Huw Davies

2788579 | WORC3219

DPhil in Music Composition

Critical Writing

Submission

Towards a more versatile dynamic-music for video games: Approaches to compositional considerations and techniques for continuous music.

Contents

DVD Contents

1. **How to Video (All-in-One)**
    i. How to: Transfer the DVD contents to your hard drive
    ii. How to: Open and use the standalone Demonstrations_Application application
    iii. How to: Open and use the standalone Interactive_Portfolio application
2. **InteractivePortfolio_Application**
    i. (open in MaxMSP Runtime)
3. **Demonstrations_Application application**
    i. (open in MaxMSP Runtime)
4. **Video Programme Notes**
    i. Video Program note for *Traversing the Centuries*
    ii. Video Program note for *Abiogenesis*
    iii. Video Program note and demonstration for *Generative Triptych of Percussive Music (MUSIC SYSTEMS)*
    iv. Video Programme note and demonstration for *Starfields (FLIGHT SYSTEMS)*
    v. Video Programme note and demonstration for *Deus Est Machina (NARRATIVE SYSTEMS)*
5. **Final Fantasy Battle System Video Demonstration**
6. **Appendix C - Unlockable versions ('code') of MaxMSP logic patches**
    i. Demonstration_Application application media, code and data files. (Open the .maxproj file in Max 6)

CD Contents List Of Recordings

1. Abiogenesis_First_Form
2. Starfields_Full
3. Starfields_Section3_Excerpt_1
4. Starfields_Section3_Excerpt_2
5. Starfields_Section3_Excerpt_3
6. Starfields_Section3_Excerpt_4
7. Starfields_Section3_Excerpt_5
8. Starfields_Section8_Excerpt_1
9. Starfields_Section8_Excerpt_2
10. Starfields_Section9_Excerpt_1
11. Starfields_Section9_Excerpt_2
12. Starfields_Section9_Excerpt_3
13. Starfields_Section12_Excerpt_1
14. Starfields_Section12_Excerpt_2
15. Starfields_Section12_Excerpt_3
16. Generative Sonata for Synthesised Drum Kit Version 1
17. Generative Sonata for Synthesised Drum Kit Version 2
18. Generative Sonata for Electronic Oscillators Version 1
19. Generative Sonata for Electronic Oscillators Version 2
20. Generative Sonata for Corrupted Sentence Version 1
21. Generative Sonata for Corrupted Sentence Version 2
22. Deus_Est_Machina

Acknowledgements

First, I would like to thank Dan Hulme and Duncan Williams for their technical aid.

Thanks to all the friends who have supported me knowingly or unknowingly during this project: Martin Evans, Kate Kennedy, Jamie Frost, Tomi Johnson, Jess Goodman, Paul Fineran, Pauline Souleau, Liz Mowforth, Liselotte Snijders, Stew McCain, Philippa Dand, Lizzie Sandis, Ally Paddock, Hannah Ryley, Jason Preece, Jim Cust, Henry Cust, Mike Taylor, Seb Fernando, Dan Jeffries, Nick DiBerardino, Nigel McBride, Chris Williams, Mike Evans, Tom Barnes, Aaron Pugh, James Harding, Dave Morgan, Jenny McParland and Sam Fernando, Nick Caplan, Catriona Munro, Tom Smart, Pete Jefferies, Kim Crangle, Lucy Nolan, without whom I could not have completed this.

Thanks to my loving family and especially to my brilliant sister, Emma, who has helped me clarify many of the issues I had taken for granted in the paper.

Katie Reeves's rallying words, her unwavering time and consideration as well as her gracious sharing of her gifted writing techniques have helped me progress through the most difficult moments of this project.

Words cannot express my gratitude to my Mum. Her endless, endless help during this doctorate is simply a theme for the endless, endless support she has given me, and continues to give me, throughout my whole life.

Abstract

This study contributes to practical discussions on the composition of dynamic music for video games from the composer's perspective. Creating greater levels of immersion in players is used as a justification for the proposals of the thesis. It lays down foundational aesthetic elements in order to proceed with a logical methodology. The aim of this paper is to build upon, and further hybridise, two techniques used by composers and by video game designers to increase further the reactive agility and memorability of the music for the player. Each chapter of this paper explores a different technique for joining two (possibly disparate) types of gameplay, or gamestates, with appropriate continuous music. In each, I discuss a particular musical engine capable of implementing continuous music.

Chapter One will discuss a *branching-music* engine, which uses a precomposed musical mosaic (or musical pixels) to create a linear score with the potential to diverge at appropriate moments accompanying onscreen action. I use the case study of the *Final Fantasy* battle system to show how the implementation of a branching-music engine could assist in maintaining the continuity of gameplay experience that current disjointed scores, which appear in many games, create. To aid this argument I have implemented a branching-music engine, using the graphical object oriented programming environment MaxMSP, in the style of the battle music composed by Nobuo Uematsu, the composer of the early *Final Fantasy* series. The reader can find this in the accompanying demonstrations patch.

In Chapter Two I consider how a *generative-music* engine can also implement a continuous music and also address some of the limitations of the branching-music engine. Further I describe a technique for an effective generative music for video games that creates musical 'personalities' that can mimic a particular style of music

for a limited period of time. Crucially, this engine is able to transition between any two personalities to create musical coincidence with the game. GMGEn (**G**ame **M**usic **G**eneration **En**gine) is a program I have created in MaxMSP to act as an example of this concept. GMGEn is available in the Demonstrations_Application.

Chapter Three will discuss potential limitations of the branching music engine described in Chapter One and the generative music engine described in Chapter Two, and highlights how these issues can be solved by way of a third engine, which hybridises both. As this engine has an indeterminate musical state it is termed the *intermittent-music* engine. I go on to discuss the implementation of this engine in two different game scenarios and how emergent structures of this music will appear. The final outcome is to formulate a new compositional approach delivering dynamic music, which accompanies the onscreen action with greater agility than currently present in the field, increasing the memorability and therefore the immersive effect of the video-game music.

Preface Introduction to the Thesis

Due to the multimedia nature of the materials submitted for this thesis I have created two full versions of the entire submission. One version takes the form of a website (dphil.huwcatchpoledavies.com) where all the multimedia elements can be displayed inline with text. The internet has evolved in part to accommodate the need to share such varied forms of media and entertainment in a widely viewable and standardised format. My thesis contained music, video, critical writing, scores, programme notes and even interactive applications. To submit these materials in hard copy requires: a CD of recordings, a DVD of Videos and Interactive applications, a paper copy of the critical writing, paper copies of the scores, paper copies of the programme notes for the musical works. As can be seen from listing these items, this is an encumbering array of material. While critical writing, scores and programme notes are standard for a thesis for a DPhil in Music Composition, my interactive application and video-game music subject matter demanded the submission of DVD materials also to form an archivable version of the thesis. Submitting these materials as a website I was able to: publish text for the critical writing and the programme notes; publish digital copies of scores; publish audio files and embed a web-player in line with text; publish video files and embed a web-player in line with text; post links to the interactive application and provide video instructions in line with text. In short, submitting as a website provides a seamless and simplified submission procedure, and it was therefore my opinion that this method of submission made the thesis more presentable for the examiners. At confirmation of status I presented this idea to my examiners (Robert Saxton and Jason Stanyek) and it was met with positivity allowing me to proceed with it further. Also, the examiners at the final viva have shown praise for this method of presentation. It is for these reasons that I see the web version of this thesis as its definitive form. It is to be noted that this choice of providing an extra way of presenting the thesis came with the burden of acquiring extra web development skills in order to achieve this for ease of communication. This necessarily cost time in both the original production and subsequent updating of materials post correction, which are not required for the typical submission. My production of the original website was in an effort to bring my submission inline with modern media and ways in which information is now communicated. While it is understandable these requirements are not yet in print at the University of Oxford, I believe they will become increasingly necessary for interdisciplinary research moving forward and am grateful for the examiners positivity toward this form of submission. For clarity, with access to the website, no hard (physical) copy need be examined as the 'hard' copy exists to satisfy standard submission criteria and not as the optimum method for communication of a thesis including varied media. The web submission is therefore an additional copy of all the materials presented in the paper/DVD/CD (physical) version of the submission.

In light of the above, this introduction provides an inventory of materials and a step-by-step guide as how to navigate the thesis for both the soft (web) copy and the hard (physical) copy.

INVENTORY

HIGH LEVEL INVENTORY OF SUBMITTED MATERIALS

At a high level, included in this thesis is the following:

- Critical Writing
    - A paper of critical writing entitled: “Towards a more versatile dynamic-music for video games: Approaches to composition considerations and techniques for continuous music.”
    - Appendix A – A discussion on the connection between the varied material submitted and how the research interests of this thesis pertain to each work of the portfolio of musical works.
    - Appendix B – Visual differences between ‘battle’ and ‘overworld’ in the early *Final Fantasy* games (VII, VIII, and IX).
    - Appendix C – a high level annotated and unlocked version of the Demonstrations_Application with accompanying instructions on how to navigate and assimilate the information herein.
    - Demonstrations_Application application
    - Video examples and demonstrations
- Portfolio of Musical Works
    - Programme notes
    - Scores for all works (including complete script for *Deus Est Machina (NARRATIVE SYSTEMS)*)
    - Video programme notes and interface demonstrations
    - 2 works of acoustic music
        - *Traversing the Centuries* – Song for Soprano and Piano
        - *Abiogenesis* – for Orchestra
    - Interactive Portfolio of digital audio experiences, help documentation and audiovisual demonstration.

- Meta work taking place in a fictional starship where the user interacts with PAMiLa, the onboard ship AI system, and selects each work from this overarching interface
    - *Starfields (FLIGHT SYSTEMS)* – digital work for one first-time operator using the Starfields digital instrument
    - *Generative Triptych of Percussive Music (MUSIC SYSTEMS)* – digital work using the generateBar digital instrument with three versions: (Synthesised Drum Kit, Electronic Oscillators, Corrupted Sentence)
    - *Deus Est Machina (NARRATIVE SYSTEMS)* – digital work for one first-time reader-listener

INVENTORY OF HARD COPY OF SUBMITTED MATERIALS

- **Paper Document 1**
    - Covering letter detailing changes made as a result of major corrections.
- **Paper Document 2**
    - This introduction detailing the inventory of submitted materials, a step-by-step guide on navigating the materials, and showing how evidence for the assertions being made are backed up by what has been submitted.
    - Critical writing entitled: "Towards a more versatile dynamic-music for video games: Approaches to composition considerations and techniques for continuous music."

    - Appendix A – A discussion on the connection between the varied material submitted and how the research interests of this thesis pertain to each work of the portfolio of musical works.
    - Appendix B – Visual differences between 'battle' and 'overworld' in the early *Final Fantasy* games (VII, VIII, and IX).
    - Appendix C – instructions accompanying the inclusion of a high level annotated and unlocked version of the Demonstrations_Application.
- **Paper Document 3**
    - Programme note and score for *Traversing the Centuries*
- **Paper Document 4**
    - Programme note and score for *Abiogenesis*
- **Paper Document 5**
    - Programme note and score for *Generative Triptych of Percussive Music (MUSIC SYSTEMS)*
- **Paper Document 6**
    - Programme note and A3 graphic score for *Interactive Portfolio and Starfields (FLIGHT SYSTEMS)*
- **Paper Document 7**
    - Programme note, score and script for *Deus Est Machina (NARRATIVE SYSTEMS)*
- **Audio CD**
    - Recordings of non-dynamic works
        - Abiogenesis
    - Multiple recordings of *Generative Triptych of Percussive Music (MUSIC SYSTEMS)*

    - Full recording with multiple excerpts of *Starfields (FLIGHT SYSTEMS)*
    - Full recording of *Deus Est Machina (NARRATIVE SYSTEMS)*
- **Data DVD**
    - Interactive_Portfolio application
        - Interactive Portfolio of digital audio experiences, help documentation and audiovisual demonstration.
            - Meta work taking place in a fictional starship where the user interacts with PAMiLa the onboard ship AI system and selects each work from this overarching interface
                - *Starfields (FLIGHT SYSTEMS)* – digital work for one first-time operator using the Starfields digital instrument
                - *Generative Triptych of Percussive Music (MUSIC SYSTEMS)* – digital work using the generateBar digital instrument with three versions: (Synthesised Drum Kit, Electronic Oscillators, Corrupted Sentence)
                - *Deus Est Machina (NARRATIVE SYSTEMS)* – digital work for one first-time reader-listener
    - Demonstrations_Application application
        - Interactive demonstrations pertaining to the critical writing
            - Tab 1 – Example of logic governing Final Fantasy Battle system's music
            - Tab 2 – Example of a Transitional Branching Engine

        - Tab 3 – Example of Generative Melody created with the mechanics discussed in the critical writing.
        - Tab 4 – Example of Generative Harmony created with the mechanics discussed in the critical writing.
        - Tab 5 – Example of GMGEn (Game Music Generation Engine), a state-based triggered transitioning generative music engine.
    - Video Program notes
        - Video Program note for *Traversing the Centuries*
        - Video Program note for *Abiogenesis*
        - Video Program note and demonstration for *Generative Triptych of Percussive Music (MUSIC SYSTEMS)*
        - Video Programme note and demonstration for *Starfields (FLIGHT SYSTEMS)*
        - Video Programme note and demonstration for *Deus Est Machina (NARRATIVE SYSTEMS)*
    - Demonstration and Instructional Videos
        - Final Fantasy Battle System Video Demonstration
        - How to: Transfer the DVD contents to your hard drive
        - How to: Open and use the standalone Demonstrations_Application application
        - How to: Open and use the standalone Interactive_Portfolio application
    - Appendix C - Unlockable versions ('code') of MaxMSP logic patches

        - Demonstration_Application application media, code and data files.

INVENTORY OF SOFT COPY OF SUBMITTED MATERIALS

- **Website – dphil.huwcatchpoledavies.com**
    - Home Tab
        - Video Introduction to the Thesis
        - Acknowledgements
        - Abstract
        - Introduction to the Thesis
    - Demos Tab
        - Demonstration and Instructional Videos
            - How to: Transfer the DVD contents to your hard drive
            - How to: Open and use the standalone Demonstrations_Application application
            - How to: Open and use the standalone Interactive_Portfolio application
    - Introduction Tab
        - Introduction to the Critical Writing
    - Chapter 1 Tab
        - Chapter 1 of the Critical Writing – High resolution musical branching applied to the early Final Fantasy battle-sequence
    - Chapter 2 Tab

        - Chapter 2 of the Critical Writing – One Compositional Technique for Generative Music for Video Games
    - Chapter 3 Tab
        - Chapter 3 of the Critical Writing – Memorable Musical Microcosm: Intermittent Music
    - Conclusion Tab
        - Conclusion to the Critical Writing
    - Bibliography Tab
        - Bibliography to the Critical Writing
    - Appendix Tab
        - Appendix A – A discussion on the connection between the varied material submitted and how the research interests of this thesis pertain to each work of the portfolio of musical works.
        - Appendix C – a high level annotated and unlocked version of the Demonstrations_Application with accompanying instructions on how to navigate and assimilate the information herein.
    - Portfolio Tab
        - Acoustic works
            - *Traversing the Centuries*
                - Text Programme Note
                - Video Programme Note
                - Notated Score
            - *Abiogenesis*
                - Text Programme Note

        - Video Programme Note
        - Notated Score
        - Audio Recording
- Interactive and Generative Works
    - Meta work taking place in a fictional starship where the user interacts with PAMiLa the onboard ship AI system and selects each work from this overarching interface
        - *Starfields (FLIGHT SYSTEMS)*
            - Text Programme Note
            - Video Programme Note and Demonstration
            - Graphical Score
            - Audio Recordings and Excerpts
        - *Generative Triptych of Percussive Music (MUSIC SYSTEMS)*
            - Text Programme Note
            - Video Programme Note, explanation of code and Demonstration
            - Source Code
            - Audio Recordings of Two compositions using each instrument variety presented (6 total contrasting compositions)
        - *Deus Est Machina (NARRATIVE SYSTEMS)*
            - Text Programme Note

- Video Programme Note, explanation of code and Demonstration
- Score diagram and full narrative script

The rest of this introduction will guide the reader through the course of the thesis. I will explain how to navigate all of the submitted materials and will show how evidence for the assertions being made are backed up by what has been submitted.

COVERING LETTER

Before exploration of the Thesis proper occurs I would direct the examiners to the covering letter detailing the changes made as part of the correction procedure. In the covering letter each requested correction's response is discussed and made explicit.

THESIS

The reader should then proceed to explore the thesis proper beginning with the Critical Writing. Reading from cover to cover takes the reader through all technical research points made and references both the video explanations and Demonstration_Application at the appropriate locations. Technical advice, should it be required, on how to open the Demonstrations_Application can be found in the demonstrations and information (in short: double click on the application icon).

The introduction discusses the state of the video game industry, the state of the academic study of video games and the state of the academic study of video-game music.

Chapter 1 discusses the early Final Fantasy battle system and its music. Referenced in this chapter is the video demonstration of the typical Final Fantasy battle-overworld transition demonstrated with gameplay from *Final Fantasy VII*. I also demonstrate the branching music engine discussed in this chapter in the Demonstrations_Application Tab 1 and 2. The former shows an example of the musical engine governing the music in the early Final Fantasy battle system where the latter shows a method of increasing the immersive capacity of the music on the listener by use of a triggered branching music engine, which is described earlier in the chapter.

Chapter 2 discusses the mechanics and methods used in building a generative music engine capable of creating static time-spaces of a particular musical character (I term these *musical personalities*). This generative music engine is also able to transition between any two personalities upon any game trigger. Demonstrations pertaining to the arguments made can be found in the Demonstrations_Application under Tab 3 and 4. Tab 5 shows a complete working prototype of GMGEn, and allows the reader to listen to the static state of a musical personality and also allows the reader to trigger a transition to a new musical personality. These demonstrations are explicit examples of the assertions I make in the text of the critical writing.

The demonstrations are artistically explored further in the digital works of the portfolio of compositions. The *Generative Triptych of Percussive Music (MUSIC SYSTEMS)* found within the InteractivePortfolo_Application is one such example, and explores the notion of larger forms of structure with indeterminate microcosms. The *Generative Triptych of Percussive Music (MUSIC SYSTEMS)* therefore acts as a proof of concept regarding the assertions made about generative rhythm and structure in a work of indeterminate nature. Further the work *Deus Est Machina (NARRATIVE*

SYSTEMS), which exploits the GMGEn instrument, provides a proof of concept for the working engine, the concept of the generative musical personality and the transitioning path finding mechanics presented in the second chapter of the thesis. *Starfields (FLIGHT SYSTEMS)* combines the work of the first two chapters of the thesis in creating an interactive audio experience applying many of the techniques discussed. For example *Starfields (FLIGHT SYSTEMS)* uses harmonic-clouds, generative rhythm and melody, large-scale definable structures with microcosmic indeterminacy and musical (and artificial intelligence) personalities. *Starfields (FLIGHT SYSTEMS)* achieves this in a work that also has the textural element of interactivity as one of the methods used to create musical juxtapositions. All the works of the digital portfolio are thereby proof for, or explorations of, the concepts highlighted and discussed throughout the course of the critical writing.

Chapter 3 discussed the concept of an intermittent music engine which is a combination and cooperation of the branching and generative engines explored (and prototyped) during chapter one and two of the critical writing. In this chapter, I argue that the intermittent engine coupled with a composer understanding the design of this engine, might act as a prevalent combination of techniques used in the composition of new continuous dynamic musics for video games.

APPENDIX

Following on from the critical writing portion of the submission I would guide the reader to continue with Appendix A. The appendix explores the impetus for the works in the portfolio of music and further gives explanations and connections between my research interests, aesthetic and technical considerations, and my conceptual aims for each work submitted.

HOW TO VIDEO

Before opening of either application the contents of the DVD should be moved to your local hard drive. The tutorial for this is included in the section 'Important Note to the Examiners'. The 'How To' video also shows how to do this if the reader is unfamiliar with this process. Further, if the reader is unfamiliar with opening, interacting, or uncomfortable with naturally exploring either the InteractivePortfolio_Application or Demonstrations_Application applications, I suggest the reader view the 'How to' video, which explains how to open and access the applications. The video also gives a demonstration of the interface in both instances. Typically, double-clicking the application icon will open the applications.

PORTFOLIO OF MUSICAL WORKS

The reader is then guided towards the portfolio of musical works. Before beginning each work the reader is advised to absorb the information provided in the video program notes and scores. While most of the information in the video program notes can be read in hard (physical) copy as a preface to each of the scores, it is not possible to include all of the demonstrable material efficiently in this format. The video programme notes allow a more assimilable format for the demonstration of interface, user interaction and compositional mechanisms inherent in the content of the musical portfolio.

Starting with the digital portfolio I recommend starting the tutorial on the first load of the portfolio. This explains the full controls available and even includes new features brought in during the corrections process. This tutorial will explain to the user where controls are and what they do. It will do this from a perspective within the fictional universe of the narrative. As the portfolio is interactive, the reader is now allowed the freedom to explore as they please; however, I will include a suggested path.

Begin exploring the interactive portfolio by selecting the FLIGHT SYSTEMS tab from the main systems control at the top of the interface. This will bring up the subsystem interface for the work *Starfields*. The engines can then be engaged beginning the work. During the work the user will be able to interact with the music using the slider on the right hand side of the interface. This slider will adjust many different parts of the musical parameters at any one moment. These parameters include many different methods of control over the textural intensity of the work. The effect of the THROTTLE slider can be viewed in the bottom right of the interface at any time. Methods used in the composition of this work are explored in the critical writing. For example the early sections heavily use the idea of harmonic-clouds. The underlying rhythm uses methods discussed relating to generative rhythm and large-scale structure during chapter 2 of the critical writing. Methods of generating melodic content, discussed in the critical writing, are also used later in the work during the 9th and 10th sections outlined in the video and text programme notes. In the latest version of the work there are also points where the user will begin MiniGames. These MiniGames are new areas of musical control and user interaction. They take the form of simple games some of which must be completed before progression occurs. The MiniGames were included to add further interaction and musical control for the user as well as provide reinforcement to the fictional narrative, which in previous versions remained hidden from the view of the player. This narrative has now been made more explicit while still allowing some ambiguous areas of hypothetical scenario for an inclined user. Also added in this version of *Starfields* is the graphical score. This acts as the navigational display inside the fictional world of the portfolio. Once *Starfields* has been completed I suggest the next tab in the main system control: MUSIC SYSTEMS.

Selecting the MUSIC SYSTEMS tab will bring up the subsystem interface for the *Generative Triptych of Percussive Music*. Here you will find three different versions of the same digital instrument used in the composition of these works. Clicking on any GENERATE button will generate and play a new work for that version of the instrument. In the new version post corrections there is now the ability to COMPOSE YOUR OWN composition where the reader is given access to the composing parameters that were available to me during the composition of the structured works. In these COMPOSE YOUR OWN versions the user is given 36 bars in which to adjust parameters. I have preset a work for each version of the digital instruments that will be loaded on startup. In other words, selecting COMPOSE YOUR OWN and then clicking GENERATE will play these preset works and allow the reader to follow along with the score in an aid to familiarisation of the mechanics of composing for these instruments. Once familiar, the user can adjust the parameters and begin the generation of the composition by pressing the GENERATE button which will now use the new parameters set by the reader. Works for all versions of this instrument are designed to illustrate macrocosmic personalities across multiple playings while remaining microcosmically indeterminate. Once the reader has finished exploring the compositional possibilities behind the digital instrument of the *Generative Triptych of Percussive Music* I suggest moving onto the last work in the digital portfolio.

Selecting the NARRATIVE SYSTEMS from the main systems menu at the top of the screen brings up the *Deus Est Machina* subsystem interface. Clicking on the

BEGIN ADVENTURE button at the bottom of the screen starts the text-based narrative and also GMGEn's 'STARS' musical personality. The text will scroll onto the screen and be readable by the user at their leisure. Once a section of text has completed there will be a CONTINUE button towards the bottom of the text display. This should be clicked when the user is ready to continue with the next section of text. GMGEn will accompany some of these continuation shifts with a transitional period of music. This transitional period will combine elements from the outgoing personality with that of the incoming personality. As the cascade of musical elements occurs the musical style of the first personality is blended with that of the second until the second aurally takes over completely, providing the new musical state for the work. During some points in the work the reader will be able to choose some directions for exploration by clicking on one tab towards the bottom of the text portion of the screen. The version of this work post corrections employs a voiceless narrator, which previously negatively competed with the musical content generated by GMGEn. Further, interactive elements have made this a simple text-based adventure as opposed to a narrative path generated solely by the artificial intelligence. In other words, the narrative path is now partly chosen by the user. On completion of the Interactive Portfolio the reader should move to the acoustic works submitted.

Abiogenesis and *Traversing the Centuries* have been included in this submission to show the beginnings of my aesthetic and methodological experimentations into the techniques I expanded upon during the thesis. In *Abiogenesis,* I used pitch-class sets to generate the harmonic content I would use in subsections of the music. This generative method enabled my choices to be bound by the pitch-class set thus affording the harmony of those sections a consistency due to the set's limitation of included pitches. This would become the harmonic-cloud technique I discuss in the critical writing and use in the digital works. *Traversing the Centuries* shows the exploration of the microscopic detail of the work, by directly excavating the music of the opening over three iterations during the work. This fascination with the micro- vs macro of a work would feature in the general way I would explore generative processes and the creation of musical personalities in the critical writing. Further, the composition of this work directly influenced my ideas on the ‘softness’ of a score and the fluxing capabilities of the composing process. This terminology is discussed further in the program notes on this work.

Important Note To Examiners

Due to the interactive nature of the material covered in this paper I have designed several interactive demonstration patches in MaxMSP to aid the reader. A large portion of my portfolio submission is also interactive and takes the form of a single MaxMSP patch. I have included a video explaining each patch to aid the reader in accessing the demonstration patches and portfolio. There are also video program notes explaining in detail all works, in particular those incorporated within the interactive portfolio. These videos also describe the use of the control system for the portfolio, which is an active part of the musical composition. Further, due to the varied media formats used to highlight certain video-game analyses, an entire web based version of this submission is available at http://dphil.huwcatchpoledavies.com should the reader prefer this format. I advise the use of this version of the thesis.

The included programs only work on Mac OSX and work consistently on a 2.3GHz Intel Core i7, 16Gb 1333 MHz DDR3, 7200rpm SSD, 512MB Intel GFX running OSX Yosemite. Testing has confirmed these patches also worked on the Oxford Music Faculty's Music Resource Center room desktop computers. OSX may select the wrong default opening program for the applications, in this case use step 6 if step 5 does not open the patch correctly. Should any unforeseen trouble occur when attempting to access the patches, please contact HuwCatchpoleDavies@gmail.com or 07960326775.

Full text instructions for opening either patch are included here:

1. From the DVD, right-click the root folder and select copy.
2. Choose a location on the hard-drive of your computer, right-click and select paste.
3. Wait for the copy to be complete.
4. Open the root DVD folder
5. Double-click the Application file within this location. Move to step 6 if step 5 resulted in a incorrect default program load. (you will see pink diagonal lines across some objects if this is the case)
6. Right-click the Application file and select 'open with' > 'Max 6.x runtime'.

Introduction

Musical engines, and the musical scores these engines employ, exist today that have some reactive agility towards changing game states, but there is scope for improvement which will allow musical scores to meet more accurately the ever-changing aesthetic needs of the modern video game. In this paper I will put forward the argument that greater levels of dynamic music increase the immersive capacity of the game on the player. I intend to build upon existing systems to increase this level of agility, by way of both creating working engines in MaxMSP and by composing music appropriate for use within them. The project will work on three distinct systems: first, a branching music system, second, a generative music system, and third, a hybrid intermittent system which incorporates elements of both other engines to further increase the reactive agility and memorability of the music for the player.

