_{v3}. The completely revised FPS-VI_{v4} GUI was biased towards the concept of the DAW's piano roll editor view. The piano roll view was a planned constraint with an expected impact: it enabled the design of various melodic and rhythmic patterns. From this followed that the FPS-VI_{v4} inspired the design of different leading musical melodies which were then used in the composition. Another planned constraint with an expected impact was the introduction of tempo changes. This new functionality inspired me to test a different tempo, with the result that the composition was produced in 90 bpm. Unlike 'The Alpha', 'The Beta', and 'The Gamma', it was the first composition in this study that was not performed in 120 bpm. Another planned constraint had an unexpected positive impact upon the musical content. 'The Delta' used pre-recorded performances from Maas, which embedded both stochastic material and single note recordings. Maas

indirectly influenced the compositional content with his pre-recorded phrases because the accompaniment was composed based on the pre-recorded phrases⁷¹.

The compositional process of 'The Delta' started with a piano sketch, and it aimed to be developed into an orchestral arrangement⁷². However, the FPS-UIC_{v3} revealed an arbitrary constraint with a negative effect on the instrumentation of the composition: the system provided insufficient possibilities to transcribe the piano sketch to an orchestral arrangement. The FPS-UIC_{v3} was built to control different musical scales, chords, and inversions and to control the FPS-VI_{v4}. However, it became apparent that the FPS-UIC_{v3} was not capable of transmitting expression related information from a piano to an orchestra. Therefore, transcribing a piano sketch to an orchestra did not work. Consequently, the compositional content excluded the use of an orchestra and focused on a piano and two guitars. Furthermore, the planned constraints of scale modulation and chord inversions had an unexpected positive impact upon the composition. In contrast to the previous two compositions, 'The Delta' did not use arbitrarily selected scale modulations because all they were all monitored in real-time and then recorded. However, in 'The Beta' and 'The Gamma', it was the other way around: the modulations (based on music theory) were first composed; then the end

⁷¹ One could argue that Maas owned 'related-rights' to 'The Delta'. Maas also listened to the 'The Delta' and he perceived the played electric guitar as his performance. However, he had never played precisely these musical phrases before. Chapter 4.5 discusses the Ethics on Sampling in more detail.

⁷² In order to create digital orchestral works, the compositional process started with a piano, which performed melodies and harmonies. Both melodies and harmonies were then replaced and performed by other orchestral instruments.

result was monitored. As a result, the harmonic progression of ‘The Delta’ sounded less arbitrary in contrast to the previous compositions, and it formed a harmonic and melodic musical coherence.

Moreover, the FPS-UIC_{v3} created new possibilities to explore new musical ideas. The FPS affected the compositional content positively in using different harmonic progressions. For example, ‘The Delta’ performed a combination of a D⁷sus4 chord to modulate three positions within the Cycle of Fifth to another key without changing any notes within the triggered D⁷sus4 chord.

Impacts of evaluation upon the software revision

Based on the result of ‘The Delta’, some minor revisions to the FPS-UIC_{v3} (called FPS-UIC_{v4}) did take place. The FPS-UIC_{v4} introduced the functionality of selecting different harmonic voicings. The routing of the voicings was routed automatically to different instruments of the orchestra. Additionally, the option of selecting between narrow voicings and wide voicings was programmed. They were called drop voicings. All new functionalities and constraints should have the effect of composing an orchestral piece of music.

5.2.5 The Omega

Compositional characteristics

The fifth and final composition called ‘The Omega’ is a piece of orchestral music. The musical melodies and musical ideas from ‘The Delta’ were reused and performed by an orchestra. The form contains a total of four segments (Table 7). ‘The Omega’ is 3:21 minutes long composition, produced in 90 bpm without any tempo changes.

Number	From	To	Segment
1	00:00	01:35	A
2	01:35	02:20	B
3	02:20	02:43	C
4	02:43	03:21	B'

Table 7. An overview of the form of 'The Omega'.

Impacts of the software on the 'The Omega'

The musical content of 'The Omega' changed significantly due to the use of the FPS-UIC_{v4}. Most of the planned constraints had their expected impacts upon the instrumentations, the used voicings, the harmonic progressions, and the melodic progressions. It was to be expected that the planned constraints would exclude the use of the FPS-VI because the sound of an electric guitar did not fit well in an orchestral musical work. The planned constraint of programming drop voicings also had an unexpected positive impact on the composition. The drop voicings offered different types of harmonisation, but it was not expected that they made it accessible to monitor different types of voicings with different instrumentations. This functionality was very inspiring and helped in choosing a voicing that fitted best to different instrumentation groups. Another planned constraint, the simultaneous recording of different instrument sections with different articulations (sustained and accented), had an unexpected positive effect upon the compositional content. As a result of this recording functionality, 'The Omega' started with long sustained notes with advanced voicings, which worked well for the introduction of a composition.

Moreover, the system also had arbitrary constraints that had unexpected positive impacts on 'The Omega'. One example was the homophonic voicing functionality of

the system, which enabled complex real-time homophonic arrangements. These complex arrangements were not possible with a conventional MIDI keyboard controller because human hands were not able to play multiple voicings, spread over several octaves, simultaneously. Unfortunately, this arbitrary constraint had also an unexpected negative impact upon the composition because this functionality was not applicable for polyphonic writing. In some situations, in the compositional process, it was desirable to use polyphonic voicings. Unfortunately, this was not possible, and I had to accept the limitations.

The FPS-UIC_{v4} also had an arbitrary positive affect on the use of harmonic chords and musical scale modulations. The FPS-UIC_{v4} shifted the focus to the ‘sound’ of the harmonic chords and not on the used music theory. It shifted the focus to the ‘sound’ because the FPS-UIC_{v4} performed complex homophonic musical chords, including nine pitch values, by clicking on one single button. In measure 36, for example, ‘The Omega’ performed an extreme key modulation from one major scale to another major scale. While composing, I paid attention to the impact of the key modulation based on my musical understanding; however, I did not pay attention to how many positions of the musical key moves there were within the Cycle of Fifth.

In summary, the evaluation of the compositions identified several types of planned or arbitrary constraints with either a positive or negative impact. Sometimes, the software was not ready yet and had too many arbitrary constraints with negative impacts. The number of negative impacts also affected the frustration level of the compositional process, and it turned out that the FPS did not always solve compositional issues, but also caused new issues. Once aware of these new issues, it was possible to circumvent most of them, which automatically constrained the compositional content to a limited set of compositional possibilities.

6. Conclusions

The introduction of this practice-based research stated that humanising the sound of virtual instruments (VIs) affected my compositions adversely. It stated that humanising the sound of a VI cost me a lot of time, interrupted my compositional process, and affected my melodic and harmonic compositional choices adversely. These factors increased the demand for an alternatively designed digital musical instrument (DMI). From this standpoint followed the research aim to produce five compositions, produced with the aid of my newly designed DMI.

This practice-based research concluded that, within the interconnected process of composing music and designing software, designing software became an act of composing music and composing music became part of designing software. The interconnected processes transformed my professional practice as a composer because the iterative design process of compositions and DMIs relocated and expanded the compositional process to the technical design of the DMIs. That being said, the process of software design changed my musical conceptions and offered me new avenues for musical creativity. However, the interconnected process of composing music and designing software did not transform all parts of my professional practice. For example, the FPS is biased towards popular music software environments and worked within my existing compositional context. Therefore, I still composed western popular and orchestral music, and the FPS had no impact upon my choice of musical genre.

The functionalities of the FPS possibly will limit composers and producers who strive to experiment with musical possibilities beyond western popular or orchestral music. I built the FPS for my compositional process and my professional practice as a commercial and movie trailer composer. The process focused on western popular and

orchestral music with an average length of three minutes and that included melodic material of two bars. I concluded that the FPS will not satisfy composers who require to work beyond the limitations of popular music.

Moreover, this research provided insights into the compositional process as well as a self-study method, the concurrent production of popular music, and a newly designed DMI for other composers. Furthermore, this thesis has contributed five compositions to the repertoire of popular music, and it has contributed two software units to the music software market.

To better understand the implications of these conclusions, future studies could address a feasibility study of the implementation of observation techniques and analysis methods within a limited timeframe. This practice-based study shows that it is feasible for a composer to conduct an autoethnographic self-study within an interconnected process of composing music and designing software; however, practitioners should consider that the implementation and analysis of self-study data costs a lot of time. Further research is needed to determine the effects of autoethnographic approaches that are more efficient; therefore, more applicable for practical compositional circumstances. Another suggestion emphasises the adaption of the principles of the FPS-VI to other musical instruments. It is highly satisfying to investigate technical feasibility in the adaption of the FPS-VI for other western instruments such as string instruments, brass instruments, woodwinds, or the human voice. Furthermore, a research suggestion includes the adaption of the iterative design process to more composers. Within this study, no other composers executed the iterative design approach. Therefore, this thesis suggests providing the methodological framework for other composers and technologists and researching the interactive design process within their professional

practice. Such research would enable the possibility to scrutinise the differences and similarities between different composers.

7. References

- Adams, A., 2016. *Perspective A Forum for Film, TV, and Media Composers* [online]. Available from: <https://perspectiveforum.net/2016/09/15/what-effect-has-the-daw-had-on-the-composition-of-modern-orchestral-film-music/> [Accessed 21 August 2019].
- Adler, S., 2002. *The Study of Orchestration*. 3rd Edition. New York: W. W. Norton & Company.
- Adrien, J.-M. and Morrison, J., 1990. Mosaic: A modular program for synthesis by modal superposition. In: Cadoz, C., ed. *Modèles physiques, création musicale et ordinateur*. Paris: Éditions de la Maison des sciences de l'homme.
- Aglić Čuvčić, M., 2018. Introducing a Dataflow visual programming language for understanding program execution. *Polytechnica: Journal of Technology Education*, 2nd Edition. Rijeka: University of Rijeka, 35-40.
- Alberge, D., 2013. *Electronic music: The Observer* [online]. Available from: <https://www.theguardian.com/music/2013/dec/15/electronicmusic-television> [Accessed 02 December 2016].
- Albright, D., 2015. *The Evolution of Music Consumption: How We Got Here*. [online] Available from: <https://www.makeuseof.com/tag/the-evolution-of-music-consumption-how-we-got-here/> [Accessed 14 August 2019].
- Arfib, D. and Le Brun, M., 1979. Digital waveshaping synthesis. *Journal of the Audio Engineering Society*, 27(4). New York: Audio Engineering Soc, 250-266.
- Ashbourn, J., *Audio Technology, Music, and Media*. 1st Edition. Cham: Springer Switzerland.

- Baalman, M.A.J., 2017. Interplay Between Composition, Instrument Design and Performance. *Musical Instruments in the 21st Century*, Springer, Singapore.
- Belkin, A., 2015. *On Musical Ideas* [online]. Available from: <https://alanbelkinmusic.com/site/en/index.php/on-musical-ideas/> [Accessed 6 May 2019].
- Bell, A. P., 2015. DAW democracy? The dearth of diversity in 'playing the studio'. *Journal of Music, Technology & Education*, 8 (2), 129-146.
- Bijsterveld, K. and Peters, P. F., 2010. Composing Claims on Musical Instrument Development: A Science and Technology Studies' Contribution. *Interdisciplinary Science Reviews*, 35 (2), 106-122.
- Bilous, J., 2016. *Concatenative Synthesis for Novel Timbral Creation* [online]. Thesis (MSc) California Polytechnic State University. Available from: <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=2753&context=theses> [Accessed 20 November 2017].
- Boden, M. A., 1990. *The Logic of Practice*. Cambridge, UK: Polity Press.
- Bonana, J. and Serra, X., 2007. Synthesis of the Singing Voice by Performance Sampling and Spectral Models. *IEEE Signal Processing Magazine* [online], 27 (2), 67-79.
- Brown, T., 2008. *Design Thinking*. Harvard Business Review, 53 (6), 84-92.
- Burgoyne, J. A., Fujinaga, I. and Downie, J., 2015. A new companion to digital humanities. In: Schreibman S., Siemens, R., and Unsworth, J., eds. *Music Information Retrieval*. Chichester: John Wiley, 213-228.
- Campelo, I., 2015. That Extra Thing - The Role Of Session Musicians In The Recording Industry. *Journal on the Art of Record Production*, (Issue 10).

- Cantini Electric Violins, 2020. *Cantinielectricviolins* [online]. Available from: <http://www.cantinielectricviolins.com/> [Accessed 30 January 2020].
- Chadabe, J., 1997. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, New Jersey: Prentice Hall.
- Chang, H., 2008. Chapter 3: Autoethnography. *Autoethnography as Method*. Walnut Creek, CA : Left Coast Press, 43-57.
- Charmaz, K., 2006a. *Constructing grounded theory: A practical guide through qualitative analysis*. London: SAGE Publications.
- Chowning, J. M., 1973. Synthesis of complex audio spectra by means of frequency modulation. *Journal of the Audio Engineering Society*, 21 (7), 526-534.
- Clauhs, M., 2020. Songwriting with digital audio workstations in an online community. *Journal of Popular Music Education*, 4 (2), 237-252.
- Collins, D., 2005. A synthesis process model of creative thinking in music composition. *Psychology of Music*, 33 (2), 193-216.
- Computer Music, 2015. *musicradar.com*. [online] Available from: <https://www.musicradar.com/tuition/tech/how-to-humanise-your-music-in-a-daw-632072> [Accessed 24 January 2020].
- Cook, P. R., 2002. *Real sound synthesis for interactive applications*. Natick (Massachusetts): A K Peters.
- Culkin, J., 1967. A schoolman's guide to Marshall McLuhan. *The Saturday Review*. 18 March 1967, 51-53.
- Demoucron, M., 2008. *On the control of virtual violins - Physical modelling and control of bowed string instruments* [online]. Thesis (PhD). Royal Institute of Technology, Stockholm. Available from: <https://tel.archives-ouvertes.fr/tel-00349920/document> [Accessed 17 March 2017].

- DeRosa, R. and Pejrolo, A., 2011. *Acoustic and MIDI orchestration for the contemporary composer*. Oxford: Focal Press.
- DeSantis, D., 2015. *Making Music: 74 Creative Strategies for Electronic Music Producers*. Berlin: Ableton AG.
- Doucet, J.-E., 2016. *Open Stage Control* [online]. Available from: <http://osc.ammd.net/> [Accessed 12 February 2019].
- Doyle, P., 2005. *Echo and Reverb*. Middletown: Wesleyan University Press.
- Drummond, J., 2009. Understanding interactive systems. *Organised Sound*, 140 (2), 124-133.
- Dubber, A., 2015. *Human-music-interaction* [online]. Available from: <https://andrewdubber.com/2015/07/human-music-interaction/> [Accessed 12 December 2016].
- Duignan, M., 2008. *Computer Mediated Music Production: A Study of Abstraction and Activity* [online]. Thesis (PhD). University of Wellington. Available from: https://pdfs.semanticscholar.org/6b4d/2e9295c13ce7463b2f22e5a7d49ec0fd4cf1.pdf?_ga=2.176716844.805678477.1581002943-748640040.1580480538 [Accessed 28 November 2017].
- Eepand, L., 1976. A phantom Orchestra at your Fingertips. *Crawdaddy*. April.
- Einbond, A., Trapani, C., Agostini, A., Ghisi, D. and Schwarz, D., 2014. Fine-tuned Control of Concatenative Synthesis with CataRT Using the Bach Library for Max. *International computer music conference (ICMC)*, Athens 14-20 September 2014. London: City University of London Institutional Repository. Available from: <https://openaccess.city.ac.uk/id/eprint/16730/1/Einbond2014-1.pdf> [Accessed on 14 September 2016].

- Eldridge, A. and Kiefer, C., 2017. The self-resonating feedback cello: interfacing gestural and generative processes in improvised performance. *Proceedings of New Interfaces for Music Expression*, 25-29. Available from: https://pdfs.semanticscholar.org/ab00/4b75fd0a39e098f1660277e0ea62b05e7954.pdf?_ga=2.38894566.455383501.1595434827-842276872.1594986629 [Accessed 26 June 2020].
- Elliott, G., 2004. *Global Business Information Technology: an integrated systems approach*. London: Pearson Education.
- Ellis, C., Adams, T. E. and Bocher, A. P., 2010. Autoethnography: An Overview. *Forum: Qualitative Social Research* [online], 12 (1).
- Emmerson, S., 2000. Crossing Cultural Boundaries Through Technology?. In: Emmerson, S., ed. *Music, Electronic Media and Culture*. Burlington: Ashgate publishing limited, 115 -137.
- Emerson, G. and Egermann, H., 2018. Exploring the Motivations for Building New Digital Musical Instruments. *Musicae scientiae* [online]. Available from: http://eprints.whiterose.ac.uk/135263/1/Emerson_Motivations_Musicae_Scientiae_inpress.pdf [Accessed 25 November 2019].
- Evanstein, M., 2019 *Composing (with) Interfaces: Analog and Digital Feedback Loops and the Compositional Process* [online]. Thesis (PhD) University of California. Available from: <https://escholarship.org/uc/item/47k5f7vv> [Accessed on 15 July 2020].
- Flaherty, K., 2016. *Diary Studies: Understanding Long-Term User Behavior and Experiences* [online]. Fremont: Nielsen Norman Group. Available from: <https://www.nngroup.com/articles/diary-studies/> [Accessed 04 March 2018].

- Glaser, B. G. and Strauss, A. L., 1967. *The discovery of grounded theory: strategies for qualitative research*. Chicao: Aldine Transaction.
- Graf, M., 1947. *From Beethoven to Shostakovich: The Psychology of the Composing Process*. New York: Philosophical Library.
- Grudin, J., 1992. Utility and usability: research issues and development contexts. *Interacting with computers*, 4 (2), 209-217.
- Guck, M. A., 1994. Rehabilitating the incorrigible. In: Pople, A., ed. *Theory, Analysis and Meaning in Music*. Cambridge: Cambridge University Press, 57-94.
- Hamm, C., Walser, R., Warwick, J. and Garrett, C., 2014. Popular Music. *Grove Music Online* [online], Available from:
<https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-1002259148> [Accessed 13 March 2021].
- Hatten, R. S., 2004. *Interpreting Musical Gestures, Topics, and Tropes Mozart, Beethoven, Schubert*. Bloomington: Indiana University Press.
- Henri, P. and Välimäki, V., 2011. Parametric Electric Guitar Synthesis. *Computer Music Journal*, 35 (3), 18-27.
- Herbst, J.-P. and Albrecht, T., 2018. The Work Realities of Professional Studio Musicians in the German Popular Music Recording Industry: Careers, Practices and Economic Situations. *Journal of the International Association for the Study of Popular Music*, 8 (2), 19-37.
- Hexler, 2008. *TouchOSC* [online]. Available from: <https://hexler.net/docs/touchosc> [Accessed 12 February 2019].
- Huber, D. M. and Williams, P., 1998. *Professional microphone techniques*. Somerset: HAYNES PUBLISHING GROUP.

- Huff, C., 2018. *Direct Boxes – What They Do, Why They’re Needed, and Why Price Matters* [online]. Available from: <https://www.behindthemixer.com/direct-boxes-why-price-matters/> [Accessed 27 August 2019].
- Hult, M. and Lennung, S.-Å., 1980. Towards a Definition of Action Research: A note and bibliography. *Journal of Management Studies*, 17(2), 241-250.
- InMusic, Inc., 2020. *Akaipro* [online]. Available from: <https://www.akaipro.com/products/ewi-series> [Accessed 30 January 2020].
- Jammy Instruments, 2019. *Playjammy* [online]. Available from: <https://playjammy.com/> [Accessed 30 January 2020].
- Johnston, W. M., Hanna, P. J. R. and Millar, R. J., 2004. Advances in Dataflow Programming Languages. *ACM Computing Surveys*, 36 (1), 1-34.
- Karray, F., Alemzadeh, M. and Sa, J. A., 2008. Human-Computer Interaction: Overview on State of the Art. *International Journal on smart sensing and intelligent systems*, 1 (1), 137-159.
- Katz, M., 1999. *The phonograph effect: The influence of recording on listener, performer, composer, 1900-1940*. Thesis (PhD). University of Michigan.
- Katz, M., 2010. *Capturing Sound*. 2nd edition. Oakland: University of California Press.
- Kefauver, A. P. and Patschke, D., 2007. *Fundamentals of Digital Audio (The Computer Music and Digital Audio Series)*. Middleton: A-R Editions.
- Kitchin, R. and Dodge, M., 2011. *Code/Space: Software and Everyday Life*. Cambridge: MIT Press.
- Kock, N., 2014. 33. Action Research. In: Soegaard, M., ed. *The Encyclopedia of Human-Computer Interaction* [online]. 2nd edition Available from: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human->

computer-interaction-2nd-ed/action-research-its-nature-and-relationship-to-human-computer-interaction [Accessed 10 October 2019].

- Lalama, D. S., 1998. *The Composition and Analysis of "Double Concerto for Saxophone and Piano"*. 1st edition. New York: New York University, School of Education.
- Lewin, K., 1946. *Action research and minority problems*. New York: Harper & Row.
- Lindemann, E., 2007. Music synthesis with reconstructive Phrase Modeling. *Signal Processing Magazine*, 24 (2), 80-91.
- Lindemann, E., 2007. *Synful Orchestra* [online]. Available from: www.synful.com [Accessed 17 December 2016].
- Macchiusi, I. A., 2017. "*Knowing is Seeing:*" *The Digital Audio Workstation and the Visualization of Sound* [online]. Thesis (PhD). York University Toronto. Available from: https://yorkspace.library.yorku.ca/xmlui/bitstream/handle/10315/34478/Macchiusi_Ian_A_2017_PhD.pdf?sequence=2&isAllowed=y [Accessed on 21 September 2019].
- Magnusson, T., 2010. Designing Constraints: Composing and Performing with Digital Musical Systems. *Computer Music Journal*, 34 (4), 62-73.
- Maestre Gómez, E., 2009. *Modeling Instrumental Gestures: An Analysis/Synthesis Framework for Violin Bowing* [online]. Thesis (PhD). Barcelona: Universitat Pompeu Fabra. Available from: https://pdfs.semanticscholar.org/eacd/87e92dff966a79e24123825a41938496b933.pdf?_ga=2.143087548.805678477.1581002943-748640040.1580480538 [Accessed 17 December 2016].

- Maestre, G. E., Ramirez, R., Kersten, S. and Serra, X., 2009. Expressive Concatenative Synthesis by Reusing Samples from Real Performance Recordings. *Computer Music Journal*, 33 (4), 23-42.
- Manzo, V. J., 2016. *Foundations of music technology*. Oxford: Oxford University Press.
- MassiveMusic, 2018. *About us* [online]. Available from: <https://www.massivemusic.com/en/about> [Accessed 15 June 2019].
- Miranda, E. R. and Wanderley, M. M., 2006. *New digital musical instruments*. Middleton: A-R Editions.
- Moon, J., 2003. *Learning journals and logs, Reflective Diaries*. Dublin: University College Dublin.
- Newman, T. U., 2008. *The creative process of Music composition: A Qualitative Self-study*. New York: New York University.
- Nielsen, J., 1993. Iterative user-interface design. *Computer*, 26 (11), 32-41.
- Norman, D. A., 1999. Affordances, Conventions, and Design. *Interactions*, 6 (3), 38-41.
- Pace, S., 2012. Writing the self into research: Using grounded theory analytic strategies in autoethnography. *TEXT* [online], 13 (1), 1-19.
- Pearce, M. and Wiggins, G. A., 2002. Aspects of a Cognitive Theory of Creativity in Musical Composition. In: Bento C., Cardoso, A. and Wiggins, G. A., eds. *Proceedings of the ECAI02 Workshop on Creative Systems*, Lyon 2002. Available from: <https://ist.psu.edu/viewdoc/summary?doi=10.1.1.3.7028> [Accessed 27 April 2020].
- Percival, G., 2013. *Physical Modelling meets Machine Learning: Performing Music with a Virtual String Ensemble* [online]. Thesis (PhD). University of Glasgow. Available from: <http://theses.gla.ac.uk/4253/1/2013percivalphd.pdf> [Accessed 21 May 2017].

- Pérez, C. A., 2009. *Enhancing Spectral Synthesis Techniques with Performance Gestures using the Violin as a Case Study* [online]. Thesis (PhD). Universitat Pompeu Fabra. Available from: <http://mtg.upf.edu/node/1578> [Accessed 23 May 2016].
- Pérez Carrillo, A. and Bonada, J., 2010. The Bowed Tube: A Virtual Violin [online]. *New interfaces for Musical Expression*, Sydney 15-18 June 2010. Sydney: University of technology Sydney. Available from: https://www.nime.org/proceedings/2010/nime2010_229.pdf [Accessed 23 April 2016].
- Polanyi, M., 1966. *The Tacit Dimension*. Chicago: University of Chicago Press.
- Puckette, M., 1991. Something Digital. *Computer Music Journal*, 15 (4), 56-69.
- Ramshaw, P., 2006. Is Music Production now a Composition Process? *First Annual Conference on the Art of Record Production*, London 17 - 18 September 2005. London: University of Westminster. Available from: <http://www.paulramshaw.com/is-music-production-now-a-composition-process/> [Accessed 23 January 2018].
- Roads, C., 1988. Introduction to granular synthesis. *Computer music journal*, 12 (2), 11-13.
- Robb, H., 2015. *Studio Bell* [online]. Available from: <https://nmc.ca/an-orchestra-at-your-fingertips-pt-2-the-new-sound/> [Accessed 7 January 2017].
- Roels, H., 2013. Data-collection methods and reflection within a self-study on the creative process of music composition. In: Verbreke, J. and Park., B., eds. *Knowing (by) designing*. Leuven: LUCA, 157-164.
- Roels, H., 2014. *The creative process in contemporary polyphonic music* [online]. Thesis (PhD). Universiteit Ghent [Accessed 13 March 2018].

- Roland Corporation, 2020. *Drums & Percussion* [online]. Available from:
https://www.roland.com/global/categories/drums_percussion/v-drums_kits/
 [Accessed 30 January 2020].
- Rotman, D., 2013. How Technology Is Destroying Jobs. *MIT Technology Review*
 [online], July/August 2013. Available from:
<https://www.technologyreview.com/s/515926/how-technology-is-destroying-jobs/> [Accessed 14 March 2018].
- Russ, M., 2004. *Sound synthesis and sampling*. 2nd Edition. Burlington: Focal Press.
- Schön, D., 2003. *The reflective practitioner: how professionals think in action*. New York: Basic Books.
- Schwartz, P. and Asher, J., 2010. *Composing With Sample Libraries: Make it “Real” or Make it ‘Good’?* [online]. Available from:
<http://www.filmmusicmag.com/?p=6145> [Accessed 30 January 2020].
- Schwarz, D., 2004. *Data-driven concatenative sound synthesis* [online]. Thesis (PhD). Ircam - Centra Pompidou. Available from:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.418.7880> [Accessed 13 November 2016].
- Schwarz, D., 2006. Concatenative Sound Synthesis: Early Years. *Journal of New Music Research*, 35 (1), 3-22.
- Schwarz, D., Beller, G., Verbrugge, B. and Britton, S., 2006. Real-Time Corpus-based Concatenative Synthesis with CataRT [online]. *Proceedings of the 9th Int. Conference on Digital Audio Effects (DAFX-2006)*, Montreal 18-20 September 2006. Montreal: McGill University. Available from: <https://hal.archives-ouvertes.fr/hal-01161358/> [Accessed 23 November 2016].

- Schwarz, D., 2014. *CataRT* [online]. Available from: <http://imtr.ircam.fr/imtr/CataRT> [Accessed 21 November 2016].
- Seashore, C., 1938. *Psychology of music*. New York: Dover Publications.
- Serra, X., 1989. *A System for Sound Analysis/ Transformation/ Synthesis based on a Deterministic plus Stochastic Decomposition*. Thesis (PhD). Stanford University.
- Serra, X., Magas, M., Benetos, E., Chudy, M., Dixon, D., Flexer, A., Gómez, E., Gouyon, F., Herrera, P., Jorda, S., Payuvi, O., Peeters, G., Schlüter, J., Vinet, H., and Widmer, G., 2013. *Roadmap for Music Information ReSearch*. Creative Commons BY-NC-ND 3.0 license.
- Shoop, J., 2013. *Advantages and Disadvantages of Object-Oriented Programming (OOP)* [online]. Available from: <https://resources.saylor.org/wwwresources/archived/site/wp-content/uploads/2013/02/CS101-2.1.2-AdvantagesDisadvantagesOfOOP-FINAL.pdf> [Accessed 02 September 2019].
- Sloboda, J. A., 1986. *The Musical Mind: The Cognitive Psychology of Music*. Oxford: Clarendon Press.
- Smith, J. O., 1991. Waveguide simulation of non-cylindrical acoustic tubes. *In: Proceedings of the 1991 International Computer Music Conference*, San Francisco: International Computer Music Association, 304-307.
- Smith, J. O., 2002. Viewpoints on the History of Digital Synthesis [online]. *In: Proceedings Int. Computer Music Conference (ICMC-91)*, Montreal October 1991. Montreal : McGill University. Available from: <https://pdfs.semanticscholar.org/f1dd/26e8a6ed7273ec2732eead94b0ea4063cdf.pdf> [Accessed 2 November 2016].

- Smith, J. O., 2010. *Physical audio signal processing* [online]. Available from:
<http://ccrma.stanford.edu/~jos/pasp/> [Accessed 3 November 2016].
- Stanton, N. A., Salmon, P. M., Rafferty, L. A., Walker, G. H., Baber, C., and Jenkins, D. P., 2013. *Human Factors Methods: A Practical Guide for Engineering and Design*. 2nd Edition. Boca Raton: CRC Press.
- Strong, A. and Karplus, K., 1983. Digital Synthesis of Plucked-String and Drum Timbres. *Computer Music Journal* , 7 (2) , 43-55.
- Sudnow, D., 1993. *Ways of the Hand: The Organization of Improvised Conduct*. Massachusetts: The MIT Press.
- Swafford, J., 2003. *More Sound!* [online]. Available from:
<https://www.theguardian.com/music/2003/mar/15/classicalmusicandopera.artsfeatures> [Accessed 6 May 2019].
- Taylor, T. D., 2001. *Strange Sounds: Music, Technology & Culture*. London: Psychology Press Ltd.
- Telestream Inc., 2019. *What is ScreenFlow?* [online]. Available from:
<http://www.telestream.net/screenflow/overview.htm#features> [Accessed 12 June 2019].
- The Interaction Design Foundation, 2014. *Action Research*. 2nd Edition. Tranbjerg: The Interaction Design Foundation.
- Tzanetakis, G., 2016. Natural Human-Computer Interaction with Musical Instruments. *In: Politis, D., Tsalighopoulos, M. and Iglezakis, I., eds. Digital Tools for Computer Music Production and Distribution*. Hershey: IGI Global, 116-136.
- Úlfarsson, H., 2018. The halldorophone: The ongoing innovation of a cello-like drone instrument. *In: Proceedings of the International Conference on New Interfaces for Musical Expression*, Blacksburg, Virginia, USA, 269-274.

- Úlfarsson, H., 2019. Feedback Mayhem: Compositional affordances of the halldorophone discussed by its Users. *In: Proceedings of the 2019 International Computer Music Conference*, New York: ICMC/NYCEMF [online]. Available from: https://www.halldorophone.info/public/img/FeedbackMayhem_ICMC_2019.pdf [Accessed 26 June 2020].
- Vaggione, H., 2001. Some Ontological Remarks about Music composition process. *Computer music journal*, 25 (1), 54-61.
- Välimäki, V. and Takala, T., 1996. Virtual musical instruments - Natural sound using physical models. *Organised Sound*, 1 (2), 75-86.
- Venners, B., 2003. *The Making of Python* [online]. Available from: <http://www.artima.com/intv/pythonP.html> [Accessed 18 December 2016].
- VERBI Software GmbH, 2019. *What is MAXQDA?* [online]. Available from: <https://www.maxqda.com/> [Accessed 12 June 2019].
- Vienna Symphonic Library, 2002. *Company: Vienna Symphonic Library* [online]. Available from: <https://www.vsl.co.at/en/Company> [Accessed 15 June 2019].
- Waisvisz, M., 1999. *Gestural Round Table* [online]. Amsterdam: STEIM Writings. Available from: <http://steim.org/steim/texts.php?id=4> [Accessed 20 March 2020].
- Wallmark, Z., 2007. *Making Music in the Digital Age: How technological developments shape the way we create and listen to music* [online]. Thesis (Master of Arts). University of Oregon. Available from: https://s3.amazonaws.com/academia.edu.documents/33238887/Wallmark_MA_Thesis.pdf?response-content-disposition=inline%3B%20filename%3DMaking_Music_in_the_Digital_Age_

How_Tech.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20200210%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20200210T161004Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=7ba5f42a79adaeff98d3957d0f2241fb1d5ae9cd29cef52942a3789aa609a092 [Accessed 27 October 2018].

Walzer, D. A., 2017. Transformational Leadership considerations in Music Technology. *Journal of Performing Arts Leadership in Higher Education*, 8 (3), 27-40.

Weeks, D., 2016. *Jazz and the phonograph* [online].

Available from: <https://edisonpapers.wordpress.com/2016/05/12/jazz-and-the-phonograph/> [Accessed 14 August 2019].

White, E. W., 1979. *Stravinsky: The Composer and His Works*. 2nd Edition. Oakland: The University of California press.

Winkler, T., 2001. *Composing Interactive Music: Techniques and Ideas Using Max*. Cambridge: MIT Press.

Winner, R., 1988. *The role of concurrent engineering in weapons system acquisition*. Alexandria: Institute for Defence Analysis.

Wishart, T. and Vassilandonakis, Y., 2009. An Interview with Trevor Wishart. *Computer Music Journal*, 33 (2), 8-23.

World Intellectual Property Organization, 2016. *Understanding Copyright and Related Rights*. 2nd Edition. Geneva: World Intellectual Property Organization.

Zagal, J., Björk, S. and Lewis, C., 2013. Dark patterns in the design of games. In: *Proceedings of the 8th International Conference on the Foundations of Digital Games, FDG 2013*, Chania, Crete, Greece, May 14-17, 2013. Society for the Advancement of the Science of Digital Games. 39-46.

Zattra, L. and Donin, N., 2016. A questionnaire-based investigation of the skills and roles of Computer Music Designers. *Musicae Scientiae*, 20 (3), 436-456.

Zimmerman, E., 2003. Play as Research The iterative design process. *In*: Laurel, B., ed. *Design Research: Methods and Perspectives*. Cambridge: Design Research, MIT Press, 176-184.

8. Appendices

TABLE OF CONTENTS

8.1 CONTENTS OF THE USB FLASH DRIVE.....	9
8.1.1 The Audio Files	9
8.1.2 The Video Files.....	9
8.2 THE TECHNICAL DESIGN OF THE FPS.....	10
8.2.1 Selecting a synthesis method	10
8.2.2 Selecting a processing environment	16
8.2.3 Selecting a user interface controller environment	17
8.3 OPEN CODING, AXIAL CODING, AND SELECTIVE CODING.....	20
8.3.1 Open Coding.....	20
8.3.2 Axial coding.....	22
8.3.3 Selective coding.....	23
8.4 FIRST ITERATIVE DESIGN PROCESS: THE ALPHA.....	25
8.4.1 Defining the problem	25
8.4.2 Designing the FPS	25
8.4.3 Composing and observing	35
8.4.4 Summary of the audio-visual analysis	36
8.4.5 Evaluation	36
8.5 SECOND ITERATIVE DESIGN PROCESS: THE BETA	40
8.5.1 Defining the problem	40
8.5.2 Designing the FPS	41
8.5.3 Composing and observing	46
8.5.4 Summary of the audio-visual analysis	47
8.5.5 Evaluation	48
8.6 THE THIRD ITERATIVE DESIGN PROCESS: THE GAMMA	53
8.6.1 Defining the problem	53
8.6.2 Designing the FPS	54
8.6.3 Composing and observing	59

8.6.4 Summary of the audio-visual analysis.....	60
8.6.5 Evaluation.....	61
8.7 FOURTH ITERATIVE DESIGN PROCESS: THE DELTA	66
8.7.1 Defining the problem.....	66
8.7.2 Designing the FPS	67
8.7.3 Composing and observing	76
8.7.4 Summary of the audio-visual analysis.....	76
8.7.5 Evaluation.....	77
8.8 FIFTH ITERATIVE DESIGN PROCESS: THE OMEGA.....	83
8.8.1 Defining the problem.....	83
8.8.2 Designing the FPS	83
8.8.3 Composing and observing	88
8.8.4 Summary of the audio-visual analysis.....	89
8.8.5 Evaluation.....	90
8.9 STEP-BY-STEP MANUALS	94
8.9.1 How to set up Cubase on a Mac computer	94
8.9.2 How to set up the FPS-UIC on an external iPad tablet	99
8.9.3 How to begin a composition with the aid of the FPS-UIC and the FPS-VI.....	103
8.9.4 How to begin a composition with the aid of the FPS-UIC and virtual strings.....	123
8.9.5 How to begin a composition with the aid of the FPS-UIC and virtual strings part 1: select the default settings and play a chord.	140
8.9.6 How to begin a composition with the aid of the FPS-UIC and virtual strings part 2: select a different scale and inversion and play different chords.....	145
8.9.7 How to begin a composition with the aid of the FPS-UIC and virtual strings part 3: select different homophonic voicings.....	152
8.9.8 How to begin a composition with the aid of the FPS-UIC and virtual strings part 4: modulate between different scales.....	160
8.10 TRANSFER OF RIGHTS AGREEMENT	170

8.11 ABOUT THE AUTHOR 172

LIST OF FIGURES

FIGURE 1. CONTENTS OF USB FLASH DRIVE	9
FIGURE 2. THE MAIN CATEGORIES OF DIGITAL SYNTHESIS TECHNIQUES.	11
FIGURE 3. THE INFORMATION STREAM OF THE FPS-VI _{v1}	26
FIGURE 4. NICO MAAS PLAYING THE ELECTRIC GUITAR.	27
FIGURE 5. A PROGRAMMED AUTOMATOR WORKFLOW THAT REORGANISES THE AUDIO UNITS.	31
FIGURE 6. THE INTERNAL MAPPING CONSTRUCTION OF THE FPS-VI GUI.	32
FIGURE 7. THE FOUR REGIONS OF THE FPS-VI _{v1} GUI.	33
FIGURE 8. THE INFORMATION STREAM OF THE FPS-VI _{v2} AND THE FPS-UIC _{v1}	41
FIGURE 9. TWO RENDERINGS OF TWO DIFFERENT SAMPLE SYNTHESIS ENGINES.	43
FIGURE 10. THE THREE REGIONS OF THE FPS-VI _{v2} GUI.	44
FIGURE 11. THE FPS-UIC _{v1}	45
FIGURE 12. CONTROLLING THE FPS-UIC _{v1} AND THE MIDI KEYBOARD CONTROLLER.	51
FIGURE 13. THE INFORMATION STREAM OF THE FPS-UIC _{v2}	54
FIGURE 14. THE SIX REGIONS OF THE FPS-UIC _{v2}	56
FIGURE 15. THE FOUR REGIONS OF THE ROLI LIGHTPAD CONTROLLER.	58
FIGURE 16. THE INFORMATION STREAM OF THE FPS-VI _{v4} AND THE FPS-UIC _{v3}	67
FIGURE 17. THE RELATIONSHIP BETWEEN STOCHASTIC AUDIO MATERIAL AND SINGLE NOTES.	68
FIGURE 18. THE FOUR REGIONS OF THE FPS-VI _{v4} GUI.	69
FIGURE 19. THE EIGHT REGIONS OF THE FPS-UIC _{v3}	71
FIGURE 20. THE MOVEMENT OF THE E CHORD WITHIN THE CYCLE OF FIFTH.	74
FIGURE 21. THE INFORMATION STREAM OF THE FPS-UIC _{v4}	84
FIGURE 22. THE FOURTH VERSION OF THE USER INTERFACE CONTROLLER.	85
FIGURE 23. DROP VOICING EXAMPLE.	86
FIGURE 24. ALL VOICING, INCLUDING FOUR DIFFERENT INVERSIONS (E MIN ⁷).	87
FIGURE 25. THE TRANSMITTED MUSICAL NOTES OF A DROP2+L2+4 VOICING TO THE DAW.	88
FIGURE 26. START CUBASE.	95
FIGURE 27. OPEN THE STEINBERG HUB.	95
FIGURE 28. SELECT 'DESKTOP' AS THE LOCATION AND OPEN A NEW PROJECT.	96

FIGURE 29. SAVE A NEW PROJECT.....	97
FIGURE 30. GIVE THE CUBASE PROJECT A TITLE.....	98
FIGURE 31. CONNECT THE USB CABLE TO BOTH THE COMPUTER AND TABLET.....	100
FIGURE 32. OPEN THE LEMUR APPLICATION ON THE TABLET.	100
FIGURE 33. IMPORT THE FPS-UIC INTO THE LEMUR APPLICATION ON THE TABLET.....	101
FIGURE 34. OPEN THE FPS-UIC.	101
FIGURE 35. THE LOADED FPS-UIC.	102
FIGURE 36. ADD INSTRUMENT TRACK.....	105
FIGURE 37. ADD REAKTOR 6 AS VIRTUAL INSTRUMENT TRACK.	106
FIGURE 38. SELECT THE FPS-VI ENSEMBLE FILE.	107
FIGURE 39. THE FPS-VI IS LOADED INTO CUBASE.	107
FIGURE 40. CHANGE THE NAME OF THE INSTRUMENT TRACK.....	108
FIGURE 41. CREATE A RHYTHMIC NOTE WITHIN THE SEQUENCER WINDOW.....	108
FIGURE 42. ADJUST THE LENGTH OF A NOTE WITHIN THE SEQUENCER WINDOW.	109
FIGURE 43. CREATE A RHYTHMIC PHRASE INCLUDING 16 TIMES AN 8 TH NOTE.....	110
FIGURE 44. PROGRAM DIFFERENT ARTICULATIONS TO CHANGE THE OVERALL LENGTH.....	111
FIGURE 45. PROGRAM THE RHYTHMIC PATTERN OF ‘PHRASE 2’.....	112
FIGURE 46. PROGRAM DIFFERENT ARTICULATIONS AND SAVE AS ‘PHRASE 2’.	114
FIGURE 47. ENABLE THE RECORDING FUNCTION IN CUBASE.	115
FIGURE 48. SELECT A MUSICAL SCALE.	116
FIGURE 49. ACTIVATE THE FPS-VI SYNTHESIS ENGINE, AND RECORD ‘PHRASE 1’ WITHIN CUBASE.	117
FIGURE 50. SWITCH TO PRESET ‘PHRASE 2’.....	118
FIGURE 51. POSITION THE CURSOR ON MEASURE 19.....	118
FIGURE 52. ACTIVATE THE FPS-VI SYNTHESIS ENGINE, AND RECORD ‘PHRASE 2’ WITHIN CUBASE.	119
FIGURE 53. SHORTEN MIDI EVENT OF ‘PHRASE 2’.....	120
FIGURE 54. SHORTEN MIDI EVENT OF ‘PHRASE 1’.....	120
FIGURE 55. MOVE MIDI EVENT OF ‘PHRASE 2’ TO MEASURE 19.....	121
FIGURE 56. ADD POST-PROCESSING EFFECT.	122
FIGURE 57. ADD INSTRUMENT TRACK.....	125
FIGURE 58. ADD NINE DIFFERENT KONTAKT 5 VIRTUAL INSTRUMENT TRACKS.....	126

FIGURE 59. OPEN KONTAKT IN THE FIRST INSTRUMENT TRACK.....	126
FIGURE 60. SELECT VIOLINS 1 ESSENTIAL.NKI FILE.....	127
FIGURE 61. CHANGE THE NAME OF THE INSTRUMENT TRACK.....	128
FIGURE 62. ROUTE THE INSTRUMENT TRACK TO THE FPS-UIC.....	129
FIGURE 63. MOVE SELECTED TRACKS TO A NEW FOLDER.....	131
FIGURE 64. RENAME THE GROUP FOLDER TO 'STRING ENSEMBLE'.....	132
FIGURE 65. SELECT A MUSICAL SCALE, VOICING, AND INVERSION.....	133
FIGURE 66. SELECT A CHORD.....	135
FIGURE 67. ENABLE RECORDING FUNCTION IN CUBASE.....	135
FIGURE 68. THE TWELVE CHORD OPTIONS OF THE FPS-UIC.....	137
FIGURE 69. CHANGE CHORD WHILE RECORDING A CHORD PROGRESSION.....	137
FIGURE 70. THE MIDI EDITOR WINDOW INCLUDING THE TWO RECORDED HOMOPHONIC CHORDS.....	138
FIGURE 71. SELECT A CHORD.....	142
FIGURE 72. ENABLE RECORDING FUNCTION IN CUBASE.....	143
FIGURE 73. THE SEVEN CHORD OPTIONS OF THE FPS-UIC.....	143
FIGURE 74. SELECT A MUSICAL SCALE, VOICING, AND INVERSION.....	147
FIGURE 75. SELECT A CHORD.....	148
FIGURE 76. ENABLE RECORDING FUNCTION IN CUBASE.....	149
FIGURE 77. THE TWELVE CHORD OPTIONS OF THE FPS-UIC.....	149
FIGURE 78. CHANGE CHORD WHILE RECORDING CHORD PROGRESSION.....	150
FIGURE 79. THE RECORDED MIDI NOTES WITHIN THE MIDI EDITOR OF CUBASE.....	150
FIGURE 80. SELECT A CHORD.....	154
FIGURE 81. THE SEVEN POSSIBLE VOICING OPTIONS.....	154
FIGURE 82. ALL SEVEN DROP VOICINGS WITH DIFFERENT INVERSIONS.....	156
FIGURE 83. ENABLE THE RECORDING FUNCTION IN CUBASE.....	157
FIGURE 84. PLAY THE CHORD AND SELECT A DIFFERENT VOICING WHILE RECORDING.....	158
FIGURE 85. THE MIDI EDITOR WINDOW INCLUDING THE TWO RECORDED HOMOPHONIC CHORDS.....	159
FIGURE 86. SELECT A CHORD.....	162
FIGURE 87. ENABLE THE RECORDING FUNCTION IN CUBASE.....	163
FIGURE 88. THE TWELVE CHORD OPTIONS OF THE FPS-UIC.....	163

FIGURE 89. THE NUMBERS DISPLAY HOW MANY POSITIONS THE SCALE MODULATES.....	165
FIGURE 90. CHANGE THE CHORD PROGRESSION AND MODULATE TO ANOTHER SCALE.....	166
FIGURE 91. SELECT THE FIRST CHORD WITHIN THE FIRST CHORD ROW.....	167
FIGURE 92. CHANGE THE CHORD PROGRESSION AND MODULATE TO ANOTHER SCALE.....	167
FIGURE 93. CHANGE CHORD WHILE RECORDING THE CHORD PROGRESSION.....	168

LIST OF TABLES

TABLE 1. OVERVIEW OF ALL USED CODES.....	20
TABLE 2. OVERVIEW OF THE LAST 10 OPEN CODES.	21
TABLE 3. EXAMPLE OF THE AXIAL CODING PROCESS.....	22
TABLE 4. SUMMARY OF ALL RECORDED SINGLE NOTE PHRASES.	30
TABLE 5. AN OVERVIEW OF ALL SELF-STUDY VIDEOS OF THE FIRST COMPOSITION.....	35
TABLE 6. AN OVERVIEW OF RECORDED RHYTHMIC VALUES CONVERTED INTO MILLISECONDS.....	42
TABLE 7. AN OVERVIEW OF ALL SELF-STUDY VIDEOS OF THE SECOND COMPOSITION.....	47
TABLE 8. AN OVERVIEW OF ALL SELF-STUDY VIDEOS OF THE THIRD COMPOSITION.....	60
TABLE 9. OVERVIEW OF ALL PITCH TRANSPOSITIONS.	70
TABLE 10. AN OVERVIEW OF THE SEVEN 7 TH CHORDS THAT OCCUR WITHIN A MUSICAL SCALE.	72
TABLE 11. AN OVERVIEW OF ALL SELF-STUDY VIDEOS OF THE FOURTH COMPOSITION.....	76
TABLE 12. AN OVERVIEW OF ALL SELF-STUDY VIDEOS OF THE FIFTH COMPOSITION.....	89
TABLE 13. SELECTED ARTICULATIONS OF ‘PHRASE 1’.....	111
TABLE 14. SELECTED ARTICULATIONS OF ‘PHRASE 2’.....	113
TABLE 15. AN OVERVIEW OF THE DIFFERENT INSTRUMENT TRACKS.....	128
TABLE 16. AN OVERVIEW OF DIFFERENT NAMES FOR DIFFERENT TRACKS.	128
TABLE 17. AN OVERVIEW OF THE ROUTING STRATEGY.....	130

8.1 Contents of the USB Flash Drive

The USB Flash drive includes the following audio-visual content: (1) Set of Audio Files; (2) Set of Video Files (Figure 1).

