

DYNAMIC PROCEDURAL MUSIC GENERATION FROM NPC ATTRIBUTES

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Megan E. M. Washburn

March 2020

© 2020
Megan E. M. Washburn
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Dynamic Procedural Music Generation
from NPC Attributes

AUTHOR: Megan E. M. Washburn

DATE SUBMITTED: March 2020

COMMITTEE CHAIR: Foaad Khosmood, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Michael Haungs, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: David Arrivé, DMA
Professor of Music

COMMITTEE MEMBER: Franz Kurfess, Ph.D.
Professor of Computer Science

ABSTRACT

Dynamic Procedural Music Generation from NPC Attributes

Megan E. M. Washburn

Procedural content generation for video games (PCGG) has seen a steep increase in the past decade, aiming to foster emergent gameplay as well as to address the challenge of producing large amounts of engaging content quickly. Most work in PCGG has been focused on generating art and assets such as levels, textures, and models, or on narrative design to generate storylines and progression paths. Given the difficulty of generating harmonically pleasing and interesting music, procedural music generation for games (PMGG) has not seen as much attention during this time.

Music in video games is essential for establishing developers' intended mood and environment. Given the deficit of PMGG content, this paper aims to address the demand for high-quality PMGG. This paper describes the system developed to solve this problem, which generates thematic music for non-player characters (NPCs) based on developer-defined attributes in real time and responds to the dynamic relationship between the player and target NPC.

The system was evaluated by means of user study: participants confront four NPC bosses each with their own uniquely generated dynamic track based on their varying attributes in relation to the player's. The survey gathered information on the perceived quality, dynamism, and helpfulness to gameplay of the generated music. Results showed that the generated music was generally pleasing and harmonious, and that while players could not detect the details of how, they were able to detect a general relationship between themselves and the NPCs as reflected by the music.

ACKNOWLEDGMENTS

Thanks to:

- Mitchell K. Miller, for his strong support, implementation know-how, help with the prototype, rubber ducking, and care
- Maia, Lucy, and David Washburn, for their love, care, and artistic input
- Nathaniel Cinnamon, for his stellar feedback, thoughtful recommendations, and great conversation; and he and Meg Sanders, always for their kind hospitality
- Professor Khosmood, for his guidance, inspiration, and great ideas
- Professors Arrivée, Kurfess, and Haungs for their encouragement and energizing brainstorming sessions
- William and Quinn for their help with getting the prototype on its feet
- Andrew Guenther, for uploading this template
- And all my playtesters for their time

TABLE OF CONTENTS

	Page
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER	
1 Introduction	1
1.1 High Level Approaches to PMGG	1
1.2 Background Organization	3
2 Background	4
2.1 Tools and Techniques	4
2.1.1 Music and Audio Libraries	4
2.1.2 Game and Web Development Libraries	5
2.2 Non-Player Characters in Video Games	7
2.3 Music Theory	7
2.3.1 Chordal Structure	9
2.3.2 Diatonic and Chromatic Composition	11
2.3.3 Musical Modes	11
2.4 Music Mood Model	12
2.4.1 Musical Subjectivity	13
3 Related Works	15
3.1 Music-Based Games	15
3.2 Early PCG Music in Games	17
3.3 Music Generation Tools	17
3.3.1 Full Featured Systems	18

3.3.2	Middleware	18
3.3.3	Libraries	19
3.4	Algorithmic Music Composition	20
3.5	Grammatical Composition	20
3.6	Evolutionary Generation	21
3.7	Machine Learning Models	21
3.7.1	Previous Surveys	23
3.7.2	Promising Paradigms	23
3.8	Music and its Arrangement in Games	24
3.8.1	Player Experience Driven	25
3.8.2	Game Narrative Based	25
3.8.3	Affective (Emotional) State Techniques	26
3.8.4	Generating Game Components Based on Music	26
3.8.5	Music Theory Driven	26
3.9	Summary	27
4	System Development	28
4.1	Composition Algorithm	28
4.1.1	Data Flow	28
4.1.2	Harmonic Structure	30
4.1.3	Chord Sequence Generation	31
4.1.4	Melodic Algorithm	32
4.1.5	Rhythmic Criteria	34
4.2	Generation Process in Practice	34
4.2.1	Non-Player Character Attribute Selection	34
4.2.2	Dynamic Attributes on Repeat Encounters	35

4.2.3	Mapping of Attribute to Musical Feature	35
4.2.4	Building and Running the Program	36
5	Experimental Design Framework	37
5.1	Case Study Hypotheses	37
5.2	Game Environment Design	37
5.2.1	NPC Attribute Selection	39
5.2.2	Player Progression Influence	39
5.3	User Study Structure	40
5.3.1	Basis Questions	40
5.3.2	Musical Quality Questions	40
5.3.3	Gameplay Questions	41
6	Results and Evaluation	42
6.1	Survey Questions	42
6.1.1	Basis Questions	43
6.1.2	Musical Quality Questions	48
6.1.3	Gameplay Questions	49
7	Conclusion	51
7.1	User Study Analysis and Hypotheses Discussion	51
7.1.1	Hypothesis I: Relevant Compositions for NPC Attributes	51
7.1.2	Hypothesis II: Musically Expressive	52
7.1.3	Hypothesis III: Dynamic Adaptation	53
7.2	Additional Analysis	53
7.2.1	Successful Immersion	53
7.2.2	Setting-Based Confusion	54
7.2.3	Beyond Hypotheses: Saved Time	55