This paper addresses the need for interdisciplinary scholarly discourse on this topic. Practical video-game music study is inescapably interdisciplinary; therefore, adequate attention to all constituent elements must be made. Most broadly, this is a study within popular and audio-visual culture. Some of the key debates it touches upon relate to audio and visual perception, aesthetic intentions and results, and the psychologically immersive power of video games and their music. Primarily, this paper offers a practical method and solution to limitations found in current dynamic musical systems. This introduction will briefly highlight these elements and will illustrate foundational concerns on which I base this thesis. First is the clarification of some basic terminology that is used throughout the paper. Second is a short discussion on the emergence and evolution of video-game music study as well as the extent to which video games have economically and culturally effected entertainment media, and their possible consideration as a form of art. Towards the end of this introduction are two key discussions that pertain directly to the aims of this paper. The first concerns the fundamental power video games have to immerse a player in a virtual world and how music reinforces this immersion. The second concerns the implications that the interactive, non-linear and temporally indeterminate nature of this medium has for musical composition. I will conclude with a detailed outline of the material covered in each chapter of this paper. I provide a short survey of the history of video-game music at the beginning of the next chapter.

Before discussion of these various avenues, clarification on some terminology is needed. As defined by Karen Collins and Rod Munday respectively a *video-game* is, 'any game consumed on video screens', and *video-game music* is music 'written for, or adapted to video-games'.[1] Further elaboration yields that video-game music is 'pre-composed and recorded for playback to accompany specific locations or events in the game'.[2] It is important to be aware of the nature of the video game as a fundamentally interactive medium and that this nature is inherent across all of its assets, including music. Therefore video games include *dynamic music,* music that is active and can change according to in-game triggers or states. This can mean that the music will inhabit an indeterminate duration of time, tonality, style (among others), based on momentary in-game parameters. This also means that this music can be considered temporally indeterminate. I will refer to any player of games as either *gamer* or *player*.

Working in this relatively young field, academics concerned have felt a challenge to establish the reputation of video games within academia. The first page of James Newman's 2004 book *Videogames,* is titled 'Taking Games Seriously',

---

[1] K Collins, *Game Sound*, MIT Press, Massachusetts, 2008, pp. 2-4; and R Munday, 'Music in Video Games', in *Music Sound and Multimedia*, J. Sexton (ed), Edinburgh University Press, Edinburgh, 2007, p. 51.

[2] Z Whalen, 'Play Along – An Approach to Videogame Music', in *Game Studies*, vol. 4, no. 1, 2004, viewed 17th February 2014, http://www.gamestudies.org/0401/whalen/

clearly implying that video games were *not* taken seriously.[3] Newman states that for the majority of the five decades of their existence, video games have been viewed as 'adolescent indulgence' and considered the 'medium of children'.[4] In 1999 Matthew Belinkie had also reflected this feeling when he subtitled his history of video games 'Not Just Kids Stuff' – and possibly these views of video games as a juvenile pre-occupation have also impacted perceptions of academic video-game study.[5] Collins recalls that, when writing in the early 2000s, 'it seemed somehow necessary to preface each article with a series of facts and figures about the importance of the game industry in terms of economic value, demographics, and cultural impact.'[6] A decade on, video-game academics no longer need to devote a considerable amount of time defending the value of their study.[7] By 2008 Collins felt that it had 'become unnecessary to quote such statistics to legitimize' this field.[8]

Lars Konzak finds that discourse on video games fall into three main categories:[9] the economic and technological product of a game;[10] the socio-cultural effect video games have on the players or audience;[11] and the aesthetics of games.[12] As my study applies technological solutions to aesthetic concerns, in an attempt to influence the degree to which a player can become immersed in the game, it relates to

---

[3] J Newman, *Videogames,* Routledge, Oxon, 2004, p. 1.
[4] Newman, p. 5.
[5] M Belinkie, 'Video Game Music: Not Just Kids Stuff', in *Video Game Music*, 15 December 1999, viewed, viewed 11th February 2014, http://www.vgmusic.com/vgpaper.shtml
[6] Preface to Collins, *Game Sound*.
[7] See A Bodine. 'Androcentrism in prescriptive grammar: singular 'they', sex-indefinite 'he', and 'he or she'', in *Language in Society,* vol. 4, Cambridge University Press, Cambridge, 1975. pp. 129-145.
[8] Preface to K Collins, *Game Sound*.
[9] L Konzack, 'Rhetorics of Computer and Video Game Research', *The Players' Realm: Studies on the Culture of Video Games and Gaming*, JP Williams and JH Smith (eds), McFarland & Co Inc., North Carolina, 2007, p.121.
[10] See K Salen, *The Game Design Reader: A Rules of Play Anthology*, MIT Press, Massachusetts, 2006.
[11] See G Crawford, *Video Gamers*, Routledge, Oxon, 2012; and CJ Ferguson, 'Blazing Angels or Resident Evil? Can Violent Video Games Be a Force for Good?', in *Review of General Psychology*, vol. 14, no. 2, 2010, pp.68-81.
[12] See J Juul, 'Games Telling stories – A brief note on games and narratives', in *Game Studies*, vol. 1, no. 1, 2001, viewed 19th February 2014, http://www.gamestudies.org/0101/juul-gts/

all of Konzak's three main areas of game study. However, the majority of this paper is concerned with the practicalities of designing musical systems and therefore can only relate to socio-cultural effects and aesthetics in a secondary fashion.

Initially, game studies neglected the essential role of music and it wasn't until the late 1980s that the earliest papers relating to video-game music study emerged.[13] Tim Knight's 1987 paper *Mastering Sound and Music on the Atari ST* and Brian Schmidt's 1989 paper *Designing sound tracks for coin-op games* were some of the first, focusing on the practicalities of composition using these hardwares.[14] Due to the speed at which the technology has advanced, a huge amount of the literature focuses on the technology and the inherent practicalities of using it and composing within its constraints.[15] Not surprisingly, due to the youth of the study much of academic writing is broadly focused, such as Zach Whalen's *Play Along* and Collins's key work *Game Sound*.[16] Studies on psychological phenomena such as perception and immersion are reasonably present such as Michiel Kamp's *Musical Ecologies in Video Games* and Sean M. Zehnder's and Scott D. Lipscomb's *Immersion in the Virtual Environment.*[17] In recent years scholars such as Kiri Miller and Roger Moseley have noted the similarities between game play and instrument play, further suggesting the need for video-game music study to be based in broad cross-

[13] Whalen, 'Play Along', *Game Studies*.
[14] T Knight, *Mastering Sound and Music on the Atari ST*, Sybex Inc., Alameda, CA, USA, 1986; and B Schmidt, *Designing Sound Tracks for Coin-op Games,* MPublishing, University of Michigan Library, 1989.
[15] See J Boer, *Game Audio Programming*, Hingham, Massachusetts, 2003; and M.D. Wilde, *Audio Programming for Interactive Games*, Focal Press, Oxford and Burlington, Massachusetts, 2004; and G.W. Childs IV, *Creating Music and Sound for Games*, Thompson Course Technology, Boston, Massachusetts, 2007.
[16] Whalen, 'Play Along', *Game Studies*; and Collins, *Game Sound*.
[17] M Kamp, 'Musical Ecologies in Video Games', in *Philosophy & Technology*, Springer, 2013, viewed 19th February 2014, http://link.springer.com/article/10.1007%2Fs13347-013-0113-z; and S. M. Zehnder and S. D. Lipscomb, 'Immersion in the Virtual Environment: The Effect of a Musical Score on the Video Gaming Experience', in *Journal of Physiological Anthropology and Applied Human Sciences*, 2004, pp.337-43.

disciplinary knowledge.[18] Video-game music study is currently moving further away from the general and more toward the specific. Case studies are one example, few of which exist in print; however, more are becoming available in online journals such as Game Studies: The International Journal of Computer Game Research.[19] Articles on video-game music have appeared in many varied journals including Music and the Moving Image, Contemporary Music Review, and the Journal of Game Design and Development Education again reflecting the interdisciplinary nature of this field.[20] While no specific journal is dedicated to the study of video-game music many focused conferences are beginning to arise, including most notably the Ludomusicology Research Group, which has held conferences across the UK at Oxford, Liverpool and Chichester Universities. Further, the inaugural North American Conference on Video Game Music held at the Dana School of Music took place in 2014. These findings suggest that the study of video-game music is active, diverse and a growing field.

Since the 1980s, the video game industry has rapidly expanded in scale, and the popularity of gaming has hugely increased. In 2013 the games industry's reported value was $66 billion with their projection for 2017 running at $78 billion.[21] In a 2001

[18] See K Miller, *Playing Along, Youtube, and Virtual Performance,* Oxford University Press, Oxford, 2012; and R Moseley, 'Playing Games with Music (and Visa Versa): Ludomusicological Perspectives on Guitar Hero and Rock Band', in *Taking it to the Bridge*, N Cook and R Pettengill (eds), University of Michigan Press, Michigan, 2013, pp. 279-318.

[19] See D Bessell, 'What's That Funny Noise? An Examination of the Role of Music in Cool Boarders 2, Alien Trilogy and Medievil 2', in G King and T Krzywinska (eds) *Screenplay: Cinema/Videogames/Interfaces*, Wallflower, London and New York, 2002, pp. 136-144; and Z Whalen, 'Case Study: Film Music vs. Video-Game Music: The Case of Silent Hill', in *Music, Sound and Multimedia: From the Live to the Virtual*, J Sexton (ed.), Edinburgh University Press, Edinburgh, 2008, pp. 68-81.

[20] See K Collins. **'**Grand Theft Audio?: Popular Music and Interactive Games**'**, in *Music and the Moving Image***,** vol. 1, no. 1, University of Illinois Press, Illinois, 2008; and K Collins 'An Introduction to Procedural Audio in Video Games', in *Contemporary Music Review*, *Special Issue on Algorithmic Generative Audio*, vol. 28, no. 1, 2009, pp. 5-15; and K Collins, U. Önen and R. Stevens, 'Designing an International Curriculum Guideline: Problems and Solutions', in *Journal of Game Design and Development Education,* vol. 1, no. 1, 2011, viewed 19th February 2014, http://www.rit.edu/gccis/gameeducationjournal/

[21] M Nayak, 'Factbox – A look at the $66 billion video-games industry', in *Thomson Reuters*, 10th June 2013, viewed 19th February 2014, http://in.reuters.com/article/2013/06/10/gameshow-e-idINDEE9590DW20130610

survey 35% of people questioned stated that video games were their preferred entertainment, television being second, and films third.[22] A 2009 National Gamers Survey concluded that '83% of the US population played video games'.[23] In September 2013, *Grand Theft Auto V* (Rockstar) was released and grossed $1 billion in just three days setting a Guinness 2014 world record for the fastest entertainment property to ever achieve this.[24] Guinness World Records editor-in-chief states 'GTA totally deserves to be recognised as an icon of modern British culture'.[25] In his study of the video games industry, Tom Chatfield argues that 'there is fast becoming no 'us' or 'them' when it comes to games…whatever your opinion on video games, they will soon be universal.'[26] Certainly video games can no longer be viewed as the domain of children and adolescents – with the average age of gamer estimated at thirty in 2013.[27] With the further expansion of video games via mobile smartphone gaming – estimated to have 6.2m players a day in April 2013 - it would seem that video games have firmly cemented their position in popular culture.[28]

The popularity of video games has even elevated their status further with BAFTA awards for games and appearances in the top five of Classic FM's 2013 Hall Of Fame.[29] Orchestras devoted to playing music from video-game scores have arisen including *Video Games Live* and *The Legend of Zelda 25th Anniversary Symphony,* the

---

[22] Newman, p. 4.
[23] T Chatfield, *Fun Inc.*, Virgin Books, Great Britain, 2011, p. xiii.
[24] K Lynch, 'Confirmed: Grand Theft Auto Breaks 6 Sales World Records', in *Guinness World* Records, 8th October 2013, viewed 19th February 2014, http://www.guinnessworldrecords.com/news/2013/10/confirmed-grand-theft-auto-breaks-six-sales-world-records-51900/
[25] Lynch, *Guinness World Records*.
[26] Chatfield, p. xiii.
[27] 'Essential Facts about the Computer and Video Game Industry', in *The Entertainment Software Association,* 2013, viewed 19th February 2014, www.theesa.com/facts/pdfs/ESA_EF_2013.pdf
[28] S Dredge, 'Mobile Gaming's Rise is a Headache for Sony and Nintendo', in *The Guardian Apps Blog*, The Guardian, 25th September 2013, viewed 19th February 2014, http://www.theguardian.com/technology/appsblog/2013/sep/25/mobile-games-apple-google-sony-nintendo
[29] 'Classic FM Conversation: Video game scores – what do you think?', in *Classic FM*, 2013, viewed 19th February 2014, http://www.classicfm.com/hall-of-fame/2013/video-game-music-hall-fame-opinion/

latter selling out all tickets for the 2011 concert series.[30] Scoring music for games has also attracted the compositional talents of established film and television composers such as Danny Elfman for *Fable*, Michael Giacchino for the *Medal of Honour* series and Jeremy Soule for his scores contributing to the *Elder Scrolls* series: *Morrowind*, *Oblivion*, and *Skyrim*.[31] This shows that games are treated with the same kind of artistic respect that is afforded to other entertainment media.

Raphael Koster, in 1999, stated that 'the public already discusses and treats games as an art form, and uses the same standards of judgment for them as they do for films or novels or any other artistic medium.'[32] Roger Scruton finds that some philosophers 'have tried to link the philosophy of art to central questions concerning meaning, understanding, and value'.[33] He explains that all of these issues relate to the ontology of art and that we must consider 'what kind of thing is a work of art? Where, or when, is art? Do works of music, works of literature, painting, sculptures and buildings all occupy a like place in our ontology?'[34] Scruton underscores that the topic of aesthetics is 'highly controversial' and, as complete discussion on the philosophical implications of considering a work 'art' is beyond the scope of this paper, I do not presume to emphasise any specific conclusion other than to suggest that video games might be considered art.[35]

---

[30] 'Welcome to Video Games Live', in *Video Games Live,* 2014, viewed 19th February 2014, http://www.videogameslive.com/ ; also 'The Legend of Zelda Symphony of the Goddess', in *Zelda Symphony*, 2014, viewed 19th February 2014, http://zelda-symphony.com/

[31] *Fable*, Xbox Game, Microsoft Studios and Feral Interactive, 2004; and *Medal of Honor*, Playstation Game, Electronic Arts, 1999; and *The Elder Scrolls III: Morrowind*, Xbox/PC Game, Bethesda Softworks and Zenimax Media Inc., 2002; and *The Elder Scrolls IV: Oblivion*, Xbox 360/Playstation 3/PC Game, 2K Games, Bethesda Softworks and Zenimax Media Inc., 2006; and *The Elder Scrolls V: Skyrim*, Xbox 360/ Playstation 3/PC Game, Bethesda Softworks and Zenimax Media Inc., 2011.

[32] R Koster, quoted in *Videogames,* p. 2.

[33] R Scruton, *Modern Philosophy – An Introduction and Survey*, Sinclair-Stevenson, London, 1994, pp. 589-90.

[34] Scruton, *Modern Philosophy,* pp. 589-90.

[35] Scruton, *Modern Philosophy,* pp. 589-90.

My own view of video games as a form of art has brought me to the conclusion that certain limitations exist currently that could be improved for the greater expressivity of the medium. It has already been found that video game music draws on many filmic tropes and that its music is designed to 'create a compelling and entertaining emulation'.[36] Due to its close relation to that of film and television, there is 'widespread adoption of filmic perspectives and techniques' in the creation, practice and study of its music and therefore, it seems to be suggested that film music study has influenced video-game music study.[37] The functional similarity to which video-game music and film music can be discussed bare such a close resemblance that it seems sensible to start from these established academic grounds. Music accompanying film has the ability to establish setting or atmosphere, draw attention to particular elements or narrative developments, contribute to the creation of emotional responses in its audience as well as to help build a 'sense of continuity' between what might seem unconnected images.[38] It is suggested that these functions encourage our 'absorption into the film by distracting us from its technological basis'.[39] Whalen agrees that in video games, music, together with other elements, also 'immerses players in a fictional space'.[40] Whalen builds a strong argument for the 'compelling and immersive properties' in three landmark video games case studies. In *Super Mario Bros.* and *The Legend of Zelda: Ocarina of Time* he claims that the music helps 'enhance our belief in the consistency of a particular emulated world' by attaching musical signatures to different environmental settings and that *Silent Hill* creates a

---

[36] Whalen, 'Play Along', *Game Studies*.
[37] Whalen, 'Play Along', *Game Studies*.
[38] K Kalinak, *Settling the Score*, University of Wisconsin Press, Wisconsin, 1992, p. 2; and Copland quoted in 'Case Study: Film Music vs. Video-Game Music: The Case of Silent Hill', in Music, Sound and Multimedia: From the Live to the Virtual, J Sexton (ed.), Edinburgh University Press, Edinburgh, 2008, pp. 68-81.
[39] K Kalinak, *A Very Short Introduction to Film Music*, Oxford University Press, Oxford, 2010, p. 1.
[40] Whalen, 'Play Along', *Game Studies*.

musically driven experience by corresponding to the 'threatening, intrusive atmosphere of the city'.[41]

The examples discussed by Whalen show that music accompanying a video game has the same potential to affect as it does when accompanying film. However, differences between the audience's experience of a film and their experience of a video game give rise to many more considerations for its musical composition beyond those taken by the film composer. Whalen attributes the difference in experience to the fact that the 'game is *played* rather than viewed'.[42] What Whalen is bringing into question is the game's linearity. Due to the presence of a player the medium of video games is narratively *non-linear*. Whereas a film (and therefore film music) has a fixed duration, the duration of a video game is to a large extent dependent on the player's behavior. The film-music composer does not need to allow for differences in temporality in the narrative when composing, an essential point of difference from the video-game composer.

Munday describes *immersion* as 'either the heightened sense of a particular aspect of a person's immediate surroundings, or the sensation of being transported into a mediated alternate reality' and goes on to explain that 'the activity of threading a needle or hammering a nail can be immersive in the former … while reading a book or watching a film can be immersive' in the latter.[43] Munday terms the first type of immersion as *cognitive* and claims it is dependent on the brain's ability to block out other stimuli when focused on a single task.[44] A well-known example relating specifically to sound is known as the 'cocktail party effect' (see I Pollack and J

---

[41] Whalen, 'Play Along', *Game Studies*.
[42] Whalen, 'The Case of Silent Hill', p. 68.
[43] Munday, p. 56.
[44] Munday, p. 55.

Pickett).[45] Munday applies the term *mythic* to his second type of immersion and sees it as the chance to actually act … as someone, or something, else.[46] Munday proposes a caveat: 'for immersion to occur, the activity or stimulus has to be sufficiently engaging to block out the normal sense impressions of one's surroundings' and 'that the real world must be dimmed in order for the mediated one to be illuminated.'[47] Video games offer the chance for a player to experience both cognitive and mythic types of immersion.

Finding immersion in video games, Rebecca Farley describes players as concentrating 'wholly on the game – on the dice or the puck or the pawn.'[48] Due to the user's ability to control the object the player becomes responsible for the object and feels positive and negative situations, which the object experiences, personally. Bruner explains this phenomenon as a type of object-attachment causality: 'when objects move with respect to one another within highly limited constraints, we see causality'.[49] This is regardless of whether the object is humanoid, like Mario in *Super Mario Bros*; a pawn, in a chess game; or a puck, an inanimate object, in a hockey (sports) game. The player becomes immersed in the experiences of the object because those experiences are self-referential.

Viewed from a scientific perspective, Sherry and Lucas suggest any combination of six psychological states that players may wish to attain when using games:

> (a) *competition,* experience defeating others; (b) *challenge,* experience success following effort; (c) *diversion*, escape an experience of stress; (d) *fantasy,* experience

---

[45] I Pollack and J. M. Pickett, 'Cocktail Party Effect', in *J. Acoust. Soc. Am.* vol. 29, no. 11, 1957, p. 1262.
[46] Munday, p. 58.
[47] Munday, p. 56.
[48] R Farley, 'Game.' in *Media Culture: A Journal of Media and Culture,* vol. 3, no. 5, 2000, viewed 19th February 2014, http://journal.media-culture.org.au/0010/game.php
[49] J Bruner, *Actual Minds, Possible Worlds,* Harvard University Press, Massachusetts, 1986, p. 17.

> novel or unrealistic stimuli; (e) *social interaction*, have a social experience; and (f) *arousal,* experience activated positive emotions such as excitement.[50]

Sherry and Lucas find that if the goals desired by any individual gamer are attained it will result in more hours of gameplay by that player.[51] We can say that any one of these states produces a certain level of the immersive qualities, described by Munday, in a player. It is suggested that attaining more of these psychological states will result in a deeper level of immersion for the player with increased 'dimming' of the real world and 'illuminating' of the virtual. From this list, the video game composer is able to aid in the goal attainment of 'experiencing novel or unrealistic stimuli'.[52]

Przybylski and colleagues describe their research as splitting 'the general immersion state into three subcomponents: … *physical presence,* feeling as if one is actually in the world; *emotional presence,* feeling that game events have real emotional weight; and *narrative presence,* having a personal investment and engagement with the story'.[53] Sound assets can help to augment the strength of presence in all three subcomponents; music helps reinforce emotional and narrative presence and sound effects aid in creating the illusion of physical presence.

It can be argued that aural stimuli have a greater effect than visual stimuli on creating a realistic virtual space in video games. For example, take the case of a non-player character talking in game; we are stimulated by, the two-dimensional image of the character on screen and the three-dimensional sound produced by hardware speakers. Munday explains that 'a virtual doorbell sounds exactly like a real one, because waves of recorded sound are perceived in exactly the same way as waves of

---

[50] J Sherry and K Lucas quoted in, Andrew Przybylski, C. Rigby and Richard Ryan. 'A Motivational Model of Video Game Engagement', in *Review of General Psychology*, vol. 14, no. 2, 2010, p. 162.
[51] Przybylski, Rigby and Ryan, p. 163.
[52] Przybylski, Rigby and Ryan, p. 162.
[53] Przybylski, Rigby and Ryan, pp. 161-162.

real-world sound.'[54] Therefore, in this situation the sound is the only stimulus that mimics the way we would perceive the character were they to exist in the real world. Both recording techniques and 3D modeling techniques allow for the virtual creation of three-dimensional virtual sounds and three-dimensional virtual objects, however it is only the stereo sound production of a typical home speaker system that is capable of reproducing the sound in three dimensions whereas the visuals remain in two-dimensions on the screen.[55] Munday has demonstrated the strong influencing and immersing capability of sound upon the player in that it produces a virtual reality.

From Munday's example it would appear that adequate technology is one of the key factors in delivering a virtual reality. Frances Dyson agrees that the idea of technology acting 'as a form of *poiesis* ... has been made for all media' types and that these technologies allow the activation of 'hitherto inert objects'.[56] Dyson explains that headphone technology augments 'by delimiting or filtering sound in the environment', therefore to a degree forcing the listener to become cognitively immersed.[57] The type of audio production equipment (for example: headphones, stereo speakers, 5.1 surround, 7.1 surround) is therefore important to the way in which sonic immersion can be established in the player. Moreover, immersive speaker configurations provide a useful tool that the audio designer can exploit to aid in the creation of the virtual world. These findings indicate there is a necessity for reactive three-dimensional sound in games.

With the modern ability to program reactive *dynamic audio* engines capable of complementing various dramatic and scenic needs within game contexts affords yet

---

[54] Munday, p. 52.
[55]It is to be noted that in the near future 3DTVs allowing stereoscopic viewing will be more ubiquitous. When this future is reached there will no longer be the limit on visual hardware for the player. The limitation will then be on the software developers to exploit this hardware within their games.
[56] F Dyson, *Sounding New Media : Immersion and Embodiment in the Arts and Culture*, Berkeley : University of California Press, London, California, 2009, p. 130.
[57] Dyson, p. 130.

greater responsibility of sound in immersing the player. Todd Winkler gives a full description of the potential offered by computer-aided composition to increase flexibility and add musical interest:

> timing of decision making, mouse movements, or keyboard strokes, could be linked to control the tempo of the music; or, clicking and dragging an object downward might cause the music to go lower, or the timbre to get darker. Constrained random values could be used to create endless variations, so that the music for one scene would always be changing but have the stylistic consistency required of the dramatic structure. Parameter changes could also be controlled by the time spent within a single scene, so that the music would intensify as the viewing continued.[58]

These control possibilities are particularly relevant to video game music, where computer interaction and on-screen movement dictated by the user are essential components. Winkler goes on to state that 'clearly, music and sound responsive to user input will increase the feeling of interaction and participation' and, I suggest when applied to video-games, player immersion.[59] It seems that it is extremely important to keep considerations of control input at the forefront of thought when designing and composing music for video games.

Jesper Kaae describes four essential considerations specific to video-game music 'that would normally not need to be taken into consideration in traditional composing.'[60] First, 'technical considerations regarding computer power/technology'; second, 'composing tools and implementation'; third, 'functional considerations regarding aesthetics and user experience'; and fourth, 'the compositional requirements

---

[58] T Winkler, 'Strategies for Interaction: Computer Music, Performance, and Multimedia' in the proceedings of the 1995 Connecticut College Symposium on Arts and Technology, p. 3.
[59] Winkler, 'Strategies for Interaction: Computer Music, Performance, and Multimedia', p. 3.
[60] J Kaae, 'Theoretical approaches to composing dynamic music for video games', in *From Pac-Man to Pop Music,* Ashgate, 2008. p. 75.

of dynamic music, which often require a completely new way of thinking about music.'[61] By focusing on the dynamic aspect of this medium this thesis meets the requirement for a new interdisciplinary approach and contributes to new ways of thinking about music.

Different compositional techniques must be employed which allow for the unpredictable onscreen action should music accompany this action. The aim of video-game music is to create the illusion of musical linearity by creating an adaptive and interactive score. In the event that the music accompanies gameplay before, during and after a transition in gameplay, this music is called *continuous*. Though it is not necessary in all scenarios to create a *continuous score,* this paper focuses on the situations where it may be used. Collins suggests a reason for continuous music in these situations, in that it 'assists in the continuity of a game … since a disjointed score generally leads to a disjointed playing experience'.[62] This paper builds an approach to the composition and reaction of continuous music accompanying transitional periods within game play to assist in the immersive continuity of play. In doing so, it will address the four considerations that Kaae described.

The discussion on the compositional requirements of dynamic music—Kaae's fourth consideration—will always be aimed towards the aesthetic experience of the user—Kaae's third consideration—and will also require knowledge of what tensions and effects, both technological and musical, the implementation into a gameworld would make—Kaae's first and second considerations. Therefore these four points are heavily intertwined, and this must be taken into account for the compositional approach.

[61] Kaae, p. 75.
[62] Collins, *Game Sound*, p. 145.

Kaae's considerations lend further weight to the interdisciplinary nature of this field of composition. The composer must be aware of visual drama and its effects on a viewer, have a conceptual knowledge of computer programming, appreciate the technological limitations of the hardware, and understand the cultural and aesthetic context for their score all before writing a note of music.

This study contributes to practical discussions on the composition of dynamic music for video games from the composer's perspective. Creating greater levels of immersion in players is a justification for the proposals of this thesis. I lay down foundational aesthetic elements in order to proceed with a logical methodology. The aim of this paper is to build upon (Chapters One and Two), and further hybridise (Chapter 3), two techniques used by composers and by video game designers to increase further the reactive agility and memorability of the music for the player. Each chapter of this paper explores a different technique for joining two (possibly disparate) types of gameplay, or gamestates, with appropriate continuous music. In each, I discuss a particular musical engine capable of implementing continuous music.

Chapter One will discuss a *branching-music* engine, which uses a pre-composed musical mosaic (or musical pixels) to create a linear score with the potential to diverge at appropriate moments accompanying onscreen action. I use the case study of the *Final Fantasy* battle system to show how the implementation of a branching-music engine could assist in maintaining the continuity of gameplay experience that current disjointed scores, which appear in many games, create. To aid this argument I have implemented a branching-music engine, using the graphical object oriented programming environment MaxMSP, in the style of the battle music composed by Nobuo Uematsu, the composer of the early *Final Fantasy* series. The reader can find this in the accompanying demonstrations patch.