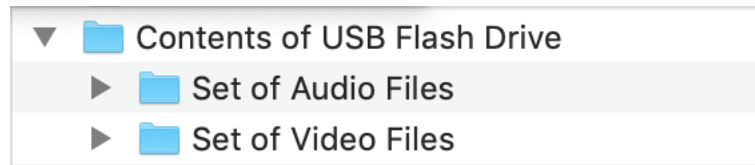


Figure 1. Contents of USB Flash Drive

8.1.1 The Audio Files

The USB Flash Drive contains audio files in MPEG-3 (.mp3) format, which ensures cross-platform compatibility. Open the audio files with any audio software that allows MPEG-3 playback. The folder ‘Set of Audio Files’ contains the following five MPEG-3 files: (1) 1_The Alpha.mp3; (2) 2_The Beta.mp3; (3) 3_The Gamma.mp3; (4) 4_The Delta.mp3; (5) 5_The Omega.mp3.

8.1.2 The Video Files

The USB Flash Drive contains audio files in MPEG-4 (.mp4) format, which ensures cross-platform compatibility. Open the video files with any audio software that allows MPEG-4 playback. The folder ‘Set of Video Files’ contains the following eight MPEG-4 files: (1) 8.9.1 How to set up Cubase on a Mac computer.mp4; (2) 8.9.2 How to set up the FPS-UIC on an external iPad tablet.mp4 (3) 8.9.3 How to begin a composition with the aid of the FPS-UIC and the FPS-VI.mp4; (4) 8.9.4 How to begin a composition with the aid of the FPS-UIC and virtual string ensembles.mp4; (5) 8.9.5 How to begin a composition with the aid of the FPS-UIC and virtual string ensembles part 1.mp4 (6) 8.9.6 How to begin a composition with the aid of the FPS-UIC and virtual string ensembles part 2.mp4; (7) 8.9.7 How to begin a composition with the aid of the FPS-

UIC and virtual string ensembles part 3.mp4; (8) 8.9.8 How to begin a composition with the aid of the FPS-UIC and virtual string ensembles part 4.mp4.

8.2 The technical design of the FPS

This section outlines the technical requirements of the FPS. Sub-section 8.2.1 discusses the selected synthesis method, 8.2.2 outlines the selected processing environment, and 8.2.3 describes the selected user interface controller in more detail.

8.2.1 Selecting a synthesis method

Virtual instruments (VIs) use synthesis methods to generate audio. Synthesis methods can be grouped into different categories depending on the type of sound generator and technique used to produce waveforms. Cook (2002) and Smith (2010) both wrote comprehensive books on physical modelling and audio signal processing, including a complete taxonomy of all digital synthesis techniques. Figure 2 provides an overview of the main categories of digital synthesis techniques and outlines three methods that are suitable for designing a digital version of an electric guitar. This figure demonstrates where the techniques originated and how they relate to other methods.

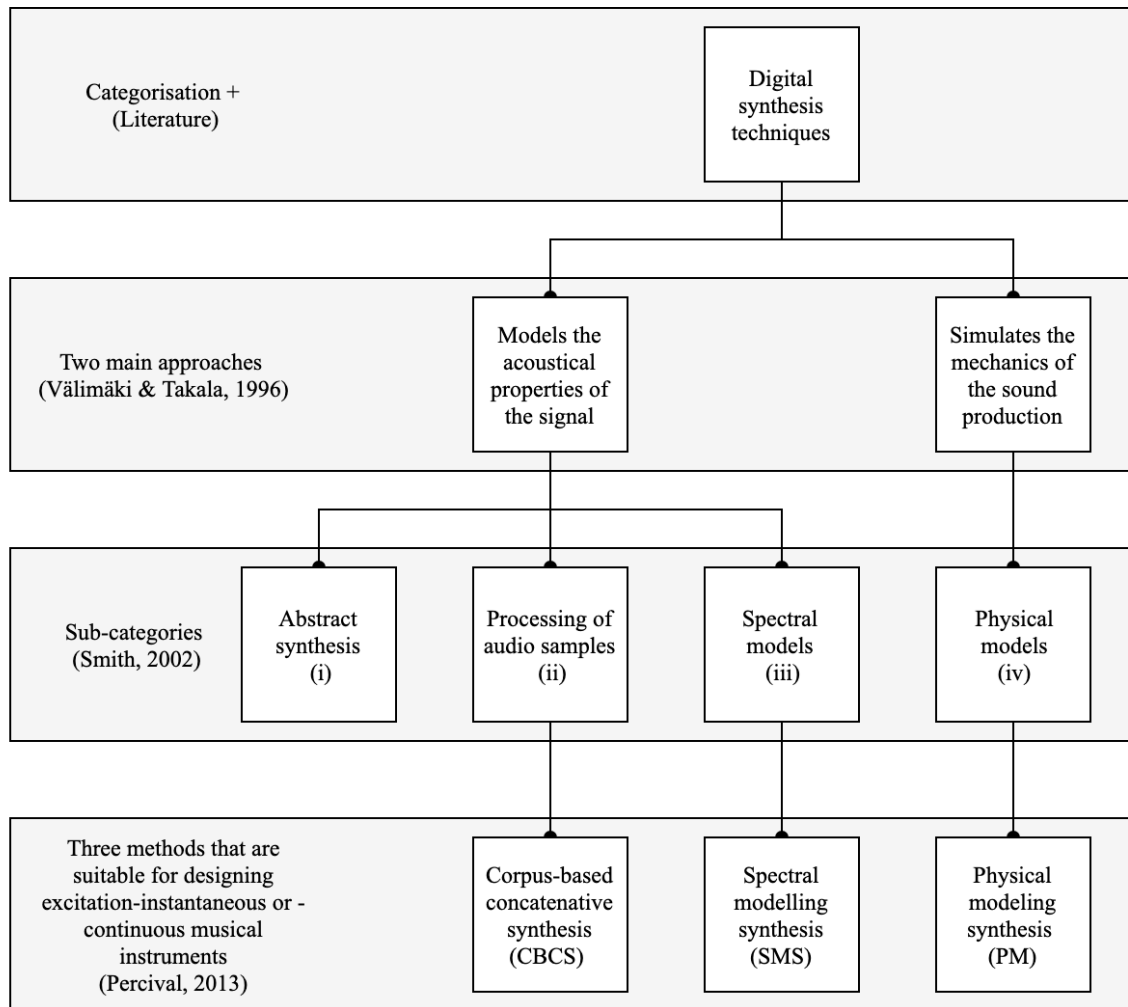


Figure 2. The main categories of digital synthesis techniques.

Figure 2 illustrates two main digital synthesis approaches (Välämäki and Takala 1996). The first approach models the acoustical properties of the signal, while the second approach simulates the mechanics of sound production. Smith (2002) classifies the first approach into three sub-categories: (i) abstract synthesis, (ii) processing of recorded samples, and (iii) spectral models. These three sub-categories include various synthesis methods. The additive synthesis method, for example, is based on the principle of adding sine waves to construct a desired spectrum. Another example is the subtractive synthesis method. The linear filtering of an input signal forms the principle of the subtractive synthesis method. Some frequency components are attenuated, and

other parts are emphasised. Another synthesis method is frequency modulation synthesis (Chowning 1973), in which one oscillator (the modulator) controls the frequency of another one (the carrier).

Another example is the waveshaping synthesis technique (Arfib and Le Brun 1979). This technique follows the principle of changing an original waveform by responding to its amplitude. The second approach includes the physical model (iv) as a synthesis approach (Välimäki and Takala 1996). Some relevant forms of physical model synthesis are, e.g. Karplus-Strong (Strong and Karplus 1983), Mosaïcing (Adrien and Morrison 1990), Waveguide synthesis (Smith 1991), and physical modelling (Demoucron 2008). According to Percival (2013), three synthesis methods are suitable for designing a digital version of an acoustic instrument: corpus-based concatenative sound synthesis (CBCS), the spectral modelling synthesis (SMS), and physical modelling (PM). Within the restrictions of this study and the requirements for the FPS, CBCS is an appropriate method to simulate the electric guitar sound.

CBCS is an appropriate method to simulate an electric guitar because it uses sampling as the synthesis method. Furthermore, Roads (1988) subdivides sample-based synthesis into CBCS and granular synthesis. Granular synthesis is comparable to CBCS; however, granular synthesis does not analyse the given sample database. The selected samples and unit sizes are determined arbitrarily.

There is a significant volume of published studies and tests describing the implementation of CBCS. Examples are the singing voices by Bonana and Serra (2007), the real-time CBCS with CataRT by Schwarz et al. (2006), the fine-tuned control of CBCS with CataRT using the Bach Library for MAX by Einbond et al. (2014), and the reconstructive phrase modelling by Lindemann (2007). A significant contribution to the topic is Schwarz's PhD dissertation (2004). According to Schwarz, CBCS is a

promising method of musical sound synthesis. The main application of the CBCS method is high-level instrument synthesis, in which selected samples from a matching context synthesise natural-sounding transitions. The CBCS method contains recorded audio samples that include nuances of the sound of the instrument.

CBCS is an appropriate method to synthesise an electro-acoustic instrument because various environments are available that support the implementation of the different tasks of CBCS. The development of a VI using the CBCS method includes five different tasks: (1) analysis; (2) database; (3) target; (4) selection; (5) synthesis.

In the first task, the recorded samples are analysed and then segmented into audio units. The segmentation can be executed automatically (using, e.g. low-, mid- or high-level feature descriptors⁷³), manually (supplying external metadata by the developer), or arbitrarily. Second, the database task provides different kinds of data information to the segmented audio units. This data information includes, for example, file references, unit descriptors, and the relationship between them. Third, the target task contains a given set of instructions that describe the synthesised sequence(s), and a symbolic MIDI score generates the instructions. However, these instructions provided

⁷³ Low-level features relate directly to signal properties and are computed according to simple formulae. Examples are the zero-crossing rate, spectral centroid, and global energy of the signal. More complex computations are characterised by mid-level features, where algorithms employed are not always successful at producing the intended results. Examples are pitches and onset times of notes. High-level features correspond to the terms and concepts used by musicians or listeners to describe various aspects of music. Examples are genre, tonality, rhythm, harmony, and mood (Serra et al. 2013).

by a MIDI score do not include expressive information⁷⁴. Fourth, the selection task chooses the audio units that best match the given target descriptors. The selected audio units that are used to synthesise are called the corpus. Fifth, the synthesis task concatenates the selected audio units. The audio units are placed and played-back at the time given by the target (musical descriptions like pitch, rhythm, dynamic, or articulation), or they are placed and played-back based on their initial duration.

In contrast to CBCS, spectral modelling synthesis (SMS) by Serra (1989) is another synthesis method used to simulate the sound of an electro-acoustic instrument. The SMS method decomposes recorded audio into a deterministic and a stochastic component. The deterministic component includes “a series of sinusoids that are described by amplitude and frequency functions” (Serra 1989, p. iii), while the stochastic component includes “a series of magnitude-spectrum envelopes that function as a time-varying filter excited by white noise” (Serra 1989, p. iii). Subsequently, the deterministic part is analysed, re-synthesised, and subtracted from the recorded audio. In the last stage, musical transformations can be applied to both components, and then the results are synthesised. SMS has engendered several volumes of published studies and tests. An example is the non-intrusive system for measuring violin performance data by Pérez (2009). A working implementation of Pérez’s system was developed in the Max/MSP processing environment by Pérez and Bonada (2010). However, there are no commercial applications that can be tested and evaluated within a digital audio

⁷⁴ See 3.3.1 for more information concerning the requirements for the FPS.

workstation (DAW) environment. The compatibility with a DAW is a crucial requirement for designing the FPS.

In contrast to CBCS, PM synthesis is another synthesis method of simulating the sound of an electro-acoustic instrument. A computer realises PM with mathematical and physical formulae, and it generates audio through modelling the physical behaviour and structure of a musical instrument (Välimäki and Takala 1996). PM synthesis contains a wide range of synthesis approaches and tests; for example, Demoucron (2008) developed a system of bowed string instruments based on PM synthesis. Graham Keith Percival (2013) expanded Demoucron's work and presented the so-called Vivi (the Virtual Violinist). The Vivi system receives sheet music, translates music notation into physical actions, creates audio with mathematical simulations and then uses feedback control with machine learning to alter the physical parameters if necessary (Percival 2013).

Moreover, Percival (2013) criticised CBCS and SMS because both synthesis methods require a massive database of recorded samples. The database relies on CPU intensive methods which would hinder other developers and researchers. However, novel consumer computer systems are capable of performing CPU intensive methods. The musical output produced by a physical model (e.g. Vivi) do not convince as it excludes the expressiveness of an instrumentalist, and it creates an unnatural and static sound. The authenticity of the sound of the musical instrument is an essential requirement for designing the FPS-VI.

8.2.2 Selecting a processing environment

Software processing environments realise the corpus-based concatenative synthesis (CBCS) method. Within the restrictions of this study and the requirements for the ‘Flexible Phrase System’ (FPS), Reaktor 6 developed by Native Instruments is a suitable signal processing environment to realise the corpus-based concatenative synthesis (CBCS) method and the graphic user interface (GUI).

Besides Reaktor 6, Kontakt 5 developed by Native Instrument is another processing environment that realises the CBCS method and GUI. Kontakt 5 is a sample-based synthesis environment, and it offers programmers the ability to customise digital instruments with the help of the Kontakt scripting (KSP) language. The interface options of Kontakt 5 are comparable to Reaktor 6. It is compatible with most existing DAWs, including a programmable GUI, a computer mouse, computer keyboard, MIDI keyboard, and the customised external user interface controller control the GUI. Within the DAW engine, the Kontakt 5 instrument receives the target as a MIDI file. The Kontakt 5 instrument analyses the MIDI file in offline modus, and the selector selects audio units that fit best to the given MIDI file. The audio units are then concatenated and synthesised within the DAW engine. Unfortunately, the KSP language does not allow for MIDI files (the target) to be played and manipulated in real-time. The real-time playback and target manipulation are essential requirements for the development of the FPS-VI.

Besides Reaktor 6, Max/MSP is another processing environment that realises the CBCS method and GUI. CataRT, a real-time CBCS (Einbond et al. 2014), is a modular system for Max/MSP. CataRT plays audio units from an extensive database of segmented and descriptor-analysed sounds according to the target position in the descriptor space (Schwarz 2014). Incoming MIDI note numbers allow audio units and

system-selected sounds to be played in real-time. Max/MSP offers the full capacity of CBCS, but it is not compatible with DAW environments like Cubase. The compatibility with Cubase is necessary because the original musical compositions are realised within this DAW.

Python is a programming language that can realise the CBCS method and a GUI. Python, developed by Guido van Rossum at the University of Amsterdam, is an interpretive, interactive object-oriented programming (OOP) language (Venners 2003). The programming language offers a full capacity of CBCS (Bilous 2016) and the development of GUIs that are executable in a DAW. However, developing a virtual instrument in Python requires a steep learning curve, and it shifts the focus onto programming the virtual instrument rather than on the development of musical compositions. Python does not provide the automatic integration of the virtual instrument within a DAW. Furthermore, it also does not provide a processing environment that links to a GUI automatically; consequently, the integration in a DAW and the GUI needs to be programmed from the ground up. Unfortunately, Python does not fulfil the requirements of the FPS and the research restriction of this study.

8.2.3 Selecting a user interface controller environment

As music and human-computer interaction (M&HCI) focuses on the interfaces between users and computers, user interface controllers (UICs) are irreplaceable, as they embody the space where interactions between humans and machines occur. According to Karray (2008), an interface mainly relies on the number and diversity of its inputs and outputs, and it enables users to interact with a computer via this interface. Interfaces are divided into three categories: (1) visual-based; (2) audio-based; (3) sensor-based. The interface approach of the FPS includes a sensor-based interface, and it focuses on the use of a

computer mouse, computer keyboard, MIDI keyboard, and customised UIC. Within the restrictions of this study and the requirements for the FPS, the Lemur programming application developed by the company Liine is an appropriate environment to design the customised FPS user interface controller (FPS-UIC) that controls the FPS-VI. The application operates on multi-touch devices, which means that the user controls the Lemur GUI with their fingers. Moreover, all of the graphical objects of the user interface are highly customizable. It enables an iterative design process of the UIC based on the functionalities of the VI. Moreover, Lemur is scriptable and provides advanced compositional and productional controlling. The application has a C-style scripting language.

In contrast to many digital musical instruments (DMI's), the FPS-UIC is not responsible for the realisation of the FPS virtual instrumentalist's musical expression. The FPS-UIC guides compositional choices. The compositional choices focus mainly on choosing harmonic voicings and switching between different musical scales. Furthermore, the expression related parameters of the FPS-VI are tuned automatically, and they are not dependent upon the expressive information given by the multi-touch device of the FPS-UIC. The composer has less or no say in the expressive information on the FPS-VI. However, the tablet computer is not a well-suited controller to perform the expression of virtual instruments that do not tune expression-related parameters automatically because the multi-touch display of a tablet is not pressure sensitive. A MIDI keyboard is still more applicable for sending expressive MIDI information to both the DAW and other virtual instruments because of its expression sensitive keys and modulation wheel.

Alternatives to the Lemur UI are the customizable multi-touch devices such as TouchOSC (Hexler 2008) and Open Stage Control (Doucet 2016). Both applications are

available for iOS and Android, and they include a graphic user interface environment which is easy to interface via MIDI with both the DAW and virtual instrument. Both applications are, comparable to the Lemur application, designed for multi-touch devices, and the user controls the machine with fingers. However, TouchOSC and Open Stage Control, are not scriptable. They also do not provide advanced compositional and productional controlling, which are essential requirements and necessary for the development of the FPS-UIC.

Lemur is an appropriate environment to design the customised FPS-UIC because it is well-suited for use with Reaktor 6. Moreover, the FPS-UIC is not responsible for the realisation of the FPS virtual instrumentalist's musical expression because the FPS-VI tunes expression-related parameters automatically. Alternatives to Lemur exist, but they do not fulfil the requirements of the FPS-UIC.

8.3 Open coding, axial coding, and selective coding

This section describes the data that is open coded, axial coded, and selective coded. The coded data describes the audio-visual recordings of the compositional process. In total, 50 hours of audio-visual observation material was produced and coded. The following table (Table 1) summarises the nine codes that are defined based on 50 hours of audio-visual self-study recordings.

Number	Codename	Coded segments	%	Documents
1	Note input (keyboard)	91	28,00%	20
2	Playback FPS	66	20,31%	15
3	Editing note input (piano roll)	49	15,08%	13
4	Note input (FPS-UIC)	29	8,92%	10
5	FPS	26	8,00%	16
6	Preparation Cubase	18	5,54%	8
7	Structuring composition	17	5,23%	11
8	Preparation composition	16	4,92%	13
9	Note input (piano roll)	13	4,00%	8

Table 1. Overview of all used codes.

8.3.1 Open Coding

In open coding, the audio-visual observation data is broken down into significant concepts. The significant concepts briefly describe the actions. The actions are then categorised based on the human-computer interaction (HCI). All audio-visual self-study observations are broken down into 325 codes. Table 2 illustrates the last 10 open codes of 'The Omega'.

Nr.	Open code	Description	Video file name	Begins	Ends
316	Note input (keyboard)	Creating unisono- octave melody Part 1	190429_The Omega_r1	01:28:30,9	01:43:27,2
317	Playback FPS	Playback everything Melody	190429_The Omega_r1	01:43:29,9	01:44:14,0
318	Note input (keyboard)	Creating unisono- octave melody Part 1	190429_The Omega_r1	01:44:15,8	01:55:59,9
319	Editing note input (piano roll)	Editing Melody Part 1	190429_The Omega_r1	01:56:01,9	01:58:26,4
320	Note input (keyboard)	Creating unisono- octave melody Part 1	190429_The Omega_r1	01:58:28,1	01:59:58,8
321	Editing note input (piano roll)	Editing Melody Part 1	190429_The Omega_r1	01:59:58,8	02:01:33,9
322	Note input (keyboard)	Editing Melody Part 1	190429_The Omega_r1	02:01:35,1	02:04:17,9
323	Playback FPS	Playback Part 1	190429_The Omega_r1	02:04:19,0	02:05:55,7
324	Note input (keyboard)	Adding dynamic Part 4	190429_The Omega_r1	02:05:57,9	02:08:06,2
325	Playback FPS	Playback Part 4	190429_The Omega_r1	02:08:05,8	02:08:57,3

Table 2. Overview of the last 10 open codes.

8.3.2 Axial coding

In axial coding, the significant concepts of the open coding process are reassembled with propositions concerning their relationships with each other. Table 3 illustrates the axial codes of 'The Omega'.

Nr.	Open code	Axial Code
296	Preparation composition	Note input segment 1
297	Playback FPS	
298	Preparation Cubase	
299	Note input (user interface controller)	
300	Editing note input (piano roll)	
301	Note input (keyboard)	
302	Note input (user interface controller)	Note input segment 2
303	Editing note input (piano roll)	
304	Editing note input (piano roll)	Note input segment 3
305	Note input (keyboard)	
306	Note input (user interface controller)	Note input segment 4
307	Editing note input (piano roll)	
308	Playback FPS	
309	Note input (keyboard)	Dynamic segment 3
310	Editing note input (piano roll)	
311	Playback FPS	
312	Note input (keyboard)	
313	Playback FPS	
314	Note input (keyboard)	

315	Editing note input (piano roll)	
316	Note input (keyboard)	Note input segment 1
317	Playback FPS	
318	Note input (keyboard)	
319	Editing note input (piano roll)	
320	Note input (keyboard)	
321	Editing note input (piano roll)	
322	Note input (keyboard)	
323	Playback FPS	
324	Note input (keyboard)	
325	Playback FPS	

Table 3. Example of the axial coding process.

8.3.3 Selective coding

In selective coding, the axial codes are selected based on the research questions of this research. The three selective codes given to the observation data of ‘The Alpha’ are: (1) Composing the melodic progression for the two electric guitars; (2) Composing harmonic building blocks; (3) Programming musical mode and root note switches. The five selective codes given to the observation data of ‘The Beta’ are: (1) Defining the musical form based on a storyline with the aid of the FPS-UIC_{v1}; (2) Choosing a musical phrase for the first electric guitar; (3) Playing MIDI notes on a MIDI keyboard; (4) Performing musical mode and root note changes on the FPS-UIC_{v1}; (5) Improvising with a second electric guitar upon the first electric guitar. The four selective codes given to the observation data of ‘The Gamma’ are: (1) Defining the musical form based on three defined emotional states; (2) Defining root notes and musical modes with the aid

of the FPS-UIC_{v2}; (3) Composing melodic material with the aid of the MIDI keyboard controller; (4) Composing harmonic progressions with the aid of the ROLI Lightpad controller. The three selective codes given to the observation data of 'The Delta' are: (1) Structuring the composition based on the storyline; (2) Performing musical scales and musical modulations with the aid of the FPS-UIC_{v3}; (3) Performing musical melodic and harmonic progressions with the FPS-VI_{v4} and the virtual piano. The three selective codes given to the observation data of 'The Omega' are: (1) Performing homophonic material; (2) Programming expressive; (3) Performing unisono-octave note input.

8.4 First iterative design process: The Alpha

The first iterative design process addresses the production of the first original composition called ‘The Alpha’ and the design of the first version of the FPS-VI_{v1}. The iterative design process recapitulates the defined problem of this research (8.4.1), describes the functional design of the FPS_{v1} (8.4.2), briefly outlines the gathered musical data and observation data (8.4.3), summarises the musical description of ‘The Alpha’, summarises the observation data (8.4.4), and evaluates the analysed data based on the research questions (8.4.5).

8.4.1 Defining the problem

This section briefly recapitulates the defined problem of this research. The initial problem describes the adverse effect of humanising the sound of virtual instruments upon composing original music. Programming musical expressions manually costs a lot of time, interrupts the compositional process, and affects melodic and harmonic choices. Therefore, a new digital musical instrument (DMI) called ‘Flexible Phrase System’ (FPS) was proposed and designed. The FPS tunes expression-related parameters automatically, and it aims to offer an interconnected composition process.

8.4.2 Designing the FPS

The first version of the ‘Flexible Phrase System Virtual Instrument’ (FPS-VI_{v1}) simulates the sound of an electric guitar, and the synthesis engine uses sample synthesis techniques. The FPS-VI_{v1} works within existing digital audio workstations (DAWs). It features a real-time playback engine, and it is compatible with MIDI controllers. Figure 3 illustrates the information stream of the FPS-VI_{v1} and the integration of the software within a DAW.

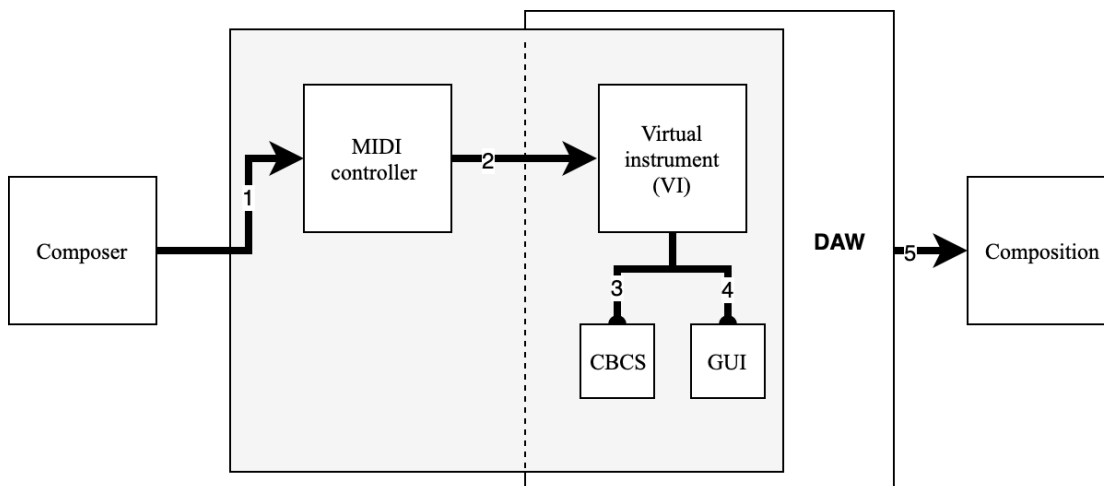


Figure 3. The information stream of the FPS-VI_{v1}.

The composer (1) uses a MIDI keyboard, computer keyboard, and computer mouse to control the FPS-VI_{v1} (2). The DAW connects to the MIDI controller by MIDI mapping strategies, and the FPS-VI_{v1} is integrated within the DAW structure. The FPS-VI_{v1} includes an audio database (CBCS) (3) and a graphic user interface (GUI) (4). The output of the interaction between composer and DAW (5) is an audio rendering of the musical composition, in this case, ‘The Alpha’.

The Reaktor 6 dataflow visual programming language (DFVPL) facilitates the FPS-VI_{v1}, including an audio database and a graphic user interface (GUI). The audio database includes a recorded and edited corpus of electric guitar samples. The corpus-based concatenative sound synthesis (CBCS) technique is used to capture the sound of the electric guitar. The audio database consists of 11408 original single note samples, and these original recordings are called audio units. The audio units keep the advantage of the following expression-related parameters: temporal aspects, dynamic, articulations, pitch, and transitions. Moreover, the audio units include 4 different articulations, 37 different pitch values, and 9 different rhythmic note lengths. Nico Maas (Figure 4), a professional electric guitarist, played the recorded electric guitar. Maas is a

jazz-guitarist who studied electric jazz guitar at the conservatory of Amsterdam. He has a lot of experience doing professional studio recordings and live performances. The recorded electric guitar is a Fender Stratocaster Anniversary 1979, and Maas plays the guitar with a plectrum. Maas plays and records the electric guitar without any effects or external amplifiers, which gives the composer more flexibility in experimenting with the sound of the electric guitar while composing and producing. For example, the composer can switch between using heavily distorted amplifiers, clean effects, re-amp the instrument, or other digital effects.



Figure 4. Nico Maas playing the electric guitar.

The recording session took place at a professional music studio between the 29th of January and the 2nd of February 2018. While recording, Maas and I were in the same room. The moment of recording the audio samples in the recording room determines Maas' musical performance contribution. The musical relationship between me as the music technologist and Maas as the music instrumentalist starts and ends at the moment of the recording of the audio units, which means that I did not ask Maas to participate

while composing the original music; however, I asked Maas to listen to the musical results at the end of the iterative design process (see 8.7.5).

The FPS-VI_{v1} audio database is built with the aid of the corpus-based concatenative sound synthesis (CBCS). The CBCS technique is used to capture the sound of the electric guitar. The digital audio workstation (DAW) Cubase 9.5 is used as both the recording and editing software environment. The electric guitar is recorded with the digital input technique (DI), and the audio recording peripherals used are an Apogee symphony AD/DA converter and the Millennia hv-3c pre-amp. A pre-amp “amplifies the mic’s low output to a voltage level that is easier to process with the rest of the recording equipment” (Huber & Williams 1998, p.111). The Millennia HV-3 amplifier series is known as a neutral and clean pre-amp and is often used for recording sample libraries. “Leading sample library creators, including Vienna Symphonic Library, Garritan Orchestral Strings, Serafine and others, have selected Millennia HV-3 preamplifiers [...] for critical applications” (Millennia Music & Media Systems 2004). The sampling rate and bit depth determine the sample reproduction quality. The sample rate is the number of audio samples executed per second, measured in Hz or kHz (Sweetwater 2013) while bit depth is the number of bits used to carry the data in each sample of audio. It describes the resolution of the sound data that is captured and stored in an audio file (Kefauver & Patschke 2007). The used sample rate is 96.000 kHz, and the selected bit depth is 32 bits. The selected sample rate and bit depth resolution allow for more flexibility when working with the recorded material.

The recorded audio material includes single note phrases. Single note phrases include one pitch value, one rhythmic value, one dynamic value, one articulation, and two-stroke direction movements. The number of played notes within one single note phrase depends on the rhythmic value: the shorter the rhythmic value, the more notes

are recorded. Playing a musical phrase including a sequence of short rhythmic values (e.g. $\frac{1}{16}$ th notes) increases the possibility of the so-called machine gun effect. The so-called machine gun effect is evoked by the lack of sonic variation since the same sample is being repeatedly played in a sequence. It sounds unnatural because no two notes played by an electric guitar, or any electro-acoustic instrument, are going to be precisely the same. A recorded musical note sounds unnatural when played repeatedly. Recording more variations of one note, including the same pitch, rhythmic, dynamic, and articulation parameters, is an efficient solution to reduce the machine gun effect. A musical note including a long rhythmic value (e.g. $\frac{1}{1}$ th or even $\frac{2}{1}$ nd) is less problematic because the excitation of a subsequent note occurs later. However, it is still advisable to record a minimum of two variations of a note that includes the same set of musical parameters.

Moreover, all single note phrases are recorded on 120 beats per minute (bpm), and they are based upon one dynamic value. Maas performed the notes not too hard and not too soft. While experimenting with different dynamics, he listened carefully to the sonic quality of the performed dynamics, and he decided to perform the musical note medium-hard. Furthermore, the audio samples are recorded based upon the pitch position on the first five of the six different guitar strings: the low e-string, a-string, d-string, g-string and b-string. Maas advised not to record the highest e-string as most musical performances are not played on the highest e-string because the majority of electric guitar performances focus on the range between the lowest E2 note up to E5. It is the most solid register on the electric guitar. All notes of the electric guitar are recorded within its context, in particular, the position in a musical phrase and the transition between two adjacent notes. Table 4 summarises all recorded single note phrases.

Rhythmic value	Different positions	E string (E2-D3)	A string (D#3-G#3)	D string (A3-F#4)	G string (G4-A#4)	B string (B4-E5)
2/1	2	22	12	20	8	12
1/1	4	44	24	40	16	24
1/2d	6	66	36	60	24	36
1/2	8	88	48	80	32	48
1/4d	12	132	72	120	48	72
1/4	18	198	108	180	72	108
1/8d	22	242	132	220	88	132
1/8	34	374	204	340	136	204
1/16	66	726	396	660	264	396
Total		1892	1032	1720	688	1032

Table 4. Summary of all recorded single note phrases.

The single note phrases are edited based on the following five steps: (1) Analysing hitpoints⁷⁵; (2) Slicing all audio recordings; (3) Naming each audio unit; (4) Colour-coding each audio unit; (5) Reorganising audio units in different maps. First, all hitpoints on every transient of every musical note are analysed with the help of the VariAudio⁷⁶ function within Cubase. Hitpoints that are not placed properly on a

⁷⁵ Hitpoints mark musically relevant positions in audio files. Cubase can detect these positions and create hitpoints automatically by analysing onsets and melodic changes of the audio (Steinberg Media Company 2012).

⁷⁶ The VariAudio features in Cubase allow [...] to edit pitch, correct timing, and intonation of individual notes in monophonic vocal recordings (Steinberg Media Company 2012).

transient of one note are removed. Second, all audio recordings are sliced in between two hitpoints, and they are exported as unique audio units. Third, each audio unit is re-named. The name includes the following unique file naming scheme: [MIDINote]_[Pitch]_[Dynamic]_[Tempo]-[RoundRobin]. Fourth, each sliced audio unit contains a unique colour-code. The colour-codes refer to the electric guitar string played. Fifth, all audio units are reorganised, and they are moved to different round-robin maps. The segmented audio units are reorganised because the corpus manager of the graphic user interface (GUI) environment is structured and organised based on the round-robin position of an audio unit. The GUI environment is not based on the position of a pitch on an electric guitar string. A simple Automator workflow⁷⁷ within the OSX Automator workflow environment is created for this task (see Figure 5).



Figure 5. A programmed Automator workflow that reorganises the audio units.

The FPS-VI_{v1} GUI connects to the audio database, and the composer controls the audio database through the GUI. The three building blocks given to the internal

⁷⁷ Automator is an Apple macOS application that enables workflows for automating repetitive tasks.

mapping of the GUI environment are: (1) Sequencer; (2) Retrigger; (3) Samples & Sampler (see Figure 6).

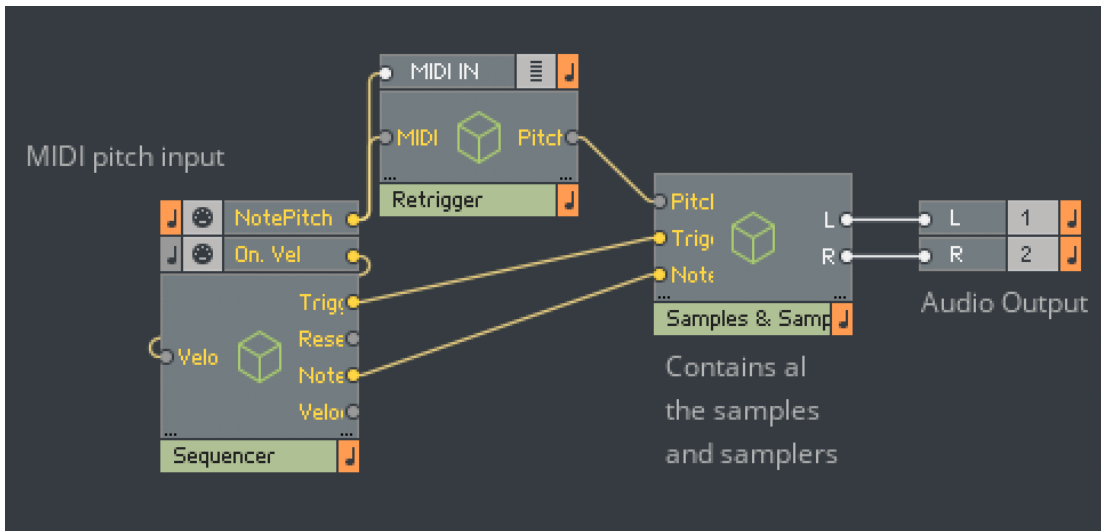


Figure 6. The internal mapping construction of the FPS-VI GUI.

The Sequencer block performs the temporal, dynamic, articulation, and transition information. It ensures that the FPS-VI performs a musical phrase rather than a single note. The GUI sequencer block pre-defines the target, and the MIDI controller controls the pitch values of the target. The pre-defined target includes temporal, dynamic, articulation, and transition information. The retrigger block ensures that a MIDI controller controls the pitch positions within the pre-defined target. The sound units are then selected and synthesised. The retrigger block also ensures that the temporal information of the virtual instrument is linked to the internal temporal information of the DAW MIDI clock. The Samples & Sampler block contains the audio database. It routes the segmented audio units to the appropriate positions that are predefined within the sequencer block, controlled by the MIDI controller. For example, a selected 8th note on the first position of the sequencer window does not automatically exclude the selection of another audio unit on the second position of the menu. The user disables or enables the excitation of a subsequent note with a left mouse click. The

sequencer of the virtual instrument determines the rhythmic values and articulations of the performed musical notes; however, the sequencer does not determine the pitch value of the performed musical note. The pitch value is determined by a triggered MIDI_On note and MIDI_Off note information, and both MIDI_On and MIDI_Off are defined by an external MIDI controller. The GUI is linked to and controlled by the Complete Kontrol MIDI controller. Both piano roll and external hardware controller enable not only the selection of different pitch note positions but also different designed phrases and different musical modes and route note positions. The different pitch note positions are selected with the help of the piano keys on the controller or by the MIDI note position within the piano roll. The different designed phrases are selected within the lower octave of the keyboard and the piano roll. The musical mode selection function allows switching between different musical modes and route notes; therefore, users can switch between different modes without changing the absolute note input.

The FPS-VI_{v1} GUI connects to the audio database (Figure 7), and the composer controls the audio database through the GUI. The GUI has four regions: (1) Sequencer window; (2) Phrase length; (3) Phrase preset menu; (4) Debugging window.

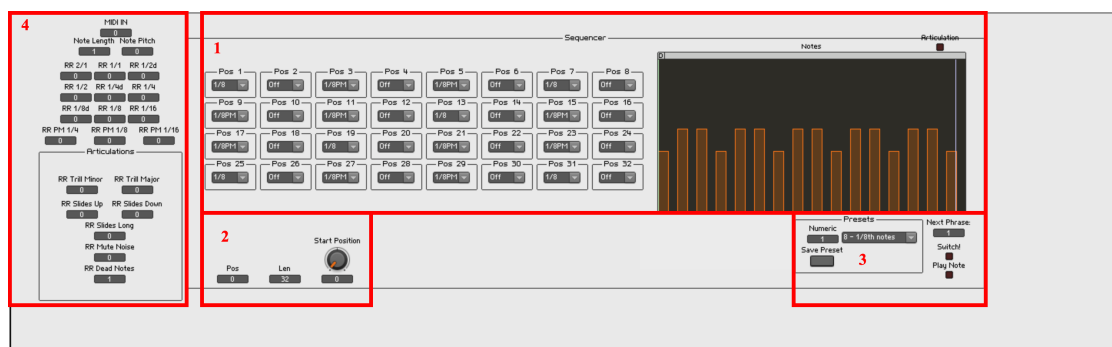


Figure 7. The four regions of the FPS-VI_{v1} GUI.

The first region includes a sequencer window, and it is the main controlling field of the FPS-VI_{v1}. The composer builds within the sequencer window musical phrases. A

musical phrase has a maximum length of 32 steps. After 32 steps, the phrase repeats. Each step represents a $\frac{1}{16}$ th musical note. The user clicks with the computer mouse on a step, and the composer then selects different rhythmical lengths and articulations via a dropdown menu. The second region includes a rotatory knob, and it determines the length of the phrase. To do so, the composer uses the computer mouse and moves the rotatory knob to a start position between the first (indicated by the number '1') and the last start position (indicated by the number '31'). A smaller number indicates a shorter phrase; a higher number indicates a longer phrase. The third region includes a save button and a preset menu. The composer uses the computer mouse and clicks on the save button to save newly designed phrases. The composer creates new phrases within the sequencer window, and the preset menu recalls the saved phrases. The fourth region includes 22 different displays and numbers. It provides an overview of all recorded audio units, and it is useful to debug and to localise wrong or unfinished audio recordings within the Reaktor 6 mapping structure. The displays and numbers do not contribute to any compositional or musical functionalities.

The internal tempo of the DAW corresponds to the tempo of FPS-VI_{v1}. Starting the playback function within the DAW starts the sequencer of the virtual instrument_{v1}. Stopping the playback engine of the DAW stops the sequencer automatically. The linked tempo and playback function of both DAW and FPS-VI_{v1} are indispensable for the compositional process. It ensures full integration of the electric guitar within the compositional process; moreover, the integration makes it possible to use the electric guitar in combination with other sounds and virtual instruments that follow all the same tempo given by the DAW. However, the FPS-VI_{v1} cannot change in tempo; consequently, the tempo stays on 120 bpm.

The Reaktor 6 DFVP realises the FPS-VI_{v1}, including an audio database and a GUI. The audio database facilitates the recorded and edited corpus of electric guitar samples, and the composer controls the audio database through the GUI.

8.4.3 Composing and observing

This section describes the collected audiovisual data of the compositional process of ‘The Alpha’. ‘The Alpha’ was composed and observed between the 5th of October 2018 and 11th of October 2018. ‘The Alpha’ was created within 17 hours. It took 8 hours to compose ‘The Alpha’ with the aid of the FPS-VI_{v1} and another 9 hours to finish the composition without the use of the FPS-VI_{v1}. Table 5 provides an overview of the seven self-study videos, and it summarises the duration of the recordings. The seven self-study video recordings focus on the human-computer interaction with the FPS-VI_{v1}. All videos were coded based on open coding, axial coding, and selective coding⁷⁸.

Number	Video title	Date	Duration
1	181005_The Alpha_r1	5 October 2018	22m 54s
2	181005_The Alpha_r2	5 October 2018	1h 42m 57s
3	181008_The Alpha_r3	8 October 2018	1h 6m 58s
4	181008_The Alpha_r4	8 October 2018	28m 30s
5	181009_The Alpha_r5	9 October 2018	1h 54m 41s
6	181009_The Alpha_r6	9 October 2018	1h 54m 16s
7	181011_The Alpha_r7	11 October 2018	37m 6s
Total			8h 7m 22s

Table 5. An overview of all self-study videos of the first composition.