8	Future Work	56
8.1	Improvement: Musical Transitions	56
8.1.1	Generative Space Considerations	57
8.2	Modes to Include	58
	BIBLIOGRAPHY	60
	APPENDICES	
A	Music Theory	66
A.1	Harmonic Functions of Mode's Chords	66
B	Basis Questions	68
B.1	Free Response Questions	68
B.1.1	<i>Tempo</i> : do you think it represented anything?	68
B.1.2	<i>Melody</i> : do you think it represented anything?	69
B.1.3	<i>Harmony / Accompaniment</i> : do you think it represented anything?	70
B.1.4	<i>Musical Mode</i> : do you think it represented anything?	71
B.1.5	<i>Instrumentation</i> : do you think it represented anything?	72
B.1.6	Layering of Voices	74
B.2	Multiple Choice Question	75
B.2.1	<i>Boss Attribute</i> : did the music help indicate any of the following?	75
C	Musical Quality Questions	77
C.1	Quality Score Questions	77
C.2	Free Response Question	77
D	Gameplay Questions	81
D.1	In-Browser Experience with Game	81

LIST OF TABLES

Table		Page
3.1	PCG Music in Games	16
3.2	PCG music papers arranged chronologically, accompanied by method	20
3.3	PMGG: Publications on Procedural Music in Video Games	24

LIST OF FIGURES

Figure		Page
2.1	Diatonic harmony inversions and their associated notations [5] . . .	10
2.2	Mapping of standard musical modes, tempo / activity onto mood model	14
4.1	Illustrates the dependencies of the core musical constructs within this system	29
4.2	Simplification of musical composition feature dependencies	29
4.3	Illustrates the dynamic relationships of game features to musical features within the generation system	36
5.1	Full Phaserquest map. Note the general vertical progression structure.	38
5.2	Final boss scenarios. View frustum of player is zoomed in, so they cannot initially see which boss they are about to face.	38
5.3	Left to right: Ogre, Spectre, Deathknight, and Lavaboss; second-to-best gear player, maxed-out player.	39
6.1	Gender distribution	42
6.2	Academic status	42
6.3	Major and / or profession distribution	42
6.4	Total responses including write-in answers	48
6.5	Did the music save you time? As in, did it indicate anything to you that other feedback in the game didn't make immediately apparent to you?	49
6.6	True: those who guessed the boss they were about to face correctly; False: those who didn't	50
6.7	Did you guess which boss you were about to face based on the music you heard?	50

8.1	Illustrates potential dynamic relationships of game features to musical features within the generation system	59
B.1	Total responses including write-in answers	76
B.2	Total responses consolidating write-in answers and <i>setting change</i> -based answers	76
C.1	Was the music pleasing?	77
C.2	Did the music sound random?	77
C.3	Was the music harmonious?	78
C.4	Was the music interesting?	78
C.5	Did the music appear to be dynamic? (As in, did it change as you progressed?)	78
C.6	Did the music appear to be change as you revisited areas and NPCs?	79

Chapter 1

INTRODUCTION

Music in video games is crucial to facilitating immersion in players [20], [41]; it is as important for conveying in-game information as it is for player enjoyment. Given the breadth of possible music theory-based and algorithmic means to create successful procedurally generated video game soundtracks, developing suitable and engaging music is a deeply compelling task. Procedurally generated music is directly suited for scoring video games, given their stochastic nature due to player action. As video games have a high degree of non-linearity, procedural methods offer designers an opportunity to coordinate music directly with player choices, while preserving thematic tone and player immersion. The rich history of procedural music in video games draws from specific concepts in both algorithmically generated music and procedural content generation (PCG), traditionally found in games. We begin by breaking down the approaches to procedural music generation in games (PMGG) along the largest dimensions found in literature.

1.1 High Level Approaches to PMGG

A simplified approach to PMGG mirrors the two main inputs in any PMGG system: player input and music input. A succinct definition of dynamic music in games was first proposed in Karen Collins’ 2009 survey of procedural music in games, in which she defines “interactive” versus “adaptive” audio within dynamic audio.

Interactive audio events are directly triggered by player action, whereas *adaptive* audio events have a broader reach: they do not respond immediately to player action,

but rather change based on the trajectory of player choices and in-game contextual events [8]. Collins uses these distinctions to define a clearer structure to her discussion of the techniques used at that time. Since then, there have been many advances in the field of procedural generation in games, algorithmic generation of music, and procedural music in games. To provide context, we'll discuss techniques, tools, games, and recent contributions and shifts in the field.

A second useful dimension in PMGG approaches is apriori music input. Here we find the work of Wooller et al. in music generation algorithms to be the most instructive. They identify “transformational,” “generative,” and “analytic” algorithms as major identifying dimensions with respect to procedural techniques in video game music [57]:

- *Transformational* algorithms take existing musical information as input and alter it; they tend not to have significant impact on either the perceived musical feel and size of the data. These are seen often in adaptive video game scores as they provide a means to react to player action without breaking continuity in the existing musical landscape of the game.
- *Generative* algorithms need no input, generally tend towards a stronger perceived musical feel in their output, and increase the data representation. These have great potential for scoring a more interactive musical experience in video games, but is both trickier to control and riskier than pre-scripting an adaptive soundtrack, and often does not see use in AAA games.
- *Analytic* algorithms reduce the potential data size and general perceived musical feel by distilling specific features from existing pieces of music. This technique is often seen in experimental or learning-based algorithms, in which a single monophonic line is extracted from a polyphonic piece [9], or when a basic musical structure is learned from a large database of pieces [27] [14] [15] [53].