In Chapter Two I consider how a *generative-music* engine can also implement a continuous music and also address some of the limitations of the branching-music engine. Further I describe a technique for an effective generative music for video games that creates musical 'personalities' that can mimic a particular style of music for a limited period of time. Crucially, this engine is able to transition between any two personalities to create musical coincidence with the game. GMGEn (**G**ame **M**usic **G**eneration **En**gine) is a program I have created in MaxMSP to act as an example of this concept. GMGEn is available in the Demonstrations_Application.

Chapter Three will discuss potential limitations of the branching music engine described in Chapter One and the generative music engine described in Chapter Two, and highlights how these issues can be solved by way of a third engine, which hybridises both. As this engine has an indeterminate musical state it is termed the *intermittent-music* engine. I go on to discuss the implementation of this engine in two different game scenarios and what emergent structures in this music will appear. The final outcome is to formulate a new compositional approach delivering dynamic music, which accompanies the onscreen action with greater agility than currently present in the field, increasing the memorability and therefore the immersive effect of the video-game music.

Chapter 1 – High resolution musical branching applied to the early *Final Fantasy* battle-sequence.

This chapter will cover the branching musical engine and apply it to the case study of *Final Fantasy VII*.[1] First to be covered will be a brief outline of some of the key developments in the music of video games and some of the major composers will be highlighted. Again, some clarification of terminology is necessary before discussion continues on the stereotypical Japanese Roll Playing Game (JRPG/RPG) style. Following this will be detailed discussion on the standard structure of the music in the *Final Fantasy* battle-sequence that reaches the point of describing the aesthetic concern this chapter addresses. Next is the main discussion on my solution to this issue, which illustrates the concept of *musical resolution*, *arch* and *capillary branches*. Finally, I discuss some possible limitations of this approach and solutions to these limitations.

Video games enjoy a rich history when considering their relative infancy. A thorough discussion is not possible here and has been provided by Collins, among others.[2] Only the first decade of video games (including *Tennis for Two* and *Spacewar*) didn't incorporate sound; the early 1970s saw the first games where sound effects had their genesis. One of the most influential games of this time was *Pong*, which Chatfield states 'transformed the world's relationship with computer

---

[1] *Final Fantasy VII,* Playstation Game, Squaresoft, Japan, 1997.

[2] For a thorough discussion on this the reader is directed to chapter 2-4 of Collins, *Game Sound*; and for a briefer account the second chapter of Chatfield, *Fun Inc*. Collins gives an extremely through picture of the first few generations of game hardware pre and post the home console including their technology, market effects, and crucially the changes made in sound. Also see M Fritsch, 'History of Video Game Music', in P Moormann (ed.), *Music and Game: Perspectives on a Popular Alliance,* Springer, 2013, pp.11-41.

technology'.[3] *Pong* had a short 'boop' sound effect when the ball (a square) hit the paddle (a, slightly greyer, rectangle). Since these early games appeared, a general trend has been the increasing realism of sound effects. Music wasn't prevalent in games during the 1950s-1970s because of hardware capacity limits and the cumbersome methods required to program music. A technique of looping the music allowed it to be continuous and somewhat reactive to the player. *Space Invaders* is one of the most iconic examples of this looping technique and is one of the first games to ever have incorporated it.[4] Music was given a lower priority to sound effects as when the single chip needed to render both simultaneously, the music would be dropped while the sound effect remained. It wasn't until multiple sound chips were available in games such as *Alpine Ski* and *Jungle Hunt* that the hardware environment became hospitable enough for music to develop.[5]

From the mid-1980s, as technology developed, video game music became more complex, and with increasing memory space the tracks could become longer. Two very influential figures of the first era of video-game music are Koji Kondo, composer of *Super Mario Bros.* and *The Legend of Zelda*, and Nobuo Uematsu, composer of the *Final Fantasy* series beginning with *Final Fantasy*.[6] Koji Kondo's 'ground theme' for *Super Mario Bros.* is widely regarded as one of the most famous pieces in video-game music history, and his main theme from *The Legend of Zelda* has reached a similar status. Nobuo Uematsu's music for the *Final Fantasy* series of games is also considered some of the most popular and well known of all video game

[3] Chatfield, p. 19; and *Tennis for Two,* Donner Model 30 analog computer Game, William Higginbotham, Brookhaven, 1958; and *Spacewar!*, PDP-1 Game, Steve Russell, MIT, 1962; and *Pong,* Arcade Game, Atari Inc., 1972.
[4] *Space Invaders*, Arcade Game, Taito, Japan, 1978.
[5] Collins, *Game Sound*, p. 15; and *Alpine Ski*, Arcade Game, Taito, North America, 1982; and *Jungle Hunt*, Arcade Game, Taito, Japan, 1982.
[6] *Super Mario Bros.,* NES Game, Nintendo, Japan, 1985; and *The Legend of Zelda*, NES Game, Nintendo, Japan, 1986; and *Final Fantasy*, NES Game, Square, Japan, 1987.

music to date. It is not surprising that once looping music became common practice the first generation of game music composers would become the proverbial fathers of all video-game music composition.

The looping of long passages of music has become an entrenched technique since these first practises. When including the independent game development scene we still have a substantial portion of games producing music with long looping tracks at the time of writing. Some examples include *Starbound*, *Starcraft 2: Heart of the Swarm*, and *Bravely Default.*[7]

Since the 1980s, hardware has improved vastly and has removed many limitations that hindered continuous music in the era of Kondo and Uematsu. Scripting or software solutions such as iMUSE (Interactive Music Streaming Engine: Lucas Arts: 1991) and FMod (Firelight Technologies) give audio designers and composers the freedom to create realistic interactive sound effects and adaptive music while not having to be highly trained programmers. Many modern games include extremely advanced systems that accurately mimic the effects that virtual objects may make on the sounds produced in game.[8] Although while I agree that sound effects are becoming increasingly interactive, I am not convinced that the same revolution is evident on the music side of game audio.

Dynamic audio production is a field that is still in its relative infancy and therefore presents much opportunity for new development and innovation. *Dynamic audio* encompasses two subdisciplines; first being *interactive audio* – 'sounds and events that react to the player's direct input'; for example, a player pressing a button

[7] *Starbound*, PC Game, Chucklefish Games, Online, 2013; and *Starcraft 2: Heart of the Swarm*, PC Game, Blizzard Entertainment, Online, 2013; and *Bravely Default,* Nintendo 3DS Game, Square Enix, Japan and North America, 2013.
[8] See *Halo: Combat Evolved*, Xbox Game, Microsoft Game Studios, 2001; and *Dishonored*, PC/Playstation 3/Xbox 360 Game, Bethesda Softworks, 2012.

making the player's avatar swing its sword producing a 'sword swing' sound.[9] The second being *adaptive audio* – 'sound that reacts to the game states, responding to various in-game parameters'; for example, when day turns to night, 'day' music ends and 'night' music begins.[10] The majority of interactive audio deals with instantaneous sound effects whereas adaptive audio can be achieved by way of musical composition.

I will be looking at a scenario where looping of long tracks (roughly twenty seconds or longer) of music is vital in creating a constant musical score to the game. This scenario is found in many Japanese Role Playing Games (JRPGs) where the game play is split between a story and battle modes. I will investigate the battle-sequences from *Final Fantasy*, widely regarded as one of the most popular and well known of all JRPG series. In analysing the stereotypical musical construction of these battle-sequences I aim to suggest an improvement by way of a working example using musical templates and MaxMSP engine prototyping. The musical template will fit the musical style of the *Final Fantasy* series so as to present the improvement in as typical a staging as possible. The study of this scenario is not diminished by the fact that the games of the *Final Fantasy* series departed from this kind of battle system in 2006 with Final Fantasy XII; the same type of musical system can still be found in many titles such as *Cthulhu Saves the World*, *Evoland*, and most notably the *Pokémon* series, even right up to the most recent *Pokémon X/Y* released in October, 2013 as well as other JRPGs (Japanese Roll Playing Games) to date.[11]

To be noted is the stereotypical formula of a *Final Fantasy* game and its music. *Final Fantasy* is known as an RPG or Role Playing Game. In a Role Playing-

[9] T M Fay and S Selfon quoted in *Game Sound*, pp. 2-4.
[10] Collins, *Game Sound*, pp. 2-4.
[11] *Cthulhu Saves the World*, PC Game, Zeboyd Games, 2010; and *Evoland*, PC Game, Shiro Games, 2013; and *Pokemon X/Y,* Nintendo 3DS Game, Nintendo, 2013.

Game the player will control one character or a group of characters and will direct the characters' movements as they are taken through the narrative. World-exploration is a common feature of an open-world RPG like *Final Fantasy*, when the player directs the characters into points of the main scripted story (or main-quest) the narrative will progress and sometimes the whole explorable world will change. Features explorable include other quests (known as side-quests) and challenges that provide their own diverging narratives occurring as parallel storytelling. In *Final Fantasy* the main-quest is a linear narrative where the player will take on the roles of a team (or party) of characters. The player will explore towns, cities, the landscape, fly ships, sail boats, drive cars, interact with objects, and will experience conversations between the played characters (PCs) and non-played characters (NPCs), which will guide the player through the narrative of the game.

In parallel with the narrative portion of the game is a battle-system. This is a mode of gameplay whereby two opponents duel in head-to-head physical and magical combat. The player's team will square off against, and strategically attempt to destroy, all the enemies before being defeated. Each battle is a small puzzle that can be completed through a series of correct choices. Though there are many ways to win there are also many ways to be defeated. In the majority of the *Final Fantasy* series these battles take place in real-time.

Appropriate terminology for the narrative portion of *Final Fantasy* is difficult to achieve. By definition it is non-battle; however, this does not adequately reflect the richness of gameplay experienced outside the battle. Colloquially, the term *overworld* is used within the gaming community to separate between the place where narrative and exploration take place and other types of gameplay. Although this term has been appropriated from games such as the *Legend of Zelda* series where the player's avatar will actually walk downstairs to a region beneath the world, thus making the contrast between underworld and overworld a literal one, the term is still applied in most RPG games to be an area that interconnects all of its levels, puzzles or locations. During this paper the term *overworld* will refer to all non-battle scenes or music relating to that scene unless otherwise stated.

The total gameplay within the early *Final Fantasy* games will be made up of a cycle between battle and overworld. For the reader unfamiliar with this overworld-battle-overworld cycle I have prepared a video demonstration in the appendix of this paper (See *Final Fantasy VII* demonstration video). This video shows the visual and audio transition between the overworld and the battle in *Final Fantasy VII* as well as an entire battle sequence.[12] When visually juxtaposed, the difference between overworld and battle is striking. Table 1 (in Appendix B) shows six images comparing the visual representation between the overworld and the battle-system across multiple games in the series, specifically *Final Fantasy VII, Final Fantasy VIII* and *Final Fantasy IX*.[13] The reader can also observe this formular in the video example (See *Final Fantasy VII* demonstration video). In complementing these two modes of gameplay, there exist two aspects of sound, which similarly use looping music with differing structures. Though the main portion of this chapter will focus on the battle-system's music, the proposed improvements are mappable onto any situation where a musical transition takes place.

The constraints of this paper only allow brief discussion of the music traditionally existing in the overworld portion of the game. Music is scored entirely (with one exception explained below) with long loops connected to areas or situations; for example: music for a named area (i.e. Cosmo Canyon/Wutai), music

[12] See also Table 1 Appendix B.
[13] Final Fantasy VIII, Playstation Game, Squaresoft, Japan, 1998; and Final Fantasy IX, Playstation Game, Squaresoft, Japan, 2000.

for an event (i.e. a chase/escape). Leitmotif is used to attach musical motifs to particular characters with the development of these themes largely only taking place during set video sequences or *cut scenes* – effectively short films – at dramatic moments during the narrative. Cut scenes provide a visual, and interactive, exception to the majority of the whole game experience. Complementing this exception is the film-like scoring of the music. During cut scenes, while the visuals span a fixed time, so too does the music. As music for these scenes is non-dynamic it will not be discussed in this paper. Shown (see Table 1 in Appendix B) are the visual formulae used across the seventh, eighth and ninth games in the series. The reader can see that in each of the images on the left side there is a single character on a 3D overworld terrain with a map in the bottom right corner. On the right side of the table is the image from within a battle where the player's party fight a party of enemies. There are menus at the bottom of the screen as well as weapons in the hands of the characters. A similar formulaic approach is also used in the music, where a particular style and execution has been consistent across all the RPG games in the series until FFXII.[14]

In discussion of the music of the *Final Fantasy* battle mode, I wish to consider the structure and the aesthetic consequences of having dynamic battle sequences scored with non-dynamic musical sequences. As in the overworld, where there is music for specific types of events or specific types of place, so too is there music for the event of battle, which possesses a similar style throughout the series. Often in a quick tempo, incorporating an irregular grouping of quavers (or semiquavers) in 4/4 meter (for example, 3+3+2 is prevalent) and is scored for either acoustic/electric instruments or synthesised versions of these instruments in the case of the earlier

[14] Final Fantasy XII, Playstation 2 Game, Square Enix, Japan, 2006.

games. Structure is consistent and includes three sections: section A, also the introduction; section B, containing the bulk of the music, which loops to maintain continuous music for a battle of any duration; and a short ending (section C), a victory music that aesthetically transitions between the visual battlefield and the post battle analysis. The game then moves back into the overworld (see *Final Fantasy VII* demonstration video). Figure 1 shows a structure diagram of the visual cues and the accompanying music with the transition and sectional markings.

Figure 1 – Structure and transitions in a *Final Fantasy* Battle Sequence.

OOB = Out of Battle

	w		**X**		**Y**		z	
overworld	| |	**Section A**	| |:	**Section B**	:| |	**Section C**	| |	overworld
OOB		Introduction		Looping		Ending		OOB

Entering battle causes a visual transition from the overworld, in the case of the below examples (see Table 1 in Appendix B), a twisting of the screen in *Final Fantasy VII,* a left-to-right oversaturated smudging in *Final Fantasy VIII,* and a virtual shattering of the screen in *Final Fantasy IX,* all revealing the battle underneath. The Introduction music, coupled with the visual cues, will always start abruptly and overpower the overworld music by way of pounding rhythmic urgency and a series-regular baseline. The looping section comprises the main bulk of compositional material and is composed with its repetition as a primary compositional feature. It incorporates subsections so as to avoid the monotony of short (less than twenty seconds) looping passages. Once the player defeats all the enemies, the battle ends. Therein, characters perform a victory animation (for example the sheathing of a sword or sighing in relief) and the player hears the victory music. The music from the looping section will be truncated and taken over by the victory music.

Herein lies an aesthetic issue. The winning of the battle and therefore the playing of the victory music is important to the player's immersion and contributes to the full understanding that the battle has been successful; however, the musical transition is abrupt and allows potential to disrupt immersion. Michiel Kamp notes the same disruption in his description of the music appearing in *Super Mario Bros.,* which displays the same structure as the music in the *Final Fantasy* battle mode. He describes that once the introduction has played the music 'proceeds to loop through a series of melodies until the player finishes the level or Mario dies, at which point a coda is played and the music stops abruptly'.[15] His opinion of these transitions as 'abrupt' aligns with my own.

Two musical solutions are available to maintain appropriate musical accompaniment and player immersion: ending the section of music, or compositionally linking the looping music to the victory music. In the current state the aesthetic value of the existing music is lost between these sections due to the fact that it does not continuously accompany the visual. Although some attempts at appropriate musical transition has been made in other games (such as *The Legend of Zelda: Skyward Sword*), in my opinion, nothing has as yet been successful enough to accompany changes in game state that happen at the speed they do in modern game scenarios.[16]

As the structure shown in Figure 1 occurs in the majority of the *Final Fantasy* series I will be referring to each transition in the plural form. Currently the successes of the transitional sections (w, X, Y and z) have different levels of competence as

[15] M Kamp, 'Musical Ecologies in Video Games'.
[16] *The Legend of Zelda: Skyward Sword,* Nintendo Wii Game, Nintendo, Japan, 2011.

transitional pieces of linear composition. The musical transitions between the overworld to the battle introduction (w transitions) are abrupt and aggressive. Though it could be argued that w transitions are *too* abrupt and *too* aggressive, this musical approach captures the essence of the unforeseen battle (the *random battle*) and mirrors the abruptness of the visual transition. If w transitions need attention then the transitions from section A to section B (X transitions) need less; these two segments are scored together, and therefore designed, to work next to each other. On this basis the only transitions needing improvement are the transitions between section B and section C (Y transitions). As has been explained above, we can see that this is a disjointed transition with no conceptual excuse for the abruptness like those that may excuse (w) transitions.

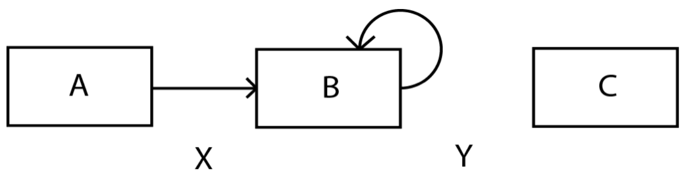
What I propose for Y transitions is the composition of potential-musics. The current system of battle music is linear where the music travels from beginning to end, regardless of need for divergence at specific player-controlled moments.[17] The potential-musics system would produce many branching lines from the main body of music to arrive at the next musical point required by the game. Similar to the way in which capillaries transport blood from arteries to many different locations in the body, and then return via the veins, so too would the music branch away from the main artery of the score and continue on to a new artery through musical branches acting as transitional capillaries. The main composition of the music would take place in the arteries (hereon *archbranches*) and different game states will have different compositional archbranches associated with them. When a musical transition is required, the appropriate capillary branch will be selected by the program and be played next. The important difference between the linear form and the branching form

[17] The structure of Figure 1 can be observed interactively in Tab 1 of the accompanying demonstration patches file.

is that from the point the music diverges from the archbranch to the point at which it enters another archbranch, the aesthetic consistency of the all musical components are neither disjunctive nor discordant. Many musical passages would be available to either link two sections together, change mood, end a section or take the place of a crossfade. This will create a fully scored musical system that reacts to the player's actions.

This particular model of dynamic music can be described using a Markov Chain. The reader is directed to Charles Ames's paper on 'The Markov Process as a Compositional Model' for a detailed explanation.[18] Figure 2 shows a Markov chain representing the current compositional linearity of the music in the *Final Fantasy* battle system. The reader can clearly see that the Y transition shows a break in musical composition.

Figure 2 - Markov Chain showing the linearity of composition during the *Final Fantasy* battle.



Two types of work must be undertaken to create a branching-music system. First, the musical composition of archbranches and capillary branches, this includes conceptualisation and realisation of many different fragments of score contributing to an aesthetic whole. Second, the programing of a system that, while playing a composition, understands its own location within that composition and has the ability

[18] C Ames, 'The Markov Process as a Compositional Model: A survey and Tutorial', in *Leonardo*, vol. 22, no. 2, 1989, pp. 175-187.

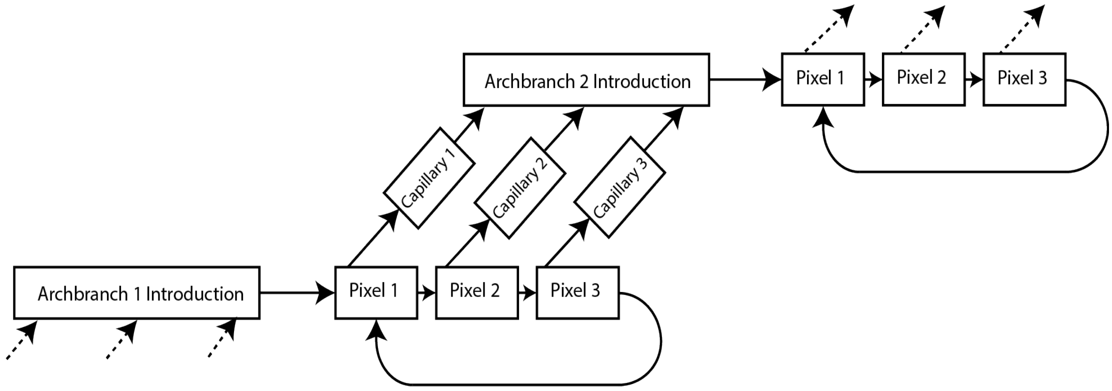
to make the appropriate choice of which branching capillary to use during archbranch transitions. Applying a visual concept to music allows the programmer to address this task. To see an image on a computer screen many individual pixels must work in conjunction to create the illusion of an unbroken image when displayed synchronously. Therefore, each linear score can be broken down into many smaller segments to become, in essence, musically pixelated. When these musical pixels are chained together, the illusion of a fully composed piece is perceived. This pixilation technique can be used by the programmer to create a tagging system.

This is not a new concept, both iMUSE and FMod, mentioned in the previous chapter, allowed the composer to mark locations in the score for the system to evaluate the game state and make appropriate musical choices based on this. Creating a unique tag for each pixel allows the program to make accurate diverging choices when a transition is required. As the number of pixels working together in a computer screen increases, the resolution and definition of an image also increases and will allow the viewer to perceive the objects onscreen with more detail and with greater fidelity. Similarly, as *musical-resolution* increases, then the points at which the music is able to diverge (node points) from its current archbranch get closer together in a relative temporal domain. A musical pixel can therefore be defined as the distance between two nodes or diverging points. Where my proposed system improves on that of iMUSE and suggests further use for composers using FMod is in the scale of the resolution.

The early *Final Fantasy* battle system has a low (zero) musical-resolution; there are no places at which it can diverge. Therefore, when the game triggers a divergence, the music simply skips to the new music without regard for aesthetic continuity of the musical line or a musical catharsis. The reader is directed toward the

Demonstrations_Application on the accompanying DVD-ROM, which illustrates this.[19] Figure 3 shows a Markov Chain diagram of this papers proposal for an improvement to this system by way of dynamic musical branching. It is feasible that capillaries could be composed to link more than two archbranches together.

Figure 3 – Markov Chain of proposed archbranch and capillary system.



Using either extremes of a musical-resolution spectrum have important implications. The primary trade-off is between maintaining the aesthetic goal of the project while avoiding large computational processing load. A low musical-resolution, where nodes are spaced a large distance apart, will not create a substantial hardware load and will produce dynamic music that will follow player's actions to some extent. However, a low musical-resolution may not be enough to produce dynamic music that reacts with enough agility to the changing game states. A high resolution, where nodes are close together, will create a music that has diverging potential at temporally closer points and thus will follow player actions more accurately. There exists a point of perfectly high musical-resolution, where nodes are so close together that they will definitively create a music that has diverging potential at any definable point and therefore, will follow player actions precisely. However,

[19] See Tab 1 in Demonstrations_Application.

such a high resolution would create an unnecessary tax on the hardware and would require a large amount of random access memory (RAM) to execute at the required speed. Further, the volume of score that would need to be composed to fill all capillary branches of a perfect-resolution engine would be staggering. A perfect resolution would require many times more score than will ever be played by the system just to accommodate the potential for all possible divergences. Economic and technological constraints of a perfect-resolution branching engine easily outweigh the viability of this ideal. There are two solutions to this problem: the first is to streamline the compositional process by, for example; making many capillary branches involved in an individual transition have similar (or the same) music (this has been done in Tab 2 inside the Demonstrations_Application). The second is simply to lower the musical resolution to an ideal compromise; a point where the moment of divergence is not noticed by the player and where the composer (or team of composers) are able to create the required quantity of score in the allotted economically-viable production time. In consideration of the latter, it is important to note that maximum player input speed is far slower than a computer's ability to react to the inputs using this branching technique.

Using MaxMSP I have built a prototype patch with an average pixel size of one bar.[20] The choice of pixel size was threefold: first, the program would have enough resolution in order to present a convincing branching music system that does not tax the hardware that was personally available. Second, it created a greater degree of musical control; creating a pixel size that links to an intrinsically musical feature (the bar line) meant I could manipulate the music occupying the pixels to have similarities in rhythm, harmony and melody, which the listener will perceive as a

[20] See 'Tab 2' in Demonstrations_Application

piece of music. Third, this eased the process of teaching the program what pixel of the whole score it was playing. Tab 2 also demonstrates that average hardware capabilities are ample for a resolution of one bar. The particular hardware system used to build and demonstrate Tab 2 would not be overburdened by increasing the musical-resolution. [21] Tab 2 demonstrates an important first stage for the implementation of this into a live video-game scenario.

A game scored using a full complement of branching-music would produce a more reactive and visually complementing sonic experience for the gamer and create the feeling that the score is being produced on-the-fly by the program. This is only possible when consideration is made over the conception and composition of all archbranch and capillary branches, an appropriate musical-resolution is chosen for the game state, and all this is incorporated into a highly advanced sample-selection program, such as Tab 2.

[21] 2.3GHz Intel Core i7 (quad-core, capable of hyperthreading a virtual core per core), 8Gb 1333 MHz DDR3, 7200rpm HDD, 512MB Intel GFX.

Chapter 2 – One Compositional Technique for Generative Music for Video Games.

This chapter explores the possibility for a generative music engine to make many of the choices needed for a reactive dynamic score for video game implementation. It also shows how this can be achieved while still having a great deal of control over the resulting output of the program and ultimately the power to create aesthetically appropriate scores for particular states within the game. I first briefly discuss the issue of authorship and what these considerations might mean for the concept of creativity. I discuss what might constitute a musical artificial intelligence and justify the use of this term within this project. The majority of this chapter discusses the founding of basic musical material and how to model these within a generative system. Discussion is grouped into subheadings with main issues including: rhythm, melodic contour and harmony. Finally I describe the Game Music Generation Engine (GMGEn) and the way in which it creates musical 'personalities' to implement a state-based musical composing engine within a video game.

As David Cope states, 'computers offer opportunities for style imitation' and 'have been successfully used in automated music (Hiller and Isaacson 1959), algorithmic composition (Fry 1984), and as stochastic probability generators (Xenakis 1972), to name a few'.[1] Generating computer-composed music shares many similarities with generating human-composed music, therefore many of the considerations of traditional composing can be applied and included in the program. The system will be designed to process random inputs to create musical composition.

[1] D Cope, 'An Expert System for Computer-assisted Composition', in *Computer Music Journal*, vol. 11, no. 4, 1987, p. 30.

Cope, the creator of a musical system, named Experiments in Musical Intelligence (*EMI*, a.k.a *Emmy*), which takes a corpus of music in a particular style and generates new music in that style, has discussed the issue of authorship. In an interview, when asked whether Cope listens to Emmy's music as he would the music of human composers he states that 'Emmy's music is 'human' as well' and that his computers simply follow his instructions.[2] He further directly states that he claims authorship over Experiments in Musical Intelligence's works.[3] Through my work with the Game Music Generation Engine (GMGEn) I do not claim direct authorship over the scores created, but do claim authorship over GMGEn itself and by indirect means, the scores it produces. A human's involvement in the sonic output of this computer-composed music is therefore indirect but remains present. The reader is directed towards Arne Eigenfeldt's article for a concise explanation of the history of interactive music.[4]

It may seem that we have reached a twilight of human-composed music if a program is able to appropriately accompany on-screen action without the input of a human. Andy Farnell writes that computer programs and algorithms such as 'expert systems, neural networks and Markov machines can codify the knowledge of a real composer but they cannot exhibit what makes a real composer, creativity.'[5] Farnell is illustrating that the computer is unable to break free of its digital make-up in this sense. Although expert systems, neural networks and Markov machines are artistically useful when within musical composition, they do not direct their output

[2] K Muscutt and D Cope, 'Composing with Algorithms: An Interview with David Cope', in *Computer Music Journal,* vol. 31, no.3, 2007, p. 18.
[3] D Cope, 'Facing the Music: Perspectives on Machine-Composed Music', in *Leonardo Music Journal,* vol. 9, 1999, pp. 81.
[4] A Eigenfeldt 'Real-time Composition or Computer Improvisation? A composer's search for intelligent tools in interactive computer music', in *Electroacoustic Studies Network,* 2007.
[5] A Farnell. 'An introduction to procedural audio and its application in computer games', in *Obewannabe,* 2007. viewed 30th October 2013, http://obiwannabe.co.uk/html/papers/proc-audio/proc-audio.pdf

towards human goals as human intelligence does. This is why, when applying small changes to their systems, it is no longer possible to guarantee that humanly expected musical congruence occurs from the output. Therefore, as Axel Berndt claims, 'the human artist… is still indispensable!'[6] Fundamentally Farnell and Berndt are correct; a computer will never do anything definitively creative in its composition of music, but will (currently) only ever follow its programming rigidly and accurately; however, I contend that absence of (musical) creativity can only ever be measured in the terms and limits of the musical observer. As the observer is human, then human limits of perceptibility must be taken into account.