⁷⁸ Appendix 8.3 provides additional information, and it includes an overview of the open coding, axial coding, and selective coding.

8.4.4 Summary of the audio-visual analysis

The compositional process of the first composition, 'The Alpha', is analysed based on the audio-visual observation data. According to the audio-visual observation data, the compositional process starts on the 5th of October 2018 and ends on the 11th of October 2018. The compositional process of 'The Alpha' is distinguished by two parts: the first part focuses on the compositional process with the aid of the FPS-VI_{v1}, and the second part focuses on the musical production of the accompaniment. I realised all sounds of 'The Alpha' within 'the box', without recording other instruments. The FPS-VI_{v1} realises both electric guitars, and other virtual instruments perform the accompaniment. A MIDI keyboard controller, a computer mouse, and a computer keyboard control all sounds. Furthermore, the selective coding distinguishes three steps in the compositional process: (1) Composing the melodic progression for the two electric guitars; (2) Composing harmonic building blocks; (3) Programming musical mode and root note switches.

8.4.5 Evaluation

This section evaluates the impact of the FPS-VI_{v1}, and the evaluation concentrates on both the analysed original musical composition and the analysed self-study observation data. The outcomes of the evaluation describe possible software design revisions for the next iteration. The FPS-VI_{v1} had an impact upon the content of the original composition, the compositional process, and the development of musical ideas of 'The Alpha'.

The FPS-VI_{v1} was biased towards western popular rock music and affected the musical form of 'The Alpha'. 'The Alpha' reminded me of 80s hard rock. Initially, it was not my motivation to compose this piece of music in the style of 80s hard rock;

moreover, I had never composed a piece of music in this style before. The FPS-VI_{v1} was biased towards this western popular musical style because the FPS functioned within a digital audio workstation (DAW), and the DAW was programmed based on western musical rulesets. 'The Alpha' followed a western popular musical form, including an introduction, 1st Verse, Bridge, 1st Climax, 2nd Climax, 2nd Verse, and 3rd Climax. Moreover, the instrumentation was a band. The instruments used were two electric guitars (The FPS-VI_{v1}), an electric bass, and drums. The FPS-VI_{v1} supported fast rhythmic e-guitar solo performances, and these solo-performances were typical for western popular rock music. Furthermore, the musical melodies of 'The Alpha' were biased towards two-measure long motives because the FPS-VI_{v1} was designed based on the idea of two-measure long phrases. The phrases were then repeated several times within the composition. The harmony modulated within western musical modes, and the time-signature included exclusively 4/4, which is a typically used time signature for western popular music.

The limitations of the musical performance of the FPS-VI_{v1} database also impacted upon 'The Alpha'. The dynamic of the electric guitar continually remained at the same level because the FPS-VI_{v1} database excluded dynamic deviations; therefore, it was not able to not perform different dynamic values. Both electric guitars were performed with the aid of only two different heavily distorted guitar amplifiers because the segmented audio units of the FPS-VI_{v1} database included a lot of clicks and bugs. Consequently, I chose a heavy amp simulation effect to mask the impurities of the segmented audio unites. Other amp simulations were not able to mask the impurities.

The FPS-VI_{v1} graphical user interface (GUI) also affected 'The Alpha'. The GUI excluded the functionality to change the time-signature; therefore, the time signature of 'The Alpha' was 4/4. Moreover, the sequencer window and the pre-designed

musical phrase menu of the GUI were insufficient developed, and they affected 'The Alpha' adversely. While composing, I did not change the pre-designed musical phrases or build new phrases within the sequencer window because it would take up a lot of time. The five pre-designed phrases, including 8th and 16th notes, continuously remained the same; consequently, short notes dominated 'The Alpha'.

Furthermore, the use of the FPS-VI_{v1} affected the compositional process and the development of musical motives of 'The Alpha'. The FPS-VI_{v1} guided me to follow a particular compositional process: (1) Composing the melodic progression for the two electric guitars; (2) Composing harmonic building blocks; (3) Programming musical mode and root note switches. First, I had to choose a musical phrase within the sequencer window of the FPS-VI_{v1} GUI. It was necessary to choose or build a musical phrase to trigger the FPS-VI_{v1} database. Second, programmed MIDI notes within the DAW triggered the musical phrase. I programmed the 16th notes with the help of the piano roll of the DAW because 16th notes on 120 BPM were too fast to play on the on a regular MIDI keyboard. Moreover, it was the first time that I controlled the FPS-VI_{v1} with a MIDI controller, and I needed more time to practice the virtual instrument. I programmed the MIDI notes based on the two-measure long sequencer window of the FPS-VI_{v1}. The melodies and harmonies and musical ideas of 'The Alpha' were biased towards the two-measure length of one phrase. Third, the musical mode function and root note changes controlled the MIDI notes. The automation panel of the DAW triggered the musical modes and root note changes. However, the musical mode function interfered with the FPS-VI_{v1} musical phrase switcher, and it created awkward compositional situations. After that, the second electric guitar used the same musical phrase and the same scale and root note; however, it performed different pitch values. The compositional process was repeated several times.

In conclusion, the FPS-VI_{v1} impacted upon the original compositional content of ‘The Alpha’. The FPS-VI_{v1} was biased towards western popular rock music. The musical performance limitations of the FPS-VI_{v1} database and GUI also impacted upon the musical content of ‘The Alpha’. Moreover, the use of the FPS-VI_{v1} also affected the compositional process, and it impacted the development of musical motives. In addition, the static-dynamic envelope, the static 4/4-time signature, the use of only two different amp simulations, and the limited use of only five pre-defined musical phrases made ‘The Alpha’ too predictable and a bit boring. These compositional issues possibly had an impact on the development of the next FPS. In the next iteration, I improved the FPS-VI_{v1} database and GUI by removing the bugs and correcting the glitches of the audio units. A revised database allows deviation in amp simulations, and it makes the sound of the electric guitar more diverse, more interesting, and more dynamic. Moreover, the revised GUI incorporates a time signature function. A time signature function allows deviation between different time signature values, and it introduces several more compositional possibilities. Furthermore, I strive to prepare different musical phrases beforehand. While composing this give access to more rhythmic variations and articulations.

8.5 Second iterative design process: The Beta

The second iterative design process addresses the production of the second original composition called ‘The Beta’. It also addresses the design of the second version of the FPS-VI_{v2} and the first version of the FPS-UIC_{v1}. The iterative design process recapitulates the compositional issues of ‘The Alpha’ (8.5.1), describes the functional design of the FPS-VI_{v2} and the FPS-UIC_{v1} (8.5.2), outlines briefly the gathered musical data and observation data (8.5.3), summarises the musical description of ‘The Beta’ and observation data (8.5.4), and evaluates the analysed data based on the research questions (8.5.5).

8.5.1 Defining the problem

This section briefly recapitulates the evaluation of ‘The Alpha’. The recapitulation pays attention to the initial problem of this research. This second iteration improved both the FPS-VI_{v1} database and GUI. I removed the bugs of the audio units, implemented a time signature function, and prepared different musical phrases beforehand. Furthermore, programming musical modes and root notes affected ‘The Alpha’ adversely. In ‘The Alpha’, I changed the musical modes and root notes within the automatization window of the DAW. However, this process was awkward and time-consuming. After experimental research, I realised that the Lemur⁷⁹ software environment allows programming a user interface controller (UIC) that performs different musical modes and root notes easily and quickly. It motivated me to introduce a customised FPS-UIC_{v1}

⁷⁹ Lemur is a software application developed by the company Liine. See 4.3.5 for more information.

within the second iteration of ‘The Beta’. The FPS-UIC_{v1} connects to the FPS-VI_{v2} and is integrated within the digital audio workstation (DAW) information stream.

8.5.2 Designing the FPS

The second version of the FPS includes the second version of the FPS-VI_{v2} and the first version of the newly designed FPS user interface controller (FPS-UIC_{v1}).

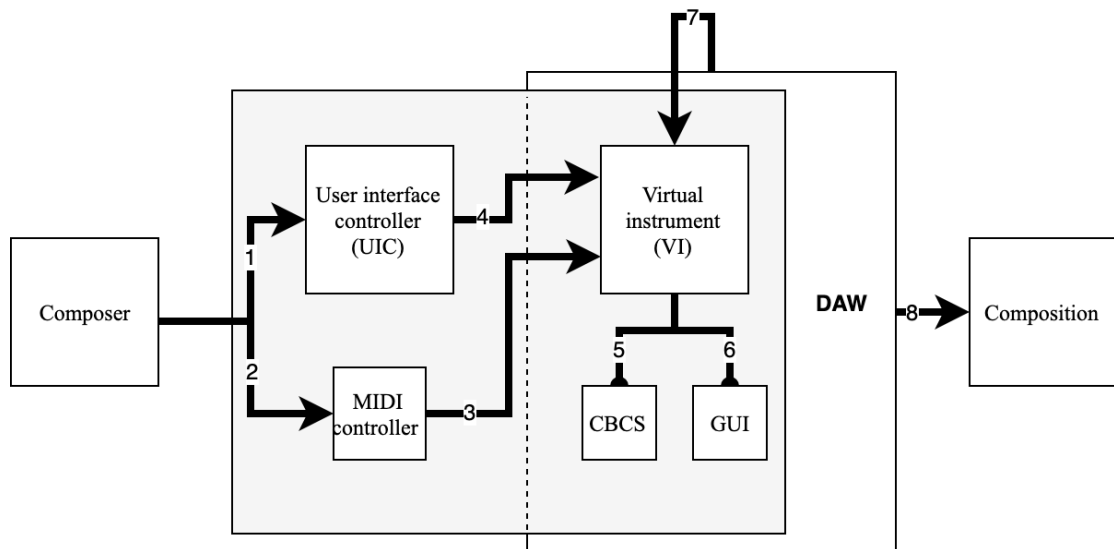


Figure 8. The information stream of the FPS-VI_{v2} and the FPS-UIC_{v1}.

Figure 8 illustrates the revised information stream of the FPS-VI_{v2}, including the FPS-UIC_{v1}. The first (1) and second (2) arrows illustrate two different ways of interacting with the FPS-VI_{v2}. The composer controls the FPS-VI_{v2} (4) via the first option with the aid of the newly designed FPS-UIC_{v1} and the FPS-VI_{v2} (3) via the second option with the aid of a MIDI keyboard controller. MIDI mapping strategies connect both the FPS-UIC_{v1} and MIDI keyboard controller to the DAW, and the FPS-VI_{v2} is integrated within the DAW structure. The FPS-VI_{v2} includes an audio database (CBCS) (5) and a graphic user interface (GUI) (6). The DAW (7) returns musical mode

and key switch information back to the FPS-VI_{v2}. The output of the interaction between the composer and the DAW (8) is an audio rendering of ‘The Beta’.

The FPS-VI_{v2} audio database is re-edited. The original single note phrases are re-segmented within the Izotope RX 6 environment. In contrast to the Cubase VariAudio function, Izotope RX 6 is a more accurate audio editor, and it ensures a better rendering process. Moreover, the audio samples are time-stretched to the perfect length of the rhythmic notes. The following rhythmic length values correspond to the following values in milliseconds (Table 6).

Rhythmic value (120 bpm)	Milliseconds
2/1	4000
1/1	2000
1/2 dotted	1500
1/2	1000
1/4 dotted	750
1/4	500
1/8 dotted	375
1/8	250
1/16	125

Table 6. An overview of recorded rhythmic values converted into milliseconds.

Moreover, the length of the recorded articulations slides down, slides up, trills major, and trills minor are linked to fixed 1/2 note length; therefore, the rhythmic values of all articulations are edited to shorter rhythmic values. From this follows that the slides up and slides down articulations are time-stretched from 1000 milliseconds to 250 milliseconds and that the trills major and trills minor recordings are time-stretched from 1000 milliseconds to both 500 milliseconds and 250 milliseconds.

The audio sampler is replaced within the FPS-VI_{v2} sequencer block. An improved audio sampler replaces the initial audio sampler. The internal sampler of the FPS-VI_{v1} inserts a short fade in on each triggered sample. The revised audio sampler performed the audio database properly. Figure 9 shows two different waveform renderings of two different sample synthesis engines. The first waveform is the result of the revised version of the FPS-VI_{v2} audio sampler, and the second waveform is the result of the FPS-VI_{v1} audio sampler. The intensity of the transient differs significantly between the two audio examples.

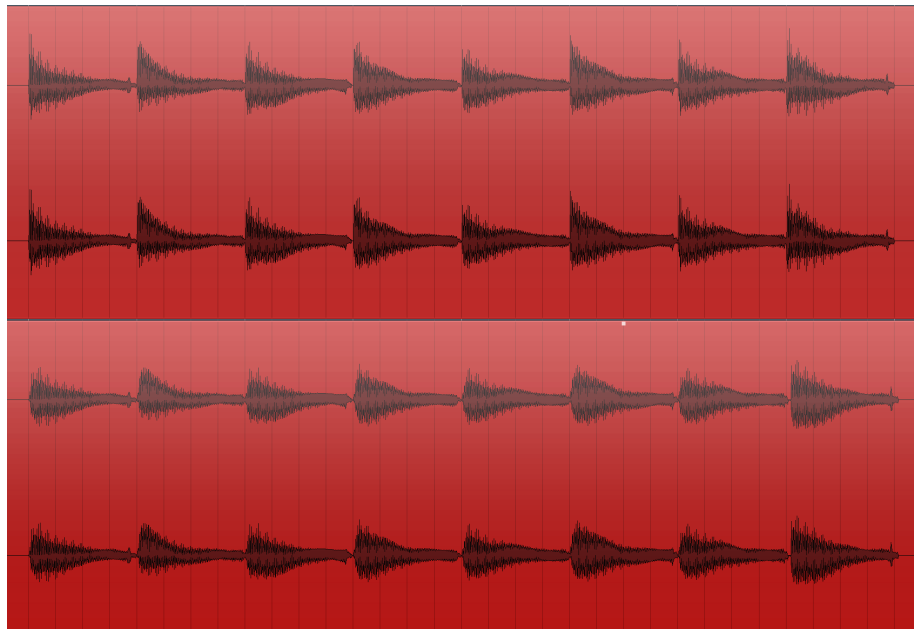


Figure 9. Two renderings of two different sample synthesis engines.

Performance recordings expand the FPS-VI_{v4} audio database. Performance recordings are 2 measure long musical electric guitar recordings. In total, 25 musical performances are recorded, and each musical performance includes four variations. The variations are linked to four different route notes. The first variations start from E, the second from G, the third variation starts from A[#], and the last variation starts from C[#]. The performances are organised based on four different musical styles: (1) Classic

Rock; (2) Funk; (3) Grunge; (4) Hard Rock. The first five phrases are played in the style of classic rock. Phrases six to ten are played in funk, phrases 11 to 20 are played in the style of grunge, and the last phrases, 21 to 25, are played in the style of hard rock.

Moreover, the deterministic and stochastic audio material of the 100 musical performance phrases are decomposed. The software environment RX 6 by Izotope was used to decompose the deterministic and stochastic part of the musical phrases; subsequently, the stochastic material is embedded within the FPS-VI_{v4} audio database. The deterministic material of the performance phrases was removed, and was not used. The small gaps in between the single notes are filled with the stochastic audio material; therefore, the stochastic material ‘glues’ the single note recordings together.

The FPS-VI_{v2} comprises a re-edited audio database and a revised GUI (Figure 10), which means that all 11408 original single note samples are re-edited and re-segmented. The re-edited audio database enables the selection of different amplifier simulations. Furthermore, the GUI is slightly improved and involves three regions: (1) Sequencer window; (2) Phrase length; (3) Phrase preset menu.

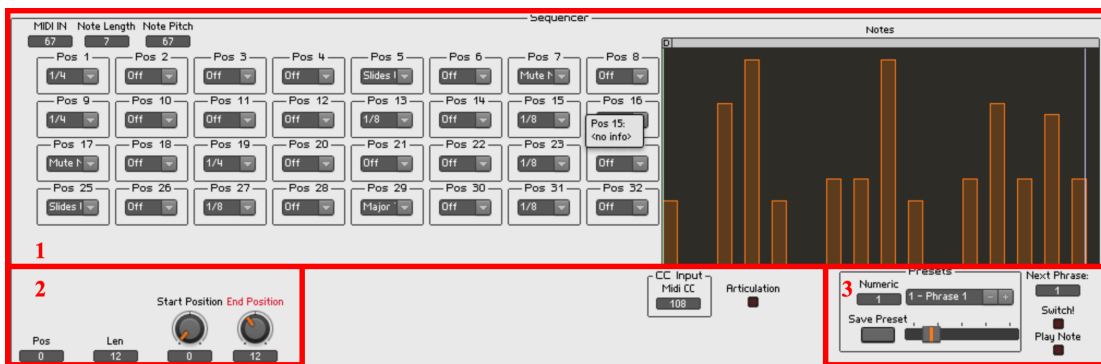


Figure 10. The three regions of the FPS-VI_{v2} GUI.

The first region includes the sequencer window. The sequencer window of the FPS-VI_{v2} is the same as the sequencer window of the FPS-VI_{v1}. The second region includes two rotatory knobs. The first rotatory knob determines the start position of the

phrase, whereas the second rotatory knob determines the end position of the phrase. Both rotatory knobs together determine the length of the phrase, and they ensure that the time signature is adjustable. The composer changes the start point and endpoint of a triggered phrase with the computer mouse. Based upon a grid resolution of $1/16^{\text{th}}$ notes, the composer can create signature values between $1/16^{\text{th}}$ and $2/1^{\text{nd}}$. In Figure 10, region 2 illustrates a start position, including the grid value of '0' and an end position including the grid value of '12'. In this example, the time signature is $12/16^{\text{th}}$, and $3/4^{\text{th}}$ respectively. The third region includes a save button and the phrase preset menu. The phrase preset menu includes several more pre-designed musical phrases. The functionality of the save button remains the same.

This section describes the concept and design of the user interface controller (FPS-UIC_{v1}). The Lemur software environment facilitates the development of the FPS-UIC_{v1}. The programming language is C, installed on an external tablet and connected as a MIDI device to the digital audio workstation (DAW) and computer. The FPS-UIC_{v1} transmits different MIDI message automation information to the DAW. The FPS-UIC_{v1} includes one central region (Figure 11, region 1), divided into 72 unique buttons.

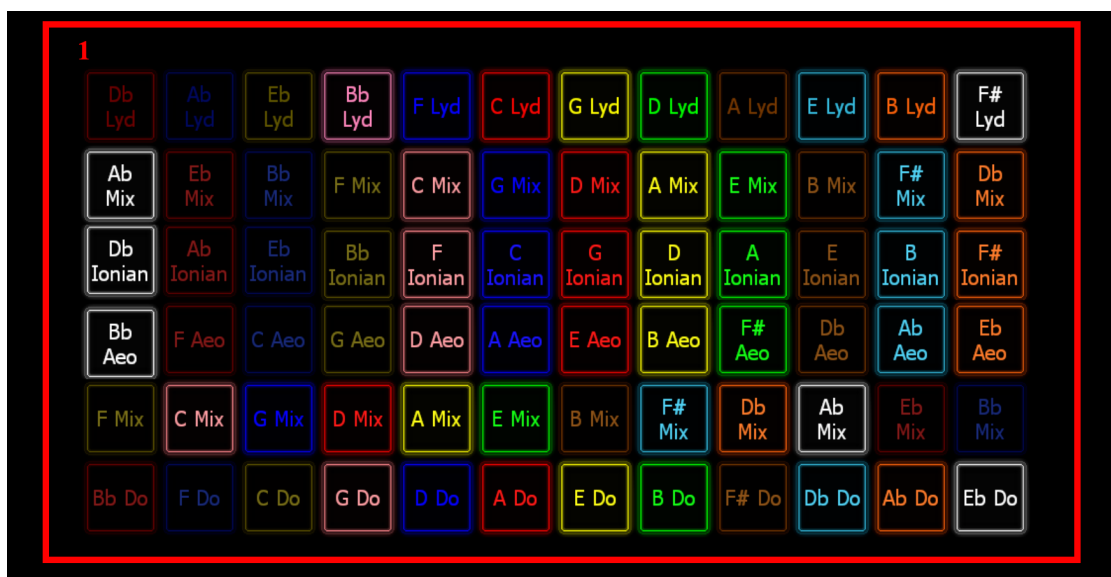


Figure 11. The FPS-UIC_{v1}.

Each button triggers a different combination of musical modes and root notes. The programmed musical modes are Dorian (Do), Lydian (Lyd), Mixolydian (Mix), Ionian (Ionian), and Aeolian (Aeo). The programmed root notes are C, D^b, D, E^b, E, F, F[#](G^b), G, A^b, A, B^b, and B. Buttons with the same colour share the same musical notes. For example, the yellow highlighted buttons with the abbreviations ‘A Mix’, ‘D Ionian’, ‘B Aeo’, ‘Ab Mix’, ‘G Lyd’, and ‘E Do’ share the musical notes D, E, F, G, A, B^b, and C.

In conclusion, the second version of the FPS includes the second version of the FPS-VI_{v2} and the first version of the newly designed FPS-UIC_{v1}. The FPS-VI_{v2} audio database is re-edited, and the GUI is revised. The first version of the FPS-UIC_{v1} is programmed and designed⁸⁰.

8.5.3 Composing and observing

This section describes the collected audio-visual data of the compositional process of ‘The Beta’. ‘The Beta’ was composed and observed between the 6th of December 2018 and 14th of December 2018. I composed the second composition within 11 hours and 30 minutes. It took 6 hours and 30 minutes to compose ‘The Beta’ with the aid of both the FPS-VI_{v2} and FPS-UIC_{v1}, and it took around 5 hours to procure and to add accompaniments to the song. Table 7 provides an overview of the six self-study videos,

⁸⁰ Appendices 8.2.1 and 8.2.2 provide technical information including about the audio database, the GUI and the UIC.

and it summarises the duration of the recordings. In this case, the observation method recorded the whole compositional process. The last two videos do not include the human-computer interaction with the second version of the FPS; consequently, the first four videos were coded based on open coding, axial coding, and selective coding⁸¹.

Number	Video title	Date	Duration (h:m:s)
1	181206_ The Beta _r1	6 December 2018	1h 16m 16s
2	181206_ The Beta _r2	6 December 2018	3h 13m 26s
3	181207_ The Beta _r3	7 December 2018	1h 45m 42s
4	181213_ The Beta _r4	13 December 2018	22m 20s
5	181213_ The Beta _r5	13 December 2018	3h 12m 57w
6	181214_ The Beta _r6	14 December 2018	1h 43m 32s
Total			11h 34m 13s

Table 7. An overview of all self-study videos of the second composition.

8.5.4 Summary of the audio-visual analysis

The compositional process of ‘The Beta’ is analysed based on the audiovisual observation data. The observation data shows that the compositional process of ‘The Beta’ started on the 6th of December 2018 and ended on the 14th of December 2018. All sounds of ‘The Beta’ were realised within ‘the box’, and no other instruments were recorded. Both the MIDI keyboard controller and the FPS-UIC_{v1} control and performed all virtual instruments. The left-hand controls the FPS-UIC_{v1}; the right-hand controls the

⁸¹ Appendix 8.3 provides additional information, and it includes an overview of the open coding, axial coding, and selective coding.

MIDI keyboard controller. Moreover, the selective coding distinguishes five steps in the compositional process: (1) Defining the musical form based on a storyline with the aid of the FPS-UIC_{v1}; (2) Choosing a musical phrase for the first electric guitar; (3) Playing MIDI notes on a MIDI keyboard; (4) Performing musical mode and root note changes on the FPS-UIC_{v1}; (5) Improvising with a second electric guitar upon the first electric guitar.

8.5.5 Evaluation

This section evaluates the impact of the FPS-VI_{v2} and the FPS-UIC_{v1}. The analysed original musical composition and the analysed self-study observation data define the evaluation. The outcomes of the evaluation describe possible software design revisions for the next iteration. The FPS-VI_{v2} and the FPS-UIC_{v1} impacted upon the original composition, the compositional process, and the development of musical ideas of ‘The Beta’. The FPS-VI_{v2} and the FPS-UIC_{v1} are biased towards western popular jazz music, and it affected the musical form of ‘The Beta’. Just like ‘The Alpha’, ‘The Beta’ was composed within a digital audio workstation (DAW) which followed western musical rulesets. ‘The Beta’ followed a western popular musical form (A, B, C, B’, A, A’, and B’), and the instrumentation included popular western instruments. Moreover, the melodies of ‘The Beta’ were biased towards two-measure long motives because the FPS-VI_{v2} sequencer window was developed based on the length of two measures. The harmony modulated within western musical modes, and the time signature included $4/4$, $10/4$, $2/4$, and $3/4$. These are typical time signatures for western popular music.

The FPS-VI_{v2} database also affected ‘The Beta’. While composing, I did not recognise significant clicks or bugs in the sound of the revised database of the FPS-VI_{v2}. In contrast to ‘The Alpha’, the revised database allowed me to choose cleaner amp

simulations because there was no need to mask impurities. Showcasing the pure sound of the FPS-VI_{v2} was a significant motivation; therefore, I used less or even no post-processing effects. In ‘The Beta’, the musical output of the FPS-VI_{v2} still sounded natural without masking the sound by using a considerable amount of distortion; consequently, I used less time to produce the sound the FPS-VI_{v2}.

The FPS-VI_{v2} graphical user interface (GUI) affected ‘The Beta’. The first and second melodies of ‘The Beta’ were pre-designed musical phrases, produced at the very beginning of the composition. However, the two melodies of ‘The Beta’ embodied the first and second pre-designed phrases, and there were no new phrases or variations designed while composing. The FPS-VI_{v2} provided an awkward way of choosing and programming different rhythmic phrases; therefore, the unacceptable phrase switch restrained me from designing or using new phrases while composing ‘The Beta’.

Moreover, the FPS-VI_{v2} performed different time signatures, and I used this newly introduced functionality for the musical segments B and B’. Beside the $\frac{4}{4}$ -time signature, I also used different time signatures such as $\frac{10}{4}$, $\frac{2}{4}$, and $\frac{3}{4}$. In addition to using different time signatures to support the storyline of ‘The Beta’, I also wanted to ‘test’ the new functionality. However, the time signature switch functionality affected my motivation and my musical inspiration adversely because the time signature functionality was not working correctly, and it interrupted my compositional workflow. ‘The Beta’ did not include any tempo changes because the FPS-VI_{v2} was not able to perform any tempo deviations; therefore, the musical work used 120 bpm.

In addition, the FPS-UIC_{v1} affected the musical form of ‘The Beta’. The FPS-UIC_{v1} is biased towards structuring the musical form of ‘The Beta’ based on a storyline. Initially, I started every musical composition by determining the musical form. However, I had never used a storyline to structure the musical form; moreover, I had

never used a sequence of musical modes and root note switches to guide the storyline of the musical work. ‘The Beta’ was an exception: the functionality of the FPS-UIC_{v1} fascinated me because I had never used this kind of UIC before. While composing, I experimented with the UIC, and I tried to find the musical boundaries of the UIC. The outcome of this experimental approach included the use of the UIC as a compositional tool that determined the musical form based on musical mode switches.

The use of the FPS-VI_{v2} and FPS-UIC_{v1} also affected the compositional process of ‘The Beta’. The compositional process included the following five steps: (1) Defining the musical form based on a storyline with the aid of the FPS-UIC_{v1}; (2) Choosing a musical phrase for the first electric guitar; (3) Playing MIDI notes on a MIDI keyboard; (4) Performing musical mode and root note changes on the FPS-UIC_{v1}; (5) Improvising with a second electric guitar upon the first electric guitar. The compositional process was comparable to the process of ‘The Alpha’; however, the second and third steps were mostly performed simultaneously. Figure 12 illustrates a moment within the compositional process in which I played the MIDI keyboard with the right-hand (step 3) and the FPS-UIC_{v1} with the left-hand (step 4). Unfortunately, the concurrent performance of both interfaces affected my compositional process adversely. It was very tough to perform on both interfaces simultaneously because the right hand had to focus on performing musical melodies (step 2), while the left hand had to focus on performing musical mode switches (step 3). This physical interaction demanded a lot of concentration and multitasking. Interestingly, there was no need to perform both steps simultaneously; however, while composing, I did not realise that I was able to control both interfaces step-by-step.

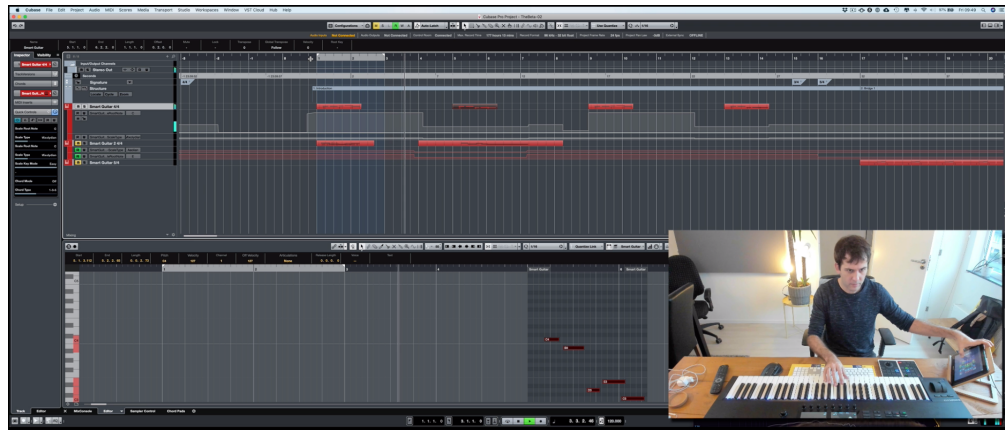


Figure 12. Controlling the FPS-UIC_{v1} and the MIDI keyboard controller.

The FPS-UIC_{v1} also affected the use of harmonic chords and musical mode modulations. Beside the 6th musical segment A', I solely used the tonic major 7th chord, including the 9th and 13th note of each musical mode. On the one hand, the musical mode switch function simplified the use of different musical modes, but on the other hand, I was overwhelmed by the mode switch and root note possibilities. Therefore, I had difficulties in defining the functionality of different chords apart from the tonic chord in each mode. It was very challenging to maintain an overview of the available harmonic possibilities. While composing, I decided to maintain straightforward harmonic progressions, and I focused solely on the tonic chord of each mode. Moreover, I exclusively used the root position of each chord, and I ignored the use of the first, second, or third chord inversion. In some cases, I wanted to use a chord inversion rather than the root position; however, the FPS-UIC_{v1} did not provide me with a sufficient overview of the available chord inversions. Moreover, the musical mode switches did not correspond with my harmonic taste. They were a set of musical instructions programmed within the UIC, and I had to improvise with musical instruments within the given set of musical modes. The improvisation impacted upon

the musical result; therefore, the harmonic progressions of ‘The Beta’ sounds a bit random.

In conclusion, the FPS-VI_{v2} and the FPS-UIC_{v1} impacted upon ‘The Beta’. The FPS was biased towards western popular jazz music, and they impacted upon the compositional process, the musical form, the harmonic chords, and the musical mode modulation. However, FPS-VI_{v2} GUI was still not intuitive enough, and it avoided designing different phrase while composing original music. The FPS-UIC_{v1} did not provide a sufficient overview, and the design was not intuitive. The compositional issues of ‘The Beta’ possibly impacted upon the development of the next FPS. For the next iteration, the FPS-VI_{v2} GUI, and the FPS-UIC_{v1} were improved. The FPS-VI_{v2} GUI includes several more pre-defined phrases. I program new phrases within the sequencer window. Moreover, the FPS-UIC_{v1} was also improved, and I tested the possibilities of using the UIC as both the musical mode switch controller and MIDI note controller. Expanding the UIC possibilities ensured that I didn’t have to focus on both the MIDI keyboard and the UIC at the same time. Moreover, the physical structure of a keyboard is not a logic interface for the FPS-VI. I did not need to express the sound of an electric guitar by playing piano keys. I assumed that it is easier for me to stay within one graphic user interface environment rather than switching between the MIDI keyboard and the tablet. Moreover, I revised the FPS-UIC GUI, and I programmed the functionality to use chord inversion rather than only using root positions. I also programmed more types of chords to a wider range of musical chord to choose from.

8.6 The third iterative design process: The Gamma

The third cyclic process addresses the design process of the third original composition called ‘The Gamma’, and it addresses the design of the FPS-UIC_{v2}. The third iterative design process recapitulates the compositional issues of ‘The Beta’ (8.6.1), describes the functional design of the FPS-UIC_{v2} (8.6.2), briefly outlines the musical data and observation data (8.6.3), summarises the musical description of ‘The Gamma’ and the observation data (8.6.4), and evaluates the analysed data based on the research questions (8.6.5).

8.6.1 Defining the problem

This section briefly recapitulates the evaluation of ‘The Beta’. The recapitulation pays attention to the initial problem of this research. For the third iteration, I improved upon the FPS-VI_{v2} GUI and the FPS-UIC_{v1}. The FPS-VI_{v2} affected the compositional process adversely because it did not include enough pre-designed flexible phrases. Therefore, the FPS-VI_{v3} GUI includes more pre-defined phrases. The FPS-UIC_{v1} affected the harmonic progression adversely. The harmonic progressions sound a bit random; therefore, the FPS-UIC_{v2} includes calculated modulations and the functionality to use chord inversion rather than only using root positions. Furthermore, the use of both the FPS-UIC_{v1} and the MIDI keyboard controller impacted upon the compositional process. I tested the possibilities of using the UIC as both the musical mode switch controller and MIDI note controller. I also tested different MIDI note controllers as an alternative to the MIDI keyboard controller.

8.6.2 Designing the FPS

The third version of the FPS includes the second version of the FPS-UIC_{v2}; however, it excludes the FPS-VI_{v3}. While composing ‘The Gamma’, I could not incorporate the FPS-VI_{v3} because the FPS-VI_{v3} could not perform different time signature deviations very well, and it could not perform triple notes. ‘The Gamma’ went beyond the limits of the FPS-VI_{v3}; therefore, I decided to not include the FPS-VI_{v3} in the iterative design process of ‘The Gamma’. Figure 13 illustrates the revised information stream, excluding the FPS-VI_{v3}, but including the FPS-UIC_{v2}.

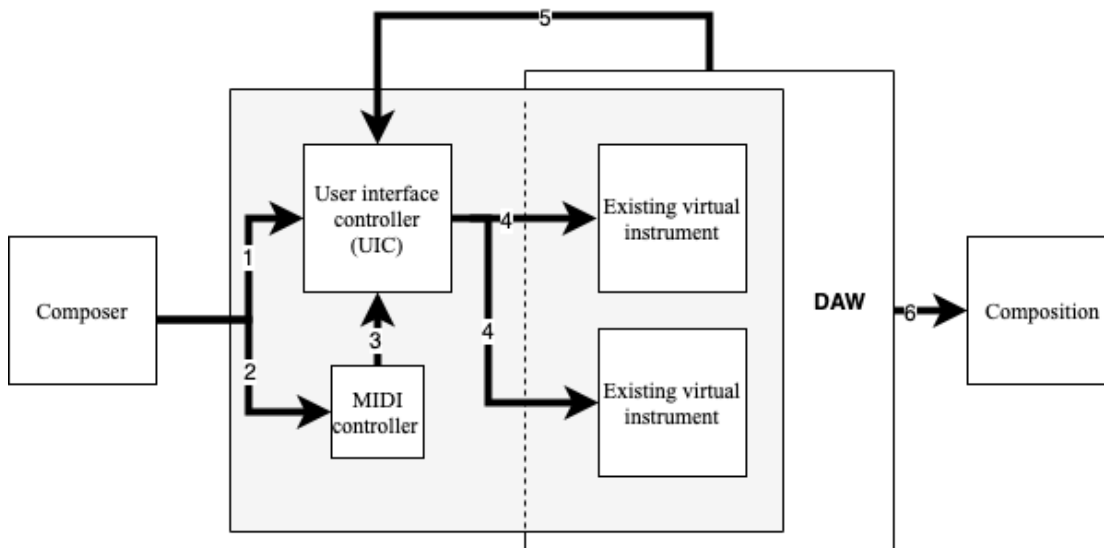


Figure 13. The information stream of the FPS-UIC_{v2}.

The first (1) and second (2) arrow illustrate two different ways of interacting with the virtual instruments. The composer controls the virtual instruments (4) with the aid of the FPS-UIC_{v2} (1), the MIDI keyboard controller (2), or both. The MIDI keyboard controller transmits Note_On and Note_Off information to the user interface controller (3). The DAW connects to the FPS-UIC_{v2} via MIDI mapping strategies; however, the MIDI keyboard controller solely communicates with the FPS-UIC_{v2}, and it is not connected directly to the DAW. The DAW returns (5) MIDI control change

information back to the FPS-UIC_{v2}, which ensures the integration of the FPS-UIC_{v2} within the DAW structure. The output of the interaction between the composer and the DAW (6) is an audio rendering of the third original composition, in this case, ‘The Gamma’.

The first evaluation of the FPS-UIC_{v1} motivated me to program a more flexible and modular interface; therefore, I revised the FPS-UIC_{v2} thoroughly. The controller calculates the musical modulations and musical modes. In contrast to the FPS-UIC_{v1}, the FPS-UIC_{v2} is less static and more flexible, which means that it is easy to add new musical chords to the FPS-UIC_{v2}. This version does not use the Komplete Kontrol software environment; therefore, it can control any virtual instrument within a DAW. The FPS-UIC_{v2} enables the performance of musical notes and musical chords that occur solely within the selected musical mode. The musical notes do not correspond to the MIDI notes; they correspond to the relative position within the selected musical mode. The FPS-UIC_{v2} has six regions (Figure 14): (1) Selecting a root note; (2) Selecting a musical chord; (3) Selecting an inversion; (4) Selecting a musical chord to modulate; (5) Selecting a flexible phrase; (6) Transposing the selected musical chord to another octave.

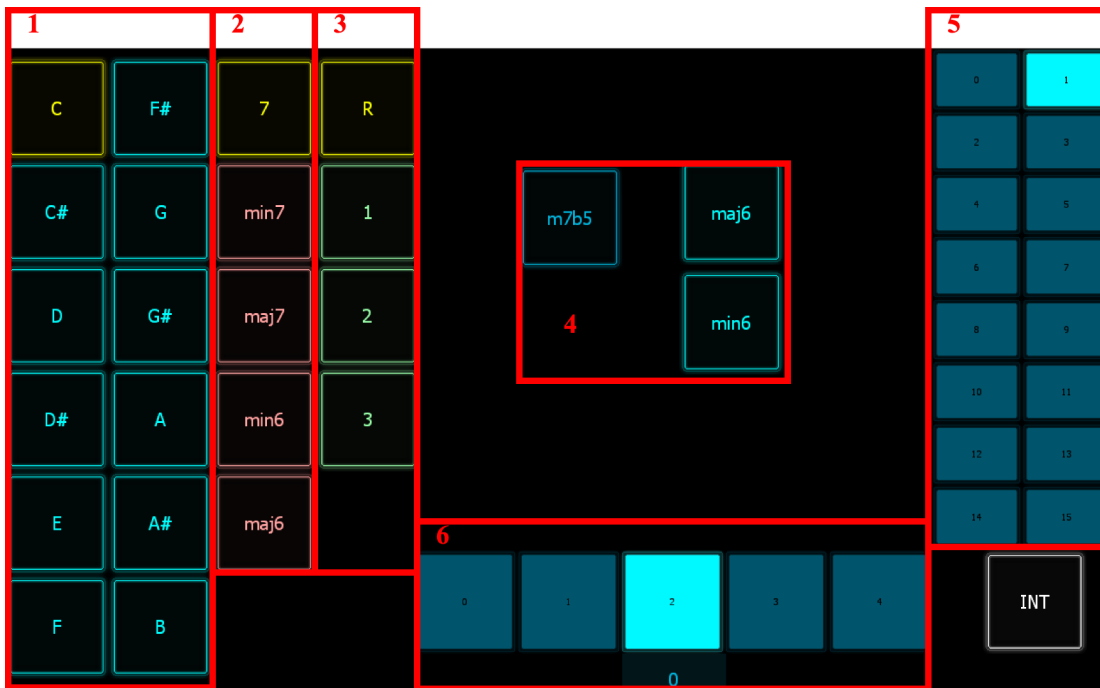


Figure 14. The six regions of the FPS-UIC_{v2}.

The first three regions define the first musical chord. In region 1, the composer selects the root note. The composer can select between 12 different root notes. In region 2, the composer selects the musical chord. The FPS-UIC_{v2} provides five different musical chord options: dominant 7, minor 7, major 7, minor 6, or major 6 chords. Moreover, the selected musical chord automatically defines the selected musical mode. The dominant 7 chord is linked to the Mixolydian mode, the minor 7 chord is linked to the Aeolian mode, the major 7 is linked to the Ionian mode, the minor 6 is linked to the Aeolian mode, and the major 6 is linked to the Ionian mode. In region 3, the composer selects the inversion of the selected chord. ‘R’ represents ‘Root position’, ‘1’ is the first inversion, ‘2’ is the second inversion, and ‘3’ is the third inversion. In this example (Figure 14), the root note ‘C’ is selected, and it is a dominant 7 chord in root position.

In region four, the composer selects the modulation chord. The FPS-UIC_{v1} uses the dominant 7th chord exclusively to modulate to another musical mode; however, the

FPS-UIC_{v2} uses a half-diminished 7 chord (M^7_{b5}) to modulate to another mode. I used the M^7_{b5} because this musical chord provides different options to modulate to other musical chords. The composer can then continue either to a major 6th chord or to a minor 6th chord. The new root note is either two semitones higher than the initial root note (introduced by a major 6th chord) or one semitone lower than the initial root note (introduced by a minor 6th chord). The composer switches continuously between the half-diminished 7th chords and the major 6th chord or the minor 6th chord; consequently, the composer cannot choose different musical chords.

The fifth region includes the phrase preset selector of the FPS-VI_{v3}. The composer clicks on one of the buttons, and then he/she selects a pre-defined phrase. In doing so, the FPS-UIC_{v2} sends control change information to the DAW and FPS-VI_{v3}. The controller then selects from the different pre-designed phrase presets of the virtual instrument. In total, the composer can switch between 15 different pre-designed phrases. The number '0' indicates that no pre-designed phrase is selected. In this example (Figure 14), the selected phrase is the first phrase preset (1). Unfortunately, 'The Gamma' excludes the use of the FPS-VI_{v3}; therefore, excluding the use of the phrase preset selector function.

In region six, the composer selects the octave in which the system performs the selected musical chord. The row includes five numbers: 0, 1, 2, 3, and 4, each inside a blue square. An additional number is located below the row. The row shifts all notes of the selected musical chord one or two octaves higher or lower. '0' transposes the selected musical chord two octaves lower, '1' transposes the selected musical chord one octave lower, '2' does not transpose, '3' transposes the selected musical chord one octave higher, and '4' transposes the selected musical chord two octaves higher. In this example, the light blue colour indicates the selected button (number 2), and the number

'0' below the row indicates that the system does not transpose the selected musical chord.

Initially, I wanted to use the FPS-UIC_{v2} as both musical mode controller and musical note performer. However, the screen space of the tablet was too small to realise musical mode switches, phrase switches, and musical note performances. The FPS-UIC_{v2} disregards the note performing option, and it focuses on calculating different musical chords and musical modulations. A MIDI keyboard controller and a newly introduced ROLI Lightpad controller (Figure 15) perform the selected musical chords and musical notes. The composer performs the selected chord with the help of the ROLI Lightpad controller. The ROLI Lightpad controller has four regions, which control four different notes of 7th chord or 6th chord: (1) is the tonic note of the chord; (2) is the third note of the chord; (3) is the fifth note of the chord; (4) is the sixth or seventh note of the chord. The FPS-UIC_{v2} defines the musical modes and musical chords while the ROLI Lightpad controller and the MIDI keyboard controller perform the musical chords and musical notes that are true to the selected musical mode. The composer controls the FPS-UIC_{v2} with the left hand and controls the ROLI Lightpad controller or MIDI keyboard controller with the right hand.

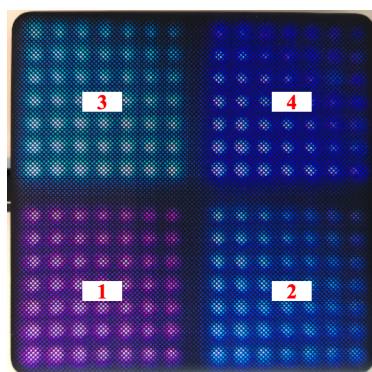


Figure 15. The four regions of the ROLI Lightpad controller.

In conclusion, the FPS-UIC_{v2} is completely re-designed with six regions. Initially, I wanted to use the FPS-UIC_{v2} as both musical mode controller and musical note performer; however, the screen space of the tablet was too small. Therefore, I introduced the Roli Lightpad controller to control the selected musical chords.

8.6.3 Composing and observing

This section describes the collected audio-visual data of the compositional process of ‘The Gamma’. ‘The Gamma’ was composed and observed between the 26th of February 2019 and the 12th of March 2019 and composed within 20 hours. Unfortunately, the majority of the compositional process focused on significant revisions of the FPS-UIC_{v2} rather than on the composition. Table 8 provides an overview of the 12 self-study videos and summarises the duration of the recordings. The self-study recordings include the whole compositional process. ‘The Gamma’ was mainly composed with the aid of the FPS-UIC_{v2} between the 4th video and the 9th video. These self-recordings record the human-computer interaction of the FPS-UIC_{v2}. The first three self-study recordings document significant issues of the FPS-UIC_{v2}, and the last three self-study recordings do not include the use of the FPS-UIC_{v2}. Therefore, videos four to nine were open coded, axial coded, and selective coded. The remaining self-study recordings were not coded⁸².