We note that these classifications apply to algorithms rather than systems; fully realized systems often contain a combination. After identifying predominant techniques used in procedural music in games, the most successful approaches will present themselves to be multi-modal combinations of these classifications; as video games present a diverse set of interactivity problems, a diverse solution to algorithmic music generation is required.

1.2 Background Organization

The background is organized as follows: establishing context for PMGG, providing useful tools, reviewing techniques for algorithmically generating music, and discussing previous applications of procedural music generation in games. The background section provides a brief discussion of music-based games to distinguish between games with a musical mechanic versus those that contain procedurally generated and dynamically-structured music. Related works will cover works ranging from David Cope's work on algorithmic music starting in 1981 to current video game systems that feature dynamic procedural music generation.

Chapter 2

BACKGROUND

2.1 Tools and Techniques

To illustrate the implementation process, this section describes the tools necessary to set up each component of the project and its user study, and provides detail on how they were used. The current best library for generative music is written in JavaScript, mainly due to the most web browser's powerful versatility and accessibility, as well as its high availability of quality audio and music related tools. The libraries involved can be primarily grouped into categories concerning music and game development.

2.1.1 Music and Audio Libraries

Tone.js

As a JavaScript framework built on top of the Web Audio API, this library is intended for interactive music creation in the browser. It allows developers to create sounds from MIDI formatted data, in this case to be played in-game. Given the extensive reach of browser-based applications, Tone.js affords developers and composers an effective means to create work that is both accessible and novel. It has served as the backbone for many interactive educational music theory applications and experimental generative music applications. For this project, it provided the tempo and time signature framework as well as the synthesized sound instrumentation.

Tonal

This library encapsulates core abstract musical concepts in an organized manner. It represents the abstract tonal constructs in music, such as chords, notes, modes, keys, intervals, and more. It is implemented in Typescript and published as a collection of Javascript NPM packages. As it is functionally programmed with no data mutation, entities are represented by data structures instead of objects, providing a bare-bones set of information to be manipulated. This served as the backbone for the more complex musical constructs that were necessary to generate the nuanced music composition algorithms for the game's NPCs.

MidiWriterJS

This library provides a layer on top of the binary data that comprises MIDI to allow for easier creation of MIDI from music theory constructs, such as those that appear in Tonal. It serves as a bridge from abstract musical constructs to a format that can be played back as audio. This library also allows for file creation, so the generated songs may also be downloaded for future use.

2.1.2 Game and Web Development Libraries

Typescript

TypeScript is a superset of JavaScript that adds type information. This helps catch errors at compile time and reduce the total amount of time needed to develop the project. However, it requires some extra effort to annotate code with types and sometimes to annotate external libraries. TypeScript code must be compiled to JavaScript before it can be run in-browser.

Webpack for Typescript

Within this project, **webpack** is responsible for statically bundling TypeScript files to JavaScript for use in-browser. As the game used for user studies was developed in JavaScript-based Phaser.io and the music generation service dependencies were largely developed in TypeScript, it was necessary to bundle the TypeScript music generation files and all dependent modules into a single JavaScript file callable from Phaser's JavaScript files. An bundling entry point is defined to **webpack**, which includes all other modules that this entry file points to via its *imports*.

Node.js and NPM

Node.js is an open-source, cross-platform JavaScript runtime environment written by Ryan Dahl that assisted in the installation, development, and deployment of this project. NPM, short for "Node Package Manager," is two things: primarily, it is an online repository for the publishing of open-source Node.js projects; secondly, it is a command-line utility for interacting with the aforementioned repository. It is the current standard for Node.js projects to be published, discovered, and developed.

Phaser.io

Phaser is an open source HTML5 2D game development framework developed by Photon Storm that uses both Canvas and WebGL renderers internally, and automatically swaps between them based on the available browser support; this allows for fast rendering across desktop and mobile. It offers game mechanic, physics, rendering, as well as rudimentary audio frameworks. JavaScript or TypeScript can be used for development, expanding its potential use cases.

2.2 Non-Player Characters in Video Games

NPCs are defined as characters in video games that are primarily controlled by the computer, rather than the player. “They range from the mundane, to the fantastic, to the distinctly non-human” [55]. They serve a broad range of functions for the player, ranging from providing aid, providing engaging challenges, enhancing player immersion, and much more. A succinct division of NPC typologies proposed by Aarseth (2012) is as follows: *deep characters* - those with believable and seemingly well-rounded personalities, *shallow characters* - those that, while they may possess a complex backstory, will rarely exhibit the same level of personalities as the former, or *bots* - those enemies (or allies) that are non-complex, forgettable characters as they lack individual identity [1].

For the purposes of this study, the target NPCs for which music is generated are *shallow enemy characters*, the player’s main antagonists or significant bosses. They fall under the category of shallow due to the fact that within the user study session, there is not enough time in to develop deep personalities for each boss; they are given backstories, but only interact with the player by attacking when aggroed. In non-user study contexts, developers will be able to use this generative system to subtly inform their players of the combat characteristics and deeper personality of these NPCs.