For the interactive situation of a video game, if a generative music program can improvise music to a level which a human listener believes it an appropriate aesthetic accompaniment to the on-screen action it may be perceived as a musical artificial intelligence. In considering the composition of interactive music, Winkler states simply that 'computers are not intelligent. They derive their appearance of intelligence only from the knowledge and experience of the person who creates the software they run.'[7] I take the suggestion from Winkler that a person skilled in computer programming as well as in the art of musical composition is preferrable to create a program capable of convincing a human listener that the dynamic computer-generated music is linear music composed by a human. Further, W. Jay Dowling agrees that 'it should be possible to integrate appropriate artificial intelligence techniques to construct an expert system which improvises.'[8] Jesper Kaae states that for a listener, 'music will always be linear, since music exists in a length of time', and

[6] A Berndt & K Hartmann. 'Strategies for Narrative and Adaptive Game Scoring', in *Audio Mostly*, 2007, viewed 19th February 2014, http://wwwpub.zih.tu-dresden.de/~aberndt/publications/audioMostly07.pdf

[7] T Winkler, 'Defining relationships between computers and performers', in *Composing Interactive Music: Techniques and Ideas Using Max,* MIT Press, 1999, pp.5-6.

[8] J Dowling, 'Tonal structure and children's early learning of music', in *Generative Processes in Music*, J. A. Sloboda (ed.), Oxford University Press, Oxford, 1988, p. 152.

so the challenge becomes only to create a convincing music during this time.[9] These methods are useful when building a generative music engine. George Lewis's interactive work *Voyager* (1993) has already demonstrated the strength of 'music-generating programs and music-generating people' working towards the same goal.[10]

When used in the dynamic context of video games, generative music holds a true key towards combining musical audio with onscreen action and aiding immersion. Tab 2 in the Demonstrations_Application demonstrated that the modern computer's capabilities extend beyond what was necessary to make a working piece of reactive dynamic music. Winkler suggests that traditional models of music making are a useful starting point for interactive relationships however, 'since the virtue of the computer is that it can do things human performers cannot do, it is essential to break free from the limitations of traditional models and develop new forms that take advantage of the computer's capabilities.'[11] Given this excess of computer processing power in a branching music system using pre-composed music, it is a natural further extension to allow the computer control over greater musical detail than simply musical structure like that in Tab 2. We can use this to adjust elements of music so as to create an aesthetically appropriate generated music for use in the video game context. Below I will discuss some of the necessary considerations that must be taken by the composer-programmer within this medium.

Musical assumptions

An approach to musical composition can occur as either top down, where the composer may design large structures before working on detail, or bottom up, where

[9] Kaae, pp. 77-78.
[10] G. E. Lewis, 'Interacting with Latter-Day Musical Automata', in *Contemporary Music Review,* vol. 18, no. 3, 1999, pp. 99-112.
[11] Winkler, 'Defining relationships between computers and performers', p. 28.

small musical segments may be built into larger structures, or a synthesis of the two approaches is possible. Programming requires a similar approach to understanding goals on the macro scale and achieving them with methods on the micro scale. During the process of programming and composing, some new features can be found to be necessary, or beneficial, while working at the micro scale that were not foreseen when designing at the macro level. When discussing Lutoslawski's use of aleatoric methods, Charles Bodman Rae defines these as a form of constrained randomness, 'a framework of limitation, a restricted range of possibilities' where 'the full range of possible outcomes is foreseen.'[12] Aleatoric methods also lend themselves well to computer programming and generating music and are therefore used heavily in the construction of GMGEn and its output.

In defining a macro goal for this project, the aim is to create a generative musical engine, an expert system, which can produce continuous music in a human controlled style. It will have the ability to combine fundamental elements of music into single musical lines. Six of these lines will be combined in a number of ways to create a rich variety of time-independent musical spaces. When complementing spaces sequentially follow one another, a consistent musical style is created which has internal variety and external reactive potential. When non-complementing spaces sequentially follow, a style transition takes place and can be used in the video game to accompany game state changes of more varied degrees to enhance the player's immersion. The macro-scale goal of this paper is achieved by completing three stages of work on the micro-scale. These stages represent the reducing of a problem 'to a set of subproblems'.[13] Simplification creates opportunity for addressing issues individually as well as the musical considerations the composer must make. In the

[12] C Bodman Rae, *The Music Of Lutoslawski*, Omnibus Press, New York, 1999, pp. 75-76.
[13] Dowling, p. 151.

first stage of work, I will lay out basic attributes, or *typicalities,* of separated musical elements that will act as a sounding board from which to elaborate when constructing a musical artificial intelligence. Second, I will discuss how these attributes can be modeled in a MaxMSP prototyping environment. Third, I will build a working generative music engine, GMGEn, which will combine all the components discussed and feature a usable user interface. It will exist as a proof of concept and a standalone work of generative art within my final portfolio. Although some scientific terminology, explanation and methods are used, the success of the project will be judged in a qualitative fashion, as is appropriate for the project in artistic terms, and upon criteria intrinsically tied to my own artistic inclinations and experience in composition.

Generative Rhythm

Typifying rhythm is probably one of the most difficult aspects of this project. Grosvenor Cooper and Leonard B. Meyer completed a comprehensive study on rhythm, which typifies it from the simple to the complex.[14] Cooper and Meyer organise their definitions into *architechtonic* levels so as to define different sub and super groups of rhythmic construction.[15] Their analysis of rhythmic architechtonics complements a bottom-up approach to the programming of this rhythmic engine. In scale order, from the smallest to largest, this section will discuss the attributes of *pulse*, *duration*, *micro-rhythm*, *meter* and *macro-rhythm* in GMGEn.

Pulse is the lowest architechtonic level of rhythm in GMGEn and constitutes an unbroken regular unit of time that is non-distinct within a single section. It is a unit of time from which all other durational values are derived and is created by setting a

---

[14] G Cooper and L. B. Meyer, *The Rhythmic Structure of Music*, Phoenix Books, University of Chicago Press, 1963.
[15] Cooper and Meyer, pp. 1-11.

metronomic impulse (in milliseconds) and broadcasting it throughout the program at any location requiring synchronisation to, or information from, the pulse.

Duration is a unit of time applied directly to a sound and is distinct from pulse in that it can hold many different values within the same section. In GMGEn, duration for a single sound is obtained by taking the value of the pulse (in milliseconds) and dividing or multiplying it by any factor required, then returning this value to the sound as its duration parameter. The sound will then play for the number of milliseconds held in its duration parameter.

Micro-Rhythm is the combination of two or more durations played in sequence by one line of music. In essence this is the level of the rhythmic detail of the piece. This is achieved in GMGEn by combining new sounds with new durations. Durational values are taken from memory or generated on-the-fly depending on the type of rhythmic output the user has selected.

Meter is 'the measurement of the number of pulses between more or less regularly recurring accents. Therefore, in order for meter to exist, some of the pulses in a series must be accented—marked for consciousness—relative to others.'[16] In GMGEn, metric groupings are achieved part by chance and part by constraint, an aleatoric method. First, a total group duration, or bar, will be determined and will be found by multiplying the value (in milliseconds) of the pulse by the number of beats in the bar (chosen by the user). Once a total bar length is obtained the program will generate a random duration from available options. This duration will then be subtracted from the total duration and another random duration will be generated and also subtracted from the total duration. This happens recursively until the total bar duration is filled.

[16] Cooper and Meyer, p. 4.

Macro-Rhythm is a concept applying to the rhythmic structure across larger architectural boundaries or phrases. In GMGEn a phrase constitutes a number of grouped bars. Larger super-phrases are possible by grouping a number of phrases together and theoretical hyper-phrases might be groups of super-phrases and so on. In GMGEn, a single phrase is obtained by allowing the program to store many filled bars of durational values, which it has generated, and recalling these values in future bars. Allowing the recall sequence to be similar or the same as that previously generated produces a phrase. This can also be done in a more immediate way by making the program regenerate a whole phrase worth of durational values at a point during playback. Adding the ability for the program to change either meter or durational values during playback creates further variety of phrases. If particular patterns were applied to either regenerations or changes of meter (or both) this could structure the music into typical western classical forms such as sonata or rondo. The reader is directed to the works in the InteractivePortfolio_Application accompanying this paper, which demonstrate the concept of this generated music within structures found in the western classical traditional.[17] GMGEn uses all of these techniques to produce every level of rhythmic architechtonics.

Generative Melodic Contour

As with rhythm, typicalities of melodic contour must be considered before attempting to program a melodic generator. In GMGEn the goal of a typical melody is to move between pitches with a majority of stepwise movement and a minority of non-stepwise, or leaping, movement. Here a stepwise movement is defined as a movement from a pitch that is one half, or one whole, tone away from the previous

[17] See ‘MUSIC SYSTEMS’ tab in InteractivePortfolio_Application. Click any ‘Generate’ button to generate a new work.

pitch. A leap being defined as a movement from a pitch that is further away than one whole-tone from the previous pitch. We can use random number generation to determine an individual pitch to be played by the program. This is achieved in GMGEn by linking particular pitches to number output from a random number generator (hereon RNG). For instance, six possible numbers can be assigned to the following pitch values: 1 = C, 2 = D, 3 = E, 4 = F, 5 = G, 6 = A. While we can assign neighboring numbers a pitch interval that equals a step we cannot determine that the RNG will generate neighbouring numbers in sequence.[18] Therefore simple random number generation does not model these goals.

From this we can determine that a higher-level function is needed for a mostly-stepwise melody to be produced in GMGEn. This function controls the difference between two consecutive number outputs, in this case constraining them to closer degrees. Using the same range as above (1 to 6) and adding a second stage hierarchical RNG with the range of (-1 to 1) will achieve this. This will be called the *step* meter. If the RNG outputs a 3 value and we take all possible values of the step meter (-1, 0, or +1) the next output can only be either 2, a 3 or a 4 value (3 ± 1) respectively. A two stage hierarchical number generator with these parameters will produce a melody that invariably moves always by step.[19] This behaviour is known as Brownian motion; the behavior of a particle of air as it moves through space and interacts with other particles. The motion of a single particle is macro-linear with small micro-deviations.

Hybridising both of the above types of generation in the correct proportions will produce the desired melodic contour. There are two methods for producing the hybridisation to suit these goals. The first method is to allow the computer control

[18] See Tab 3 in Demonstrations_Application and select 'Full Random' from the drop down menu.
[19] See Tab 3 in Demonstrations_Application and select 'Full Step' from the drop down menu.

over leap-like movements. This function would vary the step range (the second stage RNG) and thus produces phases where leaping is possible (by using a step value greater than 1) and phases where only stepwise movement is possible (by using a step value of 1). Correct tuning of the variables (the 'on' time of each phase and the step magnitude) will create the intended type of melodic contour demanded by any current game state.[20] The second method would be to create presets for leaps. For example, a pre-defined leap value that is triggered at certain points (e.g. a leap of a fifth triggered every six seconds).[21]

The first, although more hierarchically complicated, is not difficult to achieve and gives more control to the computer in the design of the melody. The second not only restricts the program's choices of leap but also predefines the leap value (this could be randomised too) meaning that increased control is given to the composer-programmer as to how the melody will develop. With each of these examples the human creator is exerting different types of control on the pitches outputted (and therefore music generated) by the program.

Generative Harmony

Harmony and harmonic language are two related constructs in Western art music. The Oxford English Dictionary defines harmony as 'the combination of simultaneously sounded musical notes to produce a pleasing effect.'[22] Eliminating any issue with the field of musical aesthetics or perception, I wish to remove the term 'pleasing' for the course of this paper. I will define harmony to be: the simultaneous sounding of more than one musical note. Any requirements of the combinations of notes exist merely to create a selective harmonic consistency. This is better described

[20] See Tab 3 in Demonstrations_Application and select 'Computer Hybrid' from the drop down menu.
[21] See Tab 3 in Demonstrations_Application and select 'Human Hybrid' from the drop down menu.
[22] *Oxford English Dictionary* definition, viewed 30th October 2013.

by the term *harmonic language,* a term often used when defining aspects of a musical work that create harmonic cohesion. Harmonic language is a higher construct that involves the collections of harmonies and how they move temporally from one pitch collection to another. The harmonic terms are analogous to their rhythmic equivalents discussed above; *harmony* relates to the micro and *harmonic language* relates to the macro.

When discussing rhythm it was appropriate to use a bottom-up approach, building larger and larger complexity from simpler functions. Here it is more appropriate to take a top-down approach. To maintain harmonic cohesion over large sections of a work the compositional technique of using pitch-class sets will be employed; pitch-class sets being the arbitrary choice made by the composer to use only a specific collection of tones to construct a whole piece or section of a piece. The harmonic language for GMGEn will be limited to four specific pitch-class sets. Implementing pitch-class sets allows the program freedom to choose any of the notes that it wishes from those provided and will produce reliable harmonic combinations based on any two (or more) of the set pitches.[23]

Figure 4 – Hexatonic Major Pitch Class Set

The hexatonic pitch class set of C, D, E, F, G and A (see figure 4) can be selected by the user within the patch interface. If many lines of music are all bound to

[23] See Tab 4 in Demonstrations_Application and explore the drop down menus.

this same pitch-class set then the music generated will be major-inclined. This means that the music produced will not evolve harmonically because it is unable to play any but the six notes provided.

The reader can see that any two-note combination taken from these six produces a chord that adheres to the certain harmonic properties contained within this pitch-class set. This is true for any trichord also.[24] It is the intention that any number of these pitches sounding simultaneously will produce chords that contribute to a consistent harmonic flavour. In essence, what will be created is an area of highly predictable, yet indeterminate, harmonic content. Winkler illustrates how the technique of constrained randomness allows a composer to set up 'improvisational situations where the music will always be different and unpredictable yet it will always reside within a prescribed range that defines a coherent musical style.'[25] The same concept applies here and allows the creation of a coherent harmonic language.

Though this area of harmony is predictable, it cannot be momentarily determined. This is analogous to the concept of an electron-cloud in a molecule or atom. An electron-cloud is the probabilistic 3D shape created due to the fact that no individual electron's location can be determined accurately. An electron here is the metaphorical equivalent of a pitch in a cloud of harmonically complementing pitches; the pitch cannot be determined accurately but its limited possible forms can be. Instead of the 3D probabilistic shape given to electron clouds, here we have a probabilistic sonic outputs. Given the analogous representation of electron-clouds to a cloud of harmonically complementing pitches I am appropriating the terminology from physical chemistry to describe *harmonic-clouds*.

[24] See Tab 4 and select 3, or more, lines from the first drop down menu.
[25] Winkler, 'Strategies for Interaction: Computer Music, Performance, and Multimedia', p. 2.

This harmonic-cloud technique is different to that of composers such as Iannis Xenakis, Krzysztof Penderecki and György Ligeti who all, Jonathan W. Bernard agrees, deal 'directly with *masses* of sound, with aggregations that in one way or another de-emphasized pitch as a sound quality, or at least as a *privileged* sound quality.' GMGEn does not create sound masses as complex as the work of the composers here. What is discussed in the section above is a cloud of *potential*, and not *actual*, sounds. The actual sound produced, because of GMGEn's limit of six simultaneous lines, can at a maximum be a chord consisting of six simultaneous pitches, a number that is far from considered a 'mass' in the same terms as those composers above.

When discussing aleatoric harmonic sections in the music of Lutoslawski, Rae notices that, similar to that proposed above, 'all such aleatory sections … share the disadvantage of being harmonically static' and further shows how Lutoslawski avoids stasis by 'composing some passages that move gradually from one chord to another.'[26] While Lutoslawski blends two of these masses together forming three distinct sections, in GMGEn I chain the end of one harmonic-cloud to the beginning of a second to create a harmonic evolution and avoiding staticity. This, in essence, is a dynamic music chord progression: areas of highly predictable potential harmonic character chained together to create harmonic-cloud progression. With rhythm we were able to regenerate small sections to produce phrases, with harmony we can control the chain of harmonic-clouds to produce our own harmonic evolution.

It is important to note the variables of note choice and harmonic-cloud duration in the case of harmonic-cloud control. The changing of these variables will adjust the output of the sound dramatically and provide an effectively infinite amount

[26] Rae, pp. 79.

of variation. These variables can be statically assigned by the composer-programmer for the computer to automate, or a further higher level of control can be built to allow the computer control over these aspects of the music. To demonstrate this, a small selection of pitch-class sets will be used for the note choices of individual harmonic-clouds. The aim is that these pitch class sets will have unique qualities that will give the harmonic-cloud they are assigned to an individual quality. Within Tab 4 of the demonstration patch there are the following four pitch-class sets which are used in GMGEn: A hexatonic major set (see figure 4), a hexatonic minor set (see figure 5), a hexatonic wholetone set (see figure 6), and an Octatonic set (see figure 7). Given these four types of pitch-class set the composer-programmer is able to switch between different sets at will or at any juncture required by the gamestate. The musical effect here is to be able to switch between musical modes at the trigger of a game parameter, and thus allowing harmonic evolution. Switching between the hexatonic major set and the hexatonic minor flattens the third and sixth notes of the preceding set meaning the overall music will change from major to a minor quality, which could be useful for a transition from a town to a castle scene for instance. Constituent pitches of any set could be adjusted creating an entirely different harmonic language for any required game scenario; Tab 4 also includes two further pitch-class sets which are less bound in the traditions of Western classical music (see figure 8 and figure 9).

The second harmonic-cloud variable of duration is also simple to control and does not have any deep areas necessary for discussion. If a harmonic-cloud has too long a duration it will become static, however, too short a duration and it won’t have time to establish a harmonic identity on the listener and will diminish from the overall harmonic consistency that a harmonic-cloud is meant to provide. Determining the lower and upper thresholds for harmonic-cloud duration will be relative to the musical context considering other elements of the music. It is likely that with some testing a reasonable formula could be produced which could then be added to a higher level of control given to the computer. This data is not yet available.

Figure 5 – Hexatonic Minor Pitch Class Set

Figure 6 – Hexatonic Whole Tone

Figure 7 – Octatonic

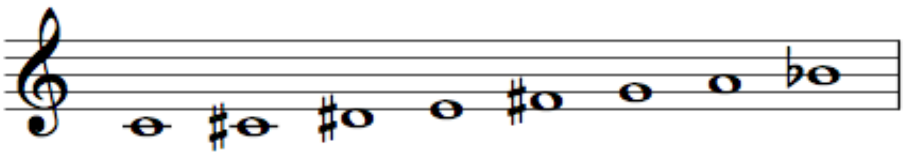


Figure 8 – Hexatonic Collection X

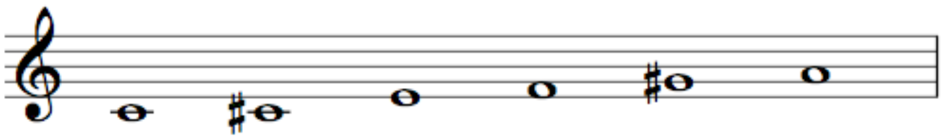
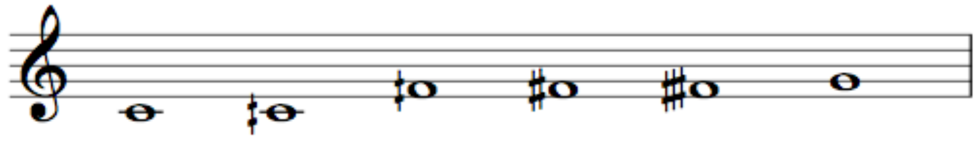


Figure 9 – Hexatonic Collection Y



GMGEn

A more detailed look at what GMGEn achieves demands discussion on which aspects of music it has control over. Partitioning a complete work of linear music into two contributing 'cosms' will appropriately describe these aspects. The first is the exact constituents of the sounds we hear, such as the pitch of a note or its duration; this level is the *microcosm* of the music. The second comes from the larger complexities that combinations of different microcosms create, and can be called the *macrocosm* of the music. Features of the macrocosm can include, for example, structure, harmonic language, musical behavior, among others.

When discussing various corpora (bodies of works) to be used in his program Experiments in Musical Intelligence, Cope notices that certain patterns occur in the works of the composers he chose. He calls these patterns *signatures* and defines them as 'contiguous note patterns that recur in two or more of a composer's works, thus characterising some of the elements of musical style.'[27] GMGEn does impose exact patterns between playings but not over small-scale aspects of the music. Cope uses the term 'signatures' to describe patterns in the microcosm of music. For patterns in the macrocosm I use the term *personality*. GMGEn is a generative music engine specifically designed for creating musical personalities. The concept of a musical personality in this sense is analogous to a non-specific musical template, which uses

[27] D Cope, 'One Approach to musical intelligence', in *IEEE Intelligent Systems*, vol 14, no. 3, 1999, pp. 21-25.

the perception principle of invariance. Invariance, is defined by Clarke as ‘the idea that within the continuous changes to which a perceiver is exposed there are also invariant properties’.[28] For example, in a twelve-bar Blues the performer knows the musical template for the structure and key of the piece; it is important that all players know what these templates are, to allow for complementing improvisatory material to be added.

GMGEn’s templates are not limited to structure as are the templates of twelve-bar Blues. GMGEn has a similar non-motivic approach to form as George Lewis’s *Voyager* does, in which he describes a ‘sonic environment is presented within which musical actions occur’.[29] In GMGEn, these actions are randomly generated by the lower level functions of programming logic. The ‘state-based’ model, Lewis confirms, is ‘particularly common in improvised music’.[30] By constraining the random values generated in the lower level functions, GMGEn can be said to be acting in an improvisatory way. Further, composition of a state-based generative system directly complements the state-based nature of a video game.

The goal of GMGEn (put forward at the beginning of this chapter) was to create a fluctuating microcosm while also creating a consistent score-like macrocosm, across multiple performances. When the program is first loaded the computer generates batches of numbers and saves them to memory. These memories are the raw material from which all music is generated in GMGEn. Based on many parameters set by the user the numbers are then manipulated through the program to produce output (See Table 2 in Appendix B below for a simplified version of this process). All white boxes in Table 2 (in Appendix B) show choice parameters that can be changed by the user (or bound into a preset), all black boxes show generations produced by GMGEn.

[28] Clarke, p. 34.
[29] Lewis, p. 105.
[30] Lewis, p. 105.

Some possible outputs are shown and used as examples through the algorithm. The lower half of the figure shows some of the effects and processes that can be placed upon the main musical line created, which itself is shown in the centre of the figure. To keep consistency throughout a performance the original memories are recalled, but can be regenerated upon beginning the piece again. The memories could also be regenerated from an in-game trigger if necessary.

Master-presets create a score-like consistency across the macrocosm by effecting any numbers generated in the memories—which are then processed through their logic pathways—in the same ways each time. A master-preset in this sense is synonymous with a specific musical personality. With consistent activation, the same master-preset will produce a similar sounding style of musical composition regardless of the microcosm of its memory, which are randomised each time they are regenerated. The building of master-presets was achieved while constantly regenerating the memories. In this way I was able to adjust variables until the program was producing music that conformed to my intuitions. This way I knew that repeated generations of the memories would create similar music and work proceeded empirically. I have created ten different personalities as a basis for the generated narrative work that accompanies this paper.[31] I contend that it should be considered GMGEn is a musical instrument rather than simply a musical piece; by way of adjusting the inner DNA of the master-presets, GMGEn's output is as customisable as that of any other acoustic or digital instrument.

The musical transitions GMGEn makes are not designed to be successful in all cases. The artistic reasoning for this is twofold. First, the narrative work it is designed for features an overconfident, arrogant, fictional AI system, called PAMiLa, who

[31] See Tab 5 in Demonstrations_Application

believes its intelligence far outstrips that of a simple human. That this fictional intelligence might occasionally be unable to compose a simple transition—regardless of confidently suggesting that it can—adds a human flaw to the artificial character. Second, as I know it is only my perception of certain transitions as 'successful' or otherwise that makes them thus, I do not expect my view of success to be the same as any other human observer. It is possible, and probable, that there is a person who perceives every transition that GMGEn makes to be a success just as there is the opposite person who perceives every transition is a failure. These observer cases would still exist even if *I* were the former individual. Coupled with my contention that artistic beauty comes from perceived perfections within an imperfect whole, or rather the order (success) that is found from disorder (failure or otherwise), leaves me satisfied sitting in the position I am: perceiving some of GMGEn's transitions as failed, which I would never dream of composing myself, and some as successes, which I could never have thought of myself.

The work for generative narrative, which is found in the InteractivePortfolio_Application, directly mimics the scenario found in an open world game such as *The Legend of Zelda*.[32] This piece takes pre-composed segments of a narrative and combines them together in a non-linear string. Each of these narratives is assigned a musical personality from GMGEn to accompany it. When a narrative segment is complete a new one is selected and the musical personality transition begins in GMGEn. It is simple to map the sentences of the narrative onto a virtual 3D game world and to see the benefit that an engine skilled at creating transitional material brings to game music scenarios. A working version of GMGEn, with ten personalities is available for the reader to explore within the

[32] See 'NARRATIVE SYSTEMS' tab in the InteractivePortfolio_Application.

Demonstrations_Application.[33]

[33] Tab 5 within the Demonstrations_Application; click any personality and wait a few moments. The user is also able to regenerate the memories on-the-fly showing a new version of all personalities.

Chapter 3 – Memorable Musical Microcosm: Intermittent Music

This chapter will discuss a way of hybridising the engines discussed in Chapter One and Two to form an engine termed the *Intermittency Engine*. In doing so I first summarise the strengths and weaknesses of the engines in the previous chapters. I also present the issue of musical memorability in both the branching and generative techniques used in the other chapters of this paper. I suggest a way to combine the strengths of both branching and generative engines to create a more immersive hybrid music engine. This solution will come in the form of a system that uses a concept of pre-composed islands of music (defined below) amid seas of generative musical personalities (as defined in the previous chapter).

The discussion on the branching and generative music engines above largely illustrated the strengths of each engine. Though some discussion on limitations has taken place, it is necessary to analyse these limitations further to provide a clearer picture on the useful qualities from each, which will then be combined within a third hybrid engine; the focus of this chapter. These qualities pertain to four areas: hardware limitations, success at transitioning between two musics, whether any long-term monotony is present, and whether the music is memorable.

Hardware limitations for a branching music engine or a generative music engine are opposed. In summary of the previous chapters, I stated that serious consideration is needed from the developers of games implementing branching music. This is due to the amounts of raw drive space needed to store all the multiple branches of pre-composed music. For modern devices, such as iPads and iPhones with small standard disk sizes this would be an issue of even greater concern for a game implementing purely branching music. With the amount of pre-composed music needing to be loaded this could even tax a processor. The generative engine, however, is more processor intensive and demands less hard drive space. Intensity here is based only on the comparison between a generative engine and a branching engine because in a game context the processor must perform many other calculations in any given moment, which are far more demanding than those of either music engine. These other calculations include those needed to accurately represent physics on objects or the rendering of 3D models. I stated that these demands are not a significant enough hindrance due to the advent of more advance technologies. These technologies include greater lossless audio compression, such as FLAC, and the increase in the average size of hard drives as well as jumps in processing speed.