⁸² Appendix 8.3 provides additional information, and includes an overview of the open coding, axial coding, and selective coding.

Number	Video title	Date	Duration (h:m:s)
1	190226_The Gamma_r1	26 February 2019	59m 34s
2	190226_The Gamma_r2	26 February 2019	1h 9m 22s
3	190301_The Gamma_r3	1 March 2019	41m 51s
4	190305_The Gamma_r4	5 March 2019	1h 39m 25s
5	190305_The Gamma_r5	5 March 2019	1h 17m 35s
6	190305_The Gamma_r6	5 March 2019	1h 25m 18s
7	190307_The Gamma_r7	7 March 2019	2h 9m 27s
8	190308_The Gamma_r8	8 March 2019	2h 12m 4s
9	190308_The Gamma_r9	8 March 2019	2h 48m 28s
10	190308_The Gamma_r10	8 March 2019	2h 8m 6s
11	190311_The Gamma_r11	11 March 2019	1h 32m 5s
12	190312_The Gamma_r12	12 March 2019	1h 34m 2s
Total			20h 3m 57s

Table 8. An overview of all self-study videos of the third composition.

8.6.4 Summary of the audio-visual analysis

The compositional process of ‘The Gamma’ is analysed based on the audio-visual observation data. The observation data shows that the compositional process of ‘The Gamma’ started on the 26th of February 2019 and ended on the 12th of March 2019. A digital simulation of an acoustic piano, realised with the help of Native Instruments’ ‘The Grandeur’ Kontakt sample library instrument, performed ‘The Gamma’. No other acoustic instruments or virtual instruments were recorded or used. The FPS-UIC_{v2}, a MIDI keyboard controller, and the ROLI Lightpad controller control the virtual piano. The left hand is used to control the FPS-UIC_{v2}, the right hand is used to control both the

MIDI keyboard controller and the ROLI Lightpad controller. The selective coding distinguishes four steps in the compositional process: (1) Defining the musical form based on three defined emotional states; (2) Defining root notes and musical modes with the aid of the FPS-UIC_{v2}; (3) Composing melodic material with the aid of the MIDI keyboard controller; (4) Composing harmonic progressions with the aid of the ROLI Lightpad controller.

8.6.5 Evaluation

This section evaluates the impact of the FPS-UIC_{v2}. The analysis of the original musical composition and the self-study observation data form the basis of the evaluation. The outcomes of the evaluation describe possible software design revisions for the next iteration. The FPS-UIC_{v2} impacted upon the original composition, the compositional process, and the development of the musical ideas of ‘The Gamma’.

The FPS-UIC_{v2} is biased towards western popular music, and it affected the musical form of ‘The Gamma’. ‘The Gamma’ is a piano ballad song since the FPS-UIC_{v2} is biased towards this western popular musical style. Like the first two compositions ‘The Alpha’ and ‘The Beta’, ‘The Gamma’ was composed within a digital audio workstation (DAW), and the DAW followed western musical rulesets. The original music followed a western popular musical form, and it included popular western instrumentation. The harmony modulated within western musical modes, and the time signature included typical western music time signatures. The musical form of ‘The Gamma’ was structured based upon a storyline. The previous compositional process of ‘The Beta’ inspired me to re-use the principle of a storyline as a creative tool to structure ‘The Gamma’. While composing, I changed some storyline interpretations due to technical software issues and frustrating production circumstances. The

storytelling of the three emotional states and interpretation of the musical ideas functioned as a creative guideline. The design process of the story was dynamic, and it was not a fixed or written story that I followed from the very beginning of the composition.

The concurrent design of the FPS-UIC_{v2} and the production of ‘The Gamma’ impacted upon the compositional process. The concurrent development of new sounds and presets for virtual instruments while composing music was a natural part of my compositional process. However, I never included the concurrent development of a UIC within my compositional process. While composing, I realised that the FPS-UIC_{v2} had too many software bugs and glitches. Therefore, I was not able to compose ‘The Gamma’ with the pre-developed FPS-UIC_{v2}. As a consequence, I was forced to revise and to improve the software functionalities of the FPS-UIC_{v2} while composing. The process of ‘computing while composing’ had an adverse effect on the duration of the compositional process, my mood, and my creative decision-making process. My frustration level was very high; therefore, my focus shifted from composing ‘The Gamma’ to debugging and ‘fixing’ the FPS-UIC_{v2}.

On the other hand, ‘computing while composing’ was also inspirational. While composing, programming the FPS-UIC_{v2} inspired me to experiment with different chord and melody programming automatization solutions. As a result, I introduced eight completely new control change automation parameters that controlled the musical modes and root notes to my compositional process. The advantage of using the eight automatization parameters was to re-record a musical melody or harmonic progression on one track without re-programming the musical modes or root notes. However, the eight automatization parameters were prone to errors. I had to be very aware of the programming process, including all the new automatization parameters. As a

consequence, I was programming and fine-tuning musical scale and root note switches. Unfortunately, the frustrating process of impersonating the sound of an instrumentalist shifted to a new and equally frustrating programming process of scale switch and root note automatization. Moreover, the FPS-UIC_{v2} GUI did not provide a sufficient harmonic overview. While composing 'The Gamma', I had no overview of all performed musical modes and harmonic chords.

In conclusion 'computer while composing' and the newly introduced programming process demanded a vast amount of productional and compositional time. The eight automatization parameters for the realisation of the musical scale and root note progressions were prone to errors. The transmitted automatization information and the different interpretations of all of the received control changes cluttered the compositional process. The designed GUI was not intuitive, and I frequently lost the compositional overview. I was forced to create an overview of the composition and the compositional process, which slowed down the compositional process continuously.

Unfortunately, this affected the duration of the compositional process adversely. Playing and practising the musical material in real-time cost more time in comparison to copying and pasting the musical material. In addition, almost all musical melodies did not start on the first beat because the eight recorded control change information of the musical modes and root notes on the first beat transmitted the control change information to the FPS-UIC_{v2} with a small delay. Excitation of a MIDI note on the first beat was impractical because my real-time improvisational performance on the MIDI Keyboard controller was inaccurate. The FPS-UIC_{v2} is biased towards not performing a musical note on the first beat due to the technical delay of FPS-UIC_{v2} and due to the inaccuracy of my real-time MIDI keyboard performance. Moreover, playing the melodic material live is biased towards the use of different time signatures. While

playing the melody, I did not pay attention to the time signatures. I performed the rhythmic and dynamic patterns based upon the musical feeling I had in that particular moment of playing the MIDI keyboard controller. As a result, 'The Gamma' changed periodically between different time signatures.

In conclusion, the FPS-UIC_{v2} had an impact on the original composition, the compositional process, and the development of musical ideas of 'The Gamma'. It is biased towards western popular music, and the concurrent design of the FPS-UIC_{v2} and the production of the original music impacted upon the compositional process. The FPS-UIC_{v2} also had an impact upon the harmonic chords, musical modes, melodic material, time signatures, and the absence of the FPS-VI_{v3}. The FPS-UIC_{v2} limited the use of musical material, and the design was not intuitive. The compositional issues of 'The Gamma' possibly affected the development of the next FPS.

For the next iteration, I improved the FPS-VI_{v3} audio database and GUI by increasing the FPS-VI_{v3} database with stochastic performance recordings. The stochastic performance recordings were extracted from guitar recordings performed by Maas. I examined the extent to which the stochastic material impacts upon the authenticity of the sound of the FPS-VI_{v4}. Furthermore, I revised the time signature function of the FPS-VI_{v3} graphic user interface (GUI). The imperfect realisation of the time signature function was one of the central reasons why I did not succeed to incorporate the FPS-VI_{v3} within 'The Gamma'. I also planned to implement a tempo deviation functionality. It allows composing the next original composition in different tempi. Moreover, the next iteration revised the FPS-VI_{v3} sequencer window. New phrases need to be programmed more intuitively within the sequencer window. Unfortunately, the user interface of the sequencer window is, for me, not user-friendly.

For the next iteration, I also improved the FPS-UIC_{v2}. I abandoned the concept of modulating between musical modes, and I introduced the concept of modulating between musical scales. As a professional composer, it is easier for me to compose and to think in musical scales rather than in musical modes. Furthermore, I expanded the harmonic chord possibilities, and I only used 7th chords to modulate to another musical scale. The 6th chords are inversions of the 7th chords. Therefore, I removed the 6th chords, and this possibly secures a better organisation of all musical chords that occur on a musical scale. Furthermore, I expanded the possibilities of modulating to another musical scale, including minor7, major7, dim7, halfdim7, and dominant7 chords. I programmed both musical scale switches and musical chords that are true to the selected musical scale. The next compositional process excludes the use of the ROLI Lightpad controller, and I embedded the functionality of the Lightpad controller within the revised FPS-UIC_{v2}.

8.7 Fourth iterative design process: The Delta

The fourth iterative design process addresses the fourth original composition called ‘The Delta’, and it addresses the design of the FPS-VI_{v4} and the FPS-UIC_{v3}. The fourth cyclic process recapitulates the compositional issues of ‘The Gamma’ (8.7.1), describes the functional design of the FPS-VI_{v4} and the FPS-UIC_{v3} (8.7.2), outlines briefly the gathered musical data and observation data (8.7.3), summarises the musical description of ‘The Delta’ and the observation data (8.7.4), and evaluates the analysed data based on the research questions (8.7.5).

8.7.1 Defining the problem

This section briefly recapitulates the evaluation of ‘The Gamma’ and pays attention to the initial problem of this research. For the fourth iteration, I improved the FPS-VI_{v3} and FPS-GUI_{v2}. The FPS-VI_{v3} affected the compositional process adversely. The imperfect realisation of the time signature function is one of the main reasons why ‘The Gamma’ did not include the FPS-VI_{v3}. Therefore, I revised the time signature functionality, the tempo deviation functionality, and the sequencer window of FPS-VI_{v3} GUI. I increased the FPS-VI_{v3} database, including stochastic performance recordings. Moreover, I improved the FPS-UIC_{v2}. The UIC affected my compositional process and the development of musical ideas adversely. Therefore, I abandoned the concept of modulating between musical modes and introduced the concept of modulating between musical scales. The controller expands the harmonic chord possibilities and uses the 7th chords to modulate to another musical scale.

8.7.2 Designing the FPS

The fourth version of the FPS includes the fourth version of the FPS-VI_{v4} and the third version of the FPS-UIC_{v3}. Figure 16 illustrates a revised information stream, and it includes the FPS-VI_{v4} and the FPS-UIC_{v3}.

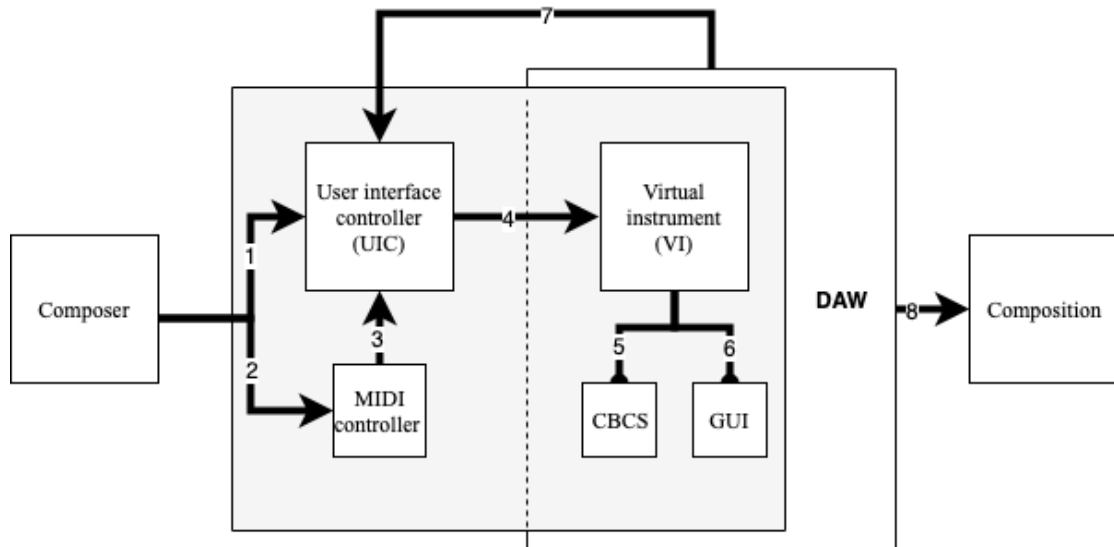


Figure 16. The information stream of the FPS-VI_{v4} and the FPS-UIC_{v3}.

The first (1) and second (2) arrow illustrate two different ways of interacting with the FPS-VI_{v4}. The composer controls the FPS-VI_{v4} (4) via the first option with the aid of the FPS-UIC_{v2} and controls the FPS-VI_{v4} via the second option with the aid of a MIDI keyboard controller. The MIDI keyboard controller transmits Note_On and Note_Off information to the user interface controller (3). Only the UIC connects to the DAW by MIDI mapping strategies. The MIDI keyboard controller alone connects to the FPS-UIC_{v3}. The DAW returns (7) MIDI control change information back to the FPS-UIC_{v3}, which ensures that the FPS-UIC_{v3} communicates appropriately with the DAW. The output of the interaction between the composer and the DAW (8) is an audio rendering of the original composition, in this case, ‘The Delta’.

The FPS-VI_{v4} includes an extension of the audio database. The audio database is extended by a sample corpus that includes 100 electric guitar phrases performed by Nico Maas. The performance phrases are 2 measures long, recorded at 120 bpm based upon one dynamic value. The recording circumstances of the performance phrases are identical to the recording circumstances of the single note database. However, the database only incorporates the stochastic audio material of the performance phrases; the decomposition removes the deterministic part. The system plays the stochastic material together with the single note recordings. The stochastic material performs the transitions between one note and another note; consequently, the sound of the electric guitar is more realistic. Figure 17 illustrates the construction and relationship between the stochastic audio material and the single note audio units.

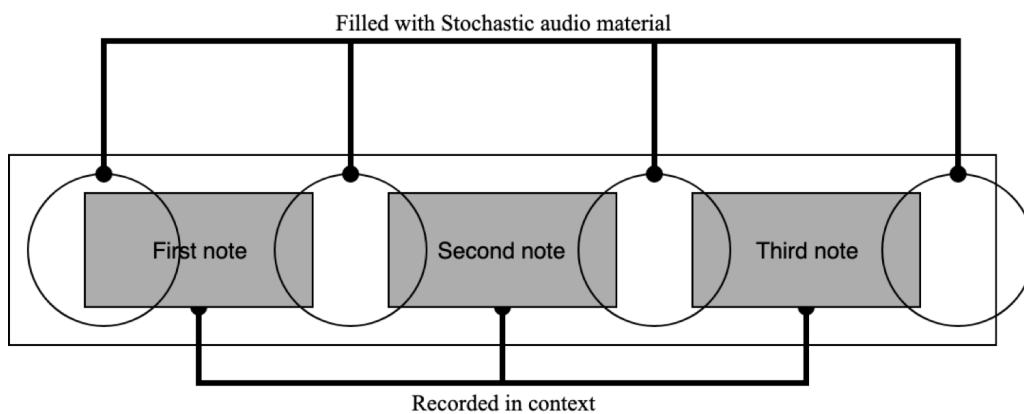


Figure 17. The relationship between stochastic audio material and single notes.

Moreover, the FPS-VI_{v4} also includes a significant revision of the graphic user interface (GUI) (Figure 18). The major revision was indispensable due to the insufficient and awkward human-computer interaction of the previous FPS-VI_{v3}. The newly designed GUI has four regions: (1) Sequencer window; (2) Phrase length; (3) Phrase preset menu; (4) Save button and clear button.

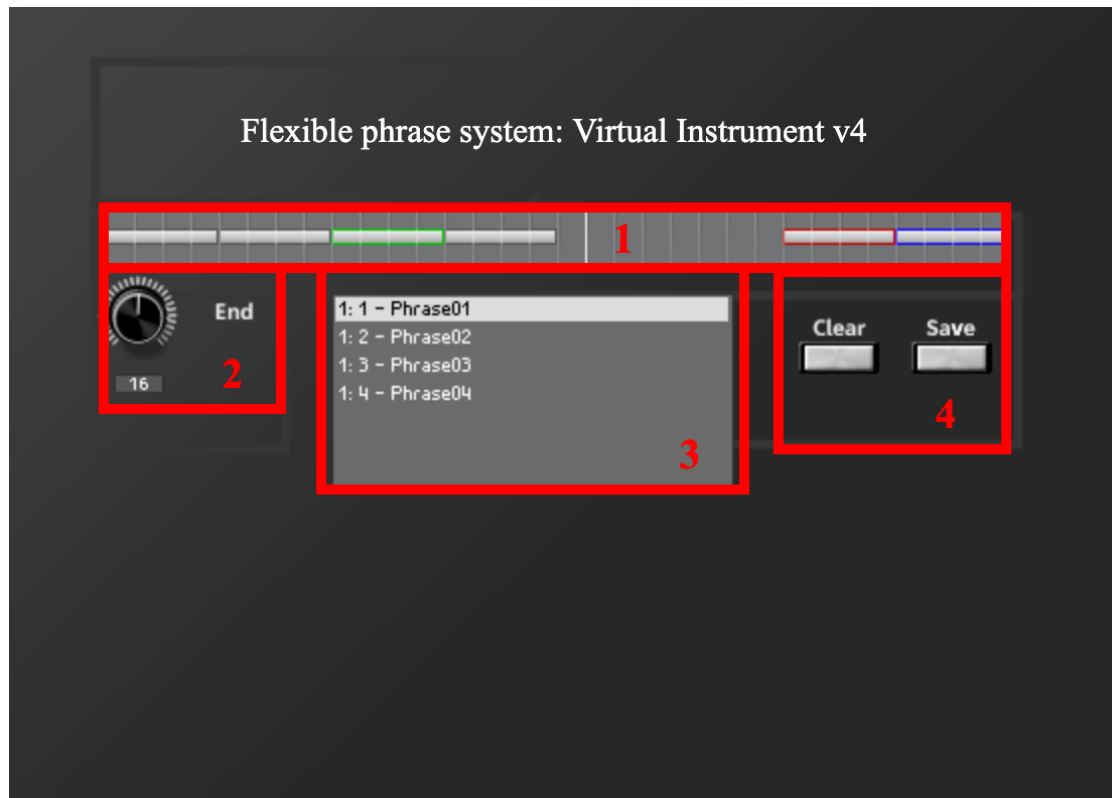


Figure 18. The four regions of the FPS-VI_{v4} GUI.

The first region includes the sequencer window, which forms the core of the FPS-VI_{v4}. The Cubase MIDI piano roll editor influences the design of the sequencer window. The grey or coloured rectangles within the sequencer window illustrate rhythmical values of musical notes, and a longer rectangle symbolises a longer rhythmic value while a shorter rectangle symbolises a shorter rhythmic value. Moreover, a different colour symbolises a different played articulation. The maximum length of the sequencer window is two measures. The phrase repeats after two measures. The second region includes a rotatory knob. The rotatory knob determines the length of the phrase, and it ensures an adjustable time signature. The functionality of the rotatory knob is comparable to the functionality of the rotatory knob in the previous versions. The third region includes the phrase preset menu, and it includes several more pre-designed musical phrases. The fourth region includes a save button and a clear button. The save

button saves newly designed phrases while the clear button removes phrases.

Furthermore, the tempo is adjustable, which means that the composer can switch between 60 bpm and 240 bpm. The composer changes the tempo within the DAW.

The second evaluation of the FPS-UIC_{v2} motivated me to revise the UIC fundamentally. The FPS-UIC_{v3} transposes incoming MIDI_On and MIDI_Off messages from an external MIDI keyboard controller. The external MIDI keyboard controller sends MIDI_On and MIDI_Off messages to the FPS-UIC; however, it does not send MIDI messages directly to the digital audio workstation (DAW). This MIDI mapping strategy ensures that the notes played on a MIDI keyboard controller are always linked to the selected root note and musical scale. In doing so, only the white keys of the MIDI keyboard controller are used, and the black keys are disabled. The white keys are transposed based on the selected root note and musical scale (Table 9).

Key	No transposition (Scale)	+1 semitone (Scale)	-1 semitone (Scale)
C	(C, G, F, B ^b , E ^b , A ^b , D ^b)	C [#] (B, D, A, E)	C ^b (G ^b)
D	(D, A, G, C, F, B ^b , E ^b)	D [#] (B, E)	D ^b (D ^b , A ^b , G ^b)
E	(E, B, A, D, G, C [#] , F)	E [#] (F [#])	E ^b (D ^b , E ^b , B ^b , G ^b)
F	(F, C, B ^b , E ^b , A ^b , D ^b , G ^b)	F [#] (F [#] , G, D, A, E, B)	-
G	(G, D, C, F, B ^b , E ^b , A ^b)	G [#] (A, E, B, F [#])	G ^b (D ^b , G ^b)
A	(A, E, D, G, C, F, B ^b)	A [#] (B, F [#])	A ^b (A ^b , D ^b , E ^b , G ^b)
B	(B, E, A, G, D, C, F [#])	-	B ^b (F, B ^b , E ^b , A ^b , D ^b , G ^b)

Table 9. Overview of all pitch transpositions.

For example, the user selects the A major scale on the FPS-UIC. The musical notes that are true to the A major scale are: A, B, C[#], D, E, F[#], and G[#]. The FPS-UIC transposes the white ‘C’ key on the MIDI keyboard controller one semitone higher to

‘C#’, the white ‘F’ key to ‘F#’, and the white ‘G’ key to ‘G#’. The user plays a white ‘C’, ‘F’ or ‘G’ key, but the user hears a transposed note; therefore, the three transposed played notes no longer correspond to the keys on the MIDI keyboard controller.

The revised FPS-UIC_{v3} has eight regions (Figure 19): (1) Selecting a root note; (2) Selecting an inversion; (3) Selecting a flexible phrase; (4) Selecting a musical chord to modulate; (5) Selecting a musical chord to start; (6) Performing the root note, the first note, the second note, or the third note of a musical chord; (7) Performing musical notes that occur within the selected musical scale; (8) Transposing the selected musical chord to another octave.

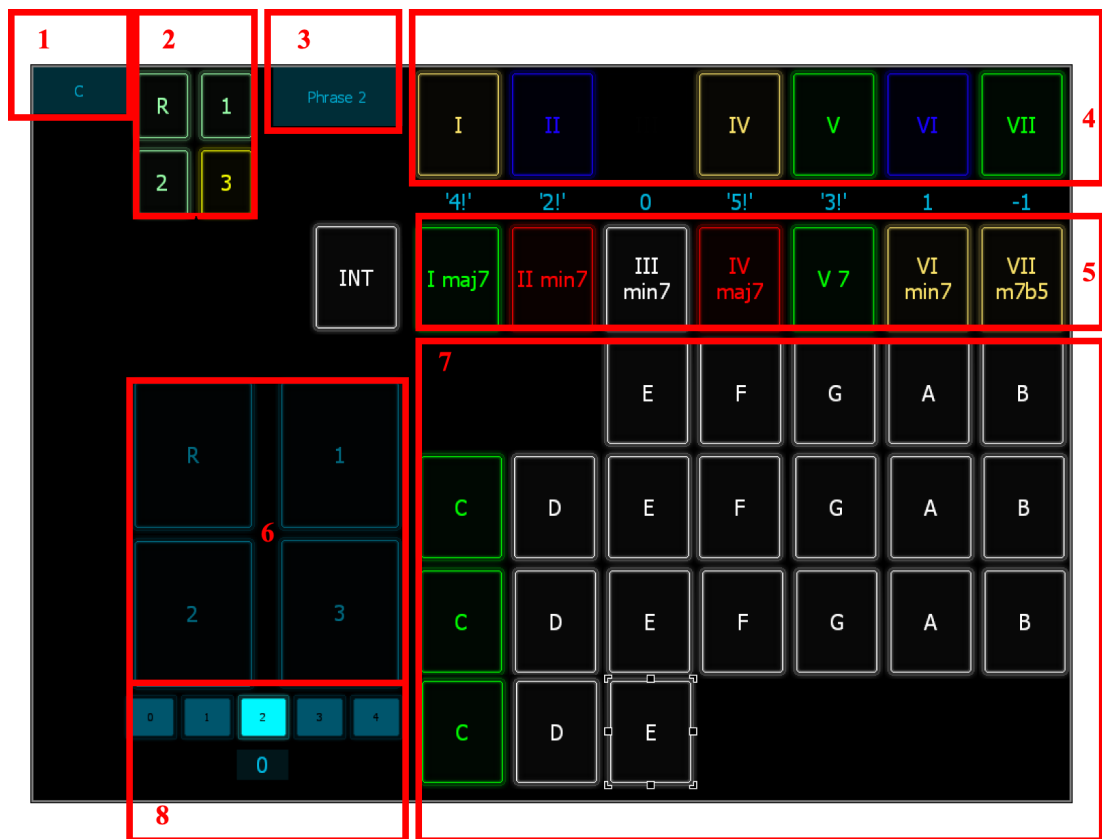


Figure 19. The eight regions of the FPS-UIC_{v3}.

The first and second regions define the first musical chord. In region 1, the composer selects the tonic and the major scale. In contrast to the previous version (FPS-

UIC_{v2}), the FPS-UIC_{v3} does not select different musical modes; instead, it selects musical scales. The musical scale is always a major scale. In this example, ‘C’ major is the selected scale. The areas in green within region 7 also highlights the tonic. In this case, ‘C’ is highlighted in green. In region 2, the composer selects the inversion of the musical chord. ‘R’ represents ‘Root position’, ‘1’ represents the first inversion, ‘2’ represents the second inversion, and ‘3’ represents the third inversion. In this example, the third inversion is selected and highlighted in green.

The third region includes the phrase preset selector of the FPS-VI_{v4}. The composer clicks on one of the buttons, and then he/she selects a pre-defined phrase. In doing so, the FPS-UIC_{v2} sends control change information to the DAW and FPS-VI_{v3}. The controller then selects different pre-designed phrase presets of the virtual instrument. In total, the composer can switch between 15 different pre-designed phrases. In this example, the second phrase preset (Phrase 2) is selected.

The fourth and fifth region define the musical chord and scale modulation. Both Region 4 and Region 5 display the seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The major, minor, dominant 7th, or half-diminished 7th chords always occur on the same position within the musical scale. Table 10 provides an overview of the seven 7th chords.

Scale degree	Relative chord	Example (region 5)
I	Maj ⁷	C maj ⁷
II	Min ⁷	D min ⁷
III	Min ⁷	E min ⁷
IV	Maj ⁷	F maj ⁷
V	Dominant ⁷	G ⁷
VI	Min ⁷	A min ⁷
VII	M ⁷ _{b5}	B m ⁷ _{b5}

Table 10. An overview of the seven 7th chords that occur within a musical scale.

Furthermore, the button that is coloured in white displays the selected chord. In this example, the third chord in Region 5 (III min⁷) is the selected chord. Both region 4 and region 5 have the function of selecting the 7th chord; however, there is one crucial difference between the first chord row (region 4) and the second chord row (region 5): The chord that is coloured in white signals both the selected chord and the selected chord row. In the example, the chord row in region 5 is selected because the third chord (III min⁷) is highlighted in white. A horizontal chord switch within a chord row including the white coloured chord affects chord switches within the selected musical scale. No matter which chord within the chord row in region 5 is selected, the musical tonic always stays the same. A vertical chord switch (moving from region 5 to region 4 or from region 4 to region 5) always creates a modulation to another musical tonic. The numbers 'in-between' region 4 and region 5 display the positions that the tonic moves by switching from region 5 to region 4 within the Cycle of Fifth (Figure 20). A positive number means a clockwise movement within the Cycle of Fifth, while a negative number means a counter-clockwise movement within the Cycle of Fifth. The tonic is changing from the perspective of the selected chord, in this case, the E min⁷ chord.

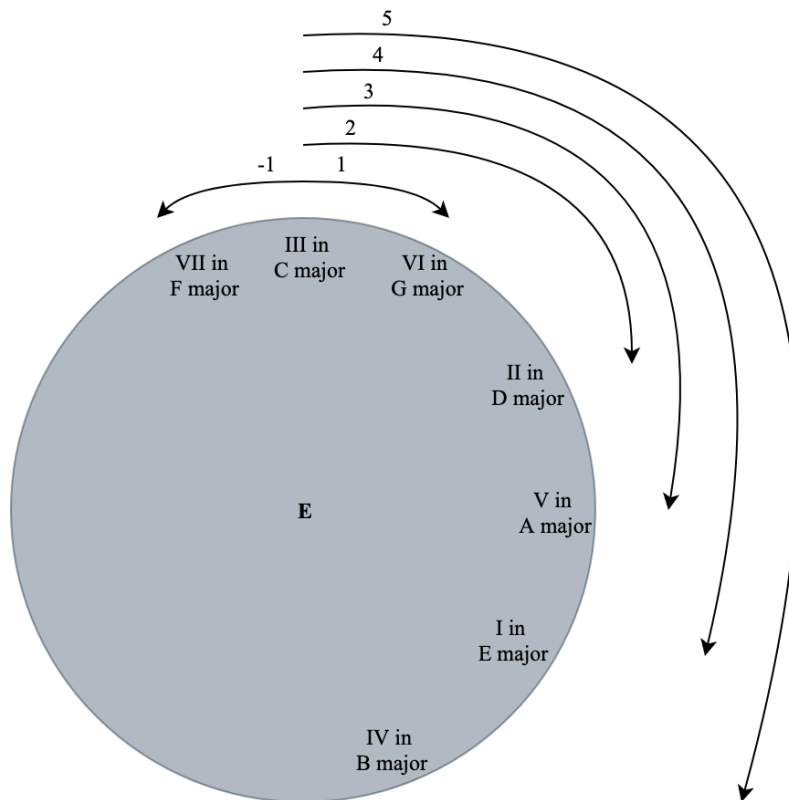


Figure 20. The movement of the E chord within the Cycle of Fifth.

Moreover, the chord buttons in region 4 and region 5 include five different colours: white, blue, green, yellow, and red. The white coloured chord button displays the selected musical chord. The colours of all the other chords change depending on the selected chord. The colours signal how many musical notes change by moving from the selected chord to another chord. The blue colour signals that none of the four musical notes changes, the green colour signals that one of the four musical notes change, the yellow colour signals that two of the four musical notes change, and red signals that three of the four musical notes change.

The sixth and seventh regions enable the performance of musical notes. In region 6, the composer controls the four different notes of a 7th chord. The composer can perform different combinations of the seventh (3), fifth (2), third (1) or tonic note (R) of the chord. These four buttons replace the functionality of the ROLI Lightpad

MIDI controller. This composition excludes the ROLI Lightpad MIDI controller. In region 7, the composer performs musical notes that occur naturally within the selected major scale. It enables monophonic, homophonic, or polyphonic performances. The lowest note starts with E₂ (at the top of the ninth region) and ends with the highest note E₅ (at the bottom of the ninth region). The displayed ambitus is based upon the recorded range of the FPS-VI_{v4} audio database because the lowest recorded note is an E₂ note and the highest recorded note is an E₅ note. The displayed musical notes change based upon the selected tonic. In this example, all musical notes that occur within the C major scale are displayed and enabled for the composer.

In region eight, the composer selects the octave in which the system performs the selected musical chord. The row includes five numbers: 0, 1, 2, 3, and 4, which are inside five blue squares. An additional number is located below the row. The row shifts all notes of the selected musical chord one or two octaves higher or lower. '0' transposes the selected musical chord two octaves lower, '1' transposes the selected musical chord one octave lower, '2' does not transpose, '3' transposes the selected musical chord one octave higher, and '4' transposes the selected musical chord two octaves higher. In this example, the light blue colour indicates the selected button (number 2), and the number '0' below the row indicates that the system does not transpose the selected musical chord.

In conclusion, the fourth version of the FPS includes the fourth version of the FPS-VI_{v4} and the third version newly designed FPS-UIC_{v3}. I expanded the FPS-VI_{v4} audio database and revised the GUI. The third version of the FPS-UIC_{v3} is also fundamentally revised.

8.7.3 Composing and observing

This section describes the collected audio-visual data of the compositional process of ‘The Delta’. ‘The Delta’ was composed and observed on the 24th, 25th, and 26th of April 2019, and realised within 9 hours. Table 11 provides an overview of the six self-study videos, and it summarises the duration of all recordings. All self-study video recordings include the use of both the FPS-VI_{v4} and the FPS-UIC_{v3}, and all self-recordings were open coded, selective coded, and axial coded⁸³.

Number	Video title	Date	Duration (h:m:s)
1	190424_The Delta_r1	24 April 2019	54m 36s
2	190424_The Delta_r2	24 April 2019	1h 27m 19s
3	190425_The Delta_r3	25 April 2019	41m 45s
4	190425_The Delta_r4	25 April 2019	2h 5m 31s
5	190425_The Delta_r5	25 April 2019	1h 21m 52s
6	190426_The Delta_r6	26 April 2019	2h 39m 46s
Total			9h 10m 49s

Table 11. An overview of all self-study videos of the fourth composition.

8.7.4 Summary of the audio-visual analysis

The compositional process of ‘The Delta’ is analysed based on the audio-visual observation data. The observation data shows that the compositional process of ‘The Delta’ starts on the 24th of April 2019 and ends on the 26th of April 2019. The FPS-

⁸³ Appendix 8.3 provides additional information, and includes an overview of the open coding, axial coding, and selective coding.

UIC_{v3} controls the FPS-VI_{v4}, and both, the FPS-UIC_{v3} and a MIDI keyboard controller control the virtual piano. The previous composition ‘The Gamma’ used the same virtual piano. I produced all sounds of ‘The Delta’ within ‘the box’, and I did not record other instruments for this composition. The left hand is used to control the FPS-UIC_{v3}, and the right hand is used to control the MIDI keyboard controller. Furthermore, the selective coding distinguishes three steps in the compositional process: (1) Structuring the composition based on the storyline; (2) Performing musical scales and musical modulations with the aid of the FPS-UIC_{v3}; (3) Performing musical melodic and harmonic progressions with the FPS-VI_{v4} and the virtual piano.

8.7.5 Evaluation

This section evaluates the impact of the FPS-VI_{v4} and the FPS-UIC_{v3}. The evaluation centres on both the analysed original musical composition and the analysed self-study observation data. The outcomes of the evaluation describe possible software design revisions for the next iteration. The FPS-VI_{v4} and FPS-UIC_{v3} had an impact on the original composition, the compositional process, and the development of musical ideas of ‘The Delta’.

The FPS-VI_{v4} and FPS-UIC_{v3} are biased towards western popular music, which affected the musical form of ‘The Delta’. ‘The Delta’ is a piano and electric guitar ballad song. ‘The Delta’ was, comparable to the first three compositions, composed within a DAW which followed western musical rulesets. The original music followed a western popular musical form, and it included two electric guitars and a piano which are western popular musical instruments. The harmony modulated within western musical scales, and the time signature included typical western music time signatures. The musical form of ‘The Delta’ was, in contrast to ‘The Beta’ and ‘The Gamma’, not

structured based on a storyline, but it was biased towards a sequence of different modulations. The musical form focused on a composed melody, and the FPS-VI_{v4} sequencer window was responsible for the design of the melody.

The FPS-VI_{v4} graphical user interface (GUI) and the musical performance limitations of the database impacted upon 'The Delta'. The completely revised FPS-VI_{v4} GUI was biased towards the concept of the DAW's piano roll editor view. The sequencer window of the fourth version enabled the fast design of musical melodies and rhythmic patterns. From this followed that the FPS-VI_{v4} inspired me to use the virtual instrument to design the leading musical melodies of 'The Delta'. However, the FPS-VI_{v4} did not provide sufficient dynamic deviations. The lack of dynamic deviations forced me to use post-processing digital signal processing frequency filters and volume envelope automation to emulate dynamic changes. Unfortunately, this cost a lot of production time. Moreover, 'The Delta' was composed in 90 bpm. Unlike 'The Alpha', 'The Beta', and 'The Gamma', it was the first composition in this study that was not performed in 120 bpm because the FPS-VI_{v4} was capable of performing different tempi. This new functionality inspired me to test a different tempo for 'The Delta'.

The sampled instrumentalist affected 'The Delta'. 'The Delta' used pre-recorded performances of Maas. The FPS-VI_{v4} embedded both stochastic material and single note recordings. Maas indirectly influenced the composition of 'The Delta' with his pre-recorded phrases. The composition was biased towards the pre-recorded phrases because the accompaniment was produced based on the pre-recorded phrases. One could argue that Maas owned 'related-rights' of 'The Delta'. Maas also listened to the 'The Delta', and he perceived the played electric guitar as his performance. However, he had never precisely played these musical phrases before.

The FPS-UIC_{v3} GUI also had an impact on ‘The Delta’. The FPS-UIC_{v3} is biased regarding the length of the musical work and the used instrumentation. The initial idea of ‘The Delta’ included a 10 minute long digital orchestral original work. In order to create digital orchestral works, I started with a piano, and the piano performed both melody and harmony. Both harmony and melody were then replaced and performed by other orchestral instruments. However, the final result of ‘The Delta’ was three minutes and 11 seconds long, and the instrumentation included a piano and two electric guitars. I started ‘The Delta’ with a piano sketch, but the FPS-UIC_{v3} provided insufficient possibilities to transcribe a piano performance to an orchestral arrangement. The FPS-UIC_{v3} was built to control the musical scales and chords and to control the FPS-VI_{v4}. However, the FPS-UIC_{v3} was not built and capable of transmitting expression-related information from a piano to an orchestra. Therefore, transcribing a piano sketch to an orchestra did not work; consequently, I stopped the composition after around three minutes.

The FPS-UIC_{v3} was biased towards controlling the FPS-VI_{v4}. The UIC combines both the performance of musical scales and root notes and the performance of musical notes of virtual instruments on one display. In contrast to the FPS-UIC_{v1} and the FPS-UIC_{v2}, the FPS-UIC_{v3} provided a better overview of the screen. The organisation and design of the user interface felt more logical to me. The note execution window visualised the ambitus of the FPS-VI_{v4}, and it provided an overview of all playable musical notes that occurred naturally in the selected musical scale. As a result, the FPS-UIC_{v3} shifted the main focus to the UIC. It avoided dividing the composer’s attention between different hardware units such as the Roli Lightpad controller or the MIDI keyboard controller. The focus on the UIC had a positive effect on my compositional process; moreover, I did not have to multitask between two or more

different musical hardware controllers. From this followed that the Roli Lightpad controller and the MIDI keyboard controller were no longer necessary to control the FPS-VI_{v4}. However, a piano sample library that excluded the automatization of expression-related parameters did not work well together with the FPS-UIC_{v3}. It is highly advisable to use a MIDI keyboard controller to control a piano sample library because a MIDI keyboard controller includes the physics of a piano, and it transmits expression information with aftertouch. In contrast to the realistic and convincing sound of the electric guitar⁸⁴, the sound of the piano, controlled with the FPS-UIC_{v3}, did not convince me. The FPS-UIC_{v3} was not primarily designed for controlling a piano sound. It was designed to control the FPS-VI_{v4} and to distinguish between different musical elements, e.g. ostinato patterns, melodies, and homophonic chords.

In conclusion, the FPS-UIC_{v3} did have an effect on the compositional process. The composer needed to use FPS-UIC_{v3} appropriately, and it was not an all-around user interface that was capable of controlling the expression of all virtual instruments. For example, the MIDI keyboard is a more logical controller for piano libraries rather than the FPS-UIC_{v3}.

The FPS-UIC_{v3} impacted upon the use of harmonic chords and musical scale modulations. In contrast to the previous two compositions, ‘The Delta’ did not use arbitrarily selected scale modulations because I first monitored the scale modulation and then record it. In ‘The Beta’ and ‘The Gamma’ it was the other way around: I first created (based on music theory) the modulations and then I monitored the musical

⁸⁴ The newly designed musical guitar phrases sound, according to Maas, incredibly realistic.

result. As a result, the harmonic progression of ‘The Delta’ sounded less arbitrary to me, and it formed a harmonic and melodic musical coherence. The FPS-UIC_{v3} accelerated the monitoring process of testing different musical ideas in different musical scales and root notes. Moreover, the FPS-UIC_{v3} created new possibilities to explore new musical ideas. The FPS guided me in using different harmonic progressions. It inspired me to use, for example, a combination of a D⁷sus4 chord to modulate three positions within the Cycle of Fifth to another key without changing any notes in the triggered D⁷sus4 chord.

In addition, the FPS-VI_{v4} and the FPS-UIC_{v3} also impacted upon the original composition, the compositional process, and the development of musical ideas of ‘The Delta’. The FPS is biased towards western popular music, and the musical performance limitations of the FPS-VI database and the graphical user interface (GUI) impacted upon ‘The Delta’. Moreover, the sampled instrumentalist and the FPS-UIC_{v3} GUI affected the fourth composition. The FPS-UIC_{v3} was biased towards the design of the FPS-VI_{v4}, and it impacted the use of harmonic chords and musical scale modulations. The compositional issues of ‘The Delta’ possibly impacted upon the development of the next FPS.

The next iteration improves the FPS-UIC_{v3}. I continued the compositional ideas of ‘The Delta’ in the next composition called ‘The Omega’. However, I did not start with a piano sketch but directly routed the voicing of the revised version of FPS-UIC_{v3} to orchestral instruments. Moreover, I included a selection of different voicings, and I automated the routing of the voicings to different instruments of the orchestra. The option of selecting a narrow voicing is insufficient for the compositional process. Furthermore, I improved the DAW implementation of the revised version of the FPS-

UIC_{v3} improves. The compositional process of the fourth composition has shown that the UIC still creates some bugs and glitches, and it interrupts the compositional process.

8.8 Fifth iterative design process: The Omega

The fifth cyclic process addresses the fifth original compositional called ‘The Omega’ and the design of the FPS-UIC_{v4}. The iterative design process recapitulates the compositional issues of ‘The Delta’ (8.8.1), describes the functional design of the FPS-UIC_{v4} (8.8.2), outlines briefly the gathered musical data and observation data (8.8.3), summarises the musical description of ‘The Omega’ and the observation data (8.8.4), and evaluates the analysed data based on the research questions (8.8.5).

8.8.1 Defining the problem

This section briefly recapitulates the evaluation of ‘The Delta’. The recapitulation pays attention to the initial problem of this research. The fifth and last iteration slightly improves the FPS-UIC_{v4}. The FPS-UIC_{v3} affected the composition, the compositional process, and the development of melodic and harmonic ideas of ‘The Delta’ adversely. Therefore, I included the selection of different voicings, and I automated the routing of the voicings to different virtual instruments. The option of selecting a narrow voicing was insufficient for my compositional process.

8.8.2 Designing the FPS

The fifth version of the FPS includes the fourth version of the FPS-UIC_{v4}. Figure 21 illustrates a revised information stream; however, it excludes the FPS-VI because ‘The Omega’ does not include the virtual instrument.

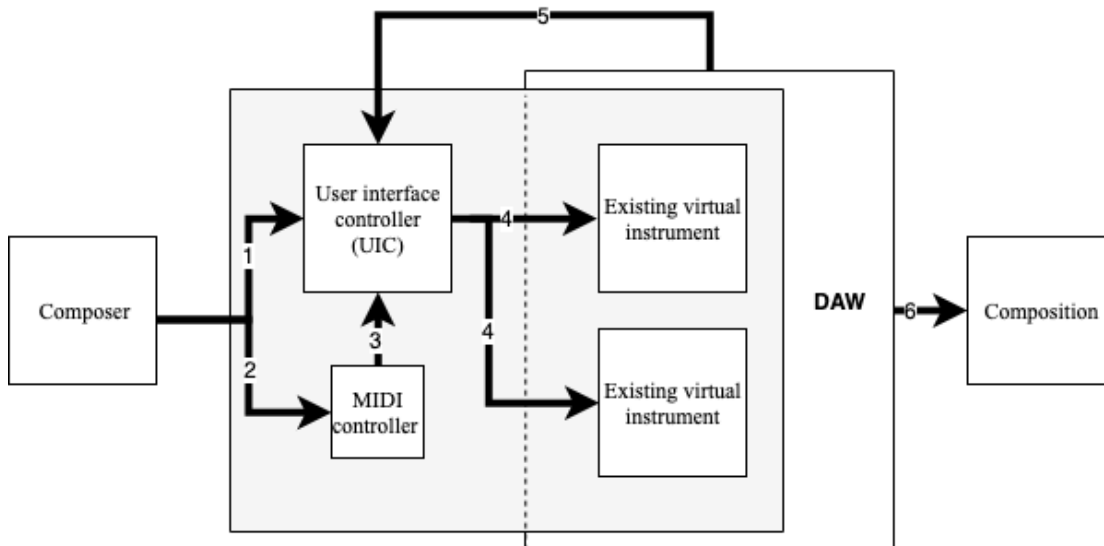


Figure 21. The information stream of the FPS-UIC_{v4}.

The first (1) and second (2) arrow illustrate two different ways of interacting with existing virtual instruments. The composer controls the existing virtual instruments (4) via the first option with the aid of the FPS-UIC_{v4}. The composer controls the existing virtual instruments via the second option with the aid of a conventional MIDI keyboard controller that transmits Note_On and Note_Off information to the user interface controller (3). Only the FPS-UIC_{v4} is connected to the DAW by MIDI mapping strategies. The MIDI keyboard controller only connects the FPS-UIC_{v4}. The DAW returns (5) MIDI control change information back to the FPS-UIC_{v4}, which ensures that the controller communicates appropriately with the DAW. The output of the interaction between the composer and the DAW (6) is an audio rendering of the fifth and final original composition ‘The Omega’.

