|  |
| --- |
| The Name of Your Project  Andrew Milne  BSc Computer Games Technology, 2019 |

|  |
| --- |
| School of Design and Informatics  Abertay University |

# i - Table of Contents

[i - Table of Contents i](#_Toc2689436)

[ii - Table of Figures ii](#_Toc2689437)

[iii - Table of Tables iii](#_Toc2689438)

[iv - Acknowledgements iv](#_Toc2689439)

[v - Abstract v](#_Toc2689440)

[vi - Abbreviations, Symbols and Notation vi](#_Toc2689441)

[1 - Introduction 1](#_Toc2689442)

[1.1 - The problem with games 1](#_Toc2689443)

[1.2 - Conventional Methods 1](#_Toc2689444)

[Procedural Music 2](#_Toc2689445)

[2 - Literature Review 4](#_Toc2689446)

[Markov Chains 4](#_Toc2689447)

[Neural Networks 4](#_Toc2689448)

[3 - Methodology 5](#_Toc2689449)

[Reading in MIDI files 5](#_Toc2689450)

[Markov Chains 5](#_Toc2689451)

[Frequency Distributions 5](#_Toc2689452)

[Choosing Notes 5](#_Toc2689453)

[4 - Results 6](#_Toc2689454)

[5 - Discussion 7](#_Toc2689455)

[6 - Conclusion 8](#_Toc2689456)

[7 - Appendices 9](#_Toc2689457)

[8 - List of References 10](#_Toc2689458)

[9 - Bibliography 11](#_Toc2689459)

# ii - Table of Figures

[Figure 1 – map thang iv](#_Toc16839)

# iii - Table of Tables

[Table 1 - table thases iv](#_Toc17002)

# iv - Acknowledgements

Thanks to google and caffeine

# v - Abstract

# vi - Abbreviations, Symbols and Notation

If required

# 1 - Introduction

## 1.1 - The problem with games

Music has always been an integral part of most art forms, from plays to blockbuster movies. When used effectively it can greatly influence how the composer wants the audience feel at any specific moment, for example in theatrical production the composer may make the music swell when a characters finally overcomes an obstacle, or become sombre when a character dies. Video games are no exception to this, but with one very important difference, the composer has no idea when a specific action will happen in the game, in essence they can’t make you press the start button.

However, the vast majority of today's games still rely on precomposed musical assets. Something clearly has to be done differently so that a game’s music in sync with what is happening in the game. It would be fairly jarring for a player if the main character died and the music did not change to reflect this, or if a player took too long in a specific part of the game and the audio file rans its course, the game would then have to either play it again, play a different song, or just stop playing music. Obviously if a player plays a game for long enough they will encounter all the available music that it has to offer which would eventually get boring.

## 1.2 - Conventional Methods

One way around this problem is to allow the game to directly influence what music is being played. One method is called ‘Horizontal resequencing’, this is when once piece of music is switched for another when a specific event occurs in the game (Phillips 2016). For example, if the player is exploring a cave the cave theme may play, but if they are attacked by an enemy the music could change to the combat theme. This allows for more flexibility

* Either have to abruptly change music - jarring
* Or fade in/out
* Difficult if tempo is different
* Worse if key is different

Vertical reorchestration

Good

* The theme is split up into several variations, i.e a rhythm section, a more instrumental version, and choral.
* Each section can be introduced or removed depending on what is happening in the game
* i.e. Less melodic rhythmic section when the player isn’t doing much
* add the instruments when a fight starts

Bad

* even more work for composer
* each part has to stand alone and work in any combination

Composer composes, but player both is audience and performer

Every playthough of a game will be different

## Procedural Music

Intro or lit rev?

One way to get around the problems with precomposed music is to have the game itself be the composer. This type of algorithmic ensures that the music will be different every time the game is played, and will continue to be different. This type of music is generally kept for ambient background music, as it lends itself quite well to this genre. However it's not overly used in games when the music has to take the forefront, and drive the intended mood of the game.

Indeterminate music:

Chance music

* As the music is being produced the notes are generated randomly
* Although performed the same
* **Composition is left up to chance**

Aleatoric music

* Premade music, performed randomly
* **Performance is left up to chance**

The only way to accurately experience a games soundtrack is to play the game

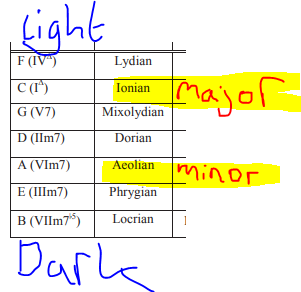
Will alter generated themes on the fly, depending on how the game plays (thematic elements)

Could have composer input a precomposed theme, then the program will then alter this as need be

Procedural Content is best when directed by human made content

<https://link-springer-com.libproxy.abertay.ac.uk/book/10.1007%2F978-3-211-75540-2>

<https://dl-acm-org.libproxy.abertay.ac.uk/citation.cfm?doid=2798084.2749466>



# 2 - Literature Review

## Markov Chains

## Neural Networks

## Player Experience

(Jayden Chan, Daza et al. Oct 2017)

# 3 - Methodology

## Reading in MIDI files

* Read in midi file
* Midi file made up of various events
  + Note start
  + Note end
  + How hard the note is pressed
* Extract note on/off events
* Created specific struct to hold the information
* As midi files don’t have specific information for rests
* Take the difference between notes as rest, create them as normal notes but the pitch is -1
* The key the inputted song is in is also known
* To make change keys easier later on each note’s pitch gets reduced down to the key of C (if original pitch is A, each note’s pitch is subtracted by 9, the semitone difference between A and C)
* Would also allow multiple songs to be combined at the read in stage

## Markov Chains

### Frequency Distributions

The next step in the process is to calculate the frequency distribution of the notes pairs in the inputted song. For each note pair it checks if it is a unique pairing, if this is not the case it increases the frequency counter for that pairing by 1, if it is unique then it creates a new instance of DependHolder and adds that to the list of note pairs. For each note it then sums the number of possible next notes.

### Choosing Notes

When choosing notes

# 4 - Results

# 5 - Discussion

# 6 - Conclusion

# 7 - Appendices

# 8 - References

JAYDEN CHAN, DAZA, J.J., KWAN, W. and BASU, A., Oct 2017Facilitating player progression by implementing procedural music in videogames, Oct 2017, IEEE, pp. 2328-2333.

PHILLIPS, W., 2016-last update, Music Game Plan: Tactics for the Video Game Composer (Part One). Available: <https://winifredphillips.wordpress.com/2016/10/04/music-game-plan-tactics-for-the-video-game-composer-part-one/> [Mar 1, 2019].

# 9 - Bibliography

If required