Transitioning between two continuous musics was the primary goal of both previously discussed engines. The transitional phase of gameplay is usually short (less that twenty seconds) and so it was only necessary to find a way in which each engine could fill these time-spaces with appropriate music that blended together two, possibly separate, musics. While manipulating different material both engines fundamentally work in similar ways. In both, the programmer has dictated the choices the computer must follow and the computer has executed as per those instructions. The transitions in branching music are given aesthetic validity by the composer, who has designed the music to faithfully portray certain scenarios by means of notated (or similar) score. Transitions in the generative music engine GMGEn are given general constraints by the user. These constraints force the random output from the computer to form particular coherences across multiple hearings due to the principle of invariance defined in the previous chapter. In GMGEn this amounts to a general scoring rather than literal scoring. In this case, transitions in GMGEn are achieved by teaching the program correct ways to adjust particular elements of the score when triggered from within the game. While the two engines perform well during transitional states, there are differing degrees of success achieved when using these engines outside of transitional periods where musical personalities may be static for longer durations (greater than twenty seconds).

In a branching music system the musical pixels of any individual archbranch are already musically designed to suit static game states. Therefore this system achieves aesthetically appropriate continuous music within both transitional periods and static periods of game play. Assuming the composer has competent skill at avoiding monotonous moments in their music, the branching music engine should never be found to be monotonous. The limit of monotony in a branching music system is therefore something intrinsic in the composed music and not a property of the engine itself.

A generative music system like GMGEn is not as successful at producing music for long static periods (time periods greater than a minute in duration) due to an eventual onset of monotony. Most game's static states will generally last for longer durations than transitional states; therefore, ways in which the system can vary the musical personality must be included to avoid monotony. Musical personalities (or *master-presets*) in GMGEn are only designed to maintain musical interest for approximately one minute, therefore upon extended listening to any individual personality the reader will notice that a limit exists after which a decline in the musical interest of the personality occurs. Though the exact times may differ between each master-preset, this generality is true for all presets.

In the construction of GMGEn efforts have been made to avoid the perception of monotony in the short-term. These efforts also indirectly increase the time for which a particular personality can be played. In GMGEn this technique assumed that an appropriate balance of both surprising and expected musical moments are needed to maintain the listener's interest. This suggests that a level of predictability in the music is desirable to meet the expectations of the listener. GMGEn takes advantage of this by looping and reusing short memory fragments within a subsection of a single musical personality, and further modulates them to avoid monotony. This achieves two things simultaneously: it creates predictability by repeating short patterns of music, while maintaining musical interest by modulating these patterns into a different harmonic-cloud. In this example, invariance is found in the pattern of the microcosm and provides a stable point for which the listener can pleasantly predict the short-term future of the music. The aleatoric modulation provides the necessary musical surprise, which allows these short-term loops to avoid becoming monotonous too quickly. Revisiting these saved patterns and juxtaposing them with other saved patterns treated in the same way creates a stable surprise-prediction balance within a single musical personality. In GMGEn, eight patterns (or *subpresets*) are composed for each musical personality (also *master-preset*). Although adding more subpresets would increase the non-monotonous duration of any master-preset the eventual onset of monotony would still be limited by the quantities and qualities of this material. The number of subpresets used for any personality will be dependent on many factors relating to the specific game state it is designed for.

There are situations when the long-term memory faculty of the listener can be utilised by the game music composer. The composer may wish to attach musical motifs to events or situations throughout the entire game in order to emotionally affect a player in future situations that may increase their level of immersion. Creating musical motifs that inhabit the same temporal position as certain characters, locations or emotions (among others), will to some degree make these motifs perceptibly inseparable from them. Once these motifs have been established it is possible to trigger the same emotional response in the listener again.[34] Clarke shows that 'the identification of a characteristic motivic/harmonic procedure…are behaviours that networks…can be shown to demonstrate after suitable exposure'.[35] Clarke's point directly applies to the network of a human brain. The listener would be remembering a previously heard music and connecting that to a previous emotional state. This type of musical prediction is a long-term version of the short-term musical prediction described above. Therefore, short-term musical memory comes from recognising the patterns of music in the present and extrapolating into a near future, while long-term musical memory is the exact memory of the microcosm of music over a large period of time, several times more than the length of a single listening. To give a typical example of this phenomenon take a person who played through *Final Fantasy VII* around its release in 1997. This player may still retain much memory of the music when replaying the game again many years later regardless of having little or no contact with it in the interim. While the reactive agility of GMGEn is positive and the design of this system fully complements the reactive situations found in almost all game scenarios, it does not create music memorable over these long-term periods. It will therefore not provide the composer with access to the emotional affecting power of motivic attachment techniques and they will ultimately not be able to use these techniques as a way of affecting the player and increasing their immersion.

The motivic attachment approach (*leitmotif*) has a secondary benefit to the video-game music culture in that it can provide a great deal of nostalgic feeling for

[34] Clarke, pp. 30-31.
[35] Clarke, pp. 30-31.

the player whether during a single playthrough or across multiple playthroughs. While the pursuit of memorable music in this paper is primarily to heighten the immersive experience for the gamer, some consideration of this nostalgic benefit is important to this discussion. Musical motifs often occur throughout multiple games in a particular series. As discussed in Chapter One, the *Final Fantasy* battle music has a very particular style associated with it and in many of the earlier games a particular bass guitar introduction can be heard. Moreover, the *Hyrule* theme is heard in almost every game that is part of the *Legend of Zelda* franchise.

The obvious cause of this long-term memorability and subsequent nostalgia associated with these scores can be attributed to the vast amount of repetition found. This is not only due to the looping of scores but also that the same tracks can be heard in many different locations or in multiple scenes and scenarios in the game. Taking *Final Fantasy VII* as a case study: four and a half hours of scored music exist but the game may take over thirty hours to complete on a first attempt. Therefore, the score will be repeated between six and seven times during this first full playthrough. Further, the listener may not hear the full four and a half hours of existing score due to them not choosing to visit (or simply not discovering) certain locations, or experiencing certain scenes where unique portions of the sound track might be played. This further contributes to the number of likely repetitions. Players drawn along by the game's plot elements are therefore required to listen to this music repeatedly resulting in the music becoming subconsciously embedded in the player's memory.

An engine with the ability to execute long-term memorable music will enable the video-game music composer with access to the immersive influence that motivic attachment techniques allow. A branching music engine's music is memorable over long periods of play and across multiple plays because it creates an exact microcosm on each play. Conversely as GMGEn cannot create these situations, the branching music engine has the edge on this criterion. An ideal system is achievable in which the user receives benefits of both long-term memorable segments of music containing static microcosms (branching) while still receiving the reactive scoring a dynamic music engine brings (generative). The obvious step made here is to create a third engine hybridising the two previous engines to exploit the strengths of both in their best situations. Therefore, a branching system populated with pre-composed music will provide the composer with the ability to emotionally affect a player over long periods of play. A generative system producing musical personalities will provide the reactive flexibility needed by the potential fluctuating game states. The generative system will be used to transition music while the branching system will be used to occupy static game states.

I appropriate terminology for this system from scientific dynamic system models resembling the proposed hybrid system. The term intermittent is used to describe systems that show periodic behavior within chaotic phases. Musical intermittency can be seen as a metaphorical representation of the mathematical system rather than an exact rendering. The ostensible semantics of this term are readily understandable and appropriate for describing the general behaviour of the proposed system; therefore, the differences between the mathematical and the musical are a negligible concern. In short, musical intermittency here will describe an engine that switches between two distinct states. Part of the disparity between the mathematical and the (now) musical intermittency models is found in the creation of the intermittent behaviour. In mathematics, intermittency is observed naturally, it is not enforced upon the system by artificial means. It simply occurs, and is therefore created by the same physical laws governing this universe. In a dynamic musical system intended for video-games, the intermittent behaviour would be artificially enforced upon the music under precisely triggered conditions.

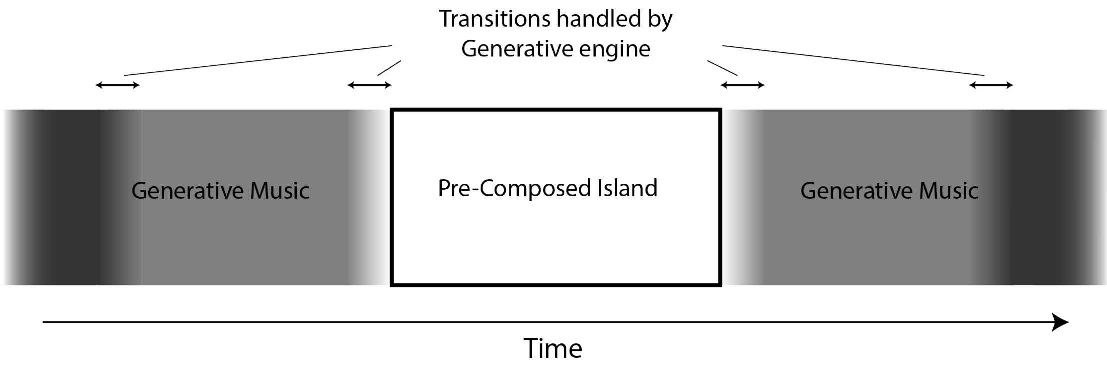
The intermittency engine will have a pre-composed branching side and a generative side constituting the engines explained in detail in Chapters One and Two. Each side of the intermittency engine will deal with one of the two different states of game play. As has been explained before these states are roughly approximated as either static or transitional. The generative side of the engine will deal with transitional states to allow reactive agility. The branching side of the engine will accompany static states to allow a long-term memorable microcosm to become established within a player.

To allow for pre-composed music to be played there must be stable periods of time (*time-spaces*) where the game state cannot change. These time spaces occur as part of a whole static state. In these time-spaces we would be waiting either for temporally known events to run their course or in a situation where, for a known duration of time, the player would be unable to trigger an event that could require a change in the music. During these stable periods of time-space it is unnecessary to change the music and therefore the music accompanying these sections can be linearly composed. It is at these points that the composer can insert affecting motifs drawing upon the long-term memory of the listener, reminding the player of particular places, people or emotions. These stable time-spaces exist in various scales and at various locations within a typical game. Their exact properties are dependent on extremely specific situations and options available to the player and therefore are unique for every game. Different programming techniques are necessary to determine exact durations for a specific static time-space; however, from the composer's perspective once the exact duration of the time-space is determined they simply need to write appropriate music for this exact duration of time. In certain situations it may be beneficial for multiple versions of the music to exist but this is a further consideration beyond the scope of this paper. The separate states of this engine and the compositional properties of the music they provide can be described using a visual analogy. As the stable time-spaces are occupied by determinable music they can be seen as *pre-composed islands*. In this analogy, the determinable music of the islands are juxtaposed by the indeterminable music of the *generative musical seas* surrounding the island.

Figure 10 shows an image of a pre-composed island of music inside the chaotic (here grey) sea of the generative music with the time domain running from left to right. This image is useful for illustrating the presence of two sides to the intermittency engine as well as consolidating the analogy above. The obvious borders of the pre-composed island are ridged to represent the stability (and definability) of the time-space and the musical microcosm contained within. Transitional periods between generative musical personalities are shown as a blending of two colours to represent a period of time where qualities of both musical personalities exist simultaneously, due to the transition. Each linear end of the diagram shows the blending, or transitional period, between two musical personalities. Two different musical personalities within the generative engine are shown as two different shades of grey.

Figure 10 – Graphic score showing the implementation of Pre-Composed Islands of

music within Generative Musical Seas, which handle transitional phases.



I wish to illustrate how the intermittency system might be implemented within the two scenarios upon which this paper has already demonstrated possible improvements: the traditional *Final Fantasy* battle system discussed in Chapter One and a more spatially (2D or 3D) explorable (open world) adventure game. In the examples below I will explore the transitions between each side of the intermittency engine. First, I discuss the transition from the pre-composed side to the generative side using the *Final Fantasy* battle system explained in Chapter One.[36] Second, I will discuss the opposite transition using the case study of a more modern game, *Rogue Legacy*.[37]

The intermittency engine would have a pronounced effect on the music of the battle system found in the earlier games in the *Final Fantasy* series. Chapter One analysed the typical structure in this scenario and found that the greatest aesthetic discontinuity was found during transitions at the end of a battle where the battle music is cut short and a new victory fanfare begins; this was illustrated in the Demonstrations_Application.[38] My aim was to connect these sections of music or

[36] *Final Fantasy VII,* Playstation Game, Squaresoft, Japan, 1997.
[37] *Rogue Legacy*, PC/Playstation 3 Game, Cellar Door Games, 2013.
[38] Tab 1 and Tab 2 of the Demonstrations_Application

design a reactive ending to the battle music.[39] The branching engine addressed the issue with the disjointed transitional music heard in this situation. Using an intermittency engine the transition between overworld and battle would still be handled by pre-composed music requiring no designed control over the transition. I argued that this particular transition was aesthetically connected to the narrative potential of battle and so does not require change. The main body of music would also be handled by pre-composed sections of score thus providing a memorable musical microcosm for the player. In the intermittency engine, the final transition from the battle music to victory music would be handled by a change in musical personality that the generative side of the engine would produce. This is because these sections of gameplay only last for a short duration. It is prudent to consider appropriate game triggers where the engine can be switched from the pre-composed side to the generative side to handle the transition.

A switch between pre-composed music and generative music must be made prior to the point at which the visual (and musical personality) transition would take place; therefore, requiring two triggers to make an full musical transition: first, the *priming point*, where the system switches to a generative side; and second, the *visual transitioning point*, where the engine begins to make the musical changes between two personalities (see Figure 11). Figure 11 shows the priming point at which the generative engine takes over and starts providing generative material similar to the pre-composed music (Generative A), and the point at which the generative music engine shifts personalities (into Generative B) synchronising with the visuals at the visual transition point. The *Final Fantasy* battle system offers many game states that could act as priming points. One suggestion is for a priming point to be based on an

[39] Tab 2 in the Demonstrations_Application

enemy-health threshold. Once the enemy's health is below a certain value, for example ten percent, a trigger message would be sent and the engine would transition between the pre-composed side to the generative side. Another threshold trigger, for a priming point, might be taken by roughly estimating how quickly the characters could end the battle based on their current strength compared with the enemy. There may be several methods for obtaining this data, which the game designers would have to decide upon during development. The latter achieves this goal to a greater extent across the entire traditional game setting as during the later stages of the game the player's characters will become much stronger than the majority of enemies they will face meaning that the characters can often execute a one-hit knock out (KO). Further, in this case, with the priming point being likely to occur straight after the opening introduction music this would mean that for one-hit KO battles, or generally shorter fights, the intermittent music engine might skip straight to the generative side after the pre-composed introduction. In consideration of this the analyst can expect the intermittency engine to write music with two emergent structures. In the event the battle takes a 'long' time (defined here as the system having enough game time to reach pre-composed music in Section B, described in Chapter One) the emergent structure would resemble that shown in Figure 10. In short battles where the system is unable to reach the pre-composed island the structure would resemble that shown in Figure 12. In this case it is preferable to employ several different priming points designed to create the largest duration possible of pre-composed music under the specific conditions of the battle. The intermittency engine also has utility outside the specific battle situation and can be used to score narrative portions of the game also. The fact that this engine tackles the root elements of all gameplay enables it to be used in many other game states and in other game genres.

Figure 11 – Priming Point and Visual Transition Point locations

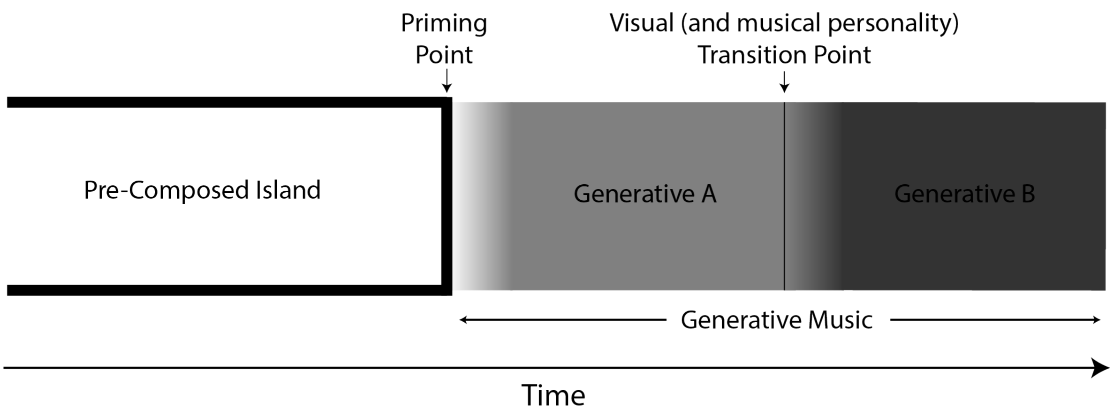
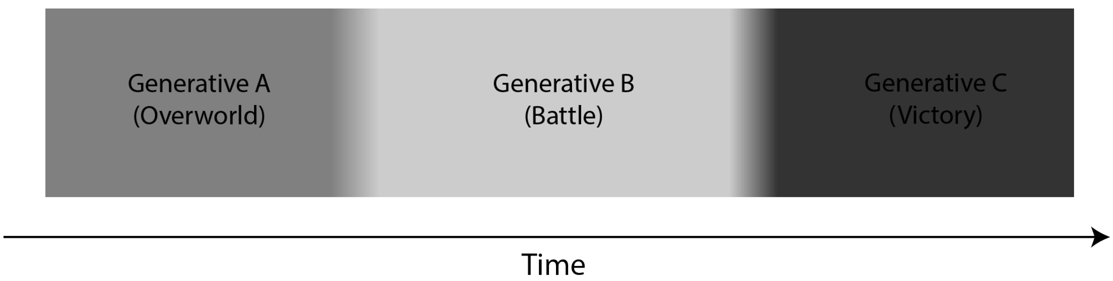


Figure 12 – Structure of music in game states unable to reach pre-composed islands due to highly transient natures (suggested personalities for *Final Fantasy* shown in parentheses).



Consider a 2D side-on platforming adventure game such as *Rogue Legacy*.[40] The generation of levels in *Rogue Legacy* is achieved in a macroscopic way. In this game the user will traverse through a land divided into four separate areas known as the Castle, Forest, Darkness and Tower (see Figure 13). The larger areas are always in the same position relative to one another and individual rooms within the larger areas are generated differently each time. An analogy can be drawn between the method

[40] *Rogue Legacy*, PC/Playstation 3 Game, Cellar Door Games, 2013.

with which level generation occurs in *Rogue Legacy* and the way in which musical personality generation occurs in GMGEn. The player starts in the castle and will find the tower upwards, the forest rightwards and the darkness downwards. Implementing an intermittency engine in this scenario would link each of these areas with their own musical personality, which has a predictable macrocosm and an unpredictable microcosm, mimicking this world's geography. These personalities would create the foundation for the transitions between zones. Although less obvious than within the *Final Fantasy* battle system above, there are still adequate moments in which a switch can occur between generative side of the engine to the pre-composed side. Figure 14 illustrates the significant points necessary for transitioning in this direction, from the generative engine to the branching engine.

Figure 13 – *Rogue Legacy's* macroscopic world generation area positioning

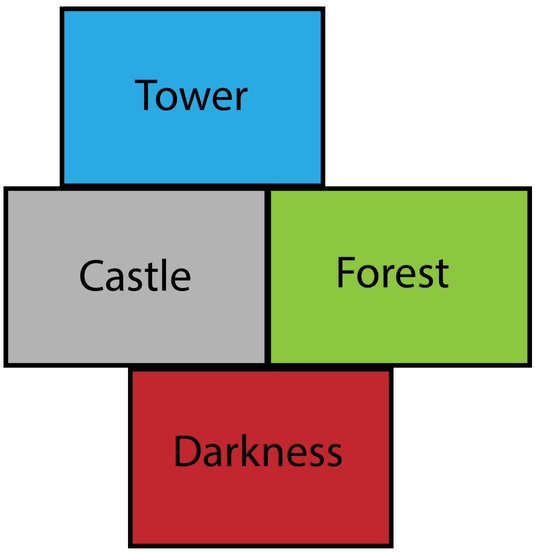
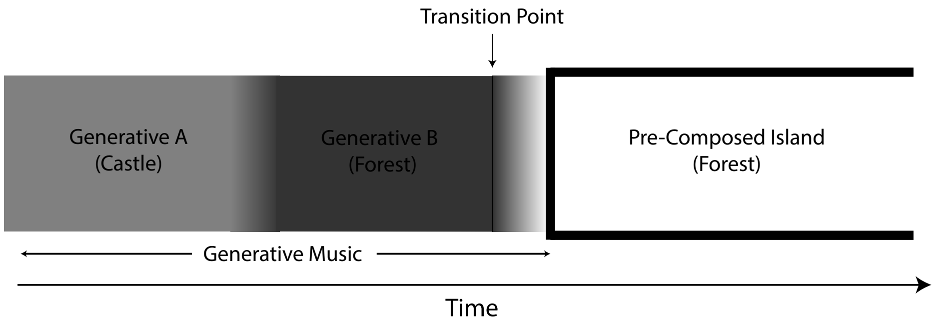


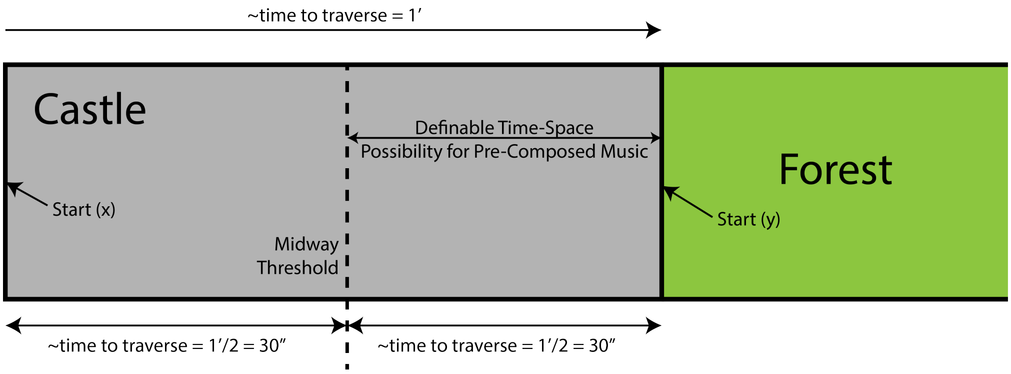
Figure 14 – Transitioning into pre-composed music within the Intermittency Engine



For the transition from the generative side to the pre-composed side of the intermittent engine to occur we must find an appropriate transition trigger within the game. In *Rogue Legacy*, the player takes time to traverse from one area to another. Giving a rough estimate for this specific scenario, the shortest time taken to get to the forest from the starting location (leftmost in the castle (see Figure 15)) takes approximately one minute. Let's assume that even given an optimum route generation built for getting to the forest as quickly as possible a user cannot get there in less time than one minute. This means that there is definable duration where the player can only be in the castle zone and under no circumstances is it possible for the player to be anywhere else that would require different music. This is a simplified test scenario and does not take into account other features of this particular game; however it is still true that if an assumption of this nature can be made consistently and accurately within any game then the proposed duration can be filled with pre-composed music. Improving the accuracy of an assumption requires a complete knowledge of all possible game state transitions but can often be simplified in a number of cases. The previous assumption about the definable duration within the castle would not be true if, for example, the user actually starts in the forest and transitions into the castle zone (start y in Figure 15). We would expect to hear castle music but we can no longer assume that it will take the user one minute to return to the forest as the player is much closer to the forest now than they were at the original starting location; therefore, we cannot simply start a one minute long piece of pre-composed music because the player may choose to turn around and re-enter the forest having only spent a very short period of time in the castle. This would result in any pre-composed 'castle' music now being unintentionally played in the forest. In this case the trigger for the switch to the pre-composed side of the engine must come at a different point. A point where the player is as far from any game-state transition as is possible.

Figure 15 illustrates a more advanced trigger able to select an appropriate point to switch the intermittency engine to the pre-composed side during a definable time space. As our starting positions we here take both the original starting location (start x in Figure 15) or the entrance to the forest (start y in Figure 15). Adding a midway threshold as a trigger allows us to calculate a definable time space from the moment the player crosses the threshold. Upon crossing this midway threshold in either direction the player now must travel for at least thirty seconds to reach the forest regardless of which location they started at. Although this creates a smaller time-space for pre-composed music than was in the first example, it is the only time-space which is definable when multiple starting locations (start y or start x) are possible. These examples also show that pre-composed music filling this space needs to be short enough for the eventuality that the player moves directly towards a game-state transition point and will therefore take the shortest amount of time possible. Again, much focus should be given to creating ample numbers of transition triggers (towards the pre-composed side) that enable an intermittency engine to produce the largest amount of pre-composed music possible to enable long-term memory of the music across playthroughs. Employing branching music techniques within pre-composed islands would create opportunities for larger islands and would provide more compositional interest to these sections.

Figure 15 – An appropriate in-game trigger to switch the intermittency engine.



Hardware demands remain trivial when implementing an intermittency music system for most conventional gaming technology. Hybridisation of the branching and generative engines combines to an averaging of their hardware demands. An intermittency engine would exploit the use of many different pre-composed sound files like that in a branching music engine but would have vastly smaller pools of those sounds. Therefore this would not create the same level of demand for drive real estate as a branching engine but would require more than a generative engine. Memory demands are greatest in the branching engine, less in the intermittency engine and lowest in the generative engine. When compared with other simultaneous demands in the video game context, processing demands are not high for either a branching system or a generative system; therefore, under a worst-case scenario an intermittent system could not exceed whichever were found to be the highest. The prime bottleneck occurring in all three of these different engine scenarios is due to the reading of the sound files from a hard disk. This occurs because of the physical time it takes the read-arm of a hard disk to read the information on the disk.

A hard disk works in a similar way to vinyl records except that instead of an extremely fine spiral over the record's surface there are concentric circles; to read a block of memory the read-arm must move from one concentric circle to another. This takes approximately ten milliseconds per move and can therefore add up if the memory blocks needing to be read are large or separated on the hard disk. At the time of writing the solid-state drive (SSD) is becoming a commercial standard in new computers. An SSD is able to read any block of memory without having to physically move a read-arm to different locations on a disk. This effectively minimizes time taken to access information. As SSDs become a standard in gaming technology the issue of memory read bottlenecks disappears. To summarise: current technology can trivially handle the loads of the most taxing musical engine discussed in this paper, the branching engine. The technology of the near future will therefore make any hardware load issues of a less taxing engine, such as an intermittency engine, negligible.

Before rigorous testing of an engine working in many different active scenarios is done it is difficult to forecast negatives onto the proposed intermittency engine. This game engine is complementary to the game scenarios proposed in this paper but will also work in a variety of others. An arguable, yet subjective, weakness can be found in those scenarios where game-states change extremely quickly. These scenarios would give the intermittency system less chance to settle into pre-composed music. In this case the intermittency engine would produce music from the generative side and would not provide the full aesthetic experience this engine is designed for (see Figure 12).

This paper has thus far only considered an intermittency engine where the pre-composed music is of a low musical resolution (discussed in Chapter One). If this resolution were raised, the intermittency engine could also suit the needs of games with particularly frequent game-state transitions. Moreover, it is in these types of scenarios that I believe the intermittency engine is likely to produce the greatest degree of success at creating memorable, aesthetic, continuous music, to provide greater immersion for the player, within this dynamic medium in the future.