The FPS-UIC_{v4} has two additional regions (Figure 22): (1) Selecting different voicings; (2) Performing the selected chord with one button.

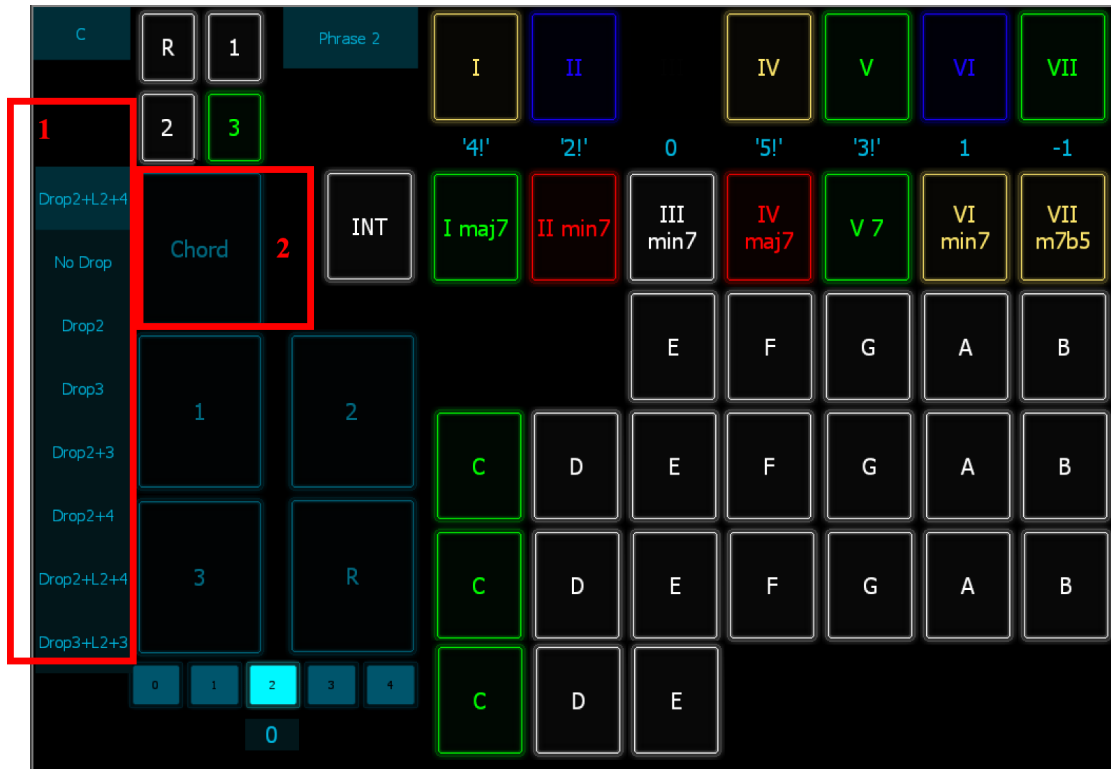


Figure 22. The fourth version of the user interface controller.

The first region enables the selection of different voicings. The composer chooses between seven different voicing options: (1) No drop; (2) Drop2; (3) Drop3; (4) Drop2+3; (5) Drop2+4; (6) Drop2+L2+4; (7) Drop3+L2+3. The voicings are designed based on drop voicings. A chord in root position and close voicing creates the drop voicing. One or more notes of the chord are dropped down an octave. For example, by dropping down the second note from the top, the voicing is called a Drop2 voicing (see Figure 23). It is important to understand that a Drop2 voicing affects the inversion of the chord. In Figure 23, the lowest note of the Drop2 E min⁷ chord is 'B' (the fifth note of the chord). The chord has now changed from a root position chord to a *Drop2 2nd inversion* chord as the fifth note is the lowest note of the chord.

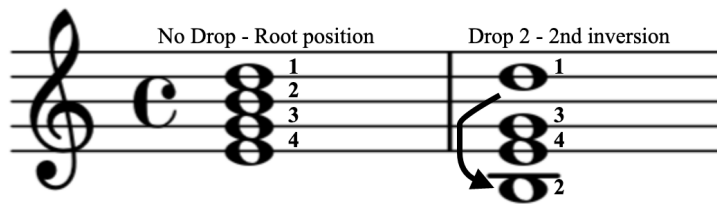


Figure 23. Drop voicing example.

The FPS-UIC_{v4} voicing options are designed based on drop voicings. The composer chooses between seven different voicing options: (1) No drop; (2) Drop2; (3) Drop3; (4) Drop2+3; (5) Drop2+4; (6) Drop2+L2+4; (7) Drop3+L2+3. The first five voicings include four notes (the first, third, fifth and seventh note of a chord). The last two voicing options (Drop2 +L2+4 and Drop3+L2+3) include nine pitches. Four notes are played in the upper register (above Middle C) while five notes are played in the lower register (below Middle C). The Drop2+L2+4 includes a Drop2 voicing in the upper register, and a Drop2+4 voicing in the lower register (marked by an 'L'). The lowest note is doubled by an octave in the lowest register. Figure 24 illustrates all voicings including different inversions with the help of the E min⁷ example.

The figure displays four rows of musical notation for the E min7 chord, each row representing a different inversion. The rows are labeled 'Root', '1. inversion', '2. inversion', and '3. inversion'. Each row contains seven columns of notes. The first five columns are labeled 'No Drop', 'Drop2', 'Drop3', 'Drop2+3', and 'Drop2+4'. The last two columns are labeled 'Drop2+L2+4' and 'Drop3+L2+3'. The notes are arranged in a way that shows the relationship between the different voicings and the effect of the 'Drop' and 'L2+4'/'L2+3' techniques.

Figure 24. All voicing, including four different inversions (E min⁷).

The second region enables the performance of a musical chord with one button. The composer triggers the ‘chord’ button, and the FPS-UIC_{v4} transmits MIDI information to the DAW. The MIDI information includes a minimum of four different pitches of a 7th chord. In the example illustrated in Figure 22, the selected tonic is ‘C’, the selected chord is the 3rd inversion of an E min⁷ chord, including a Drop2+L2+4 voicing, which means that the FPS-UIC_{v4} transmits nine notes (four different pitch values) to the DAW. Moreover, the FPS-UIC_{v4} automatically routes the nine musical notes to nine different virtual instruments. The following Figure 25 illustrates the routing of nine different musical notes.

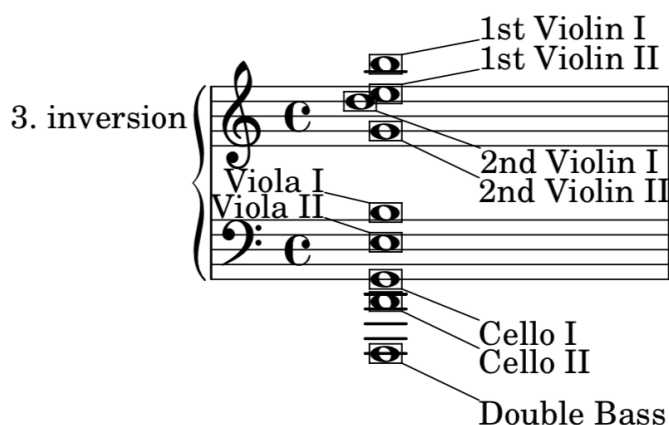


Figure 25. The transmitted musical notes of a Drop2+L2+4 voicing to the DAW.

The FPS-UIC_{v4} maps the musical notes in the upper register to the 1st Violin I, 1st Violin II, 2nd Violin I and 2nd Violin II. Moreover, the UIC maps the musical notes in the lower register to the Viola I and Viola II, Cello I, Cello II and Double Bass (five strings). The instrumentation is not limited to the String family; consequently, the FPS-UIC_{v4} can also map the voicing to other virtual instruments such as the Woodwind section, Brass section, electric guitars, or synthesisers.

In conclusion, the revised FPS-UIC_{v4} includes two additional regions. The first region enables the selection of different voicings while the second region enables the performance of a musical chord with one button. Moreover, the FPS-UIC_{v4} transmits musical notes of a musical chord to individual virtual instruments.

8.8.3 Composing and observing

This section describes the collected audio-visual data of the compositional process of ‘The Omega’. I composed ‘The Omega’ on the 29th and the 30th of April 2019, and I finished the sixth composition within six hours. Table 12 provides an overview of the self-study recording. On the 29th of April 2019, I used the FPS-UIC_{v4} for two hours. The

self-study recording includes the compositional process with the aid of the FPS-UIC_{v4}, and the self-recording was open coded, axial coded, and selective coded⁸⁵.

Number	Video title	Date	Duration (h:m:s)
1	190429_The Omega_r1	29 April 2019	2h 8m 57s
Total			2h 8m 57s

Table 12. An overview of all self-study videos of the fifth composition.

8.8.4 Summary of the audio-visual analysis

The compositional process of the last composition ‘The Omega’ is analysed based on the audiovisual observation data. According to the audiovisual observation data, the compositional process starts on the 29th of April 2019 and ends on the 30th of April. The compositional process of ‘The Omega’ is broken down into two parts:

The first part focuses on the use of the FPS-UIC_{v4}, and the second part focuses on the productional process of the virtual instruments. Moreover, ‘The Omega’ utilises all orchestral sounds within ‘the box’, controlled by both the MIDI keyboard controller and the FPS-UIC_{v4}. The compositional process of the final original composition comprises the following three steps: (1) Composing the harmonic progressions and homophonic voicings; (2) Producing dynamic movements and other expressive information of the virtual instruments; (3) Composing unison-octave melodies upon the harmonic progressions and homophonic voicings. The harmonic progression was designed and composed with the aid of the FPS-UIC_{v4}. A MIDI keyboard controller, computer

⁸⁵ Appendix 8.3 provides additional information, and includes an overview of the open coding, axial coding, and selective coding.

mouse, and computer keyboard were used to trigger expressive information such as dynamic movements.

8.8.5 Evaluation

This section evaluates the impact of the FPS-UIC_{v4}. The analysed original musical composition and the analysed self-study observation data define the evaluation. The FPS-UIC_{v4} impacted upon the original composition, the compositional process, and the development of musical ideas for ‘The Omega’.

The FPS-UIC_{v4} is biased towards western orchestral music. The original orchestral composition included strings, brass, and woodwinds. It excluded the use of the FPS-VI because, in my opinion, the sound of an electric guitar did not fit well in this orchestral musical work. ‘The Omega’ was, comparable to ‘The Alpha’, ‘The Beta’, ‘The Gamma’, and ‘The Delta’, composed within a DAW, and the DAW followed western musical rulesets. ‘The Omega’ followed a western musical form, and it included western orchestral instruments. The harmony modulated within western musical scales, and the time signature included typical western music time signatures.

The use of the FPS-UIC_{v4} biased the compositional process. The compositional process included the following three steps: (1) Composing the homophonic movement; (2) Producing dynamic movements; (3) Composing unisono-octave melodies upon the harmonic progression. The first step included the design of the homophonic chord progression. In this step, I composed the chord progression with the aid of the FPS-UIC_{v4} ‘One Button’ and the chord menu. The second step included the production of dynamic movements. The modulation wheel of the MIDI keyboard controller generated dynamic movements because the display of the FPS-UIC_{v4} tablet did not include a physical fader or a physical modulation wheel that corresponded to the dynamic

information of the virtual instruments. The third step included the production of unison-octave melodies. The single note input buttons of the FPS-UIC_{v4} performed the unison-octave melodies. Moreover, the FPS-UIC_{v4} was for the first time in the context of this study effectively integrated within the DAW, which had a significant impact on the composition process. There was no need to revise the software while composing; moreover, I perceived the human-computer interaction approach for the first time in this study as pleasant and productive.

The FPS-UIC_{v4} affected the development of harmonic voicings. The FPS-UIC_{v4} worked well as a compositional controller; moreover, it enabled multiple recording of different instrument sections, including different articulations (sustained and accented) simultaneously. It provided the option to select or mute different recorded musical sections and articulations afterwards, which was very practical and beneficial for the creation of the different voicings of the musical composition. The multiple recording strategy had an impact on the development of different musical ideas. In ‘The Omega’, the recording strategy enabled fast switches between low register instrument sections and high register instrument sections. Switching between different instrument registers fast helped to determine which instrument group has to play which musical part. Musical segment C is an excellent example of different instrument registers that share the same musical ideas. Moreover, the FPS-UIC_{v4} mainly centred on homophonic and monophonic writing. The homophonic voicing button enabled complex real-time arrangements. These complex arrangements were not possible with a conventional MIDI keyboard controller because my hands were not able to play multiple voicings, spread over several octaves, simultaneously.

Moreover, the real-time complex homophonic voicing playback function enabled the fast development of new musical ideas. It enabled the monitoring process of

different musical voicing options in real-time, and it opened the possibility to test different musical homophonic ideas very fast. It was very challenging to imagine a sophisticated voicing in my head without actually hearing it. The FPS-UIC_{v4} solved that problem, and I was able to hear sophisticated voicings in real-time. The complex voicing arrangement was beneficial to the overall audio quality of the MIDI orchestration, and it ensured, in my opinion, a sufficient and realistic orchestral sound. Unfortunately, the FPS-UIC_{v4} was less applicable for polyphonic writing; therefore, 'The Omega' did not perform polyphonic voicings.

The FPS-UIC_{v4} also affected the use of harmonic chords and musical scale modulations. The FPS-UIC_{v4} shifted the focus from the 'sound' of the harmonic chords to the music theory used. While using the FPS_{v4}, I was not aware of which harmonic chords I played. However, the display provided sufficient information to query the used harmonic information. In measure 36, for example, I composed an extreme key modulation from one major scale to another major scale. While composing, I paid attention to the impact of the key modulation based on my understanding of musical feeling; however, I did not pay attention to how many positions the musical key moves within the Cycle of Fifth. Moreover, the newly designed 'one-button' function on the FPS-UIC_{v4} performed complex homophonic musical chords, including nine pitch values with one button. This functionality affected the production of the long sustained orchestral introduction of 'The Omega' because the FPS-UIC_{v4} provided easy access to create long sustained musical notes that worked well at the beginning of this fifth composition.

In conclusion, the FPS-UIC_{v4} affected the original composition, the compositional process, and the development of musical ideas of 'The Omega'. The FPS is biased towards western orchestral music. The FPS-UIC_{v4} affected the performance of

virtual orchestral instruments, and it impacted upon the use of sophisticated homophonic voicings. The complex voicing arrangement was, in my opinion, indispensable for the authenticity of the MIDI orchestration. Moreover, it also affected the harmonic chords and musical scale modulations used.

8.9 Step-by-step manuals

8.9.1 How to set up Cubase on a Mac computer

The first step-by-step user manual explains and illustrates in detail how to set up Cubase⁸⁶, a digital audio workstation (DAW). The DAW software is installed on a Mac computer. The DAW is controlled by an Apple computer keyboard and a computer mouse that includes a left, middle⁸⁷, and right⁸⁸ button. The setup of the DAW includes the following four steps: (1) open Cubase; (2) create an empty project; (3) select a location for the project on the computer; (4) save the project on the computer.

Step 1: Open Cubase.

Click on the Cubase icon in the Dock⁸⁹ at the bottom of the computer screen (Figure 26, number 1). If the Cubase application is not localised in the Dock, go to the application

⁸⁶ Cubase is a digital audio workstation (DAW), developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple's Logic Pro or Avid's Pro Tools. However, Cubase has established itself as one of the leading DAW's for composing film scores, trailer music, or commercial music.

⁸⁷ On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

⁸⁸ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

⁸⁹ The Dock in macOS offers fast accessibility to the Finder (the filesystem), currently running applications, favourite apps, files or folders, and the Trash.

folder of the computer (HD/ User/ [USERNAME]/ Applications), and search for the Cubase application. Cubase starts, and the startup screen of the DAW opens (Figure 26, number 2).



Figure 26. Start Cubase.

Step 2: Create an empty project.

A new internal menu of Cubase called 'steinberg hub' opens (Figure 27). Click on the menu item called 'More', select 'Empty', and then click on the 'Create Empty...' button to activate a new project (Figure 27, numbers 1, 2, and 3).

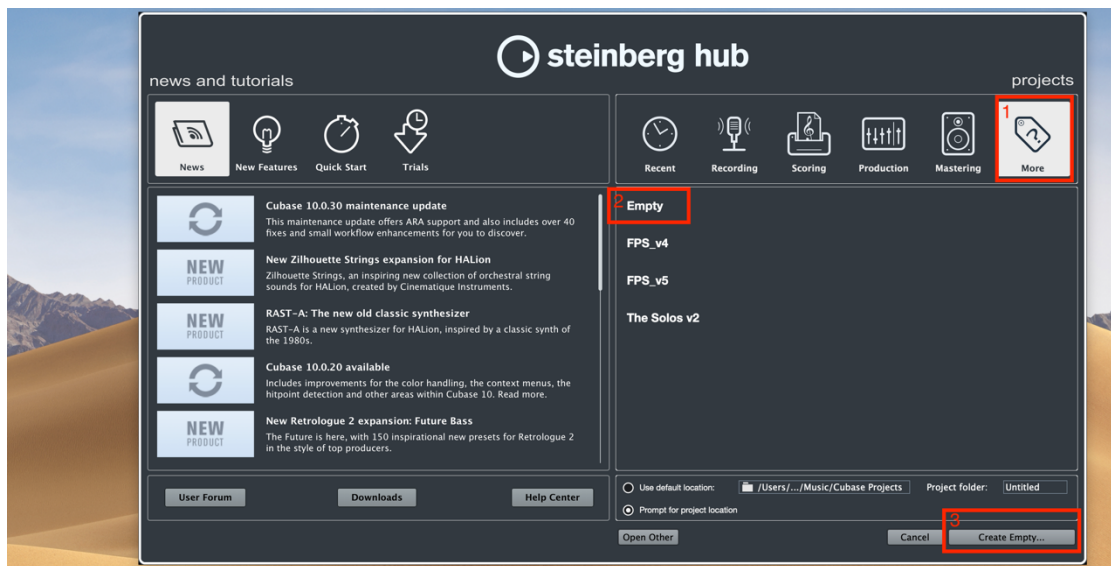


Figure 27. Open the steinberg hub.

Step 3: Select a location for the project on the computer.

Cubase prompts you to select a location on the computer to save audio⁹⁰ and MIDI⁹¹ files that are used in the project (Figure 28). In this example, choose the ‘Desktop’ folder, and subsequently click on the blue highlighted ‘Open’ button (Figure 28, number 1 and 2). A new project is created in the ‘Desktop’ folder.

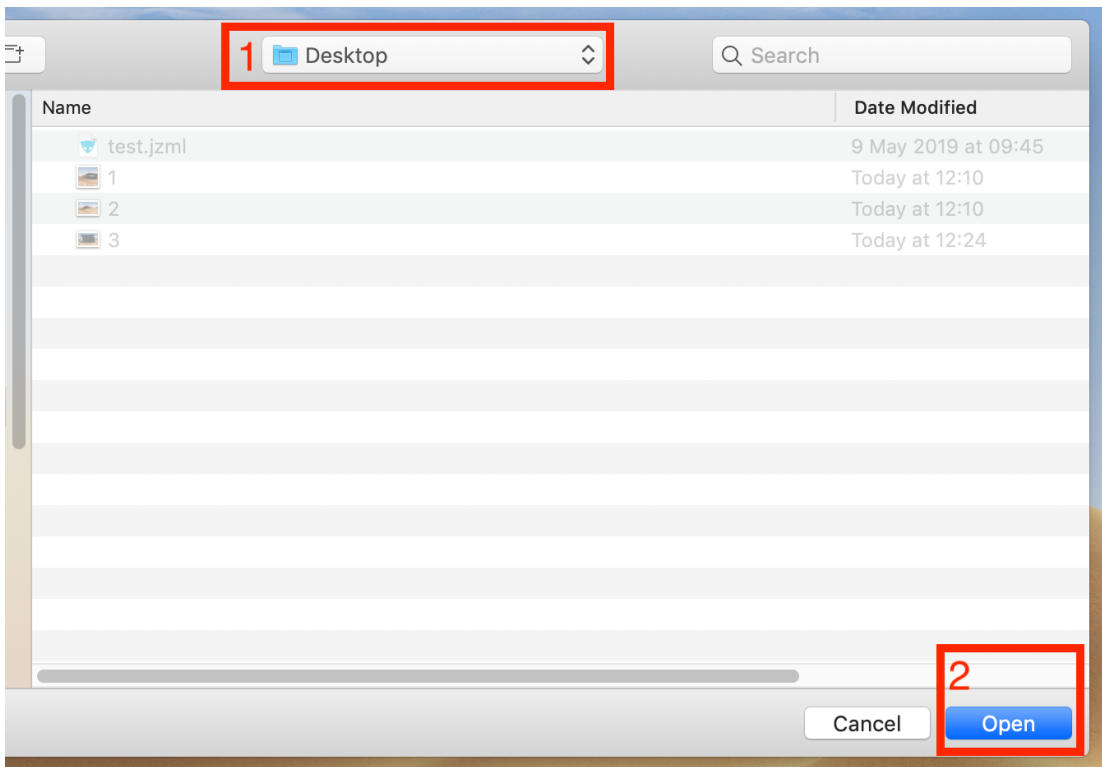


Figure 28. Select 'Desktop' as the location and open a new project.

⁹⁰ An audio file format saves digital audio data on a computer.

⁹¹ The Musical Instrument Digital Interface (MIDI) file format saves recording and playback information of music. However, the MIDI protocol does not contain actual audio data but instructional information. This instructional information includes, e.g., which notes are played, when they are played, and how long or loud each note is played. The information is created on digital synthesisers, MIDI controllers, or within a DAW software environment that supports MIDI.

Step 4: Save the project on the computer.

After Cubase opens, click on 'File' in the upper left corner of the menu bar and select 'Save' to save the project in the 'Desktop' folder (Figure 5). Alternatively, use the shortcut combination 'command and S' (hold down the command key, and type in S).

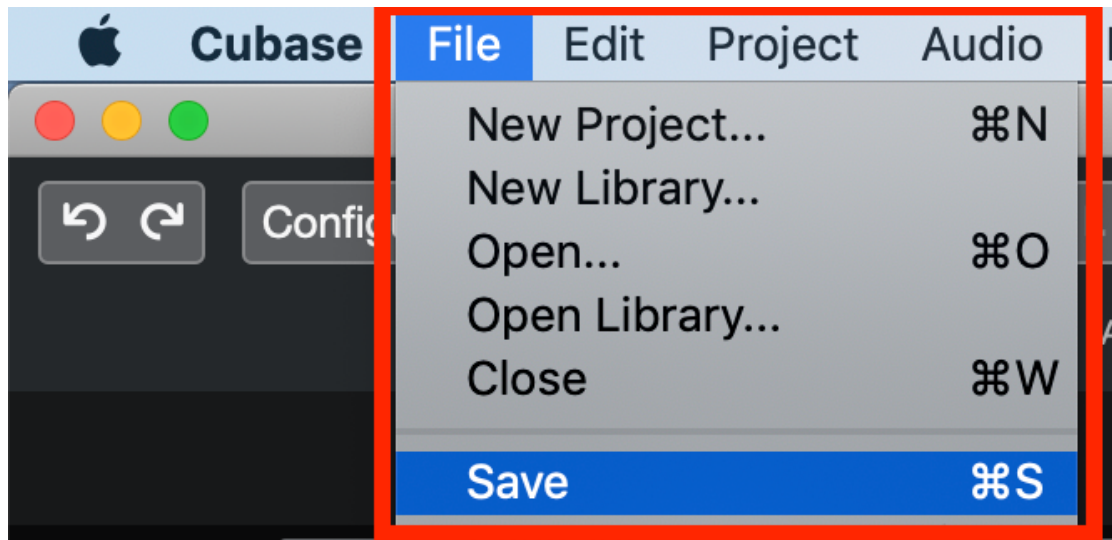


Figure 29. Save a new project.

Cubase will prompt you to give the project a title. In this example, name the project file 'Flexible Phrase System.cpr'⁹² (Figure 30).

⁹² The CPR file format is developed by Steinberg GmbH, and it is used exclusively for Cubase. This file format includes data information such as audio and MIDI tracks, effects, and other DAW settings.

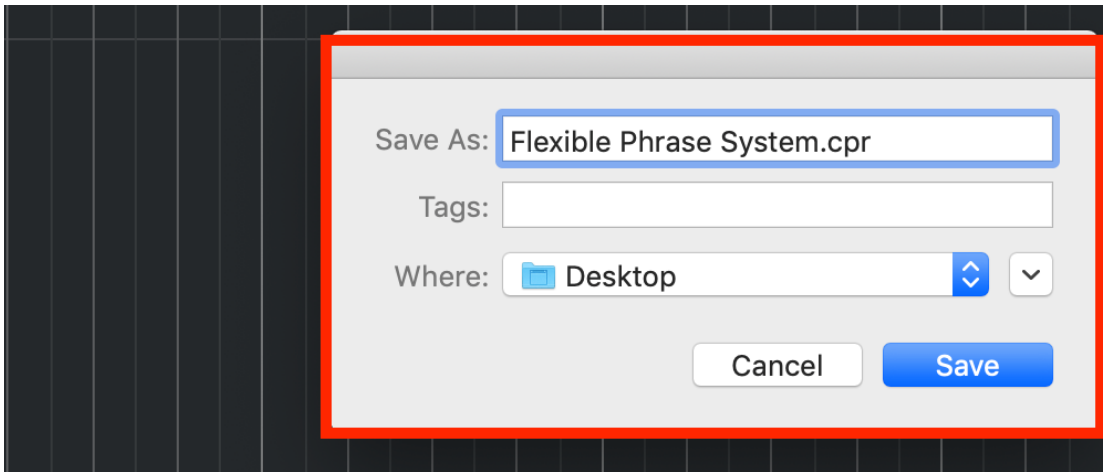


Figure 30. Give the Cubase project a title.

After starting Cubase, creating, saving, and naming the project, the DAW system is now ready for operation.

8.9.2 How to set up the FPS-UIC on an external iPad tablet

This video/manual explains and illustrates in detail how to set up and use the (FPS-UIC)⁹³. The FPS-UIC is opened in the Lemur application⁹⁴, and Lemur is installed on an iPad tablet. The iPad tablet is connected to a Mac computer with a USB cable. The set up of the FPS-UIC includes the following three steps: (1) connect the iPad to the Apple computer via a Lightning to USB cable; (2) open the Lemur application on the tablet; (3) import the FPS-UIC into the Lemur application on the iPad tablet.

Step 1: Connect the iPad to the Apple computer via a Lightning to USB cable.

Plug the USB connector into a free USB port on the Apple computer (Figure 31 number 1), and plug the lightning connector into the lightning port of the iPad (Figure 31 number 2). The iPad tablet is now connected to the computer.

⁹³ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments, and it generates musical chords and musical scales automatically.

⁹⁴ Lemur is a customizable user interface programming environment developed by the company Liine. The application is installed on a tablet and connected via MIDI with both the DAW and virtual instrument.



Figure 31. Connect the USB cable to both the computer and tablet.

Step 2: Open the Lemur application on the tablet.

Click on the Lemur app icon on the tablet display (Figure 32). The default window of the Lemur application opens on the tablet screen.

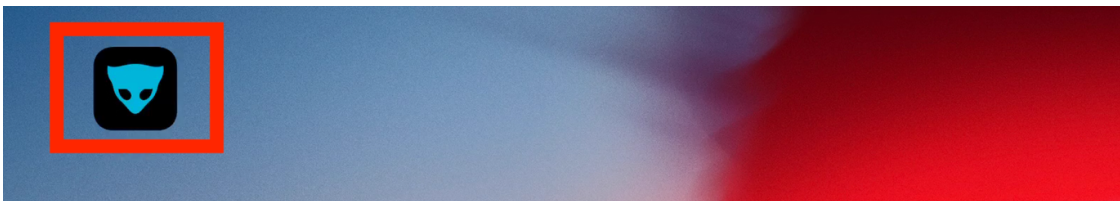


Figure 32. Open the Lemur application on the tablet.

Step 3: Import the FPS-UIC into the Lemur application on the tablet.

Next, click on the ‘gear wheel’ symbol at the top right of the Lemur window (Figure 33, number 1), and select ‘Project’ from the drop-down menu (Figure 33, number 2).

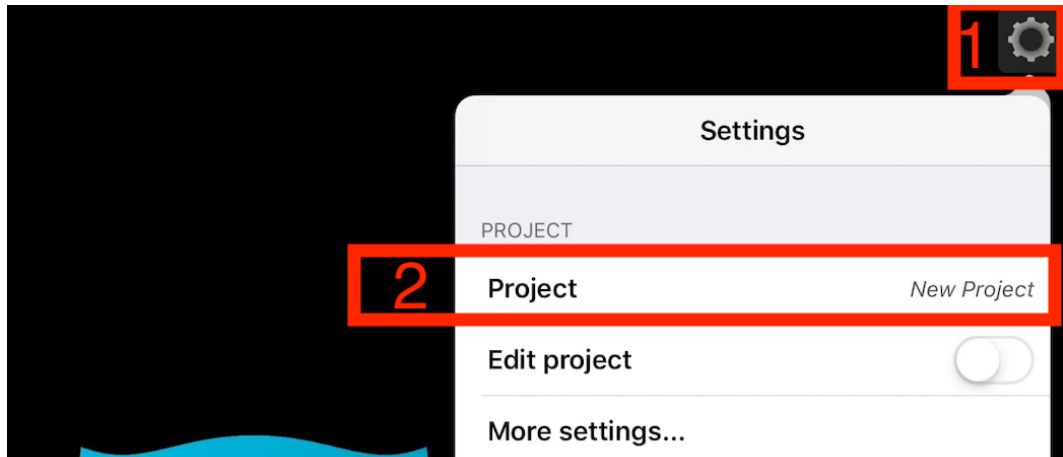


Figure 33. Import the FPS-UIC into the Lemur application on the tablet.

A new internal menu of the Lemur called ‘Project List’ opens. Click on ‘FPS-UIC’ to open the FPS application (Figure 34).

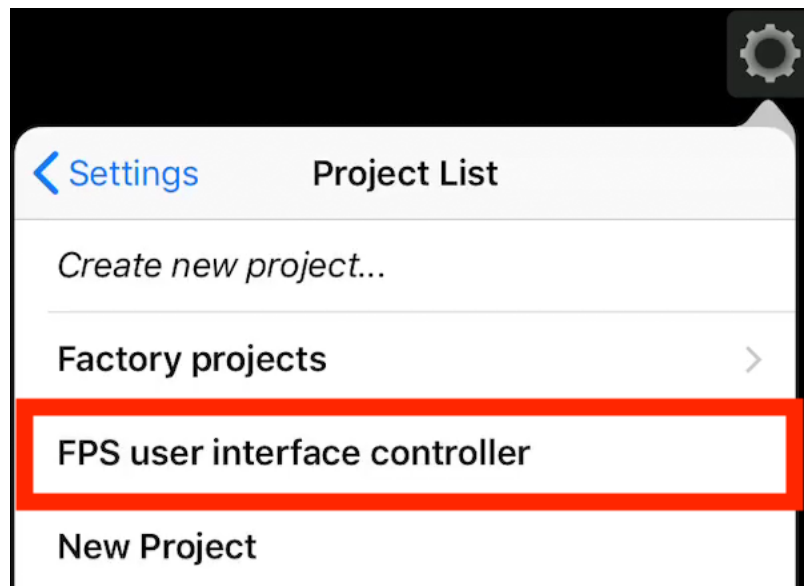


Figure 34. Open the FPS-UIC.

The FPS-UIC is now loaded into the Lemur application (Figure 35).

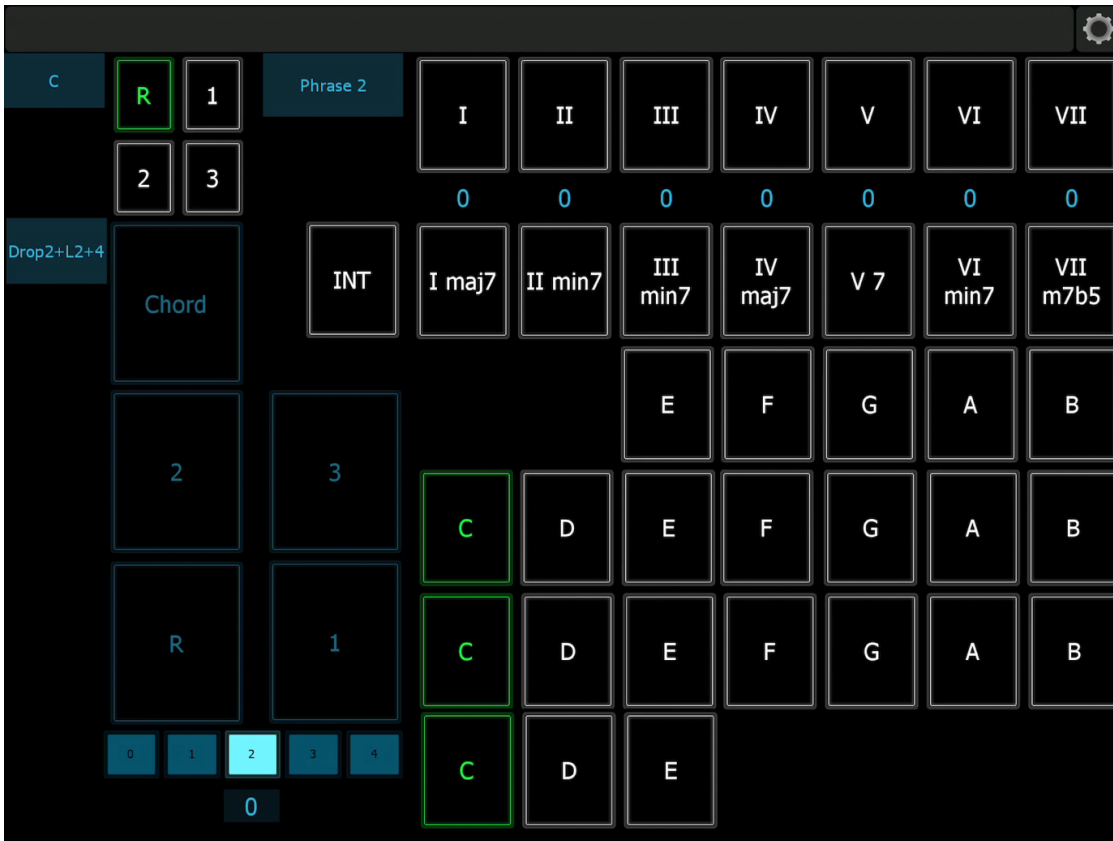


Figure 35. The loaded FPS-UIC.

After connecting the iPad to the Apple computer via a Lightning to USB cable, opening the Lemur application on the tablet, and importing the FPS-UIC into the Lemur application on the iPad tablet, the FPS-UIC is now ready for operation.

8.9.3 How to begin a composition with the aid of the FPS-UIC and the FPS-

VI

This video/manual explains and illustrates in detail how to use the FPS-UIC⁹⁵ and the (FPS-VI)⁹⁶ to start to compose original music. The FPS-VI is opened within Reaktor 6⁹⁷, and Reaktor 6 is used as a VST-plugin⁹⁸ within Cubase⁹⁹, a digital audio workstation (DAW). Optionally, the sound of the FPS-VI can be manipulated by post-

⁹⁵ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments, and it generates musical chords and musical scales automatically.

⁹⁶ The ‘Flexible Phrase System Virtual Instrument’ (FPS-VI) simulates the sound of an electric guitar. The synthesis engine of the electric guitar includes expression-related parameters that are tuned automatically, and it is responsible for the realisation of musical performances of an electric guitarist.

⁹⁷ Reaktor 6, developed by Native Instruments, is a modular audio generation and processing environment with a graphical interface environment that can be run standalone or as an instrument ensemble or effect plugin. The Reaktor 6 environment is available as a VST plug-in and is easy to interface with a DAW.

⁹⁸ Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesiser and audio effects in digital audio workstations (DAWs).

⁹⁹ Cubase is a digital audio workstation (DAW), developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

processing effects. In this case, Guitar Rig 5¹⁰⁰ is used as the post-processing effect. The software applications are installed on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The FPS-VI is controlled by the FPS-UIC¹⁰¹, an Apple computer keyboard, and a computer mouse that includes a left, middle¹⁰², and right¹⁰³ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac computer), and the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet) to continue to this step-by-step tutorial.

Starting a composition with the aid of the FPS-UIC and FPS-VI requires 20 steps: (1) add instrument track; (2) select Reaktor 6 as the virtual instrument; (3) import the FPS-VI ensemble file in Reaktor 6; (4) change the name of the instrument; (5)

¹⁰⁰ The Guitar rig 5 plug-in from Native Instrument is an amplifier simulator that emulates the colouration added to a signal by an amplifier, particularly an instrument amplifier such as those used with guitars and bass guitars. Amplifier simulators typically emulate the effects of an amp’s preamp section, power amp section, any effects (such as built-in spring reverb), and the connected speaker and its enclosure or cabinet.

¹⁰¹ Optionally, the FPS-VI can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that sends MIDI data to the DAW or other musical devices. In this example, a conventional MIDI keyboard is not used.

¹⁰² On a Mac computer, a trackpad or Apple mouse does not include a standardised middle button configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹⁰³ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

program¹⁰⁴ a rhythmic pattern of ‘Phrase 1’; (6) select articulations of ‘Phrase 1’; (7) change the overall length of ‘Phrase 1’; (8) save the programmed phrase as ‘Phrase 1’ preset; (9) program a rhythmic pattern of ‘Phrase 2’; (10) select articulations of ‘Phrase 2’; (11) save programmed phrase as ‘Phrase 2’ preset; (12) select ‘Phase 1’, and enable recording; (13) select a major scale on the FPS-UIC; (14) play notes on the FPS-UIC; (15) select ‘Phrase 2’ on the FPS-UIC; (16) enable recording of ‘Phrase 2’; (17) play notes on the FPS-UIC; (18) connect ‘Phrase 1’ and ‘Phrase 2’; (19) (optionally) add post-processing effects; (20) playback ‘Phrase 1’ and ‘Phrase 2’.

Follow these 20 steps to start an original composition with the aid of the FPS-UIC and FPS-VI.

Step 1: Add instrument track.

Move the mouse to the left, to the instrument window in Cubase, right-click, and select ‘Add Instrument Track’ from the pop-up menu (Figure 36).

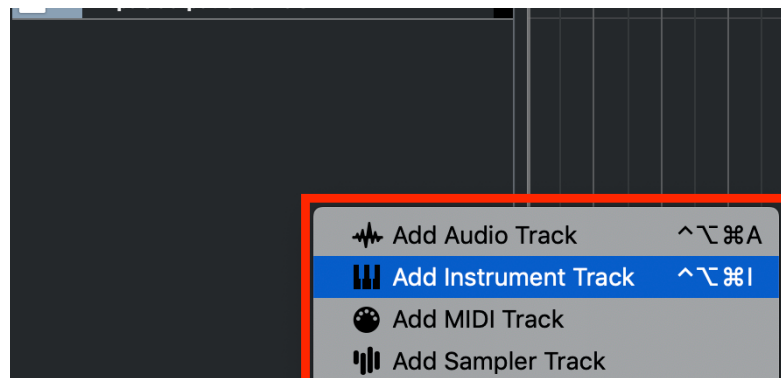


Figure 36. Add Instrument Track.

¹⁰⁴ In the context of this section, programming includes drawing different rhythmical length values and different articulations with the help of a computer mouse and computer keyboard. It does not include the process of writing or coding computer programs and applications.

Step 2: Select Reaktor 6 as the virtual instrument.

A new screen opens. Click on the ‘Instrument’ menu in the centre of the screen, select ‘Reaktor 6’, keep the audio output on ‘Stereo’¹⁰⁵, and press the ‘Add Track’ button to integrate Reaktor 6 into the DAW (Figure 37, numbers 1 and 2).



Figure 37. Add Reaktor 6 as virtual instrument track.

Step 3: Import the FPS-VI ensemble file in Reaktor 6.

After the Reaktor 6 engine opens, go to ‘Files’, ‘Documents’, and select the virtual instrument ensemble called ‘FPS_v4.ens’¹⁰⁶ within the internal Reaktor menu (Figure

¹⁰⁵ The FPS-VI is designed for Stereo Out projects. The software is not designed for surround or binaural music projects.

¹⁰⁶ The Reaktor 6 ensemble file is a software format in which virtual instruments are designed with the help of sound generators and effects.

38, numbers 1, 2, and 3). If the ensemble file is not localised in the document folder, click on the magnifier symbol in the upper-left corner to activate the search function of Reaktor 6, and type 'FPS_v4.ens'.

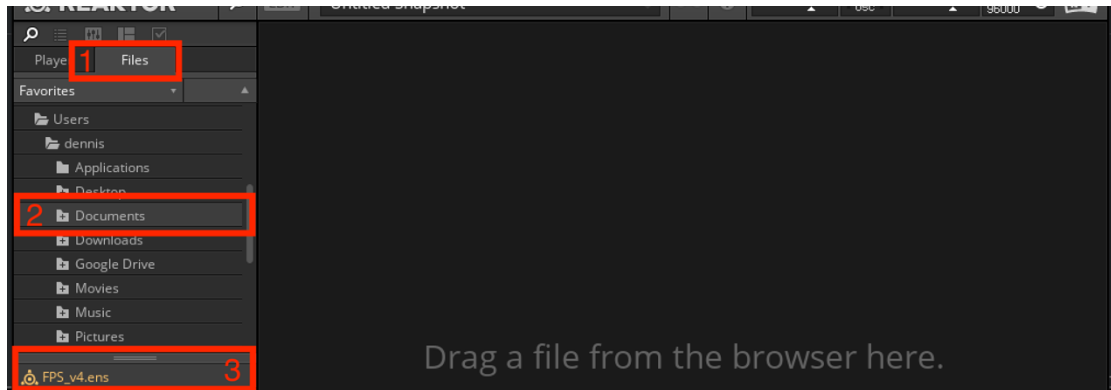


Figure 38. Select the FPS-VI ensemble file.

The FPS-VI graphical user interface (GUI)¹⁰⁷ opens, and Cubase adds the FPS-VI to the Project (Figure 39).

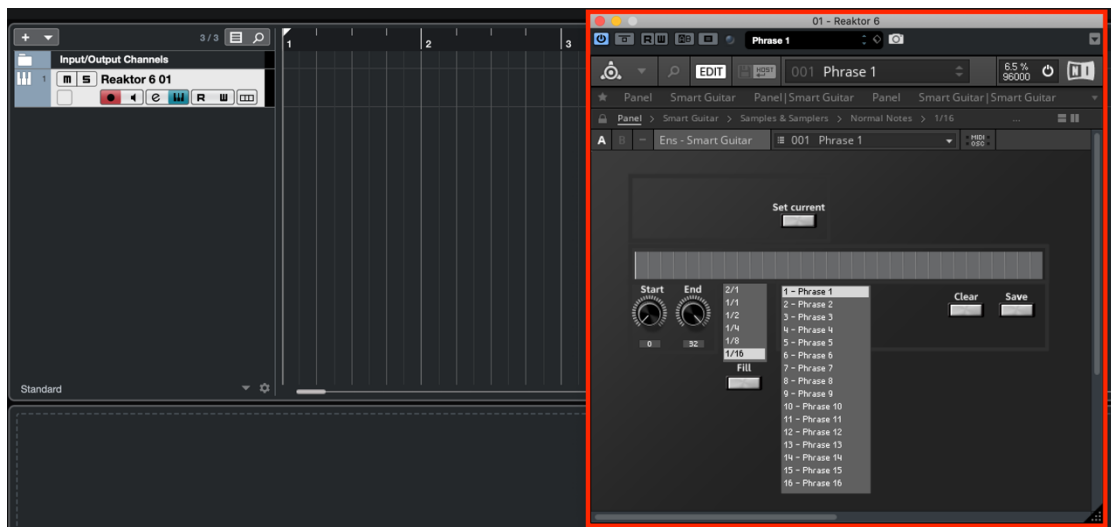


Figure 39. The FPS-VI is loaded into Cubase.

¹⁰⁷ The graphical user interface (GUI) allows music composers to interact with the FPS-VI through graphical symbols and visual indicators. The GUI is controlled with the help of a computer mouse and computer keyboard.

Step 4: Change the name of the instrument.

To identify the instrument track, double click on the track name 'Reaktor 6 01', and change the name. In this example, type 'Flexible Phrase System' (Figure 40).



Figure 40. Change the name of the instrument track.

Step 5: Program the rhythmic pattern of 'Phrase 1'.

To program a rhythmic pattern, use the sequencer window located in the centre of the FPS-VI. The sequencer window represents the length of one phrase and is divided into 32 segments. The segmentation functions as a rhythmical grid. Each segment represents a $\frac{1}{16}$ th of a note. The total length of one phrase is two measures (32 times a $\frac{1}{16}$ th of a note). Point the mouse on a thin vertical stroke and double-click. A small grey rectangle appears. The rectangle symbolises a rhythmic value of a $\frac{1}{16}$ th of a note (Figure 41).