2.3 Music Theory

To understand the algorithms that generate the music for a given unique boss, a basic understanding of chordal structure, rhythmic structure, traditional composition rules, and the function of musical modes is necessary. Each topic will be covered in detail their own respective section.

A fundamental concept here is the musical *interval*, which refers to the distance between two pitches (a *dyad*), usually measured as a number of steps along any given scale [49]. The five interval qualities that precede interval number are denoted as:

- Augmented, A: Interval that is half-step larger than *perfect* or *major* interval
- Major, M: Interval of which is part of the *major scale*
- Perfect, P: Prime / unison (first), fourth, fifth, eighth (octave) intervals
- Minor, m: Interval that is half-step smaller than *major* interval
- Diminished, d: Interval that is half-step smaller than *perfect* or *minor* interval

Perfect intervals are named as such due to their sound and because they have simple whole number frequency ratios. Generally speaking, major scales can be classified as brighter and happier sounding than their minor counterparts. The interval sequence for the basic major scale is as follows, in two different formats: whole steps = W, and half steps = H are steps between adjacent pitches in the scale, while the P1 — P8 notation indicates the interval jumps from the tonic to each pitch of the scale.

- W W H W W W H — P1 M2 M3 P4 P5 M6 M7 P8

Conversely, minor scales tend to sound darker and more melancholic in comparison to their major counterparts. With the same notation as above, the interval sequence for the natural and melodic minor scales are as follows, respectively:

- W H W W H W W — M2 m3 P4 P5 m6 m7 P8
- W H W W W W H — M2 m3 P4 P5 M6 M7 P8

Each minor scale configuration has their own benefits and applications, primarily when composing melodies. These considerations were important when designing the melody generation algorithm. This concept of associating specific emotions to each scale extends to the musical construct of modes, each of which has its own distinctive tone and perceived feel due to each modes' interval combinations.

For terminology reference, each scale degree can be assigned one of the following names, with each mode containing a different corresponding name to scale degree. In the Major (or Ionian) scale, each scale degree can be referred to as the following:

1 st : Tonic	5 th : Dominant
2 nd : Supertonic	6 th : Submediant
3 rd : Mediant	7 th : Leading Tone
4 th : Subdominant	8 th : Tonic

2.3.1 Chordal Structure

Chords and the progression from one to another (aka their harmonic function) are principally responsible for the mood and feel of a piece of music. The standard *triad* is the base structure of all complex chords: they are a collection of three notes consisting of any given root note and two other notes, one that is a third interval above the root and the other being a fifth above the root. Triads have four basic assignable qualities: major, minor, diminished, and augmented, each with their own characteristic sound that can be used to manipulate the mood of any given piece when used appropriately. The assigned quality depends on the relative intervals above the root, as follows:

- **Major Triad:** M3 and P5 above the root
- **Minor Triad:** m3 and P5 above the root
- **Diminished Triad:** m3 and d5 above the root

- **Augmented Triad:** m3 and A5 above the root

Chords become more complex as you add additional notes and introduce inversions; more qualities become available, and therefore a wider range of perceived feel. Seventh chords add a fourth note above the triad, and consist of these six qualities: **major, dominant, minor-major, minor, half-diminished, and diminished**. Inversions classify chords that have specific notes in a given chord as their bass note. For example, a *Major seventh chord in first inversion* would contain the second note (the tone that is a M3 above the root note) as the bass note of the chord, and the other P1, P5, and P7 notes anywhere above this bass M3. The second inversion contains the P5 tone as its bass note, and the third likewise has the P7 tone as its base.

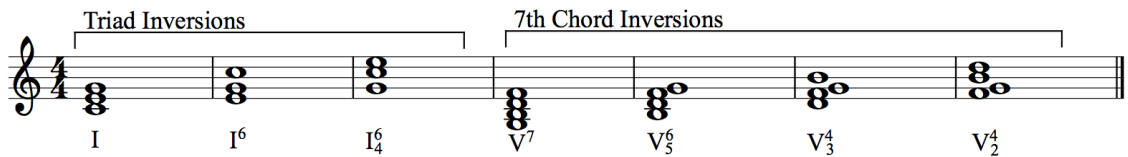


Figure 2.1: Diatonic harmony inversions and their associated notations [5]

This leads to the harmonic function of chords. When designing the harmonic accompaniment algorithm, all 7 chord possibilities built off of the 7 unique notes of a scale were considered possibilities. Though, to ensure the music sounded harmonically pleasing to the western ear, the algorithm ensures each of the chords were approached and left appropriately. There are three harmonic functions that each chord of a scale's degrees can be classified as, as follows:

Function	Chords	Triggers	Associates	Dissonances
T: Tonic	I, III, VI	1, 3	5, 6	5 (if 6 present), 7
S: Subdominant	II, IV	4, 6	1, 2	1 (if 2 present), 3
D: Dominant	V, VII	5, 7	2	4, 6

Here, triggers, associates, and dissonances are all characteristic tones of each harmonic function. In descending order, they indicate which tones are more or less characteristic of their respective function [49].