Conclusion

During this paper I have described the construction of three reactive musical engines that provide ways to produce continuous music throughout a variety of games scenarios and transitions between these scenarios. In the introduction to this paper it was shown that music has the power to aid in the creation of an immersive virtual reality for the player. It was also shown that scholars had noted an abruptness to some transitions within games where music was cut short to meet the change in the game's state. Crossfading between two musics during a transition of game state shows that some efforts have already been made in the industry to lessen the abruptive impact in these situations. Even considering the crossfade as a form of solution, the musical transition will still appear abrupt (see *Final Fantasy VII* demonstration video) as musical style can instantaneously change and therefore both still result in a lessening of the immersive effect by the game on the player. It can be seen that even though the subjects of the case studies in this paper span over three decades, this method of abrupt musical curtailment is inherent in modern games including the subject of the case study in Chapter Three, *Rogue Legacy*. The prevalence of these mechanics in modern games (see Chapter One, *The Binding of Isaac* and *FTL: Faster Than Light*), coupled with the aesthetic judgment that they are an inadequate musical and programmatic solutions, given current technology, provides justification for this study.[1] Although this study uses some scientific methodology and terminology, successes are based on my own aesthetic concerns about music and my experience, both as a composer of music, and player of video games.

The branching music system discussed in Chapter One uses a musical

[1] *The Binding of Isaac*, PC Game, Headup Games, 2011; and *FTL: Faster Than Light*, PC/iPad Game, Subset Games, 2012.

pixelation technique to separate a continuous score of music (an archbranch) into separate small segments in order to allow the program to know its location during play of the music. Coupling this technique with other archbranches and capillary branches allows pre-composed visually reactive transitioning to take place within the game's music. This builds on scripting solutions, such as iMUSE, and some of the branching capabilities of the FMod software by proposing the concept of high musical resolution where the distance between musical pixels is temporally short. It succeeds in the speed in which a branching music engine with a high resolution can adapt to the changing game states with pre-composed music. The branching music engine is potentially limited by its need for large amounts of hard drive space which may not suit gaming platforms with small hard drives, such as iOS devices or other hand held systems. A further limit arises for the composer who must produce large amounts of material for such a potential-music system; this would be a time consuming process for any single composer. Some attempt at addressing this was made in Chapter One by way of heuristic methods applied across multiple branches that may be composed similarly.

The generative music system proposed in Chapter Two suggests one method for creating static musical personalities that can be attached to locations or states within a game. The system is also able to blend between these personalities while creating music for this blend. This is dissimilar to a crossfade as a blend in musical style also takes place rather than simply a blend in amplitudes. The engine applies the concept of artificially improvised musical states to the state-based medium of video games. Its success is found in the interesting level to which a style blend can take place between two musical personalities. GMGEn can also act as a tool for creating new musical personalities which can be transitioned to, or from, within the engine.

GMGEn is limited in that it can only produce musical interest for a relatively short duration of time (less than two minutes) before that music starts to become monotonous for the listener. Due to the way it produces personalities of music, the microcosm of the music is impossible to remember for a longer period of time, such as multiple sittings or playthroughs. Due to this, GMGEn is limited in that it is unable to allow the composer the opportunity to exploit motivic attachment techniques that can increase the immersive power of the music, and therefore the game, on the player.

The intermittent music engine combines elements from the branching and generative engines designed in Chapters One and Two respectively. Hybridisation allowed the long-term memorability of the branching music system to act alongside the reactive agility provided by the generative engine artificially improvising appropriate musical personalities. As weaknesses of the individual engines are removed by the combination the intermittency engine achieves the final degree of success this paper set out to achieve. The intermittency engine produces fully reactive music that has memorable microcosms allowing the composer to use musical attachment composition techniques. Coupled with the smooth blending that the generative side of the engine can produce, these compositional techniques create a more reactive dynamic music that can enhance a player's immersion in the game.

A recent article on the Imagine Games Network (IGN) website titled '2013: The Year of the Video Game Story' shows how quickly innovation is effecting the video-game world.[2] The article demonstrates many new innovative models for narrative within games being released across a single year. Games such as *The Stanley Parable*, where the narrative is seemingly unending; *Paper's Please*, where your position as a immigration inspector makes you the indirect, narrative arbiter; and

[2] L O'Brien, '2013: The Year of the Video Game Story', in *Imagine Games Network (IGN),* 26th October 2013. viewed 30th October 2013, http://uk.ign.com/articles/2013/10/27/2013-the-year-of-the-video-game-story

The Last Of Us, where narrative communication is as much made by the body language of the characters as by the dialogue.[3] These games all prove that innovation in narrative interaction is currently very rich. This paper has discussed several techniques pertaining to the improvement of dynamic music to accompany these narratives. This is not to illustrate any failing of past games or their musical design but to highlight a potential next step for the future of dynamic music for modern titles. The use of the *Final Fantasy* case study was due both to its lasting tradition and its impact on gaming as a whole. Games immerse new players each day, some of these players may have never experienced the same levels of immersion from other forms of media. Music's power to affect in these contexts gives the designers of audio engines and the composers writing music for them a great responsibility to further progress the dynamic nature of the score. I suggest that to whatever degree we are able to affect a listener with non-dynamic music in video games, a yet greater degree is obtainable when using a more versatile dynamic-music.

[3] O'Brien; and *The Stanley Parable,* PC Game, Galactic Café, 2013; and *Papers, Please*, PC Game, Lucas Pope, 2013; and *The Last of Us,* PlayStation 3/PlayStation 4 Game, Naughty Dog, Inc., Virtuos Ltd., 2013.

Bibliography

Ames, C ‘The Markov Process as a Compositional Model: A survey and Tutorial’, in *Leonardo*, vol. 22, no. 2, 1989, pp. 175-187.

Berndt, A ‘Musical Nonlinearity in Interactive Narrative Environment’, in *International Computer Music Conference Proceedings*, Montreal, Canada, 2009, pp. 355-358.

Berndt, A, Hartmann, K, Röber, N. & Masuch, M ‘Composition and Arrangement Techniques for Music in Interactive Immersive Environments’, in *Audio Mostly 2006 Conference Proceedings,* Audio Mostly, Pitea, Sweden, 2006, pp. 53-60.

Bessell, D ‘What's That Funny Noise? An Examination of the Role of Music in Cool Boarders 2, Alien Trilogy and Medievil 2’, in G King and T Krzywinska (eds) *Screenplay: Cinema/Videogames/Interfaces*, Wallflower, London and New York, 2002, pp. 136-144

Bodine, A ‘Androcentrism in prescriptive grammar: singular ‘they’, sex-indefinite ‘he’, and ‘he or she’’, in *Language in Society,* vol. 4, Cambridge University Press, Cambridge, 1975. pp. 129-145.

Bodman Rae, C *The Music Of Lutoslawski*, Omnibus Press, New York, 1999.

Boer, J *Game Audio Programming*, Hingham, Massachusetts, 2003.

Bruner, J *Actual Minds, Possible Worlds,* Harvard University Press, Massachusetts, 1986.

Chatfield, T *Fun Inc.*, Virgin Books, Great Britain, 2011

Childs IV, G. W *Creating Music and Sound for Games*, Thompson Course Technology, Boston, Massachusetts, 2007.

Clarke, E. F *Ways of Listening,* Oxford University Press, New York, 2005.

Collins, K ‘An Introduction to Procedural Audio in Video Games’, in *Contemporary Music Review*, *Special Issue on Algorithmic Generative Audio*, vol. 28, no. 1, 2009, pp. 5-15

Collins, K *From Pac-Man to Pop Music,* Ashgate, 2008.

Collins, K *Game Sound*, MIT Press, Massachusetts, 2008.

Collins. K ‘Grand Theft Audio?: Popular Music and Interactive Games’, in *Music and the Moving Image*, vol. 1, no. 1, University of Illinois Press, Illinois, 2008

Collins, N ‘Generative music and Laptop Performance’, in *Contemporary Music Review*, vol. 22, no. 4, 2003, pp. 67-79.

Collins, N ‘Infinite Length Pieces: A User’s Guide’, Proceedings of MAXIS, Sheffield, April, 2002.

Cooper, G and Meyer, L. B *The Rhythmic Structure of Music*, Phoenix Books, University of Chicago Press, 1963.

Cope, D ‘An Expert System for Computer-assisted Composition’, in *Computer Music Journal*, vol. 11, no. 4, 1987, pp.30-46.

Cope, D ‘Facing the Music: Perspectives on Machine-Composed Music’, in *Leonardo Music Journal,* vol. 9, 1999, pp. 79-87.

Cope, D ‘One Approach to musical intelligence’, in *IEEE Intelligent Systems*, vol 14, no. 3, 1999.

Crawford, G *Video Gamers*, Routledge, Oxon, 2012.

Dowling, J ‘Tonal structure and children’s early learning of music’, in *Generative Processes in Music*, J. A. Sloboda (ed.), Oxford University Press, Oxford, 1988.

Dyson, F *Sounding New Media : Immersion and Embodiment in the Arts and Culture*, Berkeley : University of California Press, London, California, 2009.

Eigenfeldt, A ‘Real-time Composition or Computer Improvisation? A composer’s search for intelligent tools in interactive computer music’, in *Electroacoustic Studies Network,* 2007.

Emmerson, S *The Language of Electroacoustic Music,* Macmillan Press Ltd, 1986.

Ferguson, C. J ‘Blazing Angels or Resident Evil? Can Violent Video Games Be a Force for Good?’, in *Review of General Psychology*, vol. 14, no. 2, 2010.

Fritsch, M ‘History of Video Game Music’, in P Moormann (ed.), *Music and Game: Perspectives on a Popular Alliance,* Springer, 2013, pp.11-41.

Greuter, S, Parker, J, Stewart, N, and Leach, G *Undiscovered Worlds – Towards a Framework for Real-Time Procedural World Generation,* Melbourne, 2003.

Jindra, M ‘Video Game Worlds – Review Essay’, in *Society*, vol. 44, no. 4, Transaction Publishers, New Jersey, 2007, pp. 67-73.

Juul, J *Half Real: Video Games between Real Rules and Fictional Worlds,* MIT Press, 2006.

Kaae, J ‘Theoretical approaches to composing dynamic music for video games’, in *From Pac-Man to Pop Music,* Ashgate, 2008.

Kalinak, K *A Very Short Introduction to Film Music,* Oxford University Press, Oxford, 2010.

Kalinak, K Introduction to *Settling the Score*, University of Wisconsin Press, Wisconsin, 1992

Knight, T *Mastering Sound and Music on the Atari ST*, Sybex Inc., Alameda, CA, USA, 1986

Konzack, L 'Rhetorics of Computer and Video Game Research', *The Players' Realm: Studies on the Culture of Video Games and Gaming*, JP Williams and JH Smith (eds), McFarland & Co Inc., North Carolina, 2007.

Lewis, G. E 'Interacting with Latter-Day Musical Automata', in *Contemporary Music Review,* vol. 18, no. 3, 1999

Miller, K *Playing Along, Youtube, and Virtual Performance,* Oxford University Press, Oxford, 2012.

Moseley, R 'Playing Games with Music (and Visa Versa): Ludomusicological Perspectives on Guitar Hero and Rock Band', in *Taking it to the Bridge*, N Cook and R Pettengill (eds), University of Michigan Press, Michigan, 2013, pp. 279-318.

Munday, R 'Music in Video Games', in *Music Sound and Multimedia*, J. Sexton (ed), Edinburgh University Press, Edinburgh, 2007.

Muscutt, K and Cope, D 'Composing with Algorithms: An Interview with David Cope', in *Computer Music Journal,* vol. 31, no.3, 2007, pp. 10-22.

Newman, J *Videogames,* Routledge, Oxon, 2004.

Newman, J and Simons, I *100 Videogames*, British Film Institute, London, 2007.

Pollack, I and Pickett, J. M 'Cocktail Party Effect', in *J. Acoust. Soc. Am.* vol. 29, no. 11, 1957

Przybylski, A, Rigby C, and Ryan. R 'A Motivational Model of Video Game Engagement', in *Review of General Psychology*, vol. 14, no. 2, 2010, pp. 154-166.

Salen, K *The Game Design Reader: A Rules of Play Anthology*, MIT Press, Massachusetts, 2006.

Schmidt, B *Designing Sound Tracks for Coin-op Games,* MPublishing, University of Michigan Library, 1989.

Scruton, R *Modern Philosophy – An Introduction and Survey*, Sinclair-Stevenson, London, 1994, pp. 589-90.

Sexton, J *Music Sound and Multimedia*, Edinburgh University Press, 2007.

Sherman, S, R 'Perils of the Princess: Gender and Genre in Video Games', in *Western Folklore*, vol. 56, no. 4, 1997, pp. 243-258.

Solis, G and Nettl, B *Musical Improvisation*, G Solis and B Nettl (eds), University of Illinois Press, Illinois, 2009.

Squire, K 'Open-Ended Video Games: A Model for Developing Learning for the Interactive Age.' In *The Ecology of Games: Connecting Youth, Games, and Learning.* Ed. Katie Salen, MIT Press, 2008, pp.167–198.

Whalen, Z 'Case Study: Film Music vs. Video-Game Music: The Case of Silent Hill', in *Music, Sound and Multimedia: From the Live to the Virtual*, J Sexton (ed.), Edinburgh University Press, Edinburgh, 2008, pp. 68-81.

Wharton, A and Collins, K 'Subjective Measures of the Influence of Music Customization on the Video Game Play Experience: A Pilot Study', in *The international journal of computer game research,* vol. 11, no. 2, 2011.

Wilde, M. D *Audio Programming for Interactive Games*, Focal Press, Oxford and Burlington, Massachusetts, 2004.

Winkler, T 'Defining relationships between computers and performers', in *Composing Interactive Music: Techniques and Ideas Using Max,* MIT Press, 1999.

Winkler, T 'Strategies for Interaction: Computer Music, Performance, and Multimedia' in the proceedings of the Connecticut College Symposium on Arts and Technology, 1995.

Wishart, T *Audible Design*, Orpheus the Pantomime Ltd., 1994.

Wong, W. L 'Serious Video Game Effectiveness', in *Proceedings of the Internation Conference on Advances in Computer Entertainment Technology*, ACM, New York, USA, 2007, pp. 49-55.

Zagel, J, Mateas, M, Fernandez-Vara, C, Hochhalter, B,& Lichti, N 'Towards an ontological language for game analysis', in *Proceedings of the 2005 Digital Games Research Association Conference (DiGRA),* Vancouver, Canada, 2005.

Zehnder S. M and Lipscomb, S. D 'Immersion in the Virtual Environment: The Effect of a Musical Score on the Video Gaming Experience', in *Journal of Physiological Anthropology and Applied Human Sciences*, 2004

Websites

Belinkie, M 'Video Game Music: Not Just Kids Stuff', in *Video Game Music*, 15 December 1999, viewed 11th February 2014, http://www.vgmusic.com/vgpaper.shtml

Berndt A & Hartmann. K 'Strategies for Narrative and Adaptive Game Scoring', in *Audio Mostly*, 2007, viewed 19th February 2014, http://wwwpub.zih.tu-dresden.de/~aberndt/publications/audioMostly07.pdf

‘Classic FM Conversation: Video game scores – what do you think?’, in *Classic FM*, 2013, viewed 19th February 2014, http://www.classicfm.com/hall-of-fame/2013/video-game-music-hall-fame-opinion/

Collins, K U. Önen and R. Stevens, ‘Designing an International Curriculum Guideline: Problems and Solutions’, in *Journal of Game Design and Development Education,* vol. 1, no. 1, 2011, viewed 19th February 2014, http://www.rit.edu/gccis/gameeducationjournal/

Dredge, S ‘Mobile Gaming’s Rise is a Headache for Sony and Nintendo’, in *The Guardian Apps Blog*, The Guardian, 25th September 2013, viewed 19th February 2014, http://www.theguardian.com/technology/appsblog/2013/sep/25/mobile-games-apple-google-sony-nintendo

‘Essential Facts about the Computer and Video Game Industry’, in *The Entertainment Software Association,* 2013, viewed 19th February 2014, www.theesa.com/facts/pdfs/ESA_EF_2013.pdf

Farley, R ‘Game.’ in *Media Culture: A Journal of Media and Culture,* vol. 3, no. 5, 2000, viewed 19th February 2014, http://journal.media-culture.org.au/0010/game.php

Farnell, A ‘An introduction to procedural audio and its application in computer games’, in *Obewannabe,* 2007. viewed 30th October 2013, http://obiwannabe.co.uk/html/papers/proc-audio/proc-audio.pdf

Juul, J ‘Games Telling stories – A brief note on games and narratives’, in *Game Studies*, vol. 1, no. 1, 2001, viewed 19th February 2014, http://www.gamestudies.org/0101/juul-gts/

Kamp, M ‘Musical Ecologies in Video Games’, in *Philosophy & Technology*, Springer, 2013, viewed 19th February 2014, http://link.springer.com/article/10.1007%2Fs13347-013-0113-z

Lynch, K ‘Confirmed: Grand Theft Auto Breaks 6 Sales World Records’, in *Guinness World* Records, 8th October 2013, viewed 19th February 2014, http://www.guinnessworldrecords.com/news/2013/10/confirmed-grand-theft-auto-breaks-six-sales-world-records-51900/

O’Brien, L ‘2013: The Year of the Video Game Story’, in *Imagine Games Network (IGN),* 26th October 2013. viewed 30th October 2013, http://uk.ign.com/articles/2013/10/27/2013-the-year-of-the-video-game-story

‘Download the Max Runtime’, in *Cycling 74’*, 2014, viewed 25th February 2014, http://cycling74.com/downloads/runtime/

Nayak, M ‘Factbox – A look at the $66 billion video-games industry’, in *Thomson Reuters*, 10th June 2013, viewed 19th February 2014, http://in.reuters.com/article/2013/06/10/gameshow-e-idINDEE9590DW20130610

Whalen, Z ‘Play Along – An Approach to Videogame Music’, in *Game Studies*, vol. 4, no. 1, 2004, viewed 17th February 2014, http://www.gamestudies.org/0401/whalen/

Video Games

Alpine Ski, Arcade Game, Taito, North America, 1982.

Bravely Default, Nintendo 3DS Game, Square Enix, Japan and North America, 2013.

The Binding of Isaac, PC Game, Headup Games, 2011.

Cthulhu Saves the World, PC Game, Zeboyd Games, 2010.

Dishonored, PC/Playstation 3/Xbox 360 Game, Bethesda Softworks, 2012.

The Elder Scrolls III: Morrowind, Xbox/PC Game, Bethesda Softworks and Zenimax Media Inc., 2002.

The Elder Scrolls IV: Oblivion, Xbox 360/Playstation 3/PC Game, 2K Games, Bethesda Softworks and Zenimax Media Inc., 2006.

The Elder Scrolls V: Skyrim, Xbox 360/ Playstation 3/PC Game, Bethesda Softworks and Zenimax Media Inc., 2011.

Evoland, PC Game, Shiro Games, 2013.

Fable, Xbox Game, Microsoft Studios and Feral Interactive, 2004.

Final Fantasy, NES Game, Square, Japan, 1987.

Final Fantasy VII, Playstation Game, Squaresoft, Japan, 1997.

Final Fantasy VIII, Playstation Game, Squaresoft, Japan, 1998.

Final Fantasy IX, Playstation Game, Squaresoft, Japan, 2000.

Final Fantasy XII, PlayStation 2 Game, Square Enix, Japan, 2006.

FTL: Faster Than Light, PC/iPad Game, Subset Games, 2012.

Grand Theft Auto V, Xbox 360/Playstation 3 Game, Rockstar Games and Take-Two Interactive Software, Inc., 2013.

Halo: Combat Evolved, Xbox Game, Microsoft Game Studios, 2001.

Jungle Hunt, Arcade Game, Taito, Japan, 1982.

The Last of Us, PlayStation 3/PlayStation 4 Game, Naughty Dog, Inc., Virtuos Ltd., 2013.

The Legend of Zelda, NES Game, Nintendo, Japan, 1986.

The Legend of Zelda: Ocarina of Time, Nintendo 64 Game, Nintendo, Japan, 1998.

The Legend of Zelda: Skyward Sword, Nintendo Wii Game, Nintendo, Japan, 2011.

Mass Effect, PC/Xbox 360/Playstation 3 Game, Microsoft Studios and Electronic Arts, 2007.

Mass Effect 2, PC/Xbox 360/Playstation 3 Game, Electronic Arts, 2010.

Mass Effect 3, PC/Xbox 360/Playstation 3/Wii U Game, Electronic Arts, 2012.

Medal of Honor, Playstation Game, Electronic Arts, 1999.

Papers, Please, PC Game, Lucas Pope, 2013.

Pokemon X/Y, Nintendo 3DS Game, Nintendo, 2013.

Pong, Arcade Game, Atari Inc., 1972.

Rogue Legacy, PC/Playstation 3 Game, Cellar Door Games, 2013.

Silent Hill, Playstation Game, Konami, 1999.

Space Invaders, Arcade Game, Taito, Japan, 1978.

Spacewar!, PDP-1 Game, Steve Russell, MIT, 1962.

The Stanley Parable, PC Game, Galactic Café, 2013.

Starbound, PC Game, Chucklefish Games, 2013.

Starcraft 2: Heart of the Swarm, PC Game, Blizzard Entertainment, 2013.

Super Mario Bros., NES Game, Nintendo, Japan, 1985.

Tennis for Two, Donner Model 30 analog computer Game, William Higginbotham, Brookhaven, 1958.

The Walking Dead, PC/Xbox 360/ Playstation 3/iOS Game, Telltale Games and Sony Computer Entertainment America, 2012.

APPENDIX A

Introduction

The beginning of this appendix will discuss general concepts surrounding the varied material submitted as part of this DPhil. First I will discuss how this variety brings strength to this thesis. I will then discuss the impetus for submitting a body of such mixed works and the justification for the inclusion of both notated and digital interactive or generative/automatic works. I will then discuss how the path of my critical writing lead to the exploration of the later digital works and how they act as experimentations into this exploration. Finally this appendix will discuss how these research interests pertain to each piece, how they relate to aesthetic and technical ideas and what questions may emerge from this discussion.

On a first glance of the body of work constituting this thesis, the reader could be forgiven for thinking the materials do not participate towards a combined whole. This body of work combines instrumental music, digital music, interactive music, generative/automatic music, classical music, popular music, popular media, scientific analysis, and musicological analysis. Without both my childhood (and adulthood) investment in the video-game art form and my paralleled musical training, I may not have had the capacity to activate any critical response to the current state of video-game music's reactivity for which the critical writing of this thesis explores. Further too, without my classical training and my digital interests, I would have been lacking possession of the full technical framework required to knit the classical composer to the computer programmer in an effort to obtain any answer existing between these two disciplines. This generalist perspective also affords a frame of reference that can be difficult to maintain over the course of nearly half a decade's work. It is on these grounds that I believe the breadth of this thesis to be its greatest strength. This thesis contributes not only to game studies (a youthful field in rapid expansion), but also to contemporary popular music, video-game music and theory, classical music and touches spaces in-between.

With this said, the combining factor is simple: this work came from a single mind over a period of nearly half a decade. Change was inevitable. The change of musical style and musical interests is inline with my exploration into the answers required of the critical writing.

From a time just prior to the beginning of this project, I had begun to feel limitations in the traditional notated score. I was increasingly becoming unable to efficiently communicate the real sounds I wished performers to produce simply through score. I was not comfortable using descriptive methods as this would usually end up taking the form of a sentence of words, or even brief paragraph, on the opening page of the score, which, in an admittedly small sample of experience, was often not read by the time-constrained performer. Similarly any non-standard prescriptive notation would confuse performers to a point where, I found, their enthusiasm for the work diminished. While it is likely I had interacted with an unrepresentative sample group of performers I found this suggestive that the score might be a fundamental tool in this miscommunication. I could see an argument that, for traditionally trained western musicians, the score enforces limitations on pitch, time, direct interactivity for the non-performer, and could decreases interest over multiple hearings. This last point—and to an extent the second to last—is of course excluded for music with improvised content. Further, none of these statements are attributable to all musics in any sense. These comments are simply my generalised thoughts at the time, and are those which gave rise to my changing musical focuses seen throughout the portfolio. Explorations of the limits of the score can be seen in the earliest two works in this portfolio.

In *Traversing the Centuries,* notational limitations existed where I needed to write explanatory notes in the preface to describe particular sounds. I have been very pleased with all performances of this work as the singers and pianist have often given appropriate time to finding an interpretation aligned with my own. The same cannot be said of my experience with *Abiogenesis*, which consisted of a rushed sight-read workshop. As many of the parts contained quarter-tones, this provided an unreachable point for performers seeing the score for the first time. From communicating with professional classical performers I'm aware that it is not common practice to include training for on-the-spot sight reading of non-standard pitches such as quarter-tones. In these instances performers would often practise these specific moments thoroughly before a particular rehearsal. That my quarter-tone-ridden score could not be performed by sight was incompatible with the standard way that the majority of classical music is currently performed. This therefore dissuaded me from writing for this medium. At the time I felt that the sight-limit of classical performance confines composers to producing works limited to standard notation. This is in no way a ubiquitous issue and is limited to those events where performers are simply not afforded appropriate time to devote to a new work. Therefore, I sought other forms of music making not contingent on this observation.

During the same early period of study I had begun learning to produce applications with MaxMSP. With a basic technical understanding of MaxMSP, I realised how it could be manipulated to explore questions I'd had about active music in video games. It was the exploration of these questions, pertaining to the critical writing, which culminate in the works present in my compositional portfolio submission. The work *Deus Est Machina* (NARRATIVE SYSTEMS) within the Interactive Portfolio uses the completed generative transitional system (GMGEn) both proposed and built as a working proof for the discussion in the critical writing portion of the submission. The *Generative Triptych of Percussive Music* (MUSIC SYSTEMS) exists as an exploration of a generative rhythmic engine created as part of the discussion within chapter two of the critical writing. This generative rhythmic engine becomes an integral part of the Game Music Generation Engine (GMGEn). The work *Starfields* (FLIGHT SYSTEMS) contributes to the thesis in a more general way. While *Starfields* (FLIGHT SYSTEMS) has a direct contribution to the thesis by way of it being an active experimentation on the branching music work of the first chapter of the critical writing, it too uses techniques garnered later, during my research of the second and third chapters. In particular, the technique of Harmonic-clouds, generative melodies and rhythms are both used within this work. In essence, *Starfields* (FLIGHT SYSTEMS) is a work that incorporates more of the thesis than any other piece yet in a much more canvassed, rather and specific, way. While still under the correct description of 'interactive experience', *Starfields* (FLIGHT SYSTEMS) is the closest to a 'video game' that any of the works submitted become. *Starfields* (FLIGHT SYSTEMS) contribution to the portfolio is twofold as it is not only a work directly relating to the research interests of the three chapters of the critical writing, but further its inclusion affords the portfolio a more experimentational overview of interactivity, which the standard written-only thesis may not allow.

In summary, the thesis combines my compositional movements and experimentations into different methods of music making within the classical music sphere. It spans the period of time where I began to embrace technology in my works, while still maintaining techniques gained from writing notated works earlier in my portfolio. *Traversing the Centuries* shows an attempt at temporally shifting music within a work of static composition (static here defined as antithesis of dynamic – i.e. music with a relatively constructed fixed temporal length): the music of one section gets repeated and spread across a greater extent of time as the work unfolds. *Abiogenesis* contains the seed of my ideas about harmonic-clouds and their treatment to aid in a dynamic music. *Starfields* (FLIGHT SYSTEMS) and the *Generative Triptych of Percussive Music* (MUSIC SYSTEMS) combine and explore techniques directly incorporated in the Game Music Generation Engine (GMGEn) (discussed in chapter 1 and 2 of the critical writing) prior to its completion. *Deus Est Machina* (NARRATIVE SYSTEMS) uses GMGEn as its musical production. This work is completely generated based on the work put forward in the body of critical writing.

The rest of this appendix will discuss the research interests relating to each work submitted as part of the portfolio, how they relate to aesthetic and technical ideas and what questions may emerge from this discussion. As a chronological order highlights the gradual explorations and discoveries most readily, this order will be used in the questioning of the works submitted. I will therefore begin from the earliest work written and end with the most recent.