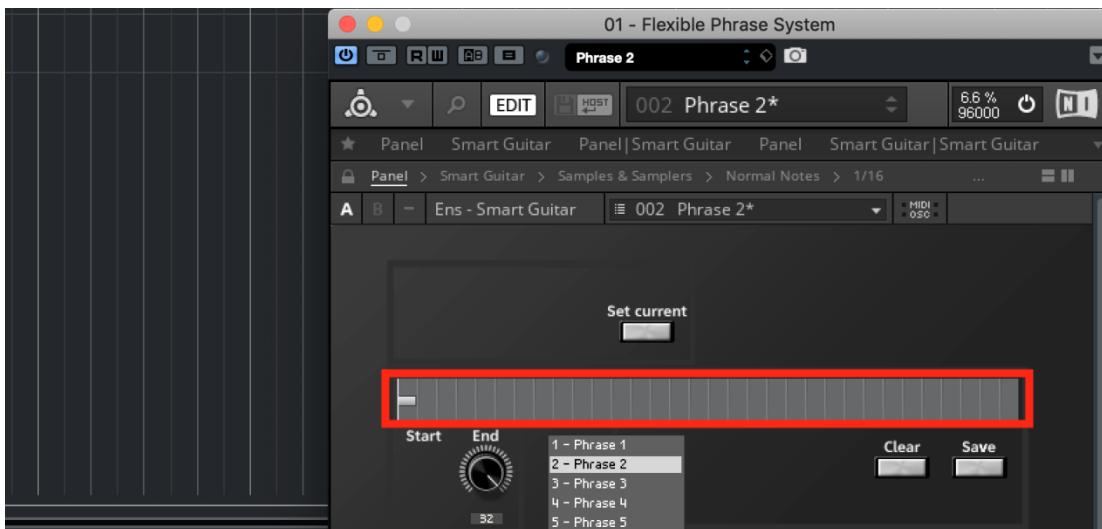


Figure 41. Create a rhythmic note within the sequencer window.

Adjust the length of a note by selecting the note (clicking), holding the right mouse button and moving the mouse to the right. Then, release the right mouse button to determine the length of the note (Figure 42).

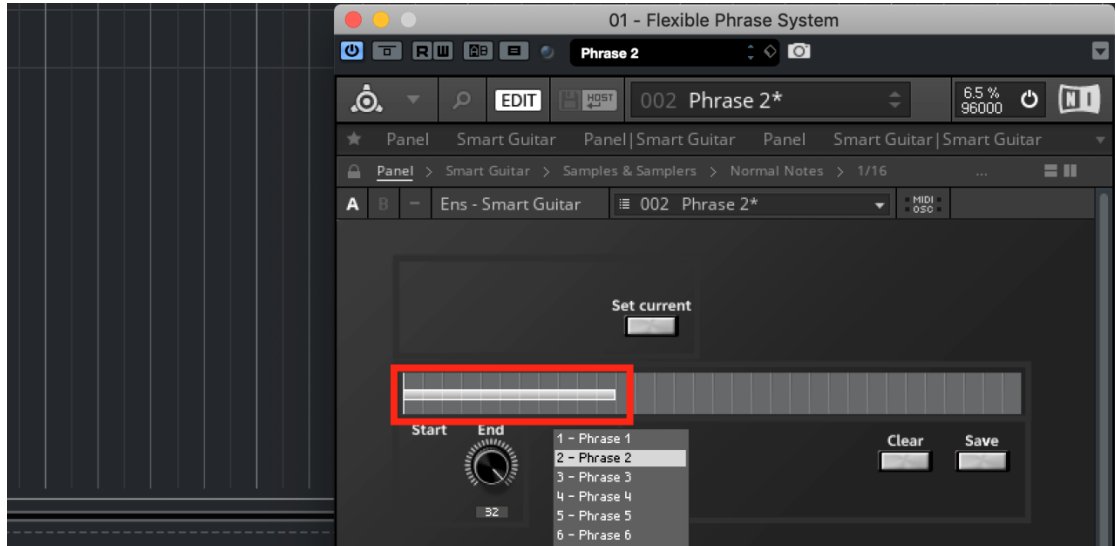


Figure 42. Adjust the length of a note within the sequencer window.

Shorten the length of a note by selecting a note (one click), holding the right mouse button and move the mouse to the left. Again, release the right mouse button to determine the length of the selected note. In this example, click 16 times the 8th note into the sequencer window. The result is a phrase, including a rhythmic value pattern of 16 times the 8th note (Figure 43).

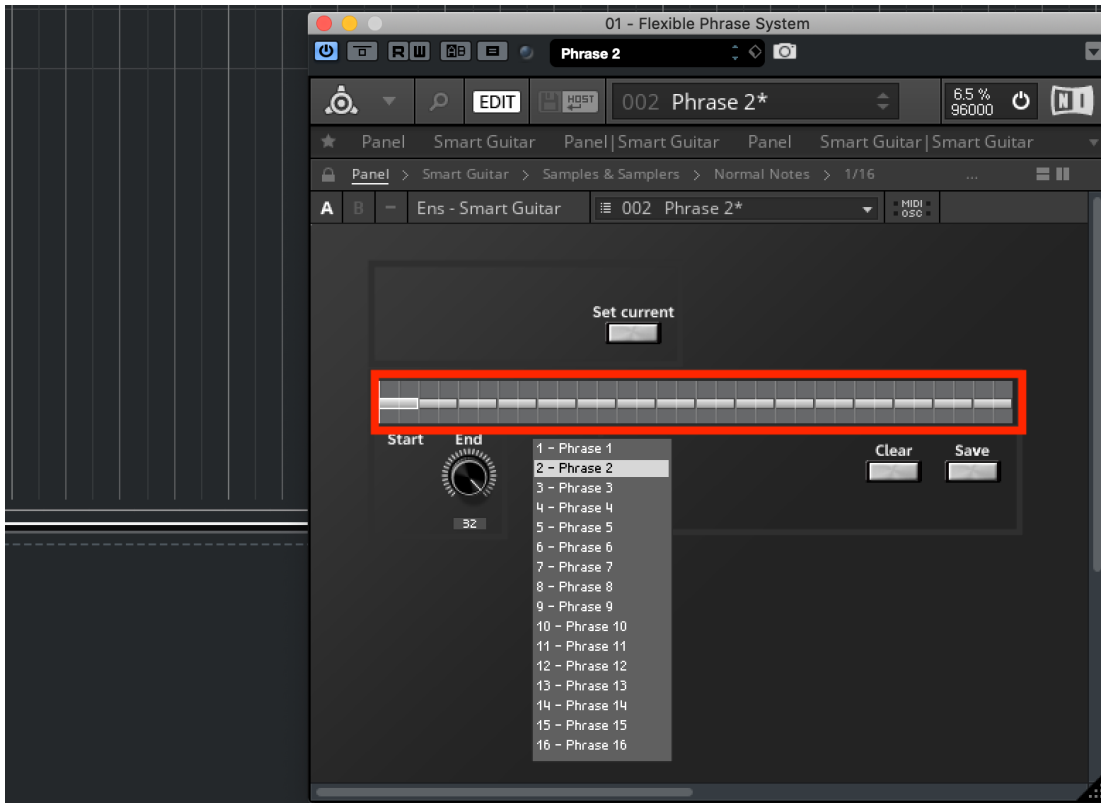


Figure 43. Create a rhythmic phrase including 16 times an 8th note.

Step 6: Select the articulations of 'Phrase 1'.

To change the articulation, move the mouse to the grey rectangle, and select a note. Click on the middle-mouse button to walk through a menu that includes different musical articulations. Every click on the middle-mouse button changes the colour of the selected note, and these colours symbolise different articulations: the first colour, grey, symbolises a plucked played note. It is the default selected articulation. The second colour, green, symbolises a palm mute, the third colour, blue, symbolises a trill articulation, and the fourth colour, red, symbolises a pitch slide. Click one time on the middle-mouse button to select a palm mute (green), two times to select a trill (blue) articulation, and three times to select a pitch slide (red). The colour changes back to grey (normal note articulation) by clicking four times on the middle mouse button. In this example (Figure 44, number 1), move the mouse to the sequencer window, and

select from left to right the following notes, including the following articulations (Table 13):

Position	Articulation (colour)
1, 2, 4, 5, 8, 12, 15, and 16	Plucked played note (Grey)
6, 7, 9, 10, 13, and 14	Palm mute (Green)
3 and 11	Trill (Blue)

Table 13. Selected articulations of 'Phrase 1'.

Step 7: Change the overall length of 'Phrase 1'.

Move the mouse to the rotatory knob (Figure 44, number 2), and turn the knob to the left (decreasing the length), or to the right (increasing the length). The maximum length of a phrase is 32 steps (2 measures), and the minimum length of one phrase is 1 step ($\frac{1}{16}$ th of a note). The number below the rotatory knob corresponds to the end position of the grid in the sequencer window. In this example, set the length of the phrase to 32 steps. Therefore, the phrase starts at the first grid position and ends at the last grid position.



Figure 44. Program different articulations to change the overall length.

Step 8: Save the programmed phrase as 'Phrase 1' preset.

Move the mouse to the 'Save' button on the right, and click on the grey button (Figure 44, number 3). In this example, the phrase is saved as 'Phrase 1', highlighted with a white background in the preset window (Figure 44, number 4). The preset window

ensures that saved phrases can be recalled at any time and that the information of programmed phrases is not lost. In total, program and save up to 128 unique phrases. Switch between presets by moving the mouse to the preset window, and subsequently clicking on different pre-designed phrases. Clear a selected phrase by clicking with the mouse on the 'Clear' button.

Step 9: Program the rhythmic pattern of 'Phrase 2'.

Click in the preset window on 'Phrase 2' (Figure 45, number 1). Move the mouse to the 'Clear' button, and click on the 'Clear' button to create an empty phrase (Figure 45, number 2). Next, move the mouse to the sequencer windows of the FPS-VI. Click 32 times the 16th note into the sequencer window. The result is a phrase including a rhythmic value pattern of 32 times the 16th note (Figure 45, number 3).

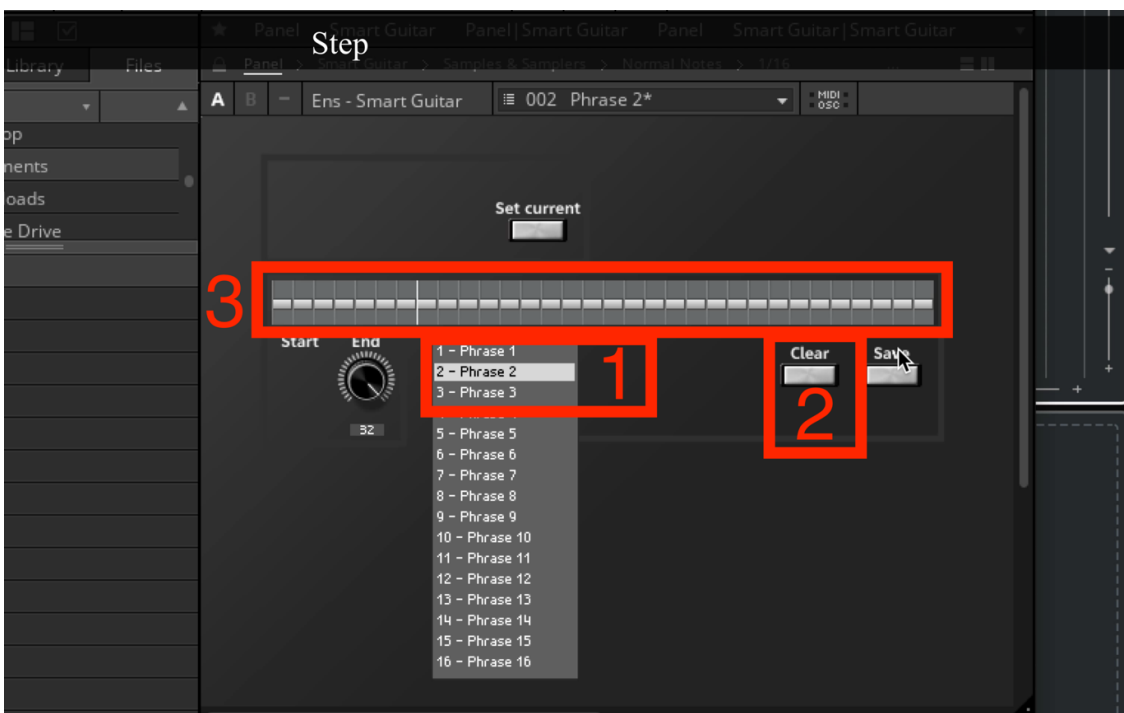


Figure 45. Program the rhythmic pattern of 'Phrase 2'.

Step 10: Select the articulations of ‘Phrase 2’.

Next, move the mouse to the sequencer window, and select from left to right the following notes, including the following articulations:

Positions	Articulation (colour)
1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, and 32	Plucked played note (Grey)
2, 3, 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23, 24, 26, 27, 29, and 30	Palm mute (Green)

Table 14. Selected articulations of ‘Phrase 2’.

Keep the length of the phrase (32 steps) unchanged (Figure 46, number 2). This particular pattern is just an example of a possible rhythmical phrase. The user is free to design another phrase that includes different rhythmic values and articulations.

Step 11: Save the programmed phrase as ‘Phrase 2’ preset.

Move the mouse to the ‘Save’ button on the right, and click on the grey button (Figure 46, number 3). The phrase is saved as ‘Phrase 2’, highlighted with a white background in the preset window (Figure 44, number 4).

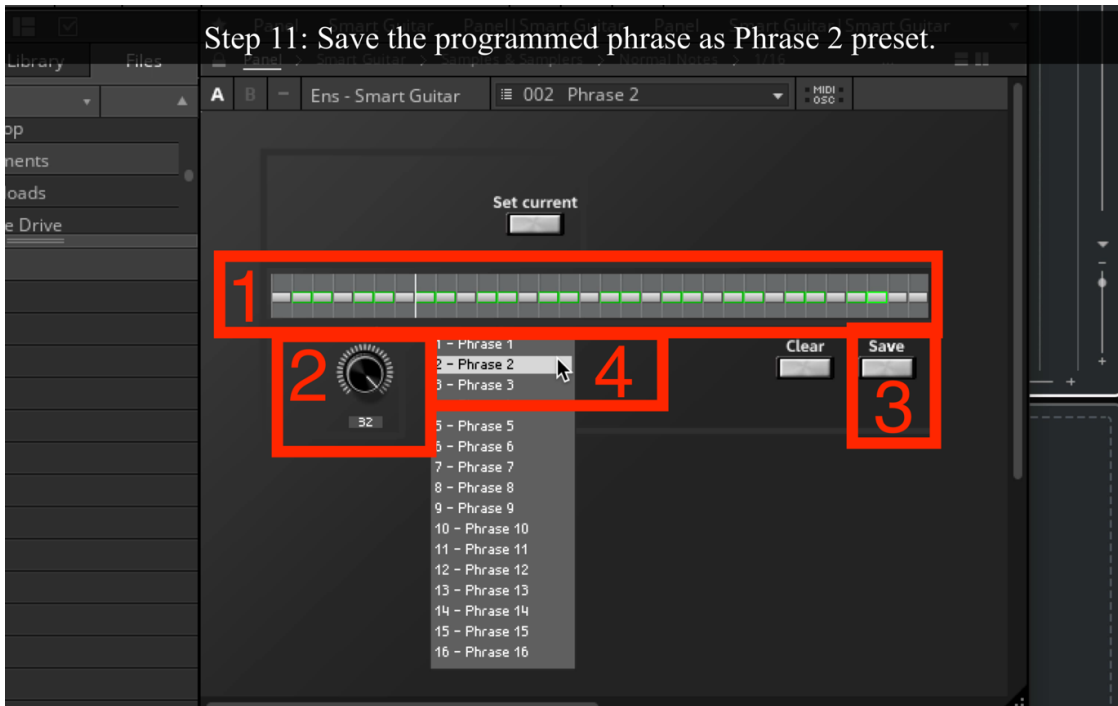


Figure 46. Program different articulations and save as 'Phrase 2'.

Step 12: Select 'Phrase 1' and enable recording.

Click on 'Phrase 1' in the preset window. Next, move the mouse to the centre bottom of the DAW window, and click on the record button. The record button turns red, and the play button to the left turns green (Figure 47, number 1). The instrument track of the FPS-VI also turns red, and the cursor¹⁰⁸ of the DAW starts to move to the right (Figure 47, number 2). The sound of a metronome click starts automatically.

¹⁰⁸ The cursor appears as a vertical line across the stave and it indicates the playback and record position of the DAW.

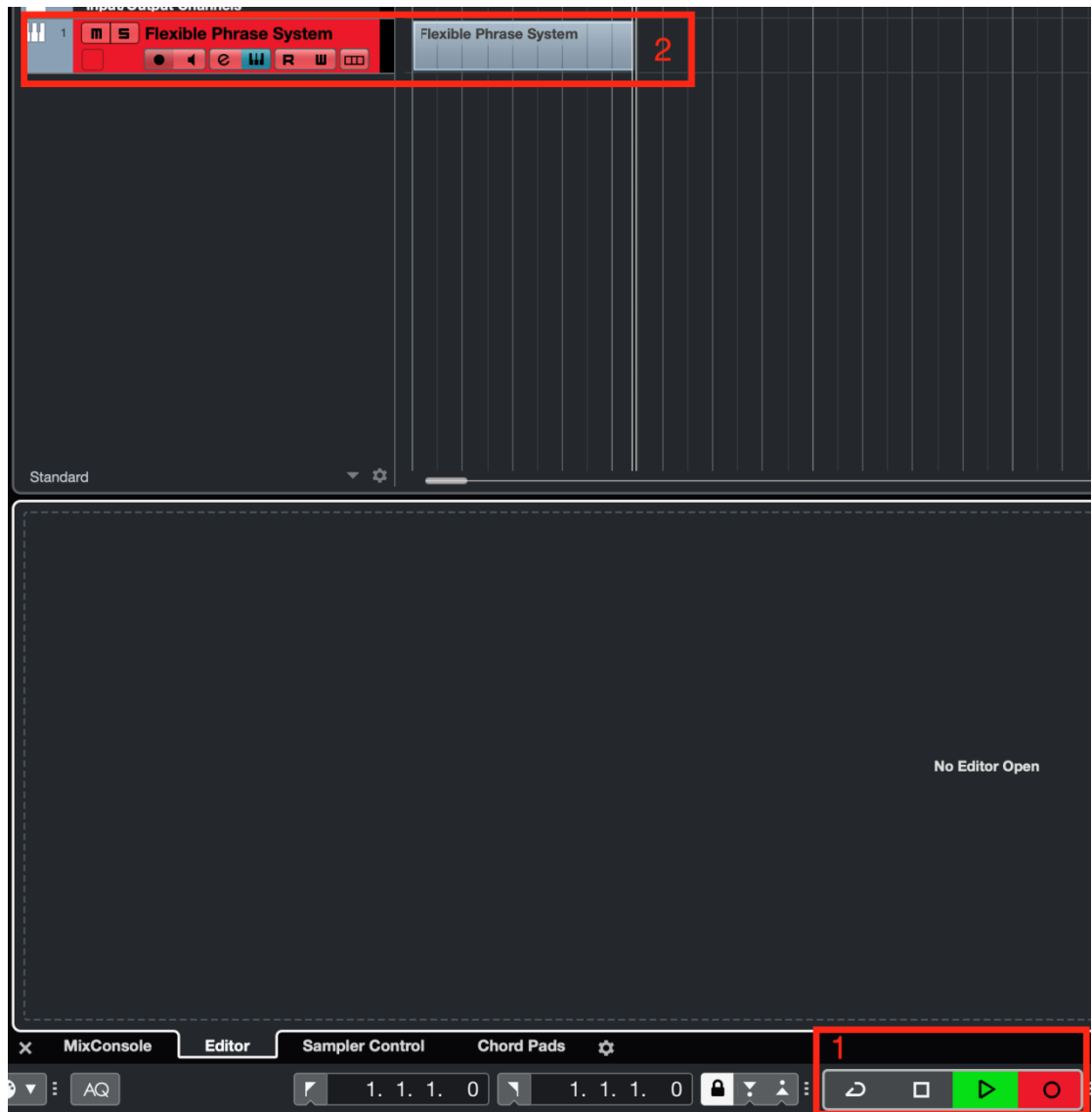


Figure 47. Enable the recording function in Cubase.

Step 13: Select a major scale on the FPS-UIC.

Go to the iPad tablet and click in the top left corner of the FPS-UIC on the scale menu. A drop-down window, including all musical major scale options, appears (Figure 48, number 1). In this example, select the C major scale.

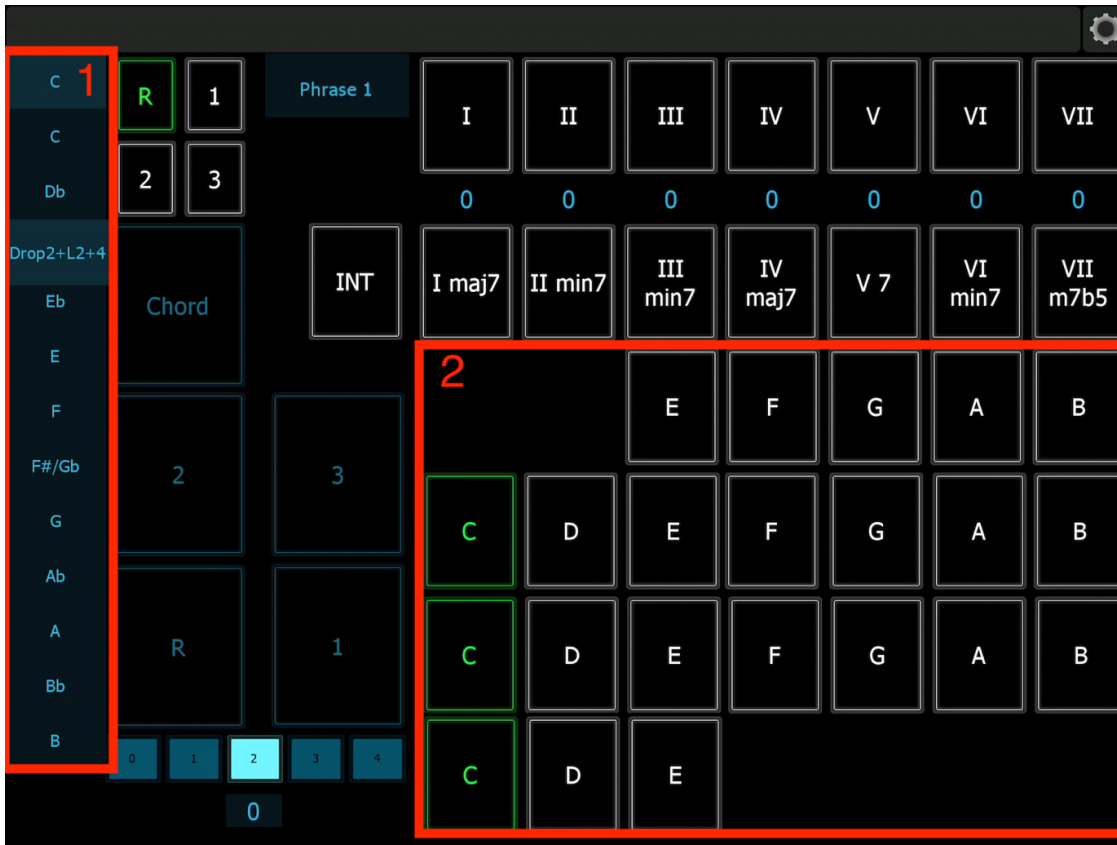


Figure 48. Select a musical scale.

Step 14: Play notes on the FPS-UIC.

In the FPS-UIC, click on different alphabetic characters (Figure 48, number 2). The letters symbolise notes within the C major scale. The playing the notes triggers the selected phrase within the FPS-VI (Figure 49, number 1), and the pitch of the played notes are repeated within the structure of the designed rhythmic pattern. After eight measures, click on the ‘Stop’ button to the left of the ‘Play’ button to stop the recording (Figure 49, number 2). The musical phrase is now recorded within the MIDI

environment of Cubase (Figure 49, number 3). If required, adjust or remove MIDI notes within the MIDI editor¹⁰⁹ window of Cubase (Figure 49, number 4).

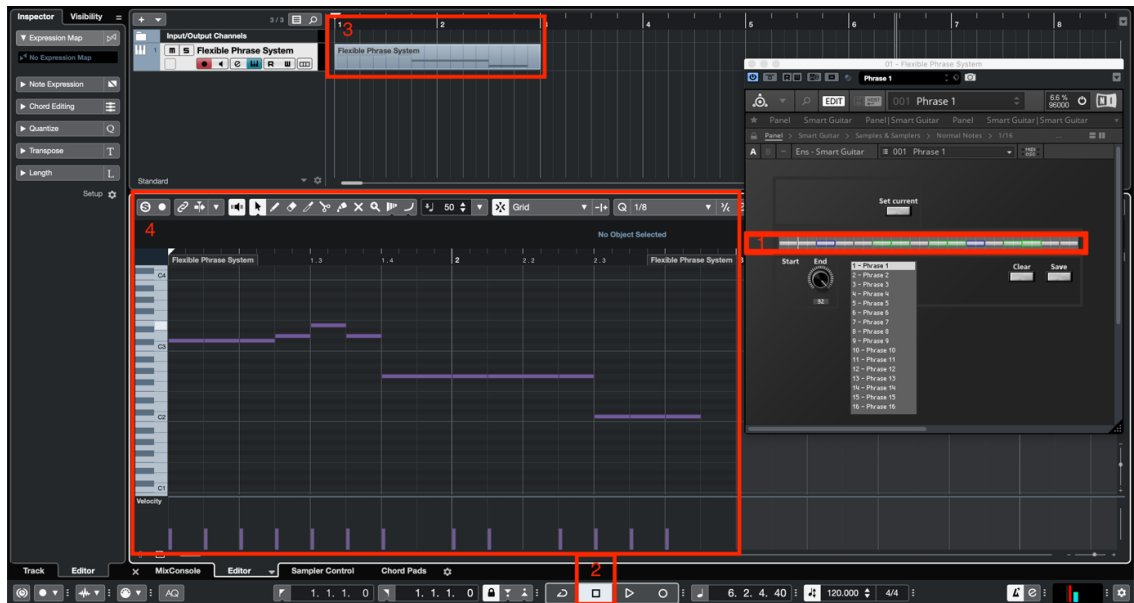


Figure 49. Activate the FPS-VI synthesis engine, and record 'Phrase 1' within Cubase.

Step 15: Select 'Phrase 2' on the FPS-UIC.

After recording 'Phrase 1', switch to preset 'Phrase 2'. Go to the FPS-UIC, and click on the drop-down menu in the upper part of the FPS-UIC (Figure 50). In this example, select 'Phrase 2'. Close the drop-down window by clicking on one of the phrases. The phrase preset switches from 'Phrase 1' to 'Phrase 2' in the FPS-VI.

¹⁰⁹ The MIDI editor window of Cubase allows you to perform MIDI editing, and it provides an overview of the context of the recorded track.

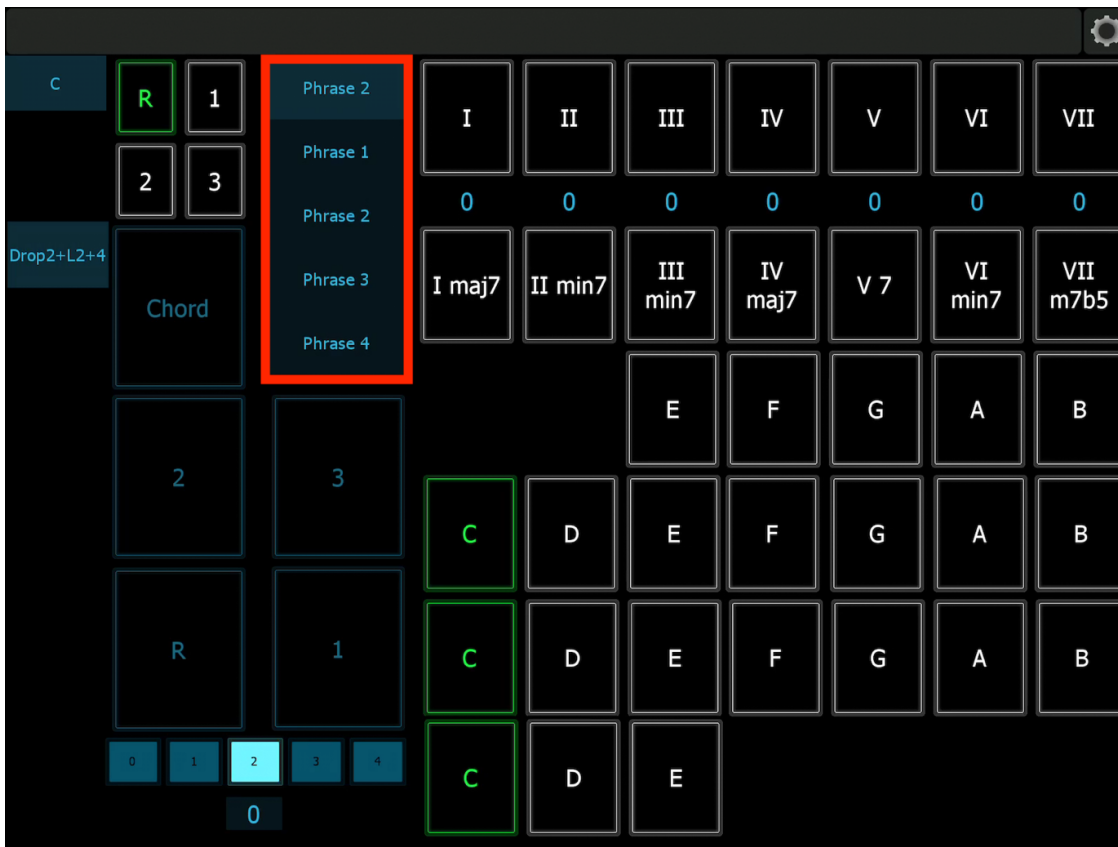


Figure 50. Switch to preset 'Phrase 2'.

Step 16: Enable recording of 'Phrase 2'.

Position the cursor of the DAW at the end of the previous recording. In this example, click in the sequencer window of the DAW on measure 19 (Figure 51).

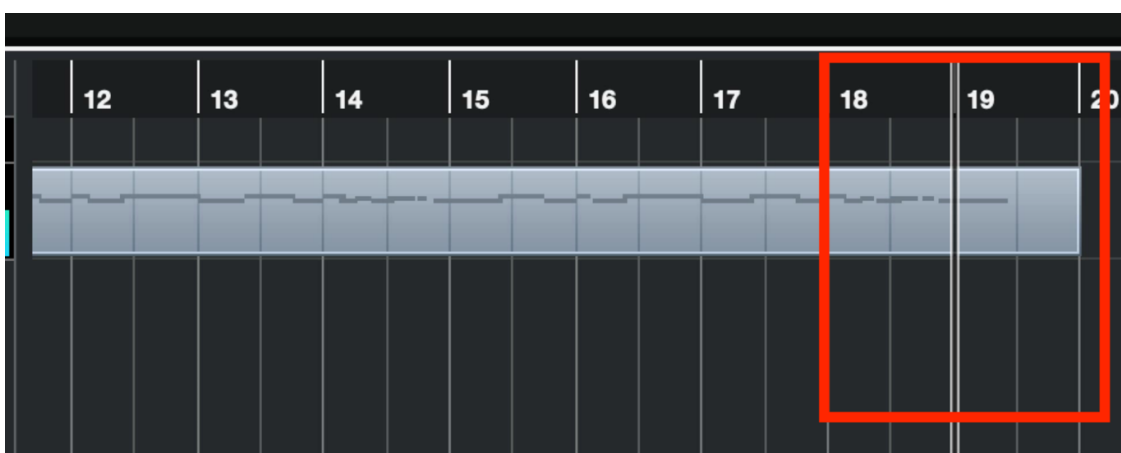


Figure 51. Position the cursor on measure 19.

Again, move the mouse to the centre bottom of the DAW window, and click on the record button. The record button turns red, and the play button to the left turns green. ‘Phrase 2’ is now looping continuously.

Step 17: Play notes on the FPS-UIC.

Again, change the pitch values of the programmed ‘Phrase 2’ ‘on the fly’ with the aid of the FPS-UIC (Figure 52, number 1). After eight measures, click on the ‘Stop’ button. The second musical phrase is now recorded within the MIDI environment of Cubase (Figure 52, number 2). This step is comparable to Step 14.

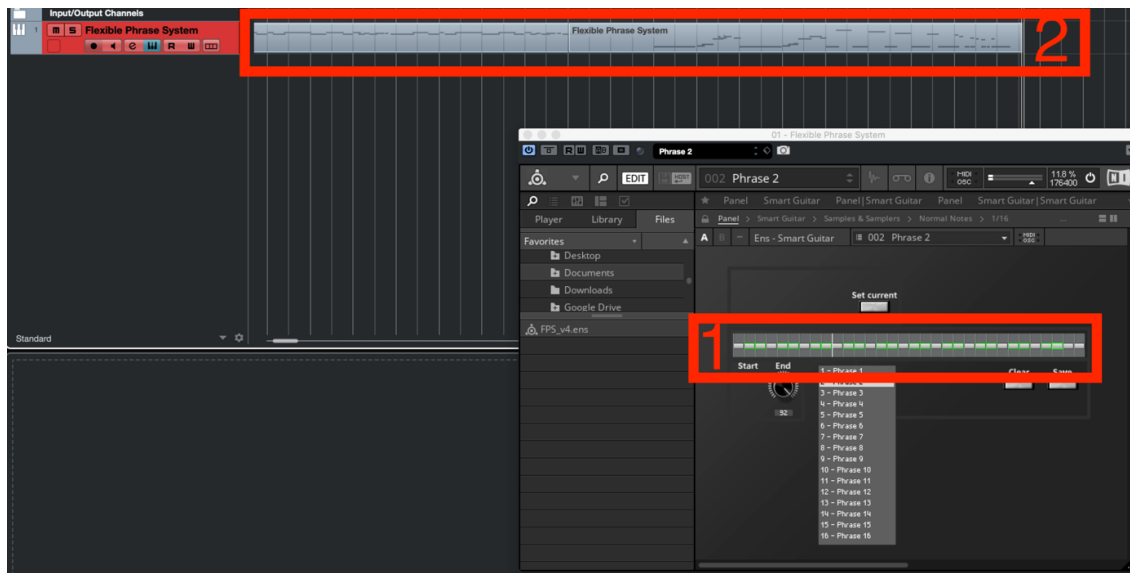


Figure 52. Activate the FPS-VI synthesis engine, and record ‘Phrase 2’ within Cubase.

Step 18: Connect ‘Phrase 1’ and ‘Phrase 2’.

The recording of the first recorded ‘Phrase 1’ stops at measure 19, and the second recorded ‘Phrase 2’ starts at measure 21. In this step, move the second recorded phrase from measure 21 to measure 19. To do so, click on the recorded MIDI event of ‘Phrase 2’, hold the left mouse button, and shorten the MIDI event from measure 21 to measure 19 (Figure 53).

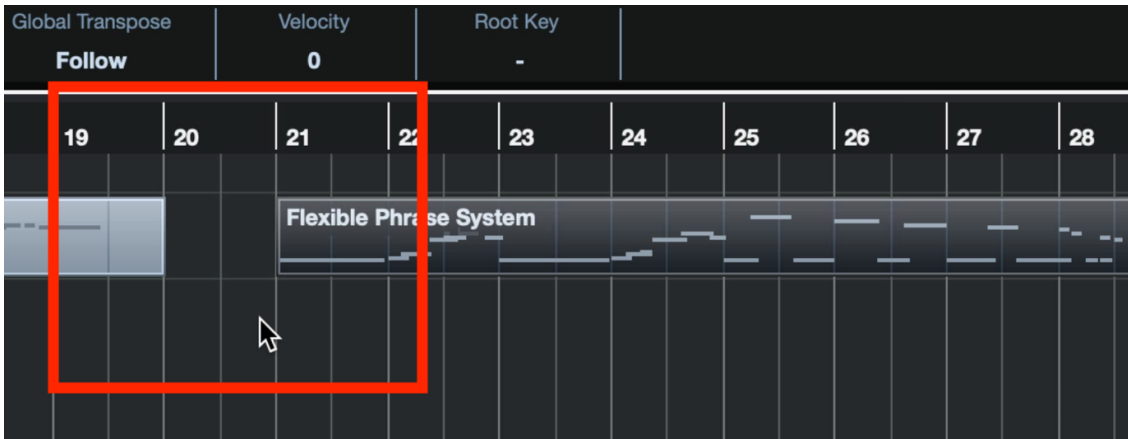


Figure 53. Shorten MIDI event of 'Phrase 2'.

Next, shorten the MIDI event of 'Phrase 1' from measure 20 to measure 19 (Figure 54). Click on the MIDI event, hold the left mouse button, and move the mouse to the left.



Figure 54. Shorten MIDI event of 'Phrase 1'.

Click on the MIDI event of 'Phrase 2', hold the left mouse button and move the start position of the MIDI event from measure 21 to measure 19 (Figure 55). Now, the second 'Phrase 2' occurs directly after the first 'Phrase 1'. 'Phrase 1' ends on measure 19, while 'Phrase 2' starts on measure 19.

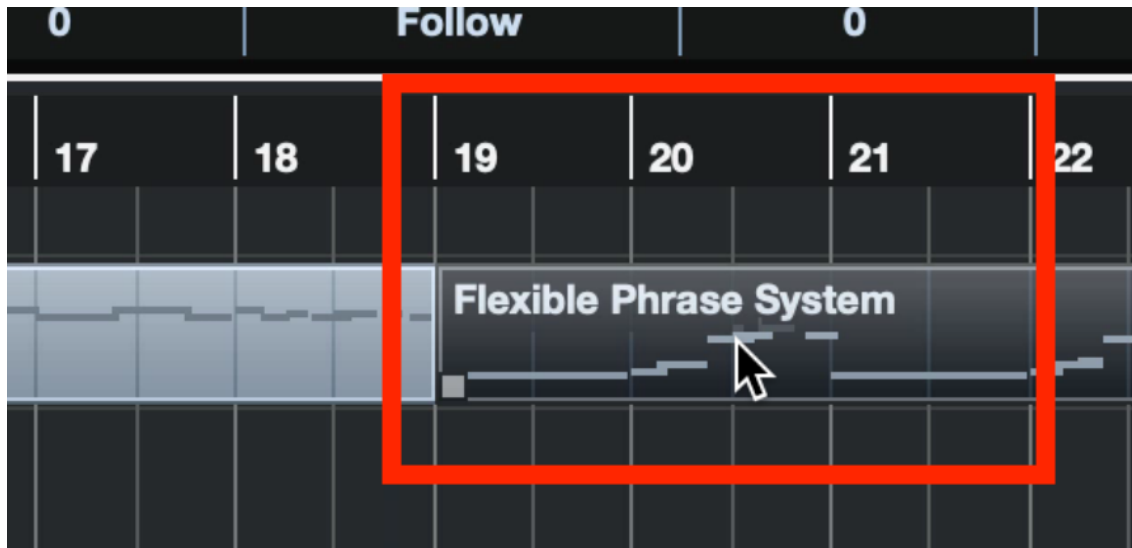


Figure 55. Move MIDI event of 'Phrase 2' to measure 19.

Step 19: (Optionally) add post-processing effects.

It is also possible to add post-processing effects to the sound of the FPS-VI. In this case, use the digital signal audio processing plugin 'Guitar Rig 5'. This effect plugin works well together with the FPS-VI because it is an amplifier simulation specifically designed for digital input of recorded electric guitars. Move the mouse to the left side of the screen. Open 'Audio inserts' in the inspector zone of Cubase, and choose 'Guitar rig 5' as an effect (Figure 56, number 1). Scroll between different presets and choose a preset that fits the composition best. In this example, choose, 'RTM2 Power' as electric guitar effect (Figure 56, number 2).



Figure 56. Add post-processing effect.

Step 20: Playback ‘Phrase 1’ and ‘Phrase 2’.

In the DAW, click on the playback button, and listen to both composed musical phrases.

Continue this composition, and repeat Step 5 till Step 20 in the same or a different order. Create more phrases within the FPS-VI and play different musical notes on the FPS-UIC. Additionally, add different instruments such as drums, keyboards, or string instruments to the composition.

8.9.4 How to begin a composition with the aid of the FPS-UIC and virtual strings.

This video/manual explains and illustrates in detail how to use the FPS-UIC¹¹⁰ and a 54-piece virtual string ensemble¹¹¹ to compose original homophonic¹¹² music. The virtual string ensemble is opened within Kontakt 5¹¹³, which is used as a VST-plugin¹¹⁴ within Cubase¹¹⁵, a digital audio workstation (DAW). The software applications are installed

¹¹⁰ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments, and it generates musical chords and musical scales automatically.

¹¹¹ The virtual string ensemble simulates the sound of a string ensemble. The virtual string ensemble, developed by Native instruments, simulates the following instrument sections: 1st Violin I (8 players), 1st Violin II (8 players) 2nd Violin I (7 players), 2nd Violin II (7 players), Viola I (6 players), Viola II (6 players), Cellos I (5 players), Cellos II (5 players), and Double Bass (4 players).

¹¹² In Homophonic music, multiple parts are moving in the same rhythm, but they are playing different pitch values.

¹¹³ Kontakt 5, developed by Native Instruments, is a sampling synthesis environment with a graphical interface environment that can be run alone or as an instrument ensemble. The Kontakt 5 environment is available as a VST plug-in and is easy to interface with a DAW.

¹¹⁴ Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesiser and audio effects in digital audio workstations (DAWs).

¹¹⁵ Cubase is a digital audio workstation (DAW), developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The virtual string ensemble is controlled by the FPS-UIC¹¹⁶, an Apple computer keyboard, and a computer mouse that includes a left, middle¹¹⁷, and right¹¹⁸ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac computer), and the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet) to continue to this step-by-step tutorial.

Composing music with the aid of the FPS-UIC and virtual string ensembles requires 13 steps: (1) add instrument track; (2) count nine times a Kontakt 5 instrument track; (3) import different string instruments; (4) change the names of the instrument tracks, (5) route instrument tracks to the FPS-UIC; (6) group instrument tracks to one string ensemble; (7) select a musical scale; (8) select a voicing; (9) select the inversion; (10) select a musical chord; (11) enable recording; (12) play a chord progression; (13) (Optional) Change dynamic progression.

¹¹⁶ Optionally, the virtual string ensemble can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that is used for sending MIDI data to the DAW or other musical devices. In this example, the modulation wheel of a conventional MIDI keyboard is used to change dynamic progressions.

¹¹⁷ On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹¹⁸ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

Follow these 13 steps to start an original homophonic composition with the aid of the FPS-UIC and a virtual string ensemble.

Step 1: Add an instrument track.

Move the mouse left to the instrument window in Cubase, right-click, and select ‘Add Instrument Track’ from the pop-up menu (Figure 36).

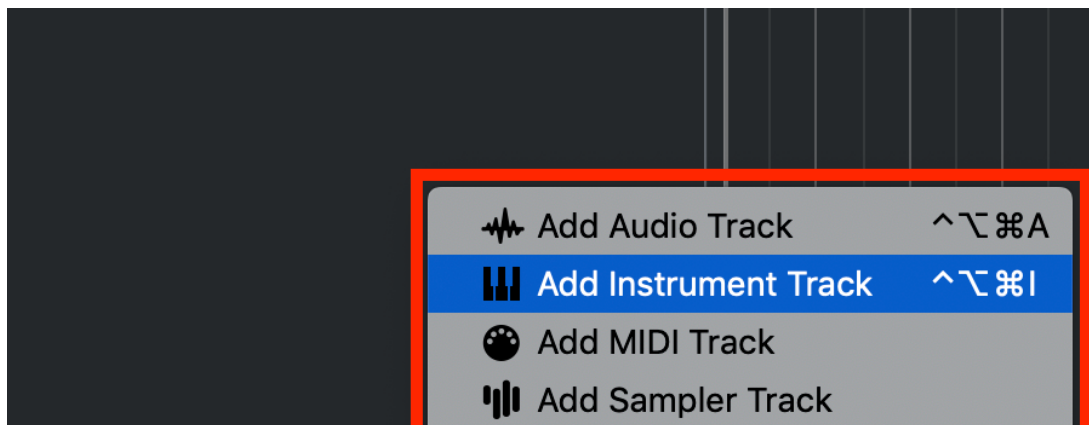


Figure 57. Add Instrument Track.

Step 2: Count nine times a Kontakt 5 instrument track.

A new screen opens. Click on the ‘Instrument’ menu in the centre of the screen, select ‘Kontakt 5’, keep the audio output on ‘Stereo’¹¹⁹, count nine times a Kontakt 5 instrument track, and press the ‘Add Track’ button to integrate nine Kontakt 5 instrument tracks into the DAW (Figure 37, number 1, 2, and 3).

¹¹⁹ The virtual string ensemble is designed for Stereo Out projects. The software is not designed for surround or binaural music projects.