2.3.2 Diatonic and Chromatic Composition

Diatonic constructs — notes that exist within the major and natural minor scales and modes — give form and structure to a composition. Though pleasing, these constructs can become bland and cause “ear fatigue” to listeners. To add more character and interest, chromatic tones can be thoughtfully added or artfully replace diatonic tones. When placed correctly, these variations can drastically change the perceived feel of a piece of music. The next section covers the musical construct that had the largest affect on the variation and uniqueness of the generated music. By factoring in well-used modes from around the world, the generated music brought more character and depth to the NPCs it was generated for.

2.3.3 Musical Modes

Modes are alternative tonalities (scales) that are largely responsible for the perceived mood and feel of a composition. The most commonly referred to modes are derived by creating a scale based on the standard scales as outlined above, beginning on a different note than the original tonic. There are many other modes that have been invented in all regions of the world, each with their own distinct, unique qualities.

To deepen the breadth of music this system generates, I looked to include only the traditional church modes for this user study due to the complexity of associated rules. My main criteria was to include those non-standard scales and modes that did not majorly overlap with one another (in terms of shared tones), to allow for the

generative space to be as minimally redundant as possible. I also excluded scales such as the Hungarian Gypsy scales and African quarter-tone scales, as the slurs and quarter tones present in these scales were impractical to implement given the tools I chose for this system. Following is a list of the modes included for the user study: **Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian** (*Note: Locrian not used due to overly-dissonant sounding results*). See this section in Future Work for the modes to be included in future implementations: 8.2.

2.4 Music Mood Model

Previous research into developing a musical mood model based on human emotional perception of musical modes (see Musical Modes) served as an essential guide when designing the algorithm to express the intended perceived mood based on NPC attributes. Earlier work in identifying the perceived emotion can be seen in Ramos et al.'s work in characterizing combinations of traditional Greek modes, tempos, and time signatures [37].

The model proposed by Robert E. Thayer has thus far achieved the best results when used to elicit the moods specified in the axes of his model. Energy (arousal) and valence are represented as axes against one another in the graphical representation of mood. The musical aspects of mode, tempo, harmonic structure, dynamics, and timbre all affect the perceived mood of a piece of music, and all can be given a general location on this axis. Though, due to the complex interaction between each of these aspects, this serves only as a rudimentary guide: for this application to video game NPCs, it works sufficiently well. Otherwise, the representation of mood and its correlation to specific combinations of musical aspects is currently an open question in need of an improved model estimation.

The following graphical representation of the circumplex model of affect aligned with musical mode (and other aspects) is based on Russell, Posner, and Peterson’s exploration into this energy-valence affect space [36] (see figure 2.2). Grekow et al. and Straehley et al. both later contributed to mapping this mood model to music; their studies into musical attributes and perceived mood helped to influence the decision of which musical attributes to map to which part on the energy-valence mood model [19] [6]. There is generally higher and lower perceived “lightness” and “darkness” of the seven basic church modes [54] that primarily influenced the placement of the modes on this mood model.

2.4.1 Musical Subjectivity

It is important to note that the interpretation of musical affective state is ultimately subjective. Depending on the societal upbringing of the listener, their opinion of the mood and feel of a particular passage of music can vary greatly. With this in mind, the aforementioned musical mood model is an estimate provided by an in-depth analysis of music psychologists with Westernized ears.

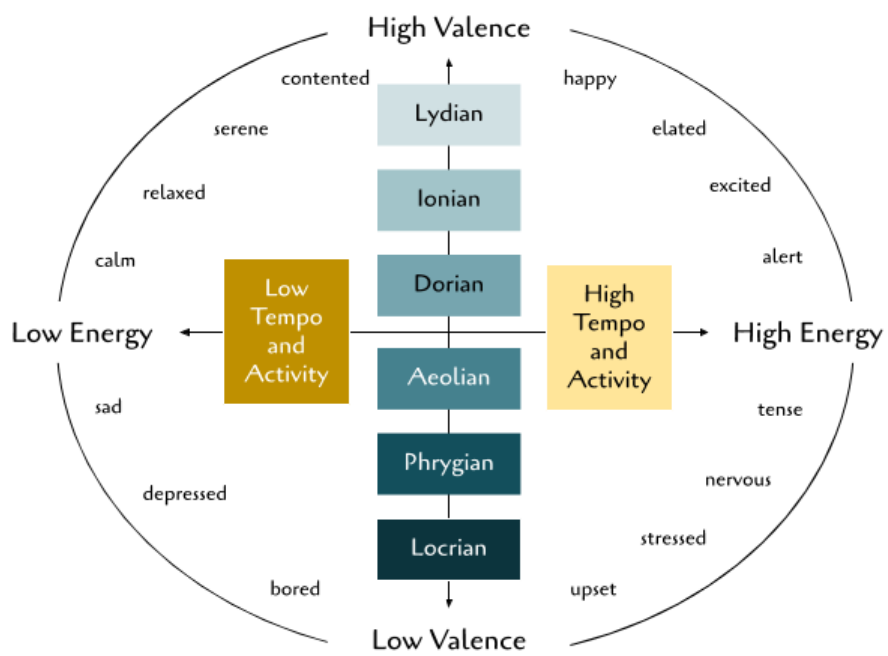


Figure 2.2:
Mapping of standard musical modes, tempo / activity onto mood model

Chapter 3

RELATED WORKS

The first instance of procedural music in a game consistently traces back to Peter Langston’s work for Lucasfilm Games *BallBlazer* games, first released for the Atari 8-bit family in 1984 [26]. LucasArts composers Michael Land and Peter McConnell used the framework laid out with the *BallBlazer* games to develop the iMUSE (Interactive Music Streaming Engine) system [52]. This system interfaced with their game engine (SCUMM), allowing the company to produce a fair list of games that included both generative and transformative musical soundtracks [25] [11].