Traversing the Centuries

In this work I attempted to explore the different viewpoints from which the text could be read. The text discusses a view of historic construction from descendants' point of reference, perhaps suggesting a present existing in a 'now' time. However, the text also firmly acknowledges that a present 'now' existed in the past for our ancestors, though it lies obscured from the direct view of their descendants. The text suggests that although this history may lie obscured that it still exerts a direct influence on our lives, whether we understand or accept this influence. The striking metaphor used by Anthony Goldhawk to shed light on the histories of our ancestors is that of human excavation or natural erosion.

I intended to blend the discovery of the ancestral histories and the present world together throughout the work, in essence gradating between the narrative of the present and the narrative of the past. As the text was visually presented in four couplets separated by a paragraph, these appeared as sedimentary layers, evoking the themes of excavation and erosion. Therefore I chose to build a work where the musical framework repeated four times. The piece therefore has a varied strophic form and becomes more complex as we move towards the close. Each time this framework was to be repeated more and more music would be discovered 'between' the notes that existed in the previous section. Therefore, the work becomes more complex during the course of the piece and highlights the viewpoint (or narrative) from the present perspective at the opening (with notes obscured from audibility), and the viewpoint from the past perspective at the close (with all of the notes fully excavated). In other words, the detail of the piece is therefore excavated or eroded to the surface much like the poem suggests. A musical motif I intended to bring out was an accelerating crescendo which in this piece trills, mostly, on the interval of a third and is cross threaded and layered in the piano lines, also occasionally in the vocal line, throughout the piece. Further to adding more pitches and augmenting gestures in the musical lines, another technique was used in the vocal line where I augmented the syllables into sub-syllabic components. For example the single syllabled word 'flesh' becomes a two-syllabled word 'fle-sh' in the music. In this example, the second syllable is achieved by blowing air through the teeth as a 'sh' sound.

While I feel the score communicates the ideas of the music well, one memory stands as contrast: the final word in the vocal part is intended to suggest a ghostly whisper from the ancestors spoken directly through the final sub-syllable of the narratively present singer. No performer of this work has correctly identified a vowel notated with a crossed note-head as identifying a whispered note. It seems pedantic to take issue here as my solution was simply to add the word 'whispered' above this note; however, I saw this solution as more of a 'work around' than an elegant answer to the miscommunication of my score.

This work was a commission from a close friend who required the particular instrumentation used. Having already begun working with MaxMSP at the time I had begun to think of ways in which the exploration of the multiple narratives of past and present could work from an interactive perspective. The example I had thought of at the time was that of a work where two pieces of music happened simultaneously but could not be heard simultaneously. In other words, one 'side' of the music could be switched off or on via some interactive trigger point. In this hypothetical piece the musics would represent, first, the narrative from the past's perspective, and second, the narrative from the present perspective. Taking the feature of contrast I used in this work—that of the obfuscation of musical detail—the two musical works would be identical in temporality. The music attaching to the detailed past of the ancestors would be rich and complex from start to finish. The music attaching to the obfuscated view from the present of the speaker of the poem, the descendant, would be the same version with many notes missing. An interactive element would be added where the user could switch between the musical layers and hear the full texture, thus interactively dipping into, or out of, the musics of the ancestors' (past) or the descendants' (present). The idea put forward here and thought up while working on

Traversing the Centuries would eventually feature in the music of *Starfields* (FLIGHT SYSTEMS) during the first MiniGame played in the first 5 minutes of the piece. In *Starfields* (FLIGHT SYSTEMS) two musical textures are played that contrast in rhythm and timbre but coalesce in harmony and textural complexity. While the version of this idea presented in *Starfields* (FLIGHT SYSTEMS) is different from the fully flowing multi-work I'd hypothesized when writing *Traversing the Centuries* they both have the same root.

As a work under the contemporary classical umbrella my experience with this work has been very positive. I've had a strong relationship with the singers and pianists who have performed the work to date. All performers involved have given generous levels of attention to master the challenges of the work. For this reason, in a large respect, this work acts almost as a scientific control against the rest of the portfolio. It stands at a point before I'd fully realised my interest and affinity for digital and interactive compositional methods. It was these fresh-seeming ideas of digital interaction, inspired through MaxMSP and the creation of this work that led me to explore these avenues further during the rest of my portfolio. My issues with notation further gave me impetus to move to working in different media, at least for a time. In summary, while this work is firmly in the domain of the non-dynamic/non-interactive, it provided the material on which my newfound digital outlook could express new creative output.

Abiogenesis

The narrative idea I'd had for *Abiogenesis* featured a creature emerging for the first time from its primordial state of non-living to become living. The chance to work with orchestral forces for *Abiogenesis* allowed me to experiment with a new method of composing. Prior to this work I'd spent a lot of my time composing at the piano to discover the pitches and harmonies I wished to use in my works. While I enjoyed the audible feedback given by the piano, for me, this came with the drawback of creating moments of inconsistencies on the page that did not match the 'score' in my head. My explanation for this is that in my exploration of many different versions of a particular musical moment, while composing at the piano, my psychological reaction to eventually having to choose only one of these moments was ultimately unsatisfying and thus lead to the disparity of the real single-score with my imagined multi-score. I would call these moments 'fluxing' as they never seemed to have completely phased into the score of the real. While an odd psychological phenomenon I'm still aware of it when composing at the piano to this day. For *Abiogenesis* I intended to completely remove the piano from my method and instead focus on just the creation of the real score.

As I was no longer bound to the piano to gain my harmonic language I instead used the technique of selecting pitch-class sets for various sections and subsections of the work. I found I naturally aimed towards this harmonic technique to consolidate harmonic consistency across small-scale events within the work. I used the overarching narrative of the work to govern my use of texture and horizontal movement; thus, I generally used the orchestra as one gestural sounding body near the opening and split the orchestra into constituent families further through the work. For me this was intended to represent the creature becoming more than just a single mass of non-complexity, and instead evolving to a complex multiply-organed being capable of breath and pulse.

At the time of writing this work I had been experimenting with creating generative harmonic consistency within a computer program. I had designed a small program that would randomly generate notes in a single musical line one after another at constrained random time intervals. Combining two or more of these generating musical-lines created harmonic interest. The question that arose was how to create a harmonic consistency between two or more randomly generating musical lines. The full discussion can be found in chapter 2 of the accompanying critical writing; however, my conclusion was to provide the computer with specific predesigned pitch-class sets which force the generative engine's decisions to be that of only concurrently pleasing options. I term these moments of potential harmony *harmonic-clouds*. These moments are macro-predictable but micro-indeterminate. This digital method of generative composition has similarity to the methods used to choose harmony and gesture in *Abiogenesis*.

With this new method of composition I succeeded in creating a work that I considered non-fluxing, based on the description of that term above. While this success may have brought me a sense of comfort in a new compositional method that allowed my music to be set by the score, this instead triggered the opposite response. I felt uncomfortable with the fact that some of the musical moments I'd discovered while composing at the piano were now being lost as the score required me to choose only one. *Abiogenesis* was therefore my final work of static (as opposed to reactive or dynamic) music that I wished to submit for this portfolio.

Generative Triptych of Percussive Music (MUSIC SYSTEMS)

The *Generative Triptych of Percussive Music (MUSIC SYSTEMS)* is an exploration of constrained randomness when dealing with structure on different levels of organisation. In the critical writing portion of this thesis I discuss how micro-level randomness can be grouped into larger sections with defined structure. This work acts as both a proof of concept for this idea and further as an exploration of the instrument I created to demonstrate this concept.

I used the ChucK coding language, developed for Princeton Laptop Orchestra (PLOrK), to design a class (a self contained object of code) which had a 'play' method (an executable block of code with a specific function). When triggered, the play method would create a single bar of generated percussive music based specifically on parameters chosen by the composer. The parameters the composer could pass to the method included: number of beats, number of times the bar is repeated and the tempo of the bar. These main parameters governed the general attributes of the bar. Further to these parameters a set of 'chance' parameters existed for each percussive instrument. To create a work out of this functionality I simply had to run consecutive calls to the play method. Each call would have a specifically composed set of parameters designed to allow a specific bar to fulfill a purpose in the greater whole of the piece. The three pieces in the *Generative Triptych* explore different avenues for working with this same play method. The mechanics that govern the 'chance' parameter are explained fully in both the text and video programme notes for this work.

The focus of research around this work was threefold. First, I wished to prove that composing at the bar level, in a digitally generative work, is directly relatable to the equivalent level in a notated work when employing constrained randomness. This was the reason for titling each piece as 'Sonata', owing to its status as one of the pinnacles (or pitfalls, depending on your perspective) of structure in the Western classical tradition. Constrained randomness established a predictability at the bar level that allowed for structured composition from above the bar level. Second, was to design a working method for generative rhythm, which I would later include in the

Game Music Generation Engine (GMGEn) for the critical writing. As has been mentioned above, GMGEn runs the music for *Deus Est Machina (NARRATIVE SYSTEMS)*. Third follows on from the discomfort with having to decide on particular versions of musical moments to go into a static work. This work was the first I'd created that resulted in multiple recognisable works of distinct music from the same score.

I believe this work is successful in creating variety over multiple playsthroughs while maintaining the essence making each composed work its own. Therefore multiple playthroughs become a feature of the work. It's important to note that it was not the intent, from my point of view with this work or any others in this portfolio, to create works of infinite interest. The term 'multiple playthroughs' here refers to an equivalent interest that might be achieved from multiple recordings of an acoustic work – greater than zero and yet non-infinite. In other words, though the number of variations a single score of these sonatas *could* produce is very large I do not see a listener's interest extending to this number of iterations. This is a psychological feature bound by humans' innate skill at pattern recognition. More research would be required to find a point at which these works no longer excite due to this biological feature.

Starfields (FLIGHT SYSTEMS)

The intent of this work was to create art music incorporating player control. This is a feature often used to varying degrees of extent and success in game audio. More personally, I wished to explore player-controlled music, in this case controlling the intensity of the music. The aim of this work was to put the listener in the position of bending some of the musical output of the piece. The player does not drastically change the overarching course of the piece, which remains largely fixed. As an extension of the transitional engines discussed in chapters one and two, Starfields sits at a point between the evolution of the first two chapters and the third. While chapter three of the critical writing discusses a hybrid engine of the first two chapters, Starfields is not an *intermittent music engine*. Starfields is a work of interactive dynamic music. The narrative of the work involves the player actively competing with the artificial intelligence PAMiLa, the antagonist of the digital portfolio meta-work.

A single slider on the right hand side of the work is used to explore the majority of the musical control the player has. Moving the slider up or down will at some moments increase or decrease the foreground texture, increase or decrease the background texture, trigger pitches on a digital harp, time-shift the work, among others. During the narrative take-over of the ships systems by the corrupted AI the player will also experience a loss of control, which I used in juxtaposition to the other forms of more direct control. The methods used to create this work are detailed more fully in the accompanying text and video programme notes.

Before completion of the work I ran a couple of user acceptance testing (UAT) sessions with friends and family. Several points were raised on obtaining user feedback. Two particular comments emerged which encouraged me to make direct changes to the work. In particular was the variety with which some players accepted the shield meter. When the shield meter falls below certain thresholds the heads up display flashes a warning and sounds an alarm to show that the shields need to be replenished. Most users commented that the REPLENISH SHIELDS button needed to be pressed too often. As a result I changed the shield decay rate and also the amount replenished by a single click; in other words, drastically decreasing the needed clicks of the REPLENISH SHIELDS button. I also made available a typed command that was previously hidden allowing the user to activate an ‘auto-shield’ feature removing the need for micromanagement of the shields altogether. Another interesting point regarding the shields was the fact that it appeared to split the demographic of players by age. Older players were very nervous about allowing the shields to deplete fully. This was seen, by these players, as a ‘losing’ condition and was therefore to be avoided. In these players the shields became a kernel of negative focus that I did not intend (and resulting in the change mentioned above). In contrast to this younger players saw the depletion of the shield meter as their first method of exploration. These players sought the answer to the question of what would happen if they take no action to seemingly preserve their avatar’s life by replenishing the shields. This is a typical phenomenon in video gaming where players will push the boundaries of their avatar’s mortality to enable a greater grasp of the rules of the game. In effect they force a lose condition to find out exactly where that lose condition stands in relation to the general universe of the game. I believe that this was the same phenomenon I was witnessing during the user acceptance testing.

The other issue that emerged later was in response to the visuals of the user interface. One commenter said they were “expecting Grand Theft Auto”. It is important to note that at the time of release *Grand Theft Auto V* (Rockstar Games 2013) was the most expensive game ever produced with a budget around 270 million USD. While I’m unsure as to the intent of the commenter’s words I felt this was an unfair comparison to make. However, this comment made me aware that if visuals exist at all then there will be a subset of players who expect extremely high standards from the visual content. The question taken from this is as to whether visuals built from MaxMSP can perform the visual role adequately. Many modern games have a ‘retro’ aesthetic and are just as popular as other AAA standard games with photo-realistic graphics. *Minecraft* (Mojang 2009) is an example illustrating this. The block-like, procedurally generated world of *Minecraft* does not seek to be photo-realistic to any arguable extent. The creator of *Minecraft*, Markus 'Notch' Persson, finds that the pursuit of graphical realism limits potential innovation achievable by games designers. The same conclusion can be drawn by looking at the styalised world of *The Legend of Zelda: Wind Waker*. This styalisation allowed the *Legend of Zelda* franchise to compete with competitors employing vastly superior graphically capable hardware, which was not available on Nintendo's Game Cube console. This entry in the *Legend of Zelda* franchise also visually appears less dated a decade on than its immediate successor *The Legend of Zelda: Twilight Princess* lending further support for styalisation. As realism isn't a graphical or artistic style, game visuals attempting to mimic reality will inevitably appear dated with the advance of visual rendering technology. It was this argument that grounded my decision towards styalisation over realism in the visuals used in *Starfields*. Further, this choice aligned with the technical, financial and temporal resources available to me.

This piece is the pinnacle of my working knowledge of MaxMSP at the time of submission on 16th May 2014. It went through a great deal of optimization to get it to run on my system with the degree of consistency it has now. This cannot compare to the normal quality assurance (QA) a software product with a dedicated studio team would receive in the commercial world and should be treated as thus; however, it is understandable that experiencing a piece of software in development can appear noticeably unfinished or unpolished when compared to software backed by the resources of a large technology company. This is owing to the fact that high levels of technological refinement have been normalised by today's technical society. Immersion into the interface was less problematic for users comfortable with gaming interfaces. The same is said for those users who are used to the everyday hacking necessities of digital laptop performance, of which this piece would be counted as a member.

Deus Est Machina (NARRATIVE SYSTEMS)

Deus Est Machina holds a twofold purpose within this portfolio. It exists as a proof of concept for the technical architecture set out for a generative transitioning system in the body of critical writing. It also exists as an compositional exploration of the instrument created. As this instrument was designed to generate music for indeterminately changing and indeterminately temporal music I designed a narrative work mimicking these scenarios, which also closely reflects the situation for which the instrument is functionally designed. Narrative 'areas' or 'zones' are set up within the personality structure of the music. These have been composed within the GMGEn instrument and amount to a detailed configuration of the way each of the personalities creates music. The logic built into GMGEn allows these zones of music (which I term *musical personalities* in the critical writing) to be transitioned to, or from, in a relatively smooth fashion. While the piece needed to incorporate all of these elements it also needed to fit into the global meta-narrative of the portfolio of compositions. The way I chose to knit these two scales together was have PAMiLa, the fictional AI, narrate the story to the listener. I wanted this piece to be more about the character of PAMiLa than of the actual narrative PAMiLa produced. The idea was to show that the personality of the artificial intelligence was more complex than one may superficially think, to the extent that PAMiLa presents human characteristics. For example, in the tutorial for the portfolio, PAMiLa shows its 'god' complex by referring to humans simply as 'bios', as well as an arrogant consideration of its own work. The confirmation of this is found in *Deus Est Machina* (God *Is* the Machine) work, for which the title was carefully chosen. The computer is in control of your path, god-like, and yet also believes it is god-like compared to a human. This resulted in the first iteration of the work which included a monotone synthesized voice which spoke the text using the basic voice synthesizer found on all commercial computer operating systems and is famously used by the physicist Stephen Hawking.

The work fully utilises GMGEn's functionality in generating static material and triggered transitional material. I also believe its success at invoking the god in the Machine, which can be read on several levels. In a self-referential way PAMiLa *is* the titular machine of the story while also being that story's fictional creator, its 'god'. This functions within itself (the piece: *Deus Est Machina*) and without by reinforcing the meta-narrative of PAMiLa as flawed storyteller to the onboard pilot (you, the listener). The original version submitted had the synthesized voice very high in the mix, which was in an attempt to focus the listener on the story. However, this created a larger divide between the music and the locations evoked in the text than was intended. This has resulted in the change to this work now submitted.

The major change I made to this piece was inspired directly from the video games of my youth. Before voice synthesis and voice acting was common in video games the music would play alongside text that would be read by the player. As GMGEn's genesis was inspired by video games from this era, it seemed to invite a natural solution to the problems presented by the inclusion of voice synthesis. In the current version of the work I have stripped the voice away and left the music play over unread text. This text can now be read, or reread, by the user at any speed. This creates a better atmosphere and greater temporal space in which the musical generation of GMGEn can thrive. It further removes the layer of separation previously acting as a barrier to the immersion of the listener. I also added more controls so that the listener can choose which locations to explore in what is now an interactive text adventure instead of a story told, and controlled, wholly by PAMiLa. PAMiLa remains in control of the protagonist’s fate and for what that character is trying to achieve on their quest; however, relinquishing some of PAMiLa’s influence over *Deus Est Machina* in this version of the work created a stronger and more interactive piece that exploits GMGEn further than the original version did.

APPENDIX B

Table 1 : Final Fantasy VII, VIII and IX Overworld and Battle-sequence – Visual style separation.

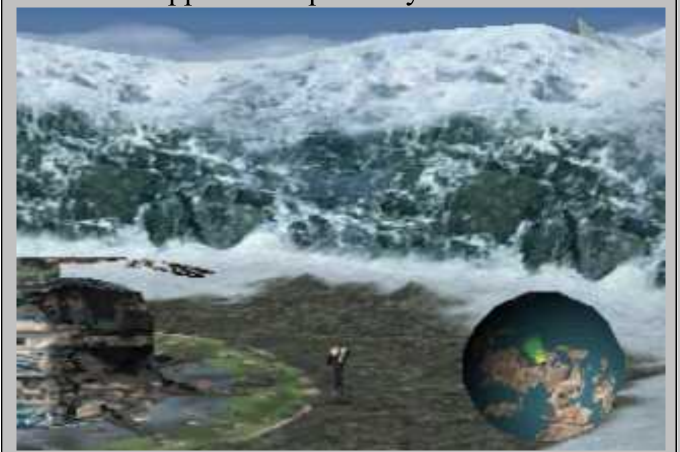
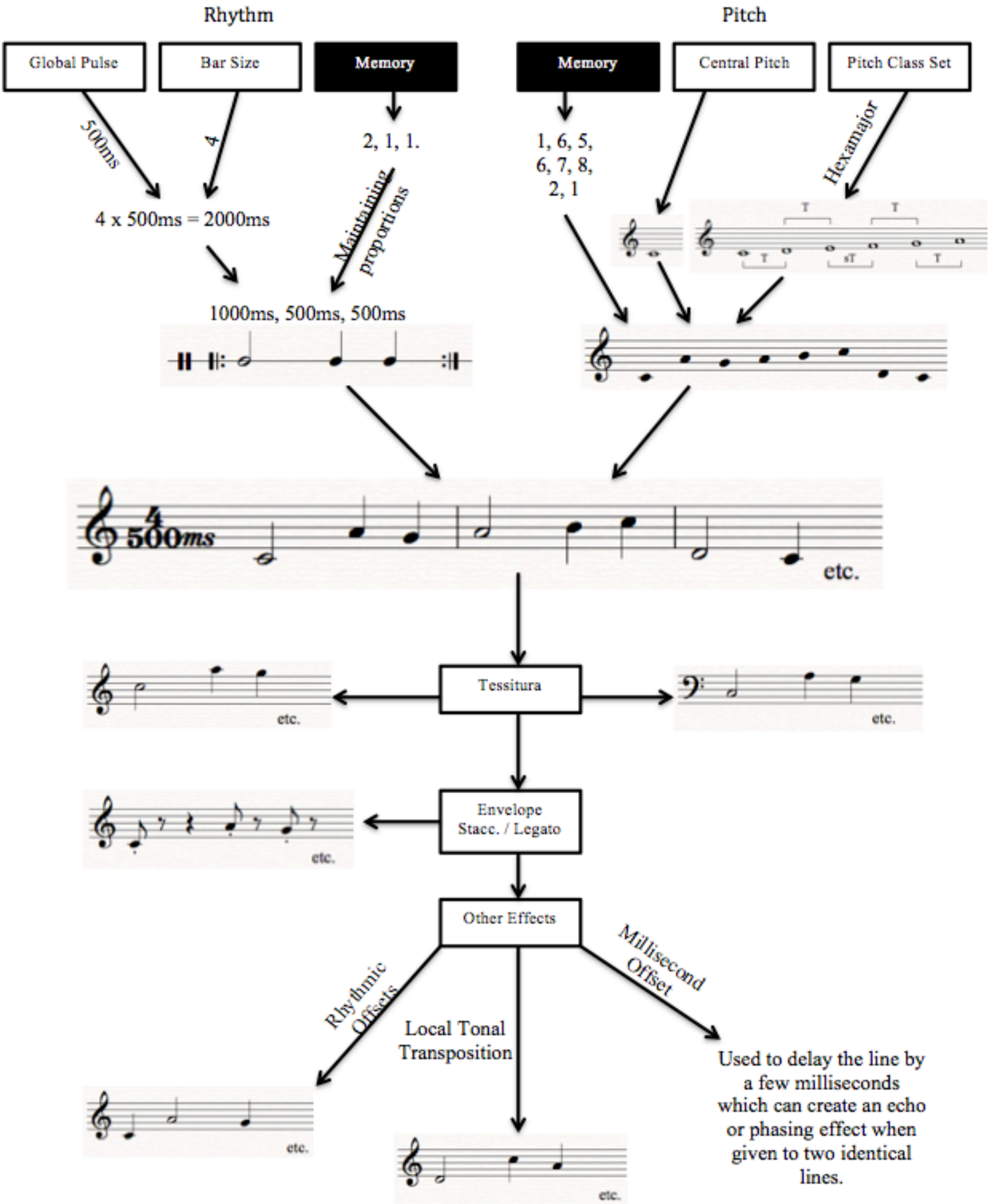
FFVII Overworld – Cloud solo. Other characters appear if required by the narrative. 	**FFVII Battle** – Cloud and allies square off against enemies. 
FFVIII Overworld – Squall solo. Other characters appear if required by the narrative. 	**FFVIII Battle** – Squall and allies square off against enemies. 
FFIX Overworld – Zidane solo. Other characters appear if required by the narrative. 	**FFIX Battle** – Zidane and allies square off against enemies. 

Table 2 — Single Line Generation Flow Logic



APPENDIX C

This appendix highlights the technical solutions I found to handling the musical transitions in response to a trigger. In the Demonstrations_Application Tabs requested by the examiners (1, 2 and 5) these triggers are all made by the click of a mouse on a particular interface button. This appendix will discuss the MaxMSP patches for each Tab (1, 2 and 5) in separate sections. As the MaxMSP patches (particularly for Tab 5) are complex, I have designed this document to explain and annotate significant points both from a technical and practical perspective. While my images and explanations give the full information required by the examiners, I have also added the location of these screenshots to the example titles should the examiners wish to explore these patches further. In the paper (physical) copy of this appendix all images will be grouped together at the end of the main body of text. On the web copy of this appendix the images will appear in line with the text.

The unlockable version of this MaxProject can be found at this location on the accompanying DVD in the root folder: "Unlockable Version of Demonstrations_Application>DemoPatches.maxproj".

Open the .maxproj file titled DemoPatches.maxproj in MaxMSP 6. This folder hierarchy needed to be maintained for Max to access all data correctly.

The patch logic that does all of the heavy work for tabs 1, 2 and 5 is contained in the small subpatches labeled in Demonstrations_Application Figure 1 below. This level of sub patch is not important to this discussion and merely shows the location of the bulk of the working logic in relation to the points being made by my critical writing; therefore, this level of zoom was used and the full patcher, which deals with other logic external to that requested by the examiners, has been omitted from this figure. The reader is advised to follow along with the unlockable version of the patch if further detail is required.

See Demonstrations_Application Figure 1

Tab 1 – Branching Example 1 Subpatch

This subpatch deals with the work logic for Tab 1, which plays an introduction to the battle music when the START button is pressed. After a few seconds this introduction music moves into the main looping phase of the battle music. This looping phase continues until the WIN button is pressed. Upon pressing the WIN button the patch will begin an introduction to the victory music. After a few seconds this victory introduction music will move into the main looping phase of the victory music. This will then continue until the STOP button is pressed. Note: the STOP button can be pressed to halt the music during either of the looping phases. The subpatcher for this logic is displayed in Demonstrations_Application Figure 2 below. This is conceptually explained further in chapter 1 of the critical writing.

See Demonstrations_Application Figure 2

Inlets in annotation 1, 2 and 3 take the input from the START, WIN, and STOP buttons from the top-level patcher and pass this into the logic below. When the patch is loaded, the messages contained in annotation 4 are passed to the sfplay~ objects in annotation 5, 6, 7, and 8. These sfplay~ objects pass all of their sound out of the outlet in annotation 9, which is linked to the digital to analogue converter (dac) above in the top-level patcher. When a STOP message comes in, the patcher is set to its initial state. This means that all gates are set to their starting positions. When a

START message comes in, the sfplay~ object in annotation 5 is triggered to start playing the bttleIntro.wav file. Once this object has finished playing, it sends a trigger out of its right outlet that starts the next sfplay~ object in annotation 6. This therefore plays the bttlLoop.wav. When this sfplay~ object finishes playing it triggers an output from its right outlet. If the WIN button has not been pressed this will just trigger itself, therefore playing another loop of the bttlLoop.wav file. However, if the WIN button has been pressed the sfplay~ object in annotation 6 will be sent a stop message and the sfplay~ object in annotation 7 will be sent a start message and the vicIntro.wav sound file will play. Once the WIN button has been pressed the play logic that has been described here continues but instead of running though the sfplay~ objects in annotation 5 and 6 the messages will run for the sfplay~ objects in annotation 7 and 8. When the user hits STOP, all soundfiles will be sent a stop message and all gates will return to their original position. Other logic that is not included in this discussion pertains to stopping the user from being able to press START twice in a row, which would result in too many ‘on’ messages getting sent through the system.

Tab 2 – Branching Example 2 Subpatch

This subpatch deals with the work logic for Tab 2, which plays a pixelated version of a piece resembling music from the *Final Fantasy* series. Separating the music into individual pixels affords the program the knowledge of its progress through the work. This can then be used for dynamically reactive purposes in video games. This patch is a working version of a branching music engine, which shows that from any point in the main body of music, an appropriate ending music can be selected. This ending was composed to aesthetically complement the exact bar in the music where the ‘win’ switch was made and therefore where the win condition was achieved by the player of the game. This results in a complementing artistic convergence of the change in game-state with an aesthetically consistent change in musical-state.

See Demonstrations_Application Figure 3

Again in this patch, the input is taken from the main top-level patcher through the inlets in annotation 1, passed through the system and results in signal output which is sent out of the outlet in annotation 2.