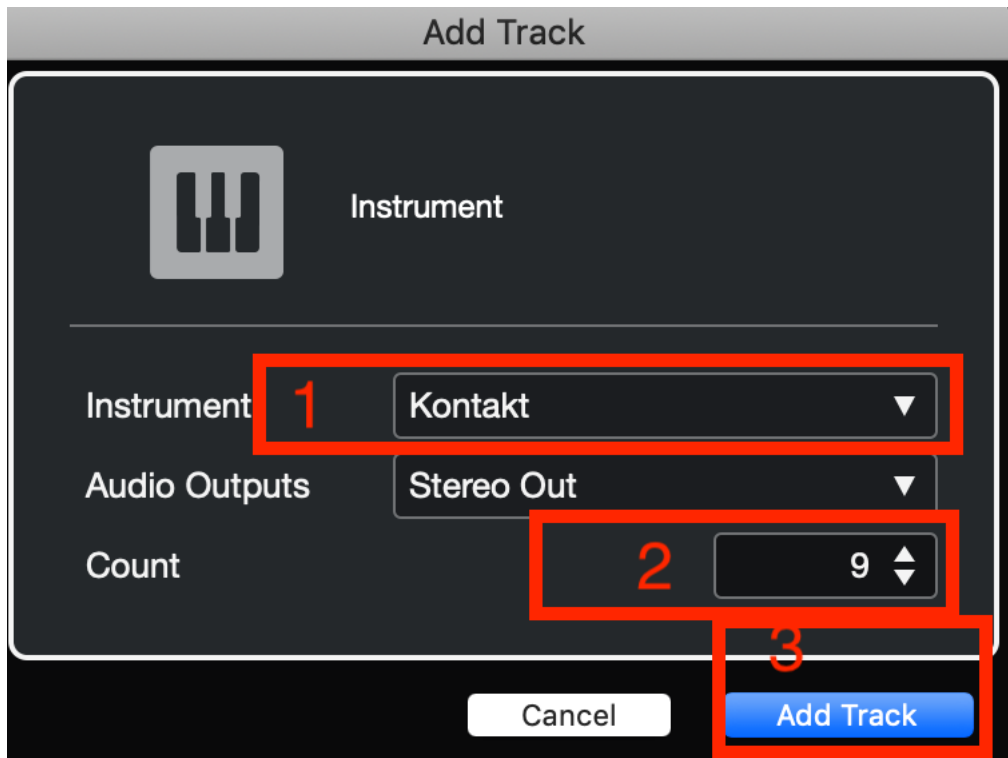


Figure 58. Add nine different Kontakt 5 virtual instrument tracks.

Step 3: Import different string instruments.

In Cubase, select the first instrument track, go to the inspector window on the left, and click on the ‘edit instrument’ icon to open Kontakt 5 (Figure 59).

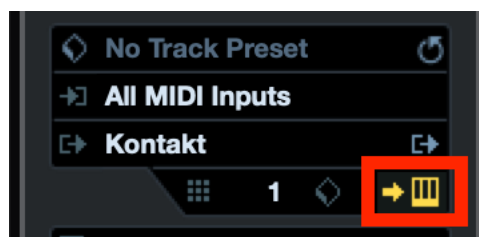


Figure 59. Open Kontakt in the first instrument track.

The Kontakt engine of the first instrument track opens. Go to ‘Libraries’, ‘Symphony Essentials String Ensemble’, and select the string instrument called ‘Violins

1 Essential.nki'¹²⁰ within the internal Kontakt menu (Figure 38, numbers 1, 2, and 3). The virtual instrument graphical user interface (GUI)¹²¹ opens. Click within the GUI at the bottom left corner on the articulation called 'Sustains'¹²² (Figure 38, number 4). The 1st Violins are now loaded including a selected articulation within Cubase.



Figure 60. Select Violins 1 Essential.nki file.

¹²⁰ The NKI file is a software format in which virtual instruments are designed with sample synthesis approaches.

¹²¹ The graphical user interface (GUI) allows music composers to interact with the virtual instrument through graphical symbols and visual indicators. The GUI is controlled with a mouse and computer keyboard.

¹²² Sustains are articulations played by string instruments.

Repeat this step for each instrument track and select different virtual instruments for each instrument track. The following table, Table 15, provides an overview of the different virtual instruments that are loaded in the different instrument tracks.

Instrument track number(s)	Virtual instrument
1 and 2	Violins 1 Essential.nki
3 and 4	Violins 2 Essential.nki
5 and 6	Violas Esssential.nki
7 and 8	Cellos Essential.nki
9	Basses Essential.nki

Table 15. An overview of the different instrument tracks.

Step 4: Change the name of the instrument tracks.

To identify the different instrument tracks, double click on the track name ‘Kontakt 01’, and change the name. In this example, select the first instrument track, and type ‘1st Violins I’ (Figure 61).



Figure 61. Change the name of the instrument track.

Rename each instrument track. The following Table 16 lists the names for the different tracks.

Instrument track number	Name
1	1 st Violins I
2	1 st Violins II
3	2 nd Violins I
4	2 nd Violins II
5	Violas I
6	Violas II

7	Cellos I
8	Cellos II
9	Double Bass

Table 16. An overview of different names for different tracks.

Step 5: Route instrument tracks to the FPS-UIC.

Click on the instrument track called ‘1st Violins I’. Go to the inspector section and click on the ‘input routing’ menu. A drop-down window opens. Select ‘Daemon Input 0’ for the first instrument track (Figure 62).

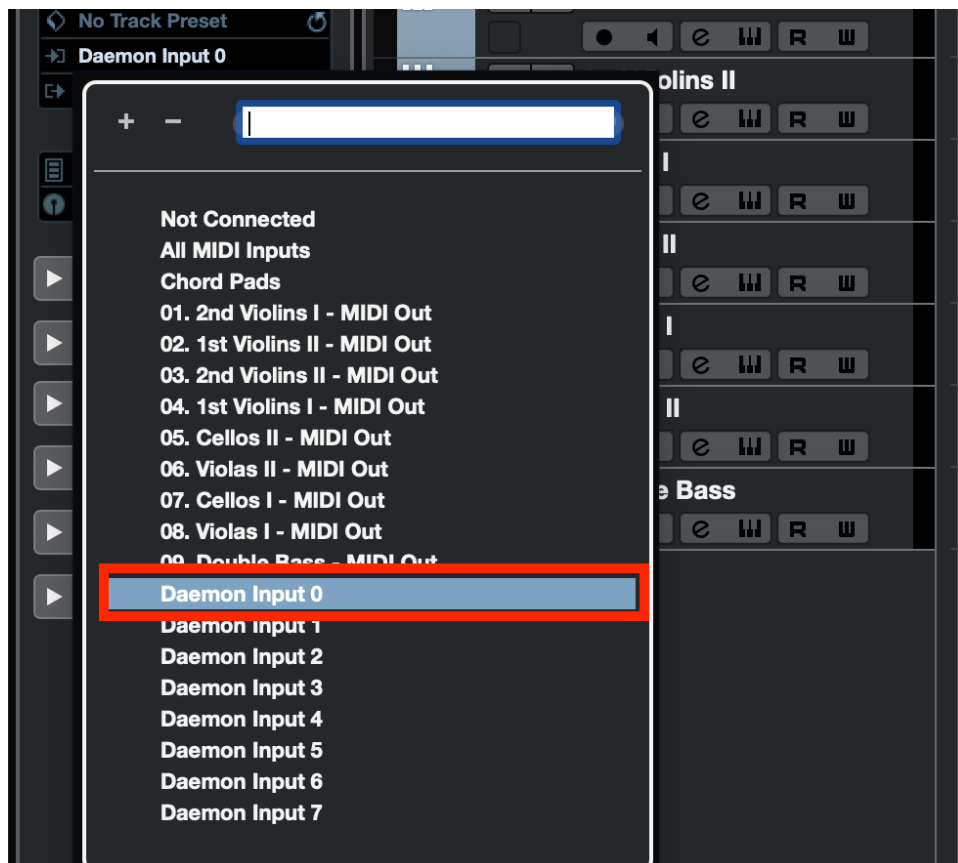


Figure 62. Route the instrument track to the FPS-UIC.

Repeat this step for each instrument track. The following Table 17 provides an overview of the routing strategy.

Name	Routing
1 st Violins I	Daemon Input 0
1 st Violins II	Daemon Input 1
2 nd Violins I	Daemon Input 2
2 nd Violins II	Daemon Input 3
Violas I	Daemon Input 4
Violas II	Daemon Input 5
Cellos I	Daemon Input 6
Cellos II and Double Bass	Daemon Input 7

Table 17. An overview of the routing strategy.

Step 6: Group instrument tracks to one string ensemble.

Select the first instrument track (1st Violins I), hold the shift button on the computer keyboard, and click on the last instrument track (Double Bass). All nine instrument tracks are now selected. Next, right-click on the instrument tracks, and a drop-down window appears. Select ‘Move Selected Tracks to a New Folder’ to group the tracks into one folder (Figure 63). Alternatively, use the shortcut combination command, option, control, and ‘v’ (hold down the command key, option key, control key on the Apple keyboard, and type in ‘v’) to group the instrument tracks.

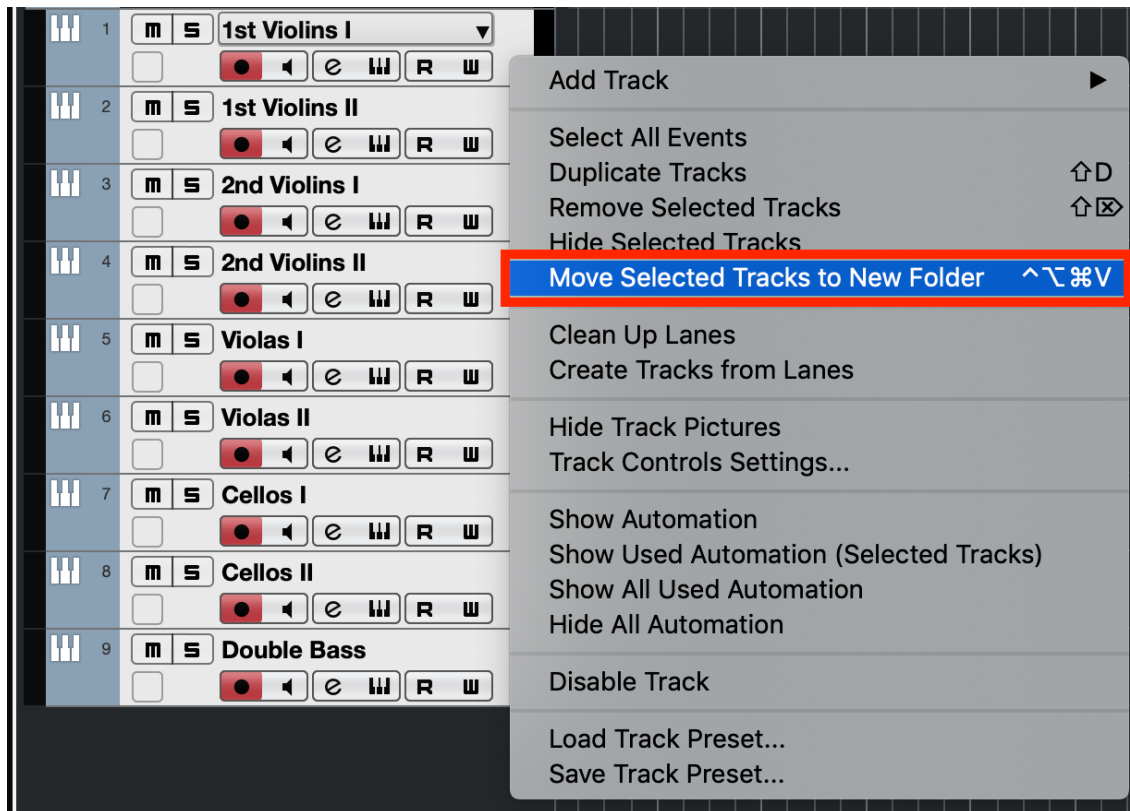


Figure 63. Move selected tracks to a new folder.

Next, rename the instrument group folder. Click on the name 'Folder' and change the name to 'String ensemble' (Figure 64). Click on the 'folder' icon (Figure 64) to collapse the folder view of the instrument tracks.



Figure 64. Rename the group folder to 'String Ensemble'.

Step 7: Select a musical scale.

Go to the iPad and select a musical major scale. To do so, click in the top left corner of the FPS-UIC on the scale menu. A drop-down window including all musical major scale options appears. In this example, select the C major scale (Figure 65, number 1).

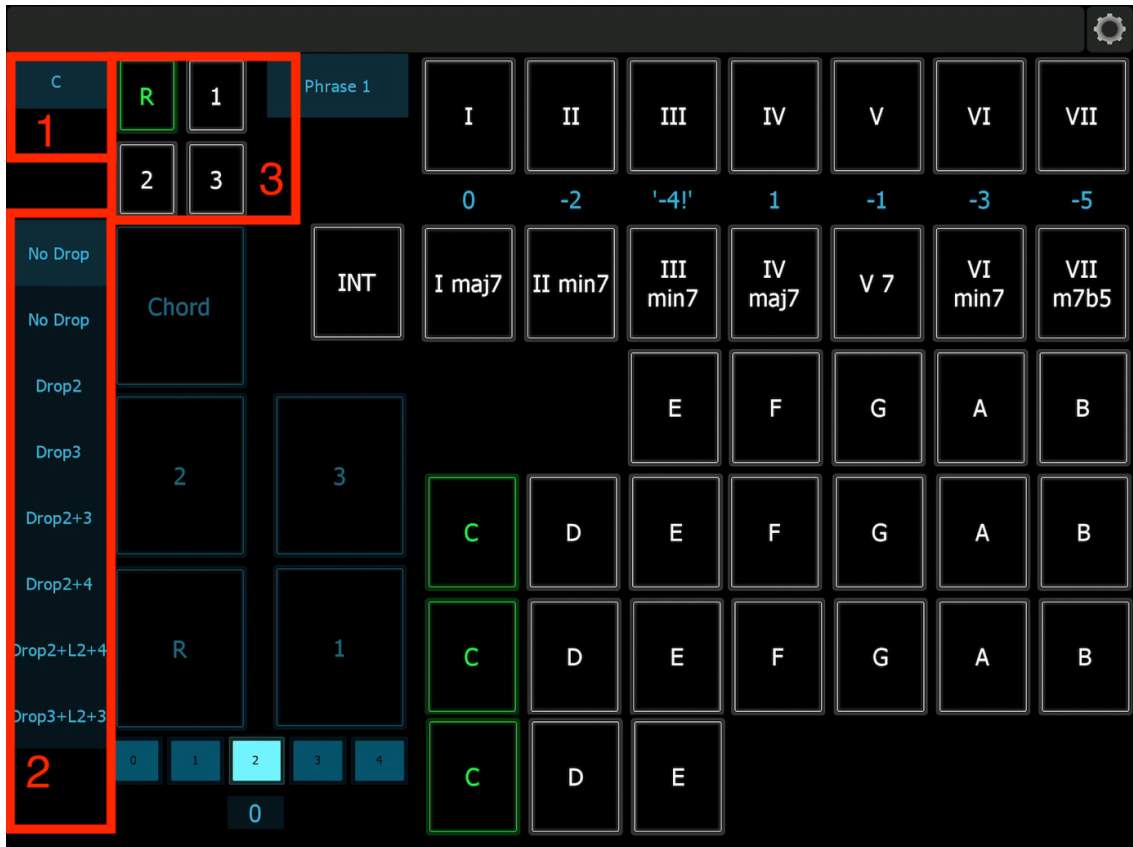


Figure 65. Select a musical scale, voicing, and inversion.

Step 8: Select a voicing.

Next, click on the ‘voicing’ menu (Figure 65, number 2). A drop-down window appears, including the following seven voicing options: No Drop, Drop2, Drop3, Drop2+3, Drop2+4, Drop2+L2+4, and Drop3+L2+3. The first five voicings include four notes (the first, third, fifth and seventh note of a chord). The last two voicing options, the Drop2 +L2+4 and the Drop3+L2+3 voicing, include nine pitches (three times the first, two times the third, two times the fifth and two times seventh note of a chord). A drop voicing is created from a chord in root position and close voicing. One or more notes of the chord are transposed down an octave. The default selected voicing is ‘No Drop’. ‘No Drop’ means that no musical notes are transposed an octave lower. The Drop2 voicing transposes the second note one octave lower, while the Drop3 voicing transposes the third note of a chord one octave lower. The Drop2+3 voicing

transposes both the second and third note one octave lower, while the Drop2+4 transposes both the second and fourth note of a chord one octave lower. The Drop2+L2+4 includes a Drop2 voicing in the upper register (above Middle C), and a Drop2+4 voicing in the lower register (below Middle C). The lowest note of the Drop2+4 voicing is doubled by an octave in the lowest register. The Drop3+L2+3 includes a Drop3 voicing in the upper register, and a Drop2+3 voicing in the lower register (below Middle C). The lowest note of the Drop2+3 voicing is doubled by an octave in the lowest register. In this example, select Drop2+L2+4 as the voicing. The drop-down window closes automatically.

Step 9: Select inversion.

Select the inversion of the selected voicing (Figure 48, number 3). ‘R’ represents ‘Root position’, ‘1’ includes the first inversion, ‘2’ the second inversion, and ‘3’ is the third inversion. Click on the ‘R’ to select the root position of the chord. The ‘R’ lights up in green.

Step 10: Select a musical chord.

Select a chord in the upper right area of the window (Figure 66, number 1). Choose between seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The first and fourth chords are Major 7 chords. The second, third, and sixth chords are Minor 7 chords. The fifth chord is a Dominant 7 chord, and the seventh chord is a half diminished 7 chord. In this example, choose the first chord (I maj7). The selected button lights up in white.

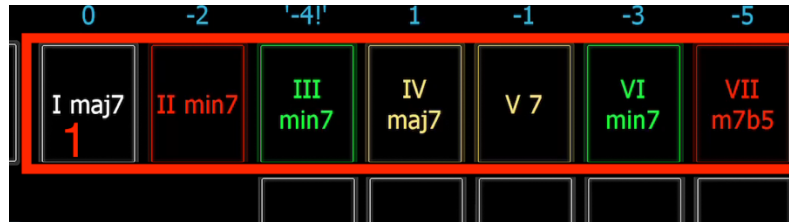


Figure 66. Select a chord.

Step 11: Enable recording.

Move the mouse to the centre bottom of the DAW window and click on the record button. The record button turns red, and the play button to the left turns green (Figure 67, number 1). The instrument track of the virtual string ensemble also turns red, and the cursor¹²³ of the DAW starts to move to the right (Figure 67, number 2).



Figure 67. Enable recording function in Cubase.

Step 12: Play a chord progression.

Go to the iPad, and hold the ‘Chord’ button on the FPS-UIC (Figure 68, number 2) to play the selected chord. Release the ‘Chord’ button to stop playing the selected chord.

¹²³ The cursor appears as a vertical line across the stave and it indicates the playback and record position of DAW.

While recording, change the selected chord by clicking on one of the other twelve chords. The chord that is coloured in white signals both the selected chord but also the selected chord row. The second chord row (Figure 68, number 1) is selected because the first chord (I maj7) is selected and highlighted in white.

A **horizontal chord switch** within the selected chord row effects chord switches within the selected musical scale. No matter which chord within the chord row is selected, the musical tonic always keeps the same. A **vertical chord switch** (Figure 68, Number 1 to Figure 68, Number 3) always creates a modulation to another musical tonic. The selected chord determines the tonic switch. The numbers 'in between' both chord rows display the positions that the tonic moves. A positive number means a clockwise movement within the Cycle of Fifth, while a negative number means a counterclockwise movement within the Cycle of Fifth. The tonic is changing from the perspective of the selected chord, in this example, the C maj7 chord. The colours signal how many musical notes within a chord are changing by moving from the selected chord to another chord: the blue colour (Figure 68, number 3, 'IV') means that **none of the four** musical notes within the chord change by switching from the selected chord to the blue highlighted chord. The green colour (Figure 68, number 1, 'III min7' and 'VI min7', and Figure 68, number 3, 'V') signals that **one of the four** musical notes change. The yellow colour (Figure 68, number 1, 'IV maj7' and 'V 7', and Figure 68, number 3, 'II', 'III', and 'VI') signals that **two of the four** musical notes change. The red colour (Figure 68, number 1, 'II min7' and 'VII m7b5', and Figure 68, number 3, 'VII') means that **three of the four** musical notes within the chord change.



Figure 68. The twelve chord options of the FPS-UIC.

In this example, click on the second chord (Figure 69, number 1). The button will change colour from red to white. Also, the chord inversion changes from ‘Root position’ to ‘3rd inversion’ (Figure 69, number 2). This is because the FPS-UIC calculates the most appropriate voice leading automatically. Hold the ‘Chord’ button again to play the newly selected chord (Figure 69, number 3). Release the ‘Chord’ button to stop playing the selected chord.

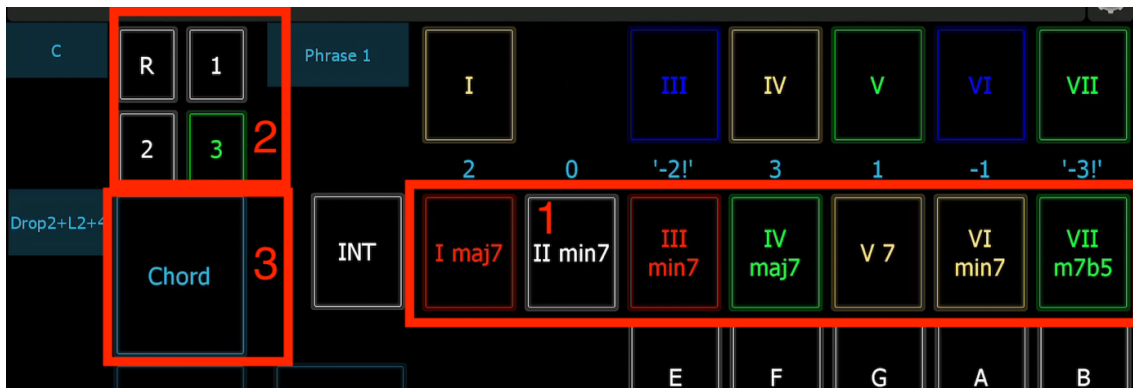


Figure 69. Change chord while recording a chord progression.

In Cubase, click on the ‘Stop’ button to the left of the ‘Play’ button to stop the recording. The different musical MIDI notes of the two homophonic chords are now

recorded within the MIDI environment of Cubase (Figure 70). If required, adjust or remove MIDI notes within the MIDI editor¹²⁴ window of Cubase.

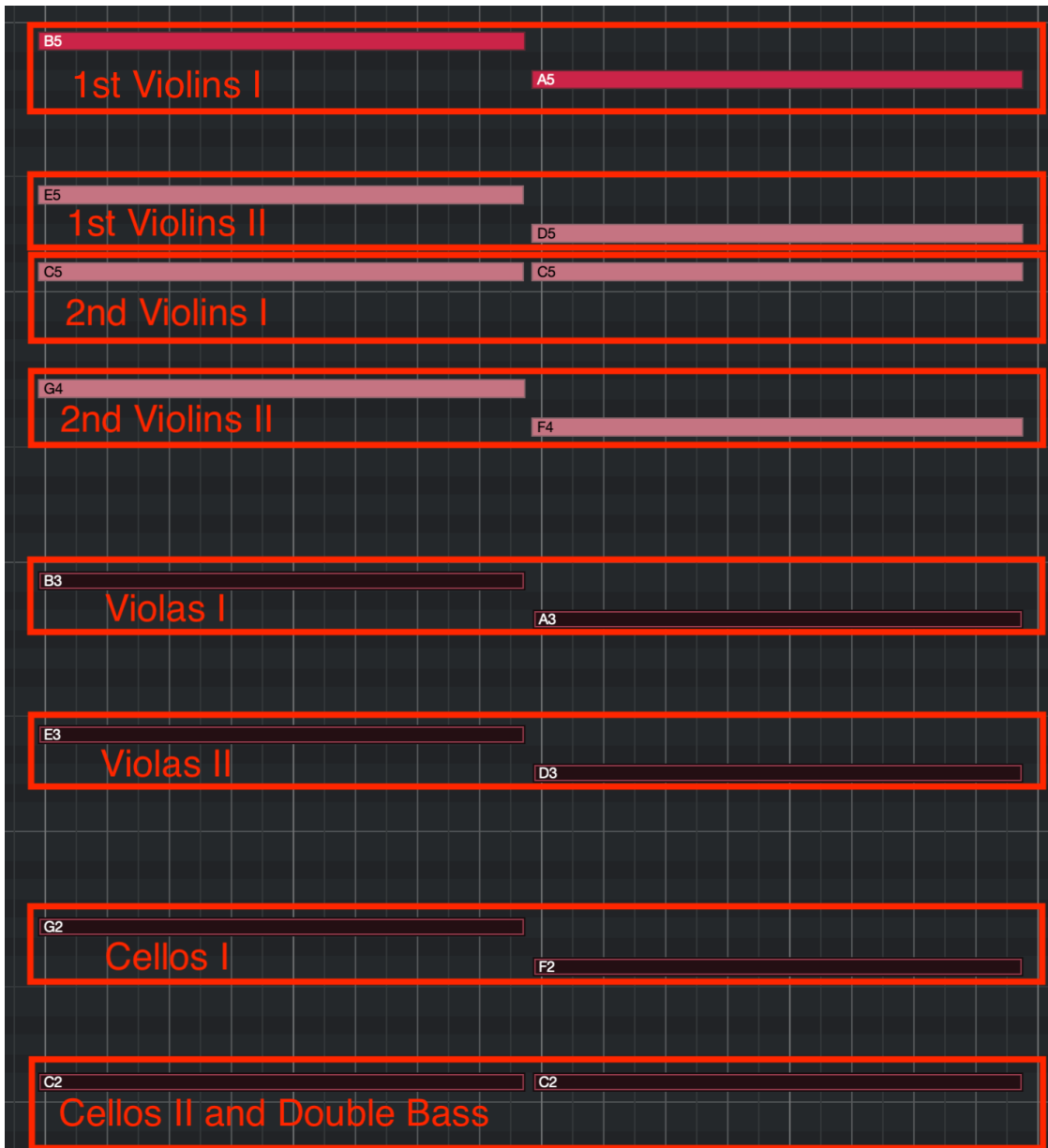


Figure 70. The MIDI editor window including the two recorded homophonic chords.

¹²⁴ The MIDI editor window of Cubase allows MIDI editing and provides an overview of the context of the recorded track.

Continue this composition, and repeat Step 7 till Step 12 in the same or different order. Play different homophonic chord progressions on the FPS-UIC. Additionally, add different instruments such as woodwinds, brass, or percussion instruments to the composition.

8.9.5 How to begin a composition with the aid of the FPS-UIC and virtual strings part 1: select the default settings and play a chord.

The fifth step-by-step user manual explains and illustrates in detail how to use the FPS-UIC¹²⁵ and a 54-piece virtual string ensemble¹²⁶ to start an original homophonic¹²⁷ composition. The virtual string ensemble is opened within Kontakt 5¹²⁸, and Kontakt 5 is used as a VST-plugin¹²⁹ within Cubase¹³⁰, a digital audio workstation (DAW). The

¹²⁵ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments, and it generates musical chords and musical scales automatically.

¹²⁶ The virtual string ensemble simulates the sound of a string ensemble. The virtual string ensemble, developed by Native instruments, simulates the following instrument sections: 1st Violin I (8 players), 1st Violin II (8 players), 2nd Violin I (7 players), 2nd Violin II (7 players), Viola I (6 players), Viola II (6 players), Cellos I (5 players), Cellos II (5 players), and Double Bass (4 players).

¹²⁷ In Homophonic music, multiple parts are moving in the same rhythm, but they are playing different pitch values.

¹²⁸ Kontakt 5, developed by Native Instruments, is a sampling synthesis environment with a graphical interface environment that can be run standalone or as an instrument ensemble. The Kontakt 5 environment is available as a VST plug-in and is easy to interface with a DAW.

¹²⁹ Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesisers and audio effects in digital audio workstations (DAWs).

¹³⁰ Cubase is a digital audio workstation (DAW) developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

software applications are installed on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The virtual string ensemble is controlled by the FPS-UIC¹³¹, an Apple computer keyboard, and a computer mouse that includes a left, middle¹³², and right¹³³ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac), the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet), and the fourth step-by-step tutorial (4. How to set up a virtual string ensemble in Cubase) to continue to this fifth step-by-step tutorial.

In this first part, composing music with the aid of the FPS-UIC and a virtual string ensemble requires 5 steps: (1) Select a chord; (2) enable recording; (3) play the selected chord; (4) play notes of the selected chord separately; (5) play notes of the selected scale separately. Follow these 5 steps to start an original homophonic composition with default settings with the aid of the FPS-UIC and a virtual string ensemble.

¹³¹ Optionally, the virtual string ensemble can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that is used for sending MIDI data to the DAW or other musical devices. In this example, the modulation wheel of a conventional MIDI keyboard is used to change dynamic progressions.

¹³² On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹³³ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

Step 1: Select a chord.

Select a chord in the upper right area of the window (Figure 71, number 1). Choose between seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The first and fourth chords are Major 7 chords. The second, third, and sixth chords are Minor 7 chords. The fifth chord is a Dominant 7 chord, and the seventh chord is a half diminished 7 chord. In this example, choose the first chord (I maj7). The selected button lights up in white.

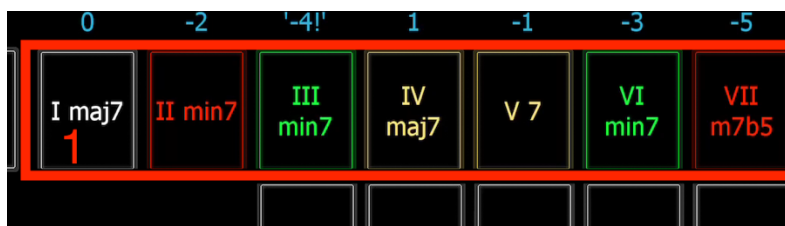


Figure 71. Select a chord.

Step 2: Enable recording.

Move the mouse to the centre bottom of the DAW window and click on the record button. The record button turns red and the play button to the left turns green (Figure 72, number 1). The instrument track of the virtual string ensemble also turns red, and the cursor¹³⁴ of the DAW starts to move to the right (Figure 72, number 2).

¹³⁴ The cursor appears as a vertical line across the staff, and it indicates the playback and record position of DAW.

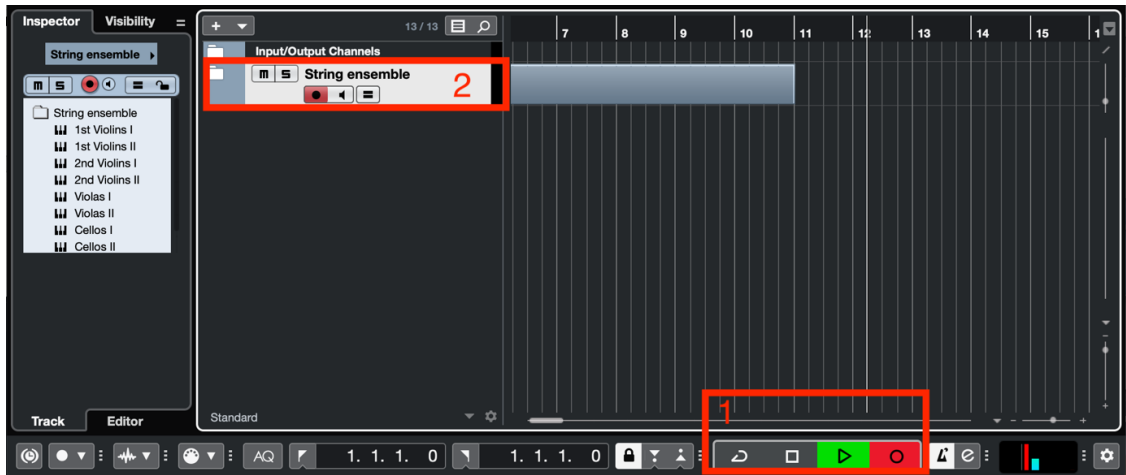


Figure 72. Enable recording function in Cubase.

Step 3: Play the chord.

Go to the iPad, and hold the 'Chord' button on the FPS-UIC (Figure 73, number 2) to play the selected chord. Release the 'Chord' button to stop playing the selected chord.

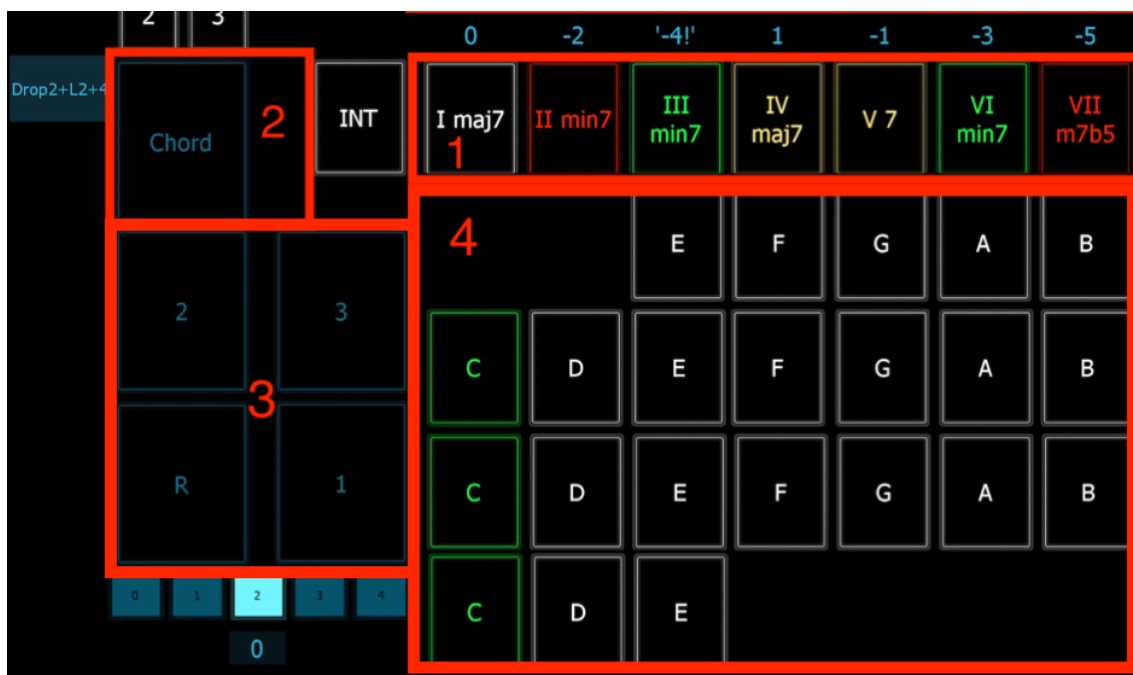


Figure 73. The seven chord options of the FPS-UIC.

Step 4: Play notes of the selected chord separately.

Hold the 'R' button on the FPS-UIC (Figure 73, number 3) to play the root note of the selected chord separately. The button lights up. Release the 'R' button to stop playing

the root note. Furthermore, hold the ‘1’, ‘2’, or ‘3’ button to play the first, second, or third note of the selected chord, respectively.

Step 5: Play notes of the selected scale separately.

On the iPad, go to the single note area (Figure 73, number 4), and hold one of the displayed buttons. The button lights up. For example, hold and release the ‘C’, ‘D’, and ‘E’ button, respectively. In Cubase, click on the ‘Stop’ button to the left of the ‘Play’ button to stop the recording. The different musical MIDI notes are now recorded within the MIDI environment of Cubase. If required, adjust or remove MIDI notes within the MIDI editor¹³⁵ window of Cubase.

Continue this composition, and repeat Step 1 till Step 5 in the same or a different order. Play different chords and notes on the FPS-UIC. Additionally, add different instruments such as woodwinds, brass, or percussion instruments to the composition.

¹³⁵ The MIDI editor window of Cubase allows MIDI editing, and it provides an overview of the context of the recorded track.

8.9.6 How to begin a composition with the aid of the FPS-UIC and virtual strings part 2: select a different scale and inversion and play different chords.

This video/manual explains and illustrates how to use the FPS-UIC¹³⁶ and a 54-piece virtual string ensemble¹³⁷ to start an original homophonic¹³⁸ composition. The virtual string ensemble is opened within Kontakt 5¹³⁹, and Kontakt 5 is used as a VST-plugin¹⁴⁰ within Cubase¹⁴¹, a digital audio workstation (DAW). The software applications are

¹³⁶ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments and generates musical chords and musical scales automatically.

¹³⁷ The virtual string ensemble simulates the sound of a string ensemble. The virtual string ensemble, developed by Native instruments, simulates the following instrument sections: 1st Violin I (8 players), 1st Violin II (8 players), 2nd Violin I (7 players), 2nd Violin II (7 players), Viola I (6 players), Viola II (6 players), Cellos I (5 players), Cellos II (5 players), and Double Bass (4 players).

¹³⁸ In Homophonic music, multiple parts are moving in the same rhythm, but they are playing different pitch values.

¹³⁹ Kontakt 5, developed by Native Instruments, is a sampling synthesis environment with a graphical interface environment that can be run standalone or as an instrument ensemble. The Kontakt 5 environment is available as a VST plug-in and is easy to interface with a DAW.

¹⁴⁰ Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesisers and audio effects in digital audio workstations (DAWs).

¹⁴¹ Cubase is a digital audio workstation (DAW), developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

installed on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The virtual string ensemble is controlled by the FPS-UIC¹⁴², an Apple computer keyboard, and a computer mouse that includes a left, middle¹⁴³, and right¹⁴⁴ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac computer), the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet), and the fourth step-by-step tutorial (4. How to set up a virtual string ensemble in Cubase) to continue to this sixth step-by-step tutorial.

In this second part, composing music with the aid of the FPS-UIC and a virtual string ensemble requires 5 steps: (1) select a musical scale; (2) select an inversion, (3) select a chord, (4) enable recording, (5) play a chord progression; Follow these 5 steps to start an original homophonic composition with the aid of the FPS-UIC and a virtual string ensemble.

¹⁴² Optionally, the virtual string ensemble can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that is used for sending MIDI data to the DAW or other musical devices. In this example, the modulation wheel of a conventional MIDI keyboard is used to change dynamic progressions.

¹⁴³ On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹⁴⁴ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard and click on the single mouse button to simulate a right mouse click.

Step 1: Select a musical scale.

Go to the iPad and select a musical major scale. To do so, click in the top left corner of the FPS-UIC on the scale menu.

A drop-down window including all musical major scale options appears. In this example, select the G major scale (Figure 74, number 1).

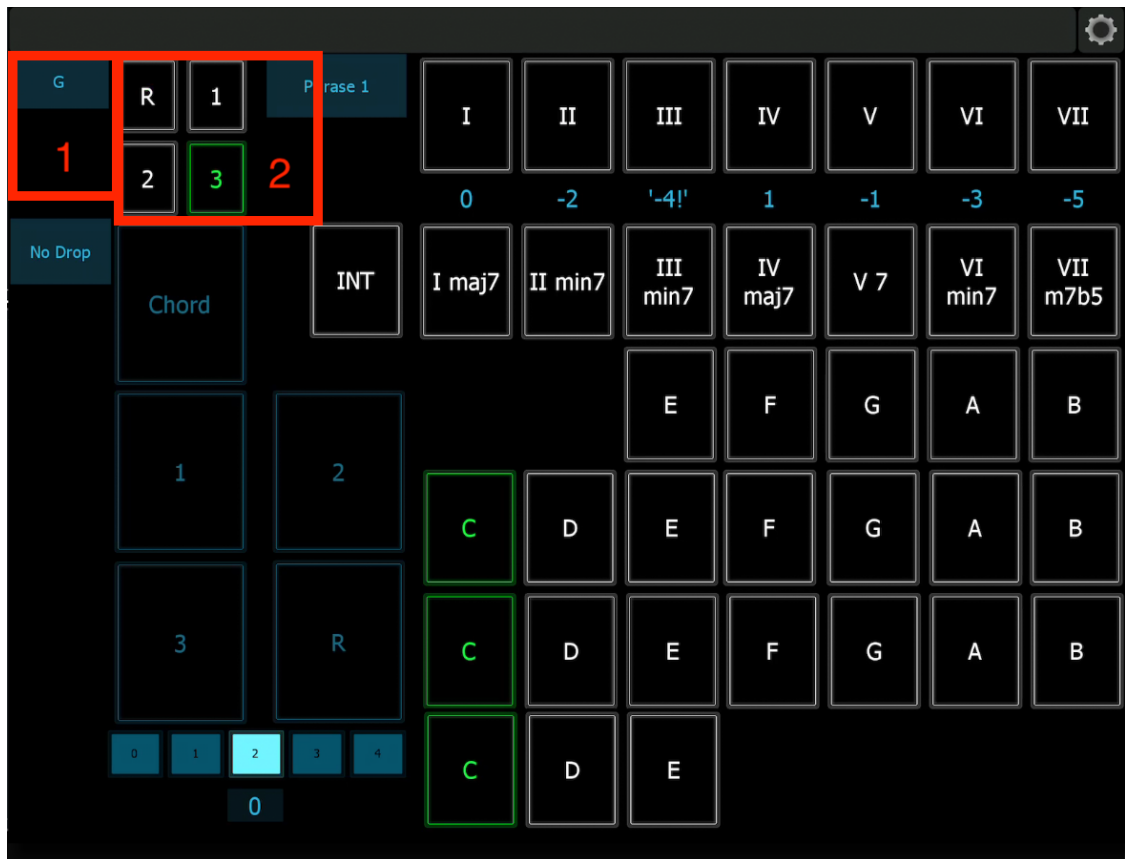


Figure 74. Select a musical scale, voicing, and inversion.

Step 2: Select an inversion.

Select the inversion of the chord (Figure 74, number 2). 'R' represents 'Root position', '1' is for the first inversion, '2' the second inversion, and '3' is the third inversion.

Click on the '3' to select the third inversion of the chord. The '3' lights up in green.

Step 3: Select a chord.

Select a chord in the upper right area of the window (Figure 75, number 1). Choose between the seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The first and fourth chords are Major 7 chords. The second, third, and sixth chords are Minor 7 chords. The fifth chord is a Dominant 7 chord, and the seventh chord is a half diminished 7 chord. In this example, choose the first chord (I maj7). The selected button lights up in white.

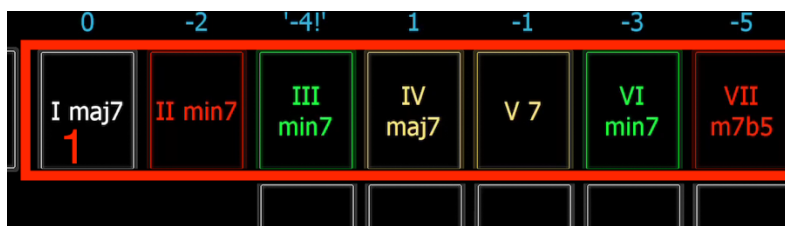


Figure 75. Select a chord.

Step 4: Enable recording.

Move the mouse to the centre bottom of the DAW window and click on the record button. The record button turns red, and the play button to the left turns green (Figure 76, number 1). The instrument track of the virtual string ensemble also turns red, and the cursor¹⁴⁵ of the DAW starts to move to the right (Figure 76, number 2).

¹⁴⁵ The cursor appears as a vertical line across the stave and indicates the playback and record position of DAW.



Figure 76. Enable recording function in Cubase.

Step 5: Play a chord progression.

Go to the iPad and hold the ‘Chord’ button on the FPS-UIC (Figure 77, number 2) to play the first selected chord. Release the ‘Chord’ button to stop playing the selected chord. While recording, change the selected chord by clicking on one of the other six chords within the row (Figure 77, number 1).

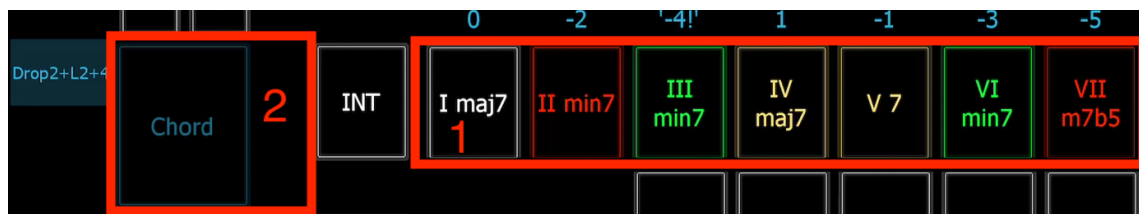


Figure 77. The twelve chord options of the FPS-UIC.

In this example, click on the second chord (Figure 78, number 1). The button changes colour from red to white. Also, the chord inversion changes from inversion ‘3’ to inversion ‘2’ (Figure 78, number 2). This is because the FPS-UIC calculates the most appropriate inversion and voice leading automatically. Hold the ‘Chord’ button again to play the newly selected chord (Figure 78, number 3). Release the ‘Chord’ button to stop playing the selected chord. Repeat this step and select the third and the fourth chord of the row.

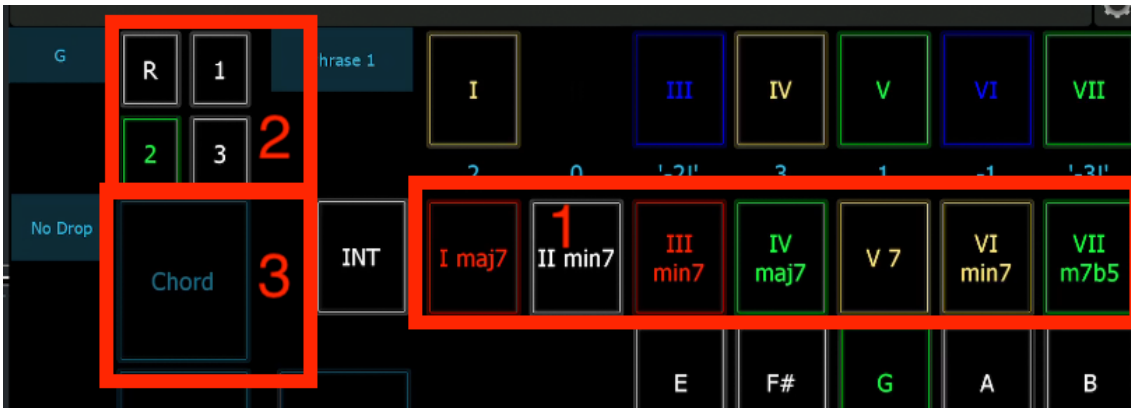


Figure 78. Change chord while recording chord progression.