Soon after, advances in platform hardware removed the limitations of sound needing to be synthesized on-board. The ability of sampling and software mixing greatly increased the capabilities of audio in games, and with it, the organization of how to best create an immersive audio landscape for players. Advancement in audio quality of horizontal and vertical stem mixing soon followed (horizontal mixing being orchestration changes to the upcoming structure of music, and vertical mixing being orchestration changes to the layers in any point of time in music) [30] [34] paving the way for more advanced systems.

3.1 Music-Based Games

Video games such as *Beat Saber*, *Guitar Hero*, *Rock Band*, *Patapon*, *PaRappa the Rapper*, and *Dance Dance Revolution* see high success due to the inherent connection their audiences have to their music track lists; these games thrive on the enhanced immersion derived from direct mapping of game events to the soundtrack. The phys-

Table 3.1: PCG Music in Games

Game Title	Developer	Year	PCG Technique(s)
BallBlazer	Peter Langston	1984	”Riffology” (Generative)
Otocky	Toshio Iwai	1987	Adaptive, Generative
Monkey Island 1-3	Lucasfilm Games	1991	iMuse
Star Wars: X-Wing	LucasArts	1993	iMuse
SimTunes	ToshioIwai	1996	Adaptive, Generative
.kkrieger	Farbrausch	2004	V2 Multifunc. Synthesizer
Grim Fandango	LucasArts	1998	iMuse
Electroplankton	indieszero	2005	Transformative, Generative
Spore	Brian Eno	2008	Interactive, Generative
No Man’s Sky	65daysofstatic	2016	Generative, Adaptive
Rez Infinite	Tetsuya Mizuguchi	2016	Quantization of Note Events

ical action places emphasis on the perceived affective (emotional) state of the music, affording players a unique, impactful experience. Games such as Audio Surfer utilize fast Fourier transforms to allow users to play to their music, while mods such as Beat Saver allow the community to create custom, shareable levels. Procedural music in games accesses this distinct form of immersion directly, and allows game developers more control in engineering a game suited to their vision.

Educational music games, such as Rocksmith, Opera Fatale, Music Wizard, Music Master: Chopin also stand to benefit from procedural music generation. Customizing auditory feedback for a player in an educational setting can boost their motivation to learn the content presented in game [13].

3.2 Early PCG Music in Games

As seen in Table 3.1, early generative procedural music can be found in BallBlazer and its succeeding games using the iMuse system at LucasArts. Aside from these works, procedural music in games widely stayed in the *adaptive* and *transformative* realms as algorithmic music composition was being explored by musicians and composers. By the early 2000s, improvements in vertical and horizontal stem mixing allowed for developers and composers to experiment again with *generative* methods.

We now find notable feats of procedural music generation for video games that adapt to player actions and in-game events in the games *Spore*, *No Man's Sky*, and *Rez Infinite*. As existing content is easier to access than ever before, designing experiences that are uniquely custom-built for players is a valuable goal in game design, and has proven to be attainable with the help of multi-modal systems for adaptive, generative procedural music.

3.3 Music Generation Tools

The adaptivity to user preferences inherent in music composition assistant software is important to consider in light of responsive, immersive video game music. The highly-customizable control parameters for AI-assisted composition tools can be mapped to player input and game events to create powerful, online adaptive music in-game. In this section we briefly survey major music generation tools. These tools can be divided into three categories using the level of sophistication and features that they offer: full-featured systems, middleware and programming libraries.

3.3.1 Full Featured Systems

Full featured tools are commercial products that perform a variety of complex, high-level tasks for the user. AIVA very recently released its free beta version. It generates custom length pieces of a select few genres of music, and allows users to specify BPM, time signature, key signature, and constrained instrumentation. Notably interesting is its ability to take user-specified input as influence to the generation.

Amper Music is another system that generates custom pieces of music that are appropriate per genre, and is able to do so as it contains a significant amount of music theory knowledge. ORB Composer also generates music suitable for commercial use using AI. Cycling74 and Earcom offer technologies that assist in producing music for interactive games, performances, and art installations. Native Instruments, Logic Pro, and Seer Systems all offer VSTs (Virtual Studio Technologies) that are useful for use in games as a looped background track.

3.3.2 Middleware

Middleware tools are meant to be embedded into larger systems and accessed in a programmatic manner. These tools are significantly important to audio game developers. Best known titles include FMOD, Wwise, and Audiokinetic. Bridging the gap between Digital Audio Workspace (DAW) and game engine, these software continue to evolve to suit the needs of game developers to craft adaptive and generative music for their games.

3.3.3 Libraries

At the lowest level, there are libraries that allow programmers access to most basic audio generation and manipulation functions. Most higher level features must be built custom by the programmers specific for their project. For web-based applications, the Javascript library Tonal.js provides a well-organized, in-depth music theory knowledge useful for translating composition concepts to code. [12]. SuperCollider, created in 1998 and currently maintained, is an extensive, open-source library for audio synthesis and algorithmic composition available for Windows, macOS, and Linux [28]. For those interested in Java-based applications, jMusic is a viable option for generative music, instrument building, interactive performance, and music analysis [51].