Upon pressing the START button the intro sfplay~ objects in annotation 3 will be triggered in sequence and will play pixel 1 and 2 of the introduction music (intro1.wav and intro2.wav). Once this is complete the final sfplay~ object of the section will send a start signal to the first sfplay~ object in the main looping section of the work. sfplay~ objects making up this section can be found in annotation 4. These sfplay~ objects are triggered one-by-one in sequence from the left side of the patch to the right until the final one in the sequence is played. When the final sfplay~ object in this section is finished playing the first sfplay~ object in this section will be retriggered and the cycle will continue again. Coupled with the start of each of these musical pixels is the opening of individual gates linking each individual pixel with the appropriate ending found in the ‘ends’ subpatch in annotation 5. When the STOP trigger is hit by the user the gate leading to the correct ending will be open and therefore the next sfplay~ object to be given a start message will be the sfplay~ object loaded with an ending sound file compositionally linked to the previously playing musical pixel. Other logic that is not included in this discussion pertains to stopping the user from being able to press START twice in a row, which would result in too many on messages getting sent through the system.

Tab 5 – Game Music Generation Engine (GMGEn)

This subpatch deals with the work logic for Tab 5. This tab houses a scaled down user interface that acts as an interactive method mimicking the kind of transitional triggers that may occur in a video game scenario. At a high level, upon clicking one of the buttons corresponding to a musical personality, the system will play music according to preconfigured (composed) rules that align with that particular personality. This state will cycle through some further preconfigured logic as part of the same personality until a new state is triggered by the user. When a new state is triggered, a transitional period of music occurs. During this transitional period, musical elements of the first are switched to the role they will perform in the new musical personality. As this happens in a semi random way for each musical element, and further that this happens asynchronously for each element, the effect is that of a gradual stylistic shift from the old personality to the new. Each musical personality can therefore be attached to a specific game state and can be used for dynamically reactive purposes. This patch is a working version of a generative state-based musical engine that is capable of transitioning between different pre configured states given a single trigger. This results in a complementing artistic convergence of the change in game-state with the aesthetically consistent change in musical-state.

As the GMGEn patch logic is complex I will split discussion into three separate subsections. These sections will follow the three highlighted areas in Demonstrations_Application Figure 4 below.

See Demonstrations_Application Figure 4

At this level of zoom only general areas of the GMGEn patcher can be made out. This screenshot is included to both give an idea of the scale of this patcher and to highlight areas of the patch that are working towards similar purposes. This can roughly be summed to the three areas shown in figure 4. Each area is explained more fully in the sections below. Annotation 1 holds the main personality switches, which trigger a change to a new preconfigured personality. Highlighted in annotation 2 is the logic controlling large-scale features of the musical personality, for example: tempo. Highlighted in annotation 3 is one of 6 musical ‘lines’. There are six identical areas of the GMGEn patch acting as individual lines. A line is the equivalent of a single instrument in an acoustic work. In effect GMGEn is therefore a digital sextet of these musical lines. I will now discuss each of the subsections highlighted by annotations 1 – 3 in figure 4 separately.

Subsection 1

This subsection of the main GMGEn patcher controls the logic that responds to the mouseclick. This sets up the master parameters (discussed below) and the musical lines to receive the correct values, which correspond to the preconfigured musical personalities. This logic is the same for each personality used in the Demonstrations_Application.

See Demonstrations_Application Figure 5

Annotation 1 shows the ignore click messages which are sent to buttons in the 'on' position to stop them from being clicked off. This functionality needed to be built to accommodate the transitioning feature of the GMGEn system. I designed the system so that only a change to a new personality would signal a tuning 'off' of the currently active personality. Annotation 2 shows the individual personality switches. When selected to 'on', the trigger object shown in annotation 3, activates the new parameter set for the new personality and passes this to the subsections dealing with the transition phase (this information is passed to both subsection 2 and subsection 3). Annotation 4 shows part of this transfer. Each of the trigger objects in annotation 3 open this gate to a specific value, for example ESCAPE sends a 3 and therefore triggers the '20' value to be passed out of the gate. This '20' value corresponds to a more detailed collection of memory values for the 'escape' personality and will be picked up and used to recall these memory values further into the patch logic. This is explained below.

Subsection 2

This subsection of the main GMGEn patcher controls the logic governing high-level features of the current personality. These musical elements are the most important in creating a consistent personality that can affect the listener in a particular way. While these high level parameters are complemented by the lower level features that are shown in the 3rd subsection, the greatest contribution to the style of the personality comes from these high-level (or *master*) parameters. To configure a personality I needed to save off the settings of the parameters into a collection of data that could be recalled by the program to perform the same logical operations again, and thus generate music with the same feel again. The collection of data that encompasses a single personality is made up of many smaller subsets, which I called *subPresets*. Each subPreset is a snapshot of the positions of all variables of the program at once. Creating groups of complementing subPresets allows for greater interest within the framework of a single personality. Therefore, in this configuration of GMGEn, each musical personality has 8 subPresets that make up the whole. Musically the function of the subPresets is to provide different musical interest within the same personality. For example, subPreset_1 could be a short portion of a 3-part choral. Using the second subPreset allows the addition of a short portion of a 4-part section to the same choral. Cycling through these different subPresets allows the composition of a multi-sectioned personality affording the personality greater artistic scope and ability to maintain a listener's aesthetic interest over an enduring period of time.

See Demonstrations_Application Figure 6

Annotation 1 and 2 show two features governing the bar. The parameter in annotation 1 controls the smallest note duration in milliseconds and the parameter in annotation 2 controls how many beats there are in a bar. These values are used in the 3rd subsection when generating rhythm for and single bar or section of repeated bars. The parameter in annotation 3 governs the way harmonic modulations can occur. These modulations occur at specific intervals governed by the parameter in annotation 5. Shown in this example, the number 4 would here trigger a harmonic modulation to a new 'tonic' note after 4 repeats of a phrase. I called this note the *Center Pitch* as it doesn't act like a tonic in the traditional sense. In summary, the parameter in annotation 3 governs the constrained random choice of what the next center pitch can be based on this setting.

As was mentioned above, each musical personality is made up of smaller subPresets. These subPresets repeat until the next subPreset is triggered. The point at which a new subPreset is triggered is controlled by the parameter in annotation 4 and the order in which they are triggered by the method selected on the parameter shown in annotation 6. This parameter can be either CHAOTIC or CYCLIC. CHAOTIC moves through the subPresets in a random order (a musical analog to mobile form), while CYCLIC moves through the subPresets from 1 to 8 then back to 1 again until the personality is switched. Both of these modes are useful for different styles of musical personalities.

The coll objects in annotation 7 and 8 hold the data relating to the master parameters. A coll object is a collection of data stored for use later. The coll object in annotation 7 is used to recall the data snapshot and then populate the parameter boxes with this data. The coll object in annotation 8 is used when saving new data snapshots for a new subPreset. This small section of logic was used in my original configuration of the subPresets and could be used again to save new versions. This save and recall method is used throughout GMGEn to create and recall snapshots of data.

Subsection 3

GMGEn is a generative instrument that uses six musical ‘lines’ in combination to create its sound. The multitude of possible relationships between each of the musical lines allows for an abundance of generated musical content. This subsection of the main GMGEn patcher controls the logic governing the low-level features of one of the six lines for the current personality. This logic is repeated for each musical line. The set up of these musical elements is key to creating the musical detail that can be generated by the current personality. These low-level features complement the high-level features (master parameters) that are discussed above in the second subsection. The same save and recall mechanism has been used here to generate music with the same feel.

See Demonstrations_Application Figure 7

This subsection is largely split into two sections shown in annotation 1. One governs control over the rhythmic detail and the other controls melodic detail. There are two subpatchers (genRhythm1 and genMelody1), which take into account these settings and output a list of pitches to the dac to be played. Those governing rhythm include the following: bar mode, offset fraction, offset integer, aug/dim type, aug/dim speed. Those governing melody include the following: timbre, harmonic cloud, range type, envelope type, transposition mode, transposition value, notes looped. There are also some parameters that affect the line more generally, including those governing the mix level or whether the line is switched on or off.

I will now give a brief explanation of the parameters for both rhythm and melody.

Bar Mode has two settings, Bar or Free. The ‘Bar’ setting uses the master value for the smallest note length and number of beats in the bar to calculate a maximum value for the bar. For example: with 100ms as the smallest note length and 8 beats in the bar, this equals a total bar size of 800ms. The ‘bar’ mode setting will now group the randomized rhythms generated inside this 800ms boundary in segments using multiples of 100ms. This is useful when configuring lines for rhythmic texture. Using the same example master variables, the ‘free’ mode setting will still group the generated rhythms into multiples of 100ms but will not necessarily group them rigidly inside the bar construct. This is useful for attaching to lines that will perform a more soloistic role within the subPreset.

Offset fraction and offset integer are settings that apply a delay to the rhythm, pushing the rhythms out (late) by certain amounts. The offset integer setting pushes the rhythm out by integer multiples of the smallest note length in milliseconds. The offset fraction is a slider, which pushes the rhythm out by values between 0 and 1. A value of zero applies no delay to the rhythm. A value of 1 applies a one beat delay to the rhythm. Giving two musical lines the same melodic and rhythmic content (therefore making them play identical material) while applying a slight rhythmic offset to one of the two lines can produce effects such as echo or Reichian phasing.

Aug/dim type and aug/dim speed are used to create augmentations or diminutions of the generated rhythmic values based on the generated rhythmic criteria. Aug/dim type has two settings: compound and simple. These have the effect of changing any augmentation or diminution of the generated rhythmic values into either simple or complex subdivisions or multiples based on the aug/dim speed value. The aug/dim speed has 5 settings Normal, Fast, Slow, Very Fast, and Very Slow. The ‘normal’ setting applies no adjustment to the rhythm. ‘Fast’ and ‘Very Fast’ apply a diminution while ‘Slow’ and ‘Very Slow’ apply augmentations to the generated rhythmic values. This is useful for applying rhythmic interest to the same melodic passages, which may have been used previously, allowing them to be compositionally developed inside a generative system.

On the melodic side of the generative settings the Timbre matrix can mix together different proportions of four signal types to produce a variety of different electronic timbral qualities. This is a 2 dimensional slider that is saved as different values of x and y. This feature is used to create different timbral families to help accentuate a particular role the line may be performing. Coupled with this is the envelope type feature, which has several settings: sustained, legato, staccato and swell. These envelope types add further variety to the timbre matrix.

The Harmonic Cloud feature is explained in detail in the critical writing portion of the thesis. In short, this amounts to different collections of pitches organized together to perform a particular role in the creation of the mood of the personality.

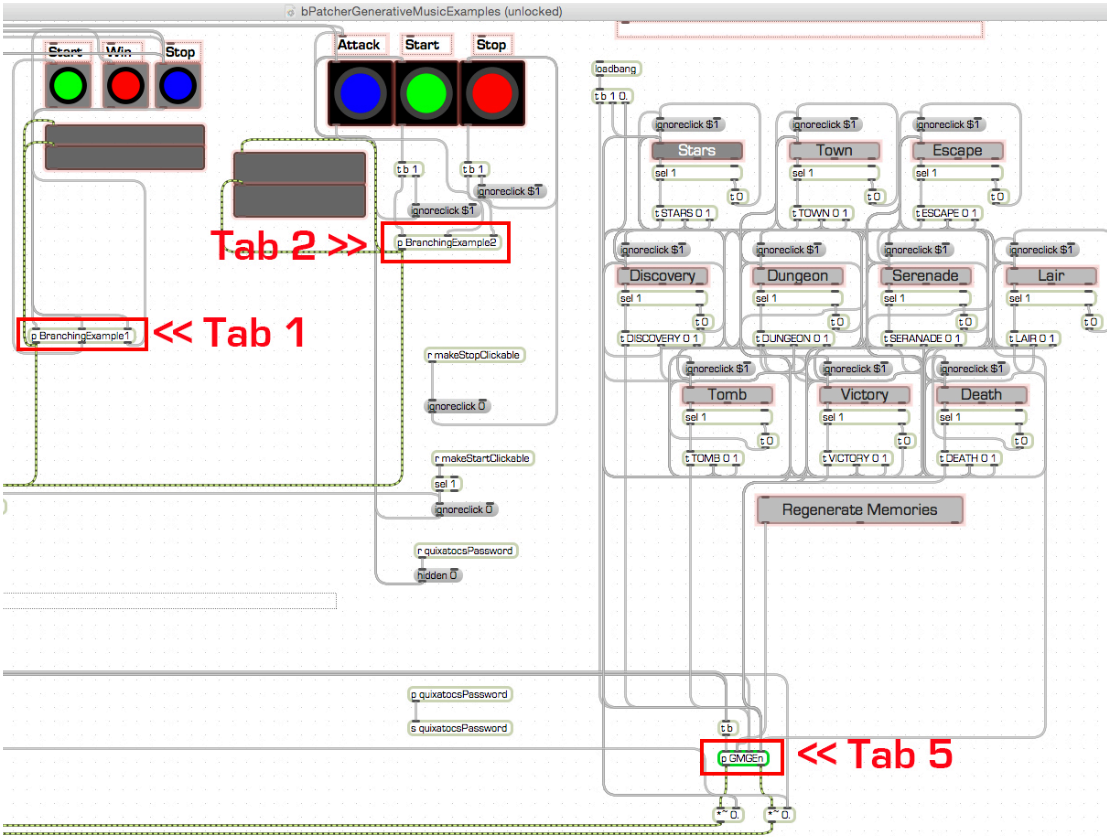
Range type shifts the musical line up or down octaves based on the voice role it will perform. Therefore, there are typical choices between bass, baritone, alto, soprano and hyper soprano. With alto as the neutral position, baritone and bass are shifted down one and two octaves respectively while soprano and hyper soprano are shifted up one and two octaves respectively from the generated midi pitch values.

The transposition mode and transposition value settings govern a shift in pitch of the generated melody line. This can be used to create chordal textures by combining two (or more) lines playing the same material with differing transposition values. The transposition value can be set higher or lower than the unaffected line to create many different chordal combinations. The transposition mode has two settings: real or tonal. Real transposes up or down the number of semitones shown in the transposition, while Tonal transposes the line up or down based on the relative tonal positions setup in the harmonic cloud. This is analogous to the terms used in the analysis of real vs tonal answers to musical phrases in renaissance and baroque polyphonic music.

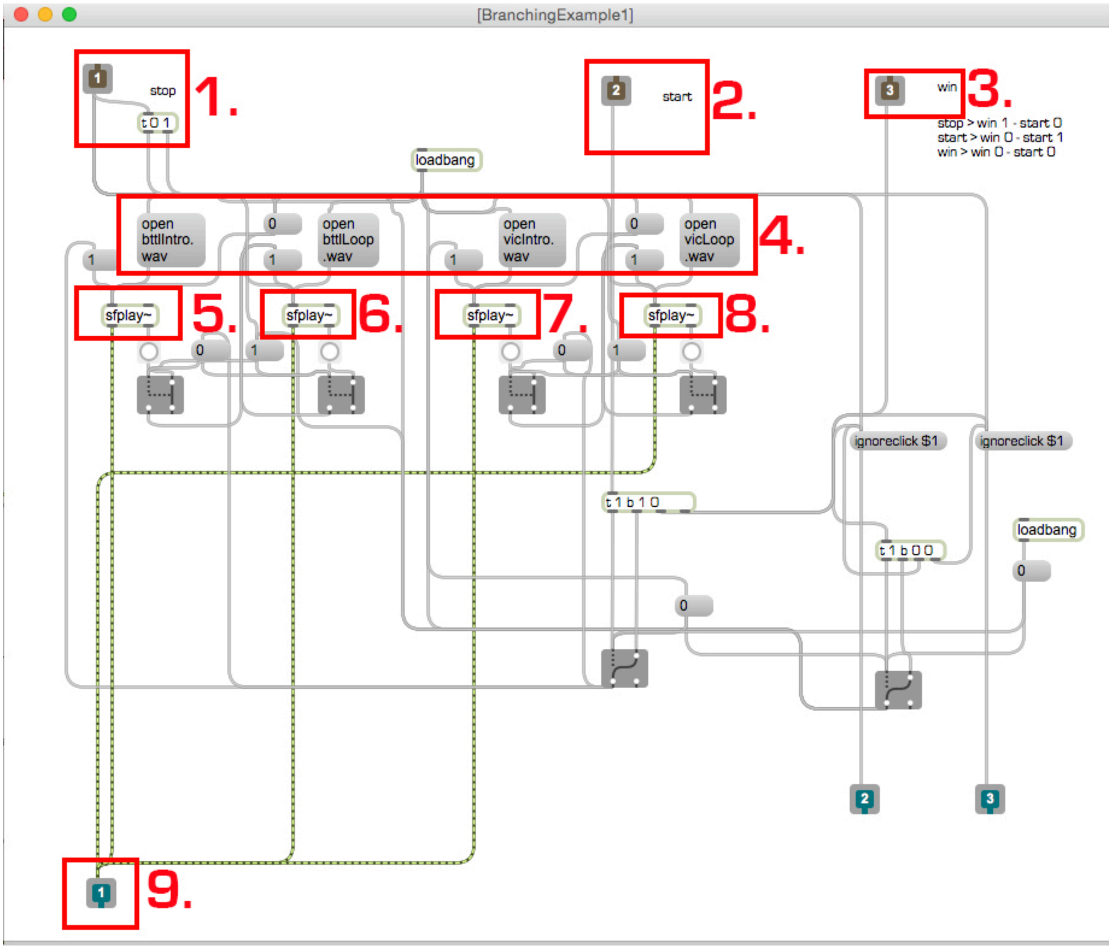
The notes looped parameter is used to set how far through the generated list of pitches the generator plays. For example, if the generated list contains the values A, B, C and D, the pitches will cycle through in order from pitch 1 (A) to pitch 2 (B) until reaching pitch 4 (D) and then will begin again. Inputting a 3 value into the notes looped parameter will truncate the pitch at index position 4 and will instead loop through A, B, C, A, B, C, A…etc. This is useful for creating rhythmic arpeggiaic lines, oscillating lines, or for creating lines of single repeated notes, which, for example, might be useful for an idiomatic bass guitar-style.

Both annotation 2 and 3 show the save and recall part of the patch logic. The save logic is the same as that described in subsection 2, however, the recall logic has a fundamental difference that allows GMGEn to trigger stylistic transitions between two different personalities. When a new set of parameters is recalled by a personality change the new values are passed through this cascade logic. This logic withholds the new parameter setting for a time before passing it out to the rest of the system. It withholds each new parameter setting for a different amount of time and thus allows some of the settings of the previous personality to merge with that of the new personality. This cascading feature is the mechanism allowing the stylistic transition from one personality to another within GMGEn.

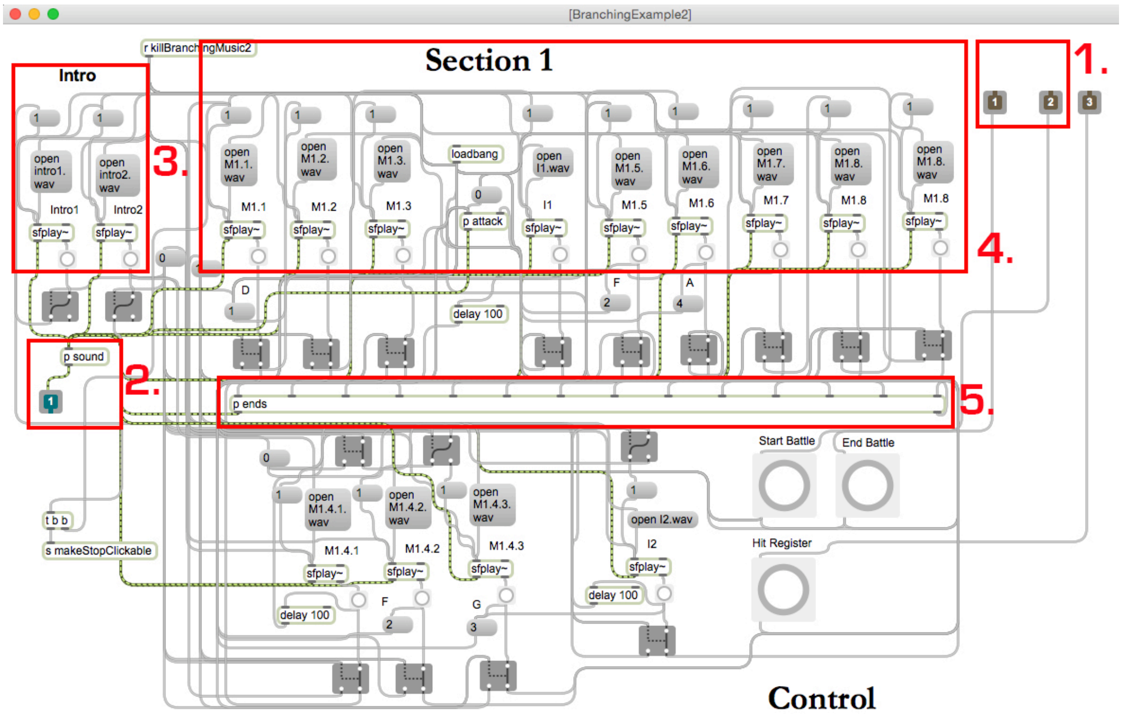
Demonstrations_Application Figure 1 – Top-level patcher of the bPatcherGenerativeMusicExamples.maxpat file



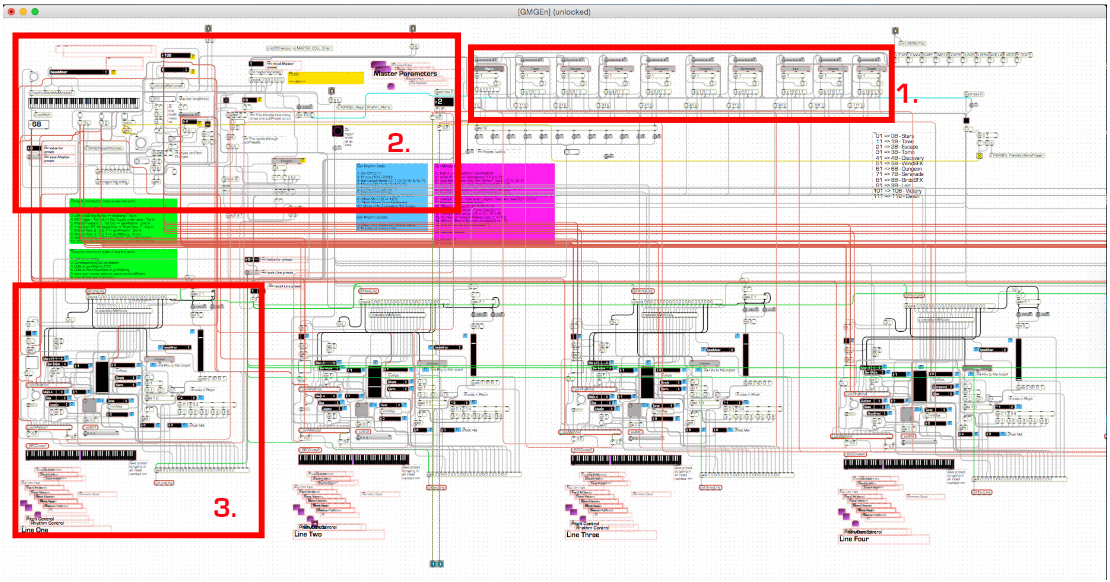
Demonstrations_Application Figure 2 – Found in the BranchingExample1 subpatcher in the bPatcherGenerativeMusicExamples.maxpat file top-level patcher.



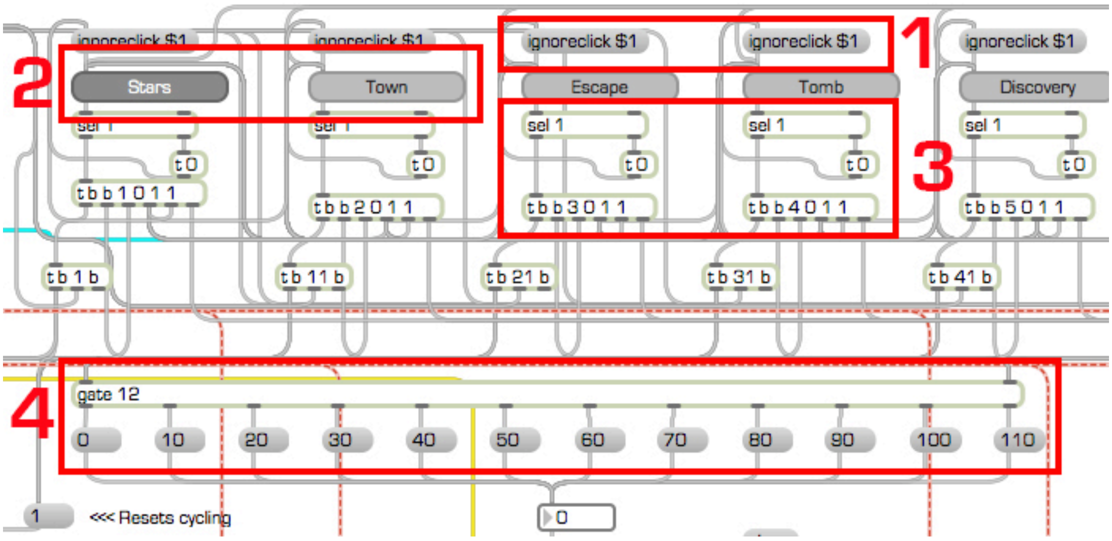
Demonstrations_Application Figure 3 – Found in the BranchingExample2 subpatcher in the bPatcherGenerativeMusicExamples.maxpat file top-level patcher.



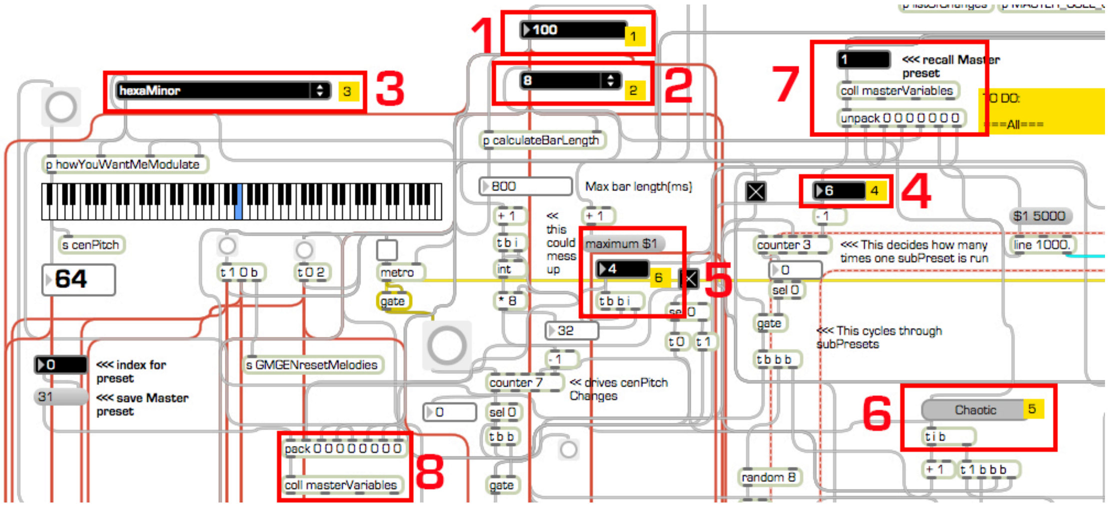
Demonstrations_Application Figure 4 – General overview of the main subpatcher for GMGEn found in the GMGEn subpatcher in bPatcherGenerativeMusicExamples.maxpat file top-level patcher.



Demonstrations_Application Figure 5 – Subsection 1 of the main subpatcher for GMGEn found in the GMGEn subpatcher in bPatcherGenerativeMusicExamples.maxpat file top-level patcher.



Demonstrations_Application Figure 6 – Subsection 2 of the main subpatcher for GMGEn found in the GMGEn subpatcher in bPatcherGenerativeMusicExamples.maxpat file top-level patcher.



Demonstrations_Application Figure 7 – Subsection 3 of the main subpatcher for GMGEn found in the GMGEn subpatcher in bPatcherGenerativeMusicExamples.maxpat file top-level patcher.