In Cubase, click on the 'Stop' button to the left of the 'Play' button to stop the recording. The different musical MIDI notes of the two homophonic chords are now recorded within the MIDI environment of Cubase. If required, adjust or remove MIDI notes within the MIDI editor¹⁴⁶ window of Cubase (Figure 79).

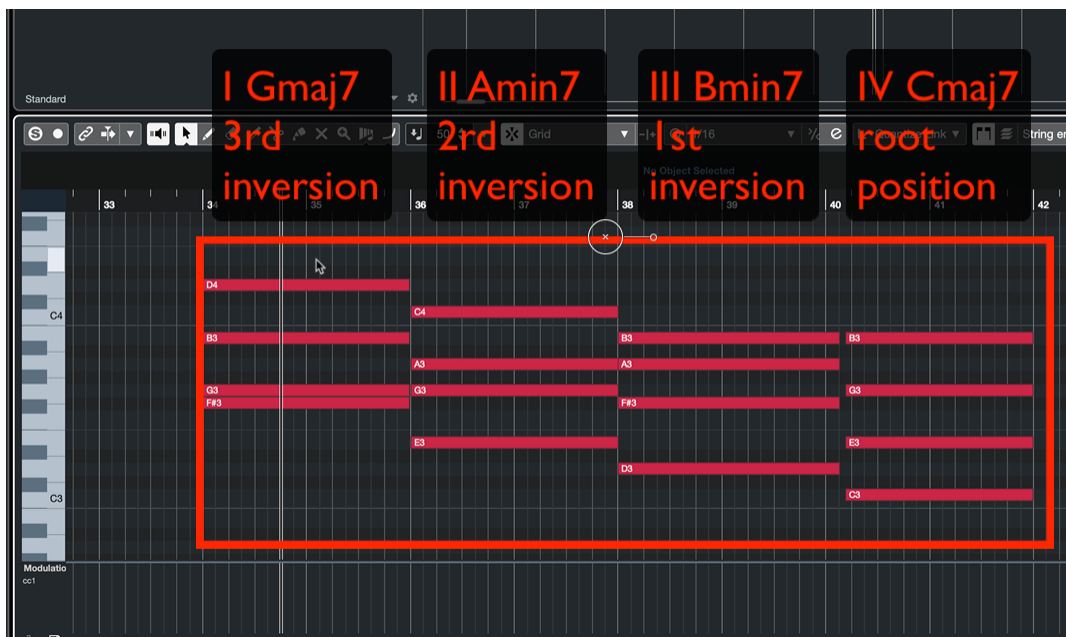


Figure 79. The recorded MIDI notes within the MIDI editor of Cubase.

¹⁴⁶ The MIDI editor window of Cubase allows MIDI editing and provides an overview of the context of the recorded track.

Continue this composition and repeat Step 1 till Step 5 in the same or different order. Play different homophonic chord progressions on the FPS-UIC. Additionally, add different instruments such as woodwinds, brass, or percussion instruments to the composition.

8.9.7 How to begin a composition with the aid of the FPS-UIC and virtual strings part 3: select different homophonic voicings.

The seventh step-by-step user manual explains and illustrates in detail how to use the FPS-UIC¹⁴⁷ and a 54-piece virtual string ensemble¹⁴⁸ to compose original homophonic¹⁴⁹ music. The virtual string ensemble is opened within Kontakt 5¹⁵⁰, and Kontakt 5 is used as a VST-plugin¹⁵¹ within Cubase¹⁵², a digital audio workstation

¹⁴⁷ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments and generates musical chords and musical scales automatically.

¹⁴⁸ The virtual string ensemble simulates the sound of a string ensemble. The virtual string ensemble, developed by Native instruments, simulates the following instrument sections: 1st Violin I (8 players), 1st Violin II (8 players), 2nd Violin I (7 players), 2nd Violin II (7 players), Viola I (6 players), Viola II (6 players), Cellos I (5 players), Cellos II (5 players), and Double Bass (4 players).

¹⁴⁹ In Homophonic music, multiple parts are moving in the same rhythm, but they are playing different pitch values.

¹⁵⁰ Kontakt 5, developed by Native Instruments, is a sampling synthesis environment with a graphical interface environment that can be run standalone or as an instrument ensemble. The Kontakt 5 environment is available as a VST plug-in and is easy to interface with a DAW.

¹⁵¹ Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesisers and audio effects in digital audio workstations (DAWs).

¹⁵² Cubase is a digital audio workstation (DAW) developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

(DAW). The software applications are installed on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The virtual string ensemble is controlled by the FPS-UIC¹⁵³, an Apple computer keyboard, and a computer mouse that includes a left, middle¹⁵⁴, and right¹⁵⁵ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac computer), the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet), and the fourth step-by-step tutorial (4. How to set up a virtual string ensemble in Cubase) to continue to this seventh step-by-step tutorial.

In the third part, composing music with the aid of the FPS-UIC and a virtual string ensemble requires 4 steps: (1) select a chord; (2) select a voicing; (3) enable recording; (4) play a chord progression.

Follow these four steps to start an original homophonic composition, including the selection of different voicings and different chords, which is done with the aid of the FPS-UIC and a virtual string ensemble.

¹⁵³ Optionally, the virtual string ensemble can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that is used for sending MIDI data to the DAW or other musical devices. In this example, the modulation wheel of a conventional MIDI keyboard is used to change dynamic progressions.

¹⁵⁴ On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹⁵⁵ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

Step 1: Select a chord.

Select a chord in the upper right area of the window (Figure 80, number 1). Choose between the seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The first and fourth chords are Major 7 chords. The second, third, and sixth chords are Minor 7 chords. The fifth chord is a Dominant 7 chord, and the seventh chord is a half diminished 7 chord. In this example, choose the third chord (III min7). The selected button lights up in white.



Figure 80. Select a chord.

Step 2: Select a voicing.

Click on the 'voicing' menu (Figure 81, number 1). A drop-down window appears with the following seven voicing options: No Drop, Drop2, Drop3, Drop2+3, Drop2+4, Drop2+L2+4, and Drop3+L2+3.



Figure 81. The seven possible voicing options.

The first five voicings include four notes (the first, third, fifth, and seventh note of a chord). The last two voicing options, the Drop2 +L2+4 and the Drop3+L2+3 voicing include nine pitches (three times the first, two times the third, two times the fifth, and two times seventh note of a chord). A drop voicing is created from a chord in root position and close voicing. One or more notes of the chord are transposed down an octave. The default selected voicing is 'No Drop'. No Drop means that no musical notes are transposed an octave lower. The Drop2 voicing transposes the second note one octave lower, while the Drop3 voicing transposes the third note of a chord one octave lower. The Drop2+3 voicing transposes both the second and third note one octave lower, while the Drop2+4 transposes both the second and fourth note of a chord one octave lower. The Drop2+L2+4 includes a Drop2 voicing in the upper register (above Middle C), and a Drop2+4 voicing in the lower register (below Middle C). The lowest note of the Drop2+4 voicing is doubled by an octave in the lowest register. The Drop3+L2+3 includes a Drop3 voicing in the upper register, and a Drop2+3 voicing in the lower register (below Middle C). The lowest note of the Drop2+3 voicing is doubled by an octave in the lowest register. In this example, select Drop2+4 as the voicing (Figure 82, number 1). The drop-down window closes automatically.

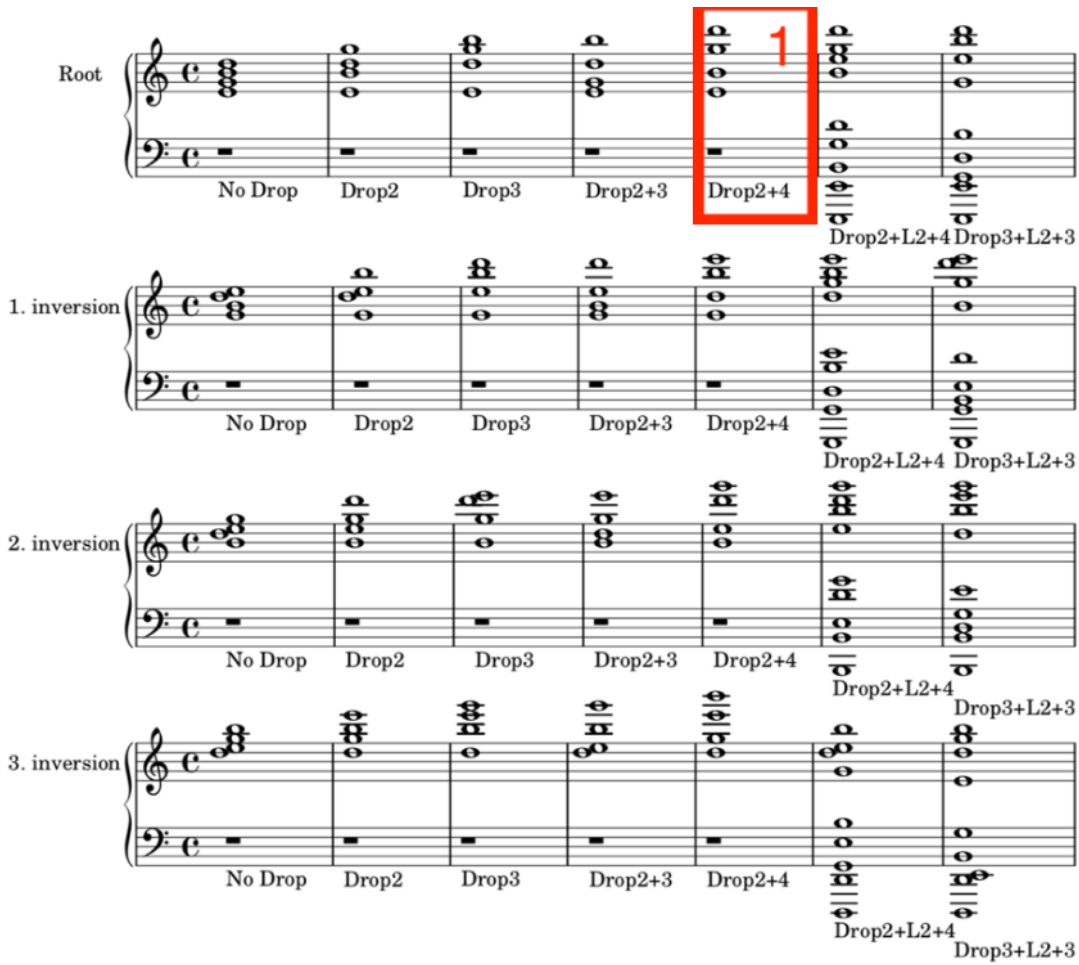


Figure 82. All seven drop voicings with different inversions.

Step 3: Enable recording.

Move the mouse to the centre bottom of the DAW window and click on the record button. The record button turns red and the play button to the left turns green (Figure 83, number 1). The instrument track of the virtual string ensemble also turns red and the cursor¹⁵⁶ of the DAW starts to move to the right (Figure 83, number 2).

¹⁵⁶ The cursor appears as a vertical line across the staff and indicates the playback and record position of DAW.



Figure 83. Enable the recording function in Cubase.

Step 4: Play a chord.

Go to the iPad and hold the 'Chord' button on the FPS-UIC (Figure 84, number 1) to play the selected chord and Drop2+4 voicing. Release the 'Chord' button to stop playing the selected chord.

Step 5: Select a different voicing.

While recording, change the selected voicing by clicking on one of the other seven voicings. In this example, open the voicing menu, and then click on the Drop2+L2+4 voicing. Again, click on the chord button (Figure 84, number 1) and play the second selected chord, including the Drop2+L2+4 voicing.

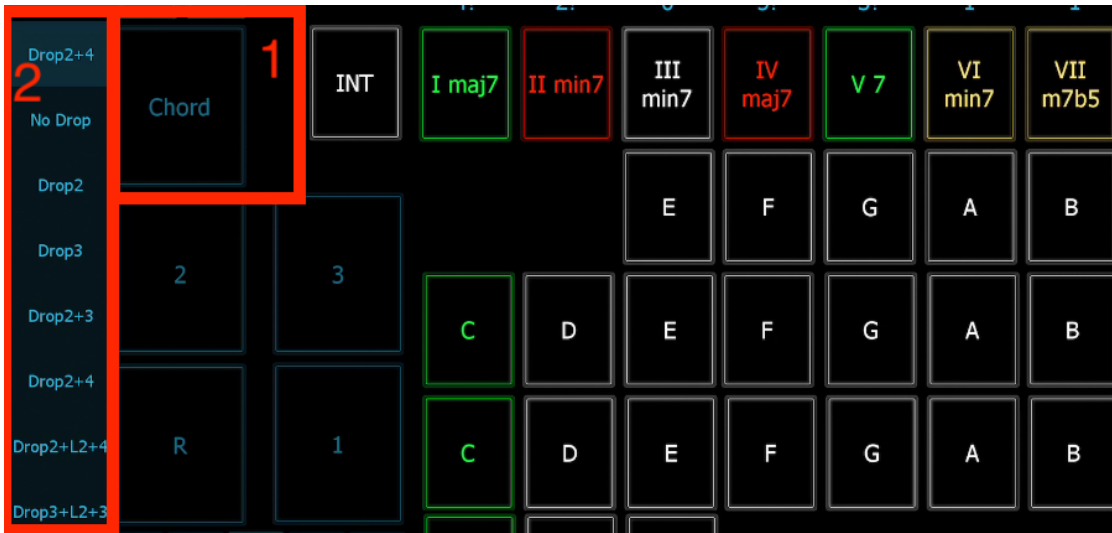


Figure 84. Play the chord and select a different voicing while recording.

In Cubase, click on the 'Stop' button to the left of the 'Play' button to stop the recording. The different musical MIDI notes of the two homophonic chords are now recorded within the MIDI environment of Cubase (Figure 85). If required, adjust or remove MIDI notes within the MIDI editor¹⁵⁷ window of Cubase.

¹⁵⁷ The MIDI editor window of Cubase allows MIDI editing and provides an overview of the context of the recorded track.

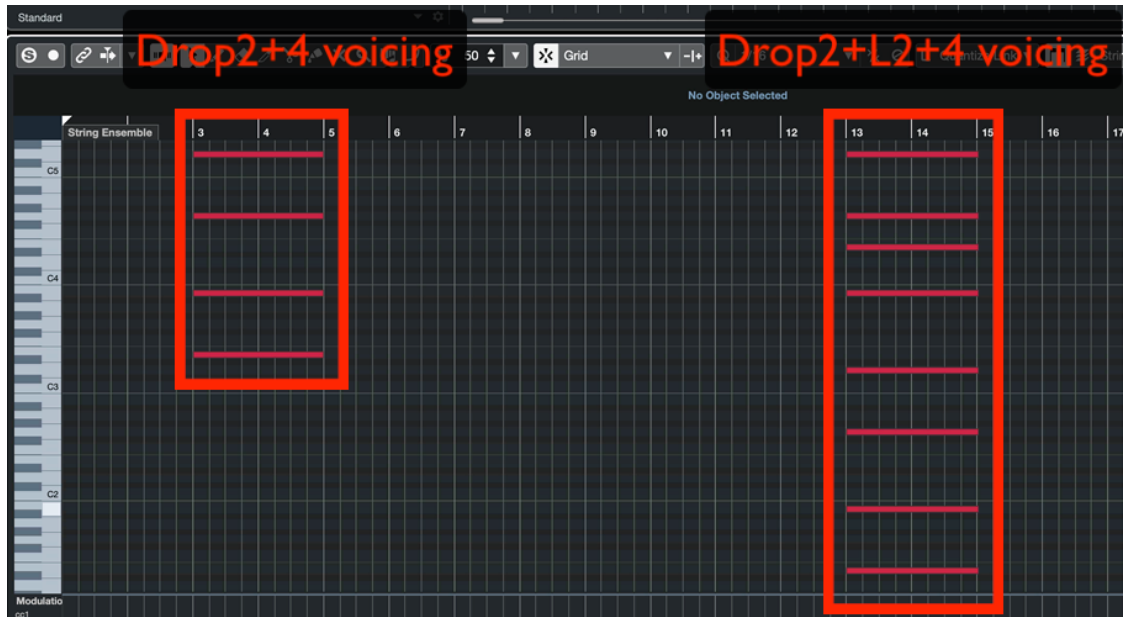


Figure 85. The MIDI editor window including the two recorded homophonic chords.

Continue this composition and repeat Step 1 to Step 4 in the same or a different order. Play different homophonic chord progressions on the FPS-UIC. Additionally, add different instruments such as woodwinds, brass, or percussion instruments to the composition.

8.9.8 How to begin a composition with the aid of the FPS-UIC and virtual strings part 4: modulate between different scales.

The eighth step-by-step user manual explains and illustrates in detail how to use the FPS-UIC¹⁵⁸ and a 54-piece virtual string ensemble¹⁵⁹ to start an original homophonic¹⁶⁰ composition. The virtual string ensemble is opened within Kontakt 5¹⁶¹, and Kontakt 5 is used as a VST-plugin¹⁶² within Cubase¹⁶³, a digital audio workstation (DAW). The

¹⁵⁸ The ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) is a MIDI controller that sends MIDI data to the digital audio workstation (DAW). It controls the pitch values of virtual instruments and generates musical chords and musical scales automatically.

¹⁵⁹ The virtual string ensemble simulates the sound of a string ensemble. The virtual string ensemble, developed by Native instruments, simulates the following instrument sections: 1st Violin I (8 players), 1st Violin II (8 players), 2nd Violin I (7 players), 2nd Violin II (7 players), Viola I (6 players), Viola II (6 players), Cellos I (5 players), Cellos II (5 players), and Double Bass (4 players).

¹⁶⁰ In Homophonic music, multiple parts are moving in the same rhythm, but they are playing different pitch values.

¹⁶¹ Kontakt 5, developed by Native Instruments, is a sampling synthesis environment with a graphical interface environment that can be run standalone or as an instrument ensemble. The Kontakt 5 environment is available as a VST plug-in and is easy to interface with a DAW.

¹⁶² Virtual Studio Technology (VST) is a software interface developed by Steinberg GmbH that integrates musical software synthesisers and audio effects in digital audio workstations (DAWs).

¹⁶³ Cubase is a digital audio workstation (DAW) developed by Steinberg GmbH. The DAW software environment is used for recording, editing, and producing music and sounds. Aside from Cubase, many other DAWs are available on the market, such as Apple’s Logic Pro or Avid’s Pro Tools. However, Cubase has established itself as one of the leading DAW’s for composing film scores, trailer music, or commercial music.

software applications are installed on a Mac computer, and the Mac computer is connected to the iPad tablet with a USB cable. The virtual string ensemble is controlled by the FPS-UIC¹⁶⁴, an Apple computer keyboard, and a computer mouse that includes a left, middle¹⁶⁵, and right¹⁶⁶ button. Follow the first step-by-step tutorial (1. How to set up Cubase on a Mac computer), the second step-by-step tutorial (2. How to set up the ‘Flexible Phrase System User Interface Controller’ (FPS-UIC) on an external iPad tablet), and the fourth step-by-step tutorial (4. How to set up a virtual string ensemble in Cubase) to continue to this eighth step-by-step tutorial.

The fourth part, composing music with the aid of the FPS-UIC and a virtual string ensemble requires seven steps: (1) Select a chord; (2) enable recording; (3) play the selected chord; (4) Select a chord to modulate to another scale and play the chord; (5) select a chord within the selected scale and play the chord; (6) Select a chord to modulate to another scale and play the chord; (7) select a chord within the selected scale and play the chord. Follow these seven steps to start an original homophonic

¹⁶⁴ Optionally, the virtual string ensemble can also be controlled by a standard MIDI keyboard. A conventional MIDI keyboard is an external piano-style keyboard that is used for sending MIDI data to the DAW or other musical devices. In this example, the modulation wheel of a conventional MIDI keyboard is used to change dynamic progressions.

¹⁶⁵ On a Mac computer, a trackpad or Apple mouse does not include a standardised middle mouse configuration. However, the trackpad and Apple mouse can be configured. Besides that, every standard external mouse also includes a middle mouse button option.

¹⁶⁶ In case of a computer mouse that only includes a single mouse button, hold down the control key on the Apple keyboard, and click on the single mouse button to simulate a right mouse click.

composition and modulate between different scales, with the aid of the FPS-UIC and a virtual string ensemble.

Step 1: Select a chord.

Select a chord in the upper right area of the window (Figure 86, number 1). Choose between the seven possible 7th chords (displayed in roman numerals) that occur naturally within the selected scale. The first and fourth chords are Major 7 chords. The second, third, and sixth chords are Minor 7 chords. The fifth chord is a Dominant 7 chord, and the seventh chord is a half diminished 7 chord. In this example, choose the first chord (I maj7). The selected button lights up in white.

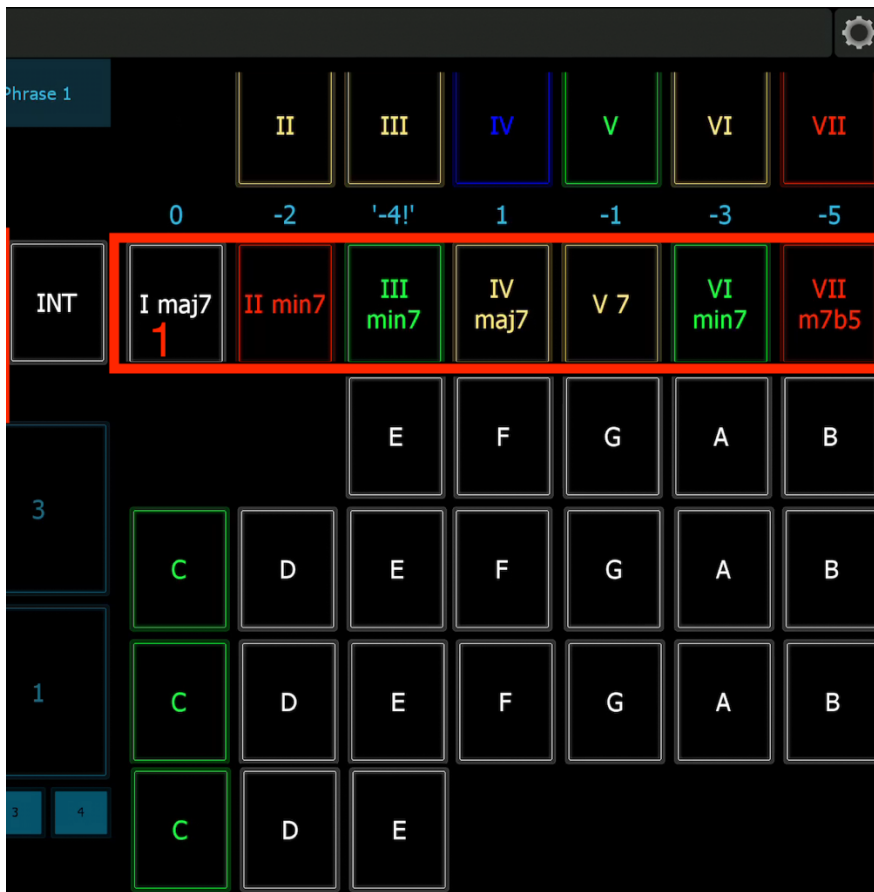


Figure 86. Select a chord.

Step 2: Enable recording.

Move the mouse to the centre bottom of the DAW window and click on the record button. The record button turns red and the play button to the left turns green (Figure 87, number 1). The instrument track of the virtual string ensemble also turns red and the cursor¹⁶⁷ of the DAW starts to move to the right (Figure 87, number 2).

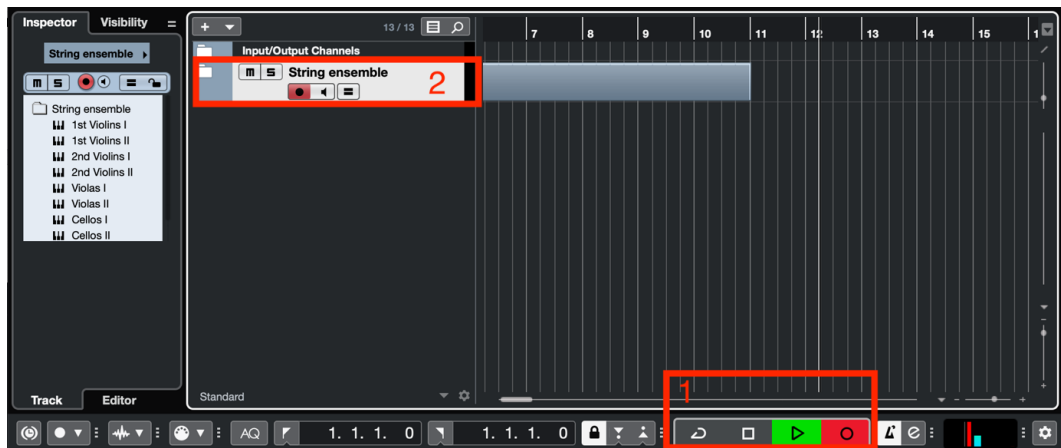


Figure 87. Enable the recording function in Cubase.

Step 3: Play the selected chord.

Go to the iPad, and hold the 'Chord' button on the FPS-UIC (Figure 88, number 2) to play the selected chord. Release the 'Chord' button to stop playing the selected chord.

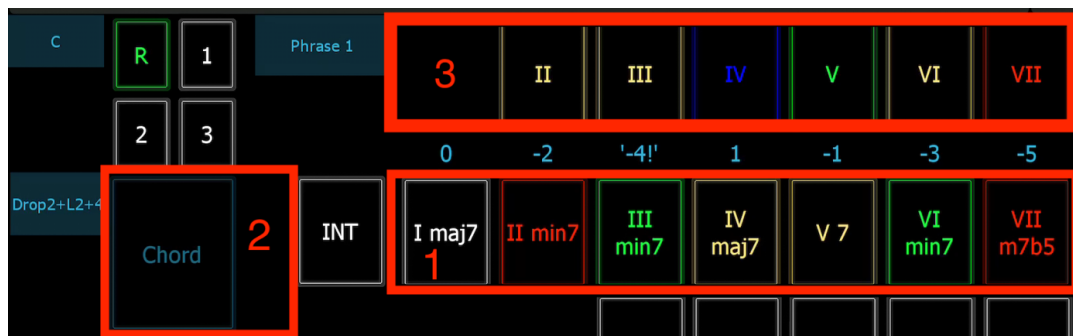


Figure 88. The twelve chord options of the FPS-UIC.

¹⁶⁷ The cursor appears as a vertical line across the staff and indicates the playback and record position of DAW.

Step 4: Select a chord to modulate to another scale and play the chord.

While recording, change the selected chord and scale by clicking on one of the other six chords in the first row (Figure 88, number 3). The chord that is coloured in white signals both the selected chord but also the selected chord row. The second chord row (Figure 88, number 1) is selected because the first chord (I maj7) is selected and highlighted in white.

A **horizontal chord switch** within the selected chord row affects chord switches within the selected musical scale. No matter which chord within the chord row is selected, the musical tonic always stays the same. However, a **vertical chord switch** (from Figure 88, Number 1 to Figure 88, Number 3) always creates a modulation to another musical tonic. The selected chord determines the tonic switch. It is essential to mention that the root note of the chord always stays the same. The numbers ‘in between’ the two chord rows display the positions that the tonic moves. A positive number means a clockwise movement within the Cycle of Fifth. A negative number means a counterclockwise movement within the Cycle of Fifth. The tonic is changing from the perspective of the selected chord, in this example, Cmaj7 (Figure 89).

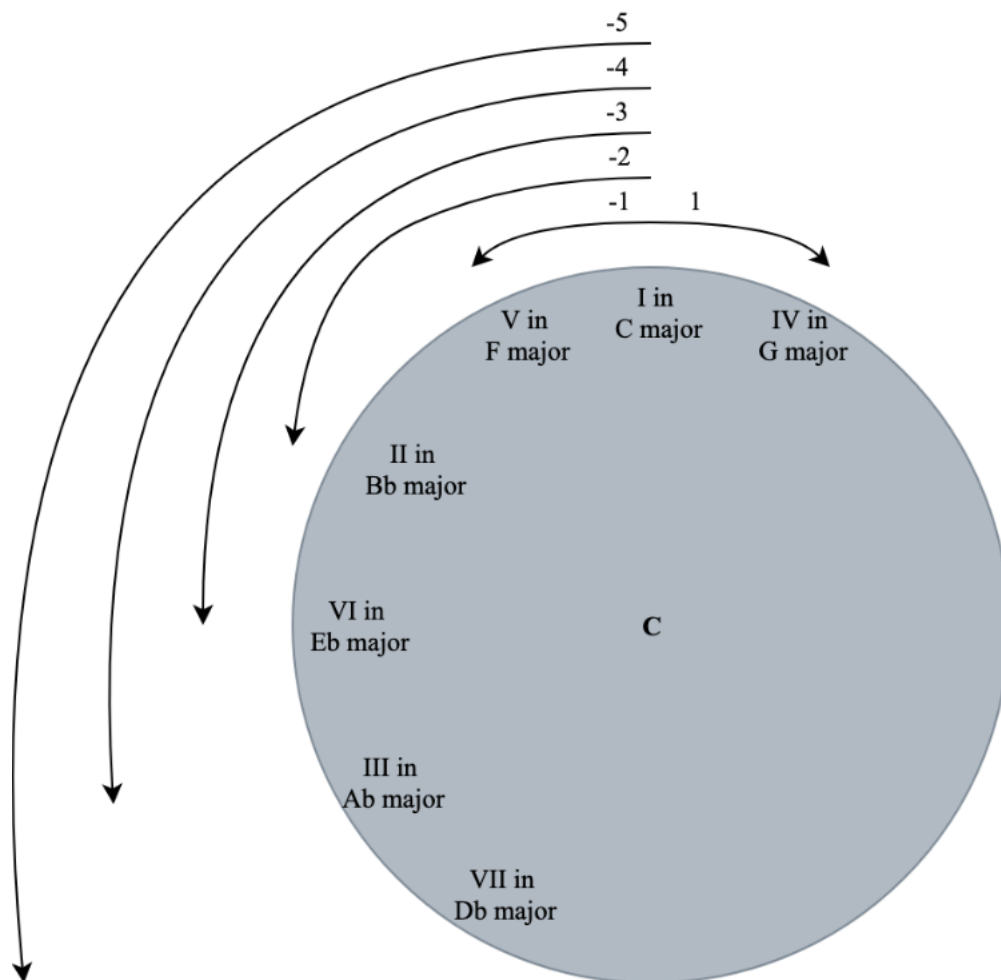


Figure 89. The numbers display how many positions the scale modulates.

The colours signal how many musical notes within a chord change by moving from the selected chord to another chord: blue (Figure 90, number 3, ‘IV’) means that **none of the four** musical notes within the chord change by switching from the selected chord to the blue highlighted chord. Green (Figure 90, number 1, ‘III min7’ and ‘VI min7’, and Figure 90, number 3, ‘V’) signals that **one of the four** musical notes change. Yellow (Figure 90, number 1, ‘IV maj7’ and ‘V 7’, and Figure 90, number 3, ‘II’, ‘III’, and ‘VI’) signals that **two of the four** musical notes change. Red (Figure 90, number 1, ‘II min7’ and ‘VII m7b5’, and Figure 68, number 3, ‘VII’) shows that **three of the four** musical notes within the chord change.

In this example, click on the fourth chord (IV) in the second row (Figure 90, number 1) to modulate to another scale. The button in the second row changes its colour from the blue colour to the white colour. The chord stays on the Cmaj7 chord; however, the musical scale changes from the C major scale to the G major scale. The displayed notes in the single note area display all notes that occur naturally within the G major scale. In this case, the note F changes to F#, and the note G is highlighted in green because G is the tonic of G major (Figure 90, number 2). Hold the 'Chord' button again to play the newly selected chord (Figure 90, number 3). Release the 'Chord' button to stop playing the selected chord.

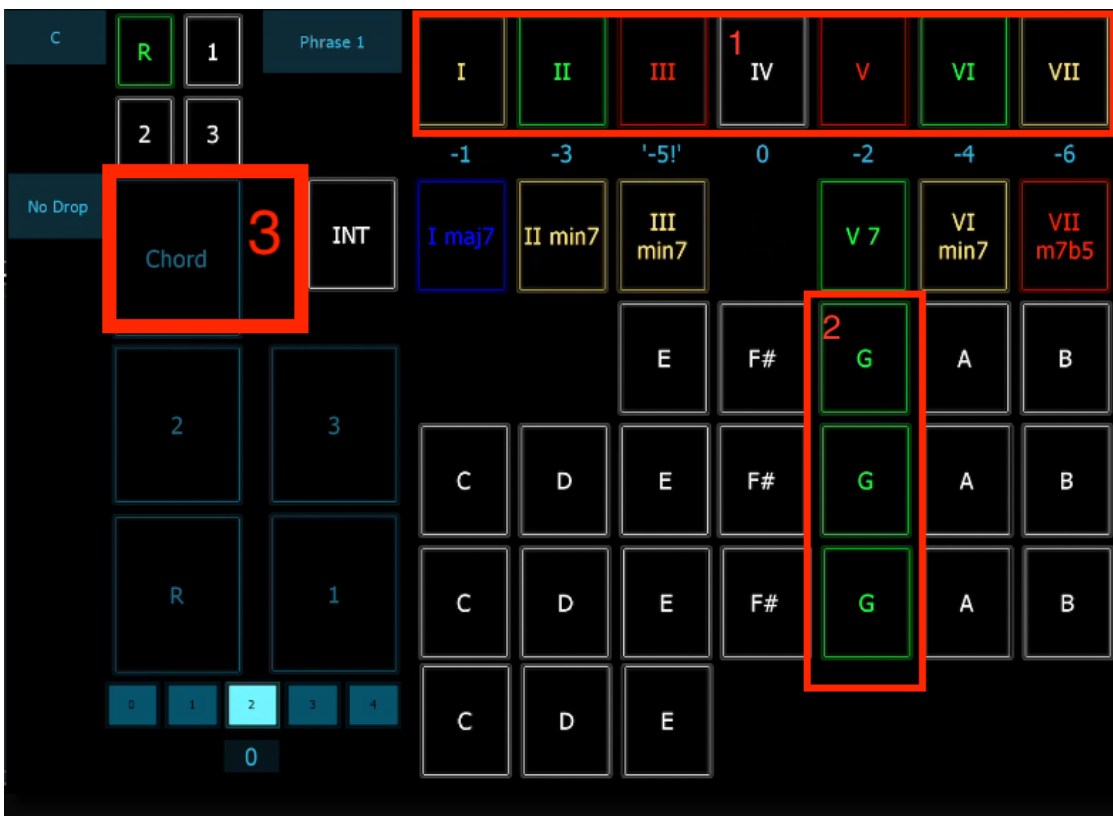


Figure 90. Change the chord progression and modulate to another scale.

Step 5: Select a chord within the selected scale and play the chord.

Next, select a chord within the G major scale. This step is comparable to Step 1; however, stay in the first chord row and do not select chord in the second chord row. In

this example, choose the first chord (I). The selected button lights up in white (Figure 91, number 1). Hold the ‘Chord’ button on the FPS-UIC (Figure 91, number 2) to play the selected chord. Release the ‘Chord’ button to stop playing the selected chord.

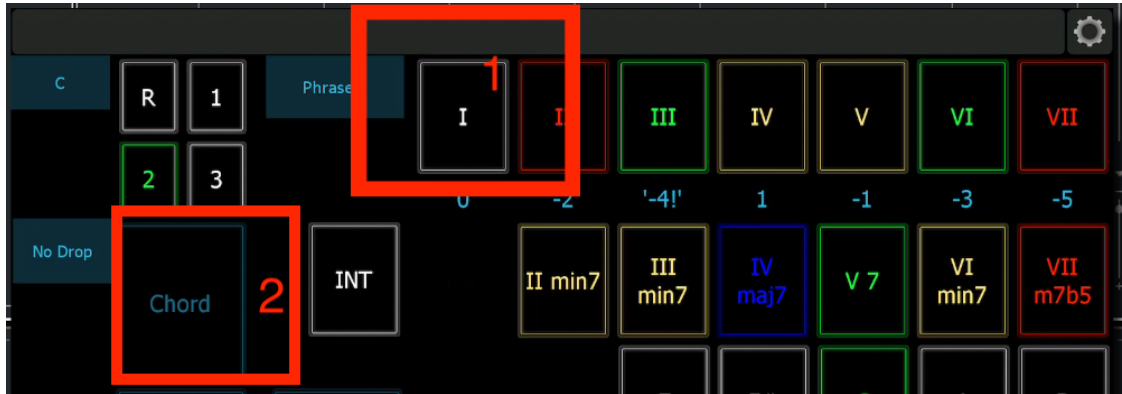


Figure 91. Select the first chord within the first chord row.

Step 6: Select a chord to modulate to another scale and play the chord.

In the sixth step, modulate again to another scale. This step is very similar to step 4. Click on the seventh chord of the second chord row (VII m7b5) to modulate from G major to Ab major (Figure 92, number 1). Click on the chord button to play the VIIm7b5 chord (Figure 92, number 2). Release the chord button to stop playing the chord.

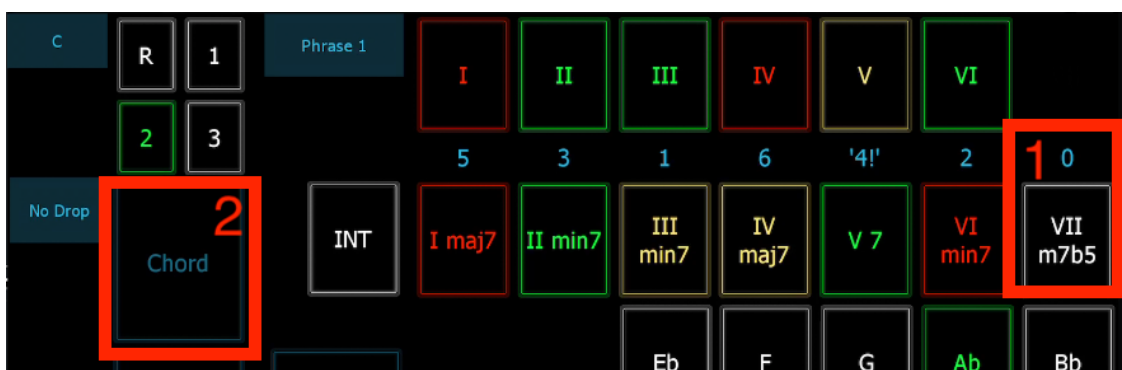


Figure 92. Change the chord progression and modulate to another scale.

Step 7: Select a chord within the selected scale and play the chord.

In this step, select the first chord within the same chord-row (Figure 93, number 1). This step is also very similar to the first and fifth step. Click on the chord button to play the selected chord and release the chord button to stop playing the chord.

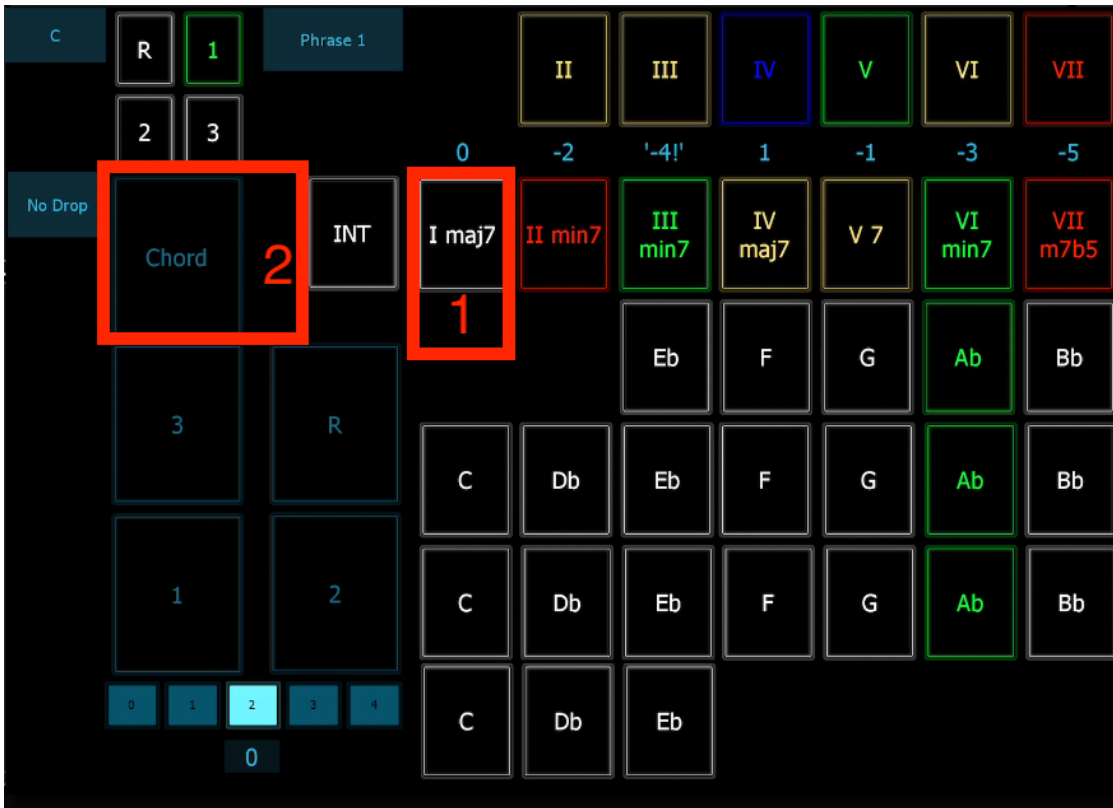


Figure 93. Change chord while recording the chord progression.

In Cubase, click on the 'Stop' button to the left of the 'Play' button to stop the recording. The different musical MIDI notes of the two homophonic chords are now recorded within the MIDI environment of Cubase. If required, adjust or remove MIDI notes within the MIDI editor¹⁶⁸ window of Cubase.

¹⁶⁸ The MIDI editor window of Cubase allows MIDI editing and provides an overview of the context of the recorded track.

Continue this composition, and repeat Step 1 till Step 7 in the same or a different order. Play different homophonic chord progressions on the FPS-UIC. Additionally, add different instruments such as woodwinds, brass, or percussion instruments to the composition.

8.10 Transfer of Rights Agreement

TRANSFER OF RIGHTS AGREEMENT

This **TRANSFER OF RIGHTS AGREEMENT** (hereinafter the "**Agreement**") is entered into

BETWEEN:

Dennis Braunsdorf

(hereinafter the "**Composer**")

AND:

Nico Maas

(hereinafter the "**Instrumentalist**")

WHEREAS INSTRUMENTALIST HAS SUPPLIED GUITAR SAMPLES AND PHRASES FOR USE IN SOFTWARE OWNED BY THE COMPOSER.

HEREINAFTER REFERRED TO COLLECTIVELY AS ("THE MATERIAL").

Instrumentalist hereby transfers to **Composer** all rights and interest in the **Material**.

Ownership of the **Material** is hereby transferred to **Composer**, effective as of the date of recording the **Material**.

Instrumentalist represents and warrants that he is the sole creator of the **Material**, that the **Material** is original and that to the best of his knowledge it is free of rights owned by another party.

Instrumentalist does hereby releases and discharges **Composer**, and his or her agents, representatives, and assignees from any and all claims and demands arising out of or in connection with the use of the material.

Instrumentalist retains the right to advertise the fact that he was the creator of the **Material** and to cite **Composer** in a list of customers.

Composer agrees to credit **Instrumentalist** on its website for the creation of the **Material** and, for this purpose, will use the credit information provided by **Instrumentalist**.

Entire Agreement. This Agreement contains the entire agreement between the parties relating to the subject matter hereof and supersedes any and all prior

agreements or understandings, written or oral, between the parties related to the subject matter hereof. No modification of this Agreement shall be valid unless made in writing and signed by both of the parties hereto.

Governing Law. This Agreement shall be governed by and construed in accordance with the laws of the Netherlands.

Instrumentalist

Name: Nico Maas

Date:

09.03.2020



Composer

Name: Dennis Braunsdorf

Date: 02.03.2020



8.11 About the Author

Dennis Braunsdorf is a music composer and music technologist. He is the founder of 'Braunsdorf Music', member of 'The Solos', and founder of 'Prolody B.V.'. 'Braunsdorf Music' and 'The Solos' specialise in musical sound design-based compositions for media, in particular for Hollywood movie trailers. In recent years, he has established a list of credits, including compositions for Hollywood movie trailers, films, and commercials. Prolody B.V. has been specialised in the development of expressive speech design. Dennis' academic path started in 2009 at the University of the Arts in Utrecht (HKU). In 2013, he received a Bachelor of Music with honours. In 2013, he also achieved a Master of Arts at the Open University of England, and in 2015 he achieved a Master of Music degree at the HKU. Since 2018, Dennis has been a board member of the professional association for media composers (BCMM) in the Netherlands and since 2020, Dennis has been a board member of Buma Cultuur.