A helpful gentle introduction to music composition using Lisp and common music theory concepts can be found in Simoni’s work for the SPO Scholarly Monograph Series, an interdisciplinary collection of original, open-access scholarly monographs and essays. [50]. For those who prefer visual feedback, or who come from a stronger music background, OpenMusic is a visual programming language based on Lisp aimed towards computer-aided music composition [3].

Author	Year	Technique
Roads et al.	1979	Generative: Grammar
Langston	1986	Generative / Transformative: Specific Music Constructs
Cope	1991	Generative: Algorithms for Style Modeling
Xenakis	1992	Generative: Formulaic
Jacob	1997	Generative: Exploration of Creative Algorithm
Papadopoulos, Wiggins	1999	Survey
Birchfield	2003	Generative: Genetic Algorithm
Jewell et al.	2005	Generative / Transformative: Multi-Modal
Wiggins	2007	Generative: Review
Nierhaus	2008	Survey

Cope	2013	Generative: KB Algorithm
Rodriguez, Vico	2014	Survey
Scirea et al.	2016	Generative / Transformative: Multi-Modal, GA
Liu, Ting	2017	Survey
Huang et al.	2017	Analytic: Convolutional Layers
Oore et al.	2017	Analytic: LSTM, RNN
Roberts et al.	2017	Analytic: Bidirectional Encoder, Hierarchical Decoder
Dong et al.	2018	Analytic: Multi Model GAN
Dong, Yang	2018	Analytic: GAN with Refiner
Sturm et al.	2019	Analytic: Survey

Table 3.2: PCG music papers arranged chronologically, accompanied by method

3.4 Algorithmic Music Composition

Generative music on its own has a rich history that inspired much of the structure for this project; it is fitting to dedicate a section to describing the previous approaches and algorithms for generative music that have inspired this system [9] [10]. Several approach groupings can be observed in the PMGG literature. We discuss each below with enough background such that the interested reader can investigate further if desired.

3.5 Grammatical Composition

One of the earliest representations of music in computational generation are grammars: by applying linguistic and semantic analysis concepts, grammars are capable of describing music theory in a structure directly suitable for procedural generation [38]. Many systems codified their target music theory paradigms with enough depth to generate structurally correct music [57] [9] [10] [56]. See 3.2 for more on methods for grammatical composition.

With the concept of machine emotional intelligence in mind, Salas et al. attempted to solve the issue with a linguistic approach, using a generative grammar to create affective states

[17]. This approach of codifying affective states in a grammar arises in many works, as specific music theory paradigms such as choice of mode, chord type, or chord progression dictate the perceived affective state significantly.

3.6 Evolutionary Generation

Given the inclination towards emergent behavior found in evolutionary algorithms, algorithms that take inspiration from biological evolution serve as strong tools for creative inspiration, as they have the ability to generate a broad space of potentially new chord progressions, harmonies, or melodies.

Birchfield et al. introduced a generative model for the creation of musical emotion, meaning, and form: their system uses a genetic algorithm (GA), that of which is proven to be ideal for their goal based on the brief background given on other GA models and how they compare. Their system is intended to model emotion and meaning in music, and is more of a compositional system than a listening experience. Improvements the team noted included “moments in the generated music where the drive seems to stall,” an issue still found often in many generative systems [4]. The addition of stronger music theory structures such as chordal structures to create an overarching song form to provide direction in the piece is an example of an appropriate remedy to this issue.

3.7 Machine Learning Models

Huang and Cooijmans work in 2017 is one of the early successes that proves the efficacy of machine learning (ML) applied to music composition: they trained a convolutional neural network to generate valid counterpoint compositions [23]. Given the ruleset inherent to composing a counterpoint piece, a neural network utilizing convolutional layers to analyze the chronological patterns found in counterpoint pieces is an ideal application. While

successful, this system does not allow for interactive adaptation of the orchestration of a musical piece. Adaptability is crucial to video games, rendering this form of ML as not ideal as a solution to orchestrating an adaptive, immersive video game soundtrack. Regardless, ML as applied to music still offers much potential to solve other issues for PMGG, and therefore earns its place for discussion in this paper.

There is significant work in creating machine music that contains expressivity as would be found in a *performance* [31]. Google’s Maganta team has trained on a database of prolific piano performances, to inject appropriate pauses, accelerandos, and dynamics to a musical score. If applied to video game music generation tasks, this work poses potential for higher quality, nuanced scores to improve player immersion.

To address the issue of generating coherent music over prolonged spans of time, Roberts et al. applied ML to codify long-term structures found often in and characteristic of music [39]. It is important to determine a way to better algorithmically generate music with cohesive long-term structure to combat ear fatigue. Methods such as layered adaptive music or well-crafted musical grammars are a cleaner means to address this issue; these methods are core parts of successful multi-modal systems for generating video game music.

Accompaniment is significant to establishing the general tone of a musical piece, and if done well, can boost player immersion by subtly establishing the intended mood. Dong et al. applied a triple-model GAN to the issue of generating quality multi-track piano rolls for use as accompaniment, labeling the three GAN models Jamming, Composer, and Hybrid respectively [14] [15]. This structure allowed for both novelty and music theory to influence their generator, aspects important to combat ear fatigue in listeners.

By analyzing previous successful machine learning models in music creation, Sturm et al. present a prediction of important ML research to come in the following years [53]. Given a more robust review of previous projects, this paper would present a compelling argument for ML music research that is pertinent to both fields.

3.7.1 Previous Surveys

One of the earliest survey publications presented six techniques for algorithmic music composition, seeds of algorithmic generation to come: stochastic binary subdivision, riffology, auditals, key phrase animation, and fractal interpolation. Langston was the driving developer behind LucasArt’s BallBlazer, the earliest game to ignite work for procedural music in games [26]. For more content inspired by his work, see table 3.1.

Further definition of appropriate algorithms for music generation such as mathematical models, knowledge-based systems, grammars, evolutionary methods, systems which learn, and hybrid systems were defined in 1999 by Papadopoulos et al.[32]. The following survey discussed Markov models, generative grammars, transition networks, chaos and self-similarity, genetic algorithms, cellular automata, and preemptive artificial neural network (ANNs) systems in 2008 [29]. Popular AI methods maintained to be grammars and symbolic representations, knowledge-based systems, Markov chains, ANNs, evolutionary and other population-based methods, and self-similarity and cellular automata [40]. Each method excels in addressing individual problems that the multi-modal system of composing music presents, but still does not solve the problem of achieving quality generation for all aspects of music in one system in an adaptive manner.

A more recent survey conducted in 2017 broadly covers computational intelligence techniques that saw relative success, reviewed in light of predominant music composition aspects, “wit”, musical form, melody, and accompaniment. Leading into the slew of machine-learning based methods of generating music, Liu et al. also reviews their subjective best evolutionary algorithms and neural networks designed for music composition [27].

3.7.2 Promising Paradigms

Multi-modal systems pose the highest potential for suitable, generative, adaptive music in video games. Prior systems such as Jewell et al.’s work in 2005 and Scirea et al.’s work on

MetaCompose in 2016 gear themselves towards adaptability in video games [24] [48]; their work support multi-agent systems as the ideal paradigm for improving the current state of procedural music in games.

MetaCompose presents a promising system design to integrate to game development systems for adaptive, fully generative music in games [48]. It is a significant step in the direction of achieving immersive, affective music while maintaining compositional quality. Emphasis is placed on the distribution of musical tasks to three different subsystems in their design to generate the composition (chord sequence, melody, and accompaniment), and on what musical constructs based in music theory best create the intended perceived emotional state.

Author	Year	Technique
Farnell	2007	Survey
Collins	2008	Survey
Collins	2009	Survey
Plans and Morelli	2012	Player Experience-Driven
Holtar, Nelson, Togelius	2013	Co-Generation of Game and Music
Scirea et al.	2014	Narrative-Driven
Adam	2014	Algorithmic and Affective
Hoover, Togelius, Yannakis	2015	Co-Generation of Game and Music
Chan et al.	2017	Player Experience-Driven
Gillespie and Brown	2017	Music Theory-Driven
Scirea et al.	2018	Evolutionary Algorithms for Affective State

Table 3.3: PMGG: Publications on Procedural Music in Video Games

3.8 Music and its Arrangement in Games

A useful early discussion of procedural audio in video games by Farnell defines four types of sound found in procedural audio for games: synthetic, generative, stochastic, and algorithmic [16]. This marks an early attempt to better classify procedural audio in games, and is followed by Collins’s 2009 survey of procedural generation music for video games.

Her definitions and classifications serve as important distinctions to better understand the topic, and are referenced throughout the analysis presented here [8]. Given that there have been no comprehensive publications covering the topic since then, this background covers almost a decade’s worth of contributions to the field of procedural audio in games. See 3.3 for a comprehensive list of publications on PMGG.

3.8.1 Player Experience Driven

The Experience-Driven Procedural Content Generation (EDPCG) framework outlined by Yannakakis et al. in a previous publication [33] serves as the basis of structural design in Plans et al. player experience-driven procedural generation of music [35]. They chose to encode *frustration*, *challenge*, and *fun* as their player metrics to serve as the parameters of generation which follows a traditional GA implementation with a harmonic sequence generator and period builder.

Coming from the perspective of aiding player progression rather than just improving their experience, Chan et al. turn to procedural music in games to do so [7]. Their work introduces an evaluation system to prove whether or not procedural music aids players in progressing through a video game. Their findings ensure that a system with procedural audio to assist in the completion of levels lowered their completion time in addition to increasing players’ engagement.

3.8.2 Game Narrative Based

Marco Scirea conducted research into mood-expressing, procedurally generated music specifically with the intent of preemptively alerting players of upcoming events in-game [44]. Their evaluation proved that appropriately corresponding musical cues to upcoming events positively impacted perceived consistency between the game’s music and narrative; there were interesting cases in which *false* (or mismatching) foreshadowing increased player enjoyment.

3.8.3 Affective (Emotional) State Techniques

The theory on affective state present in MetaCompose and previous work by Marco Scirea references Jacek Grekow's work on defining emotion in music by mapping their descriptors to a space composed of the axes *valence* and *arousal*, the former corresponding to a scale of positivity and negativity, the latter to high to low energy [47] [19]. Earlier works also used this scale to structure metrics with which to generate music [2]. Scirea et al. built off their system to develop experiments to further research into expressive music specifically in interactive environments such as video games [46].

3.8.4 Generating Game Components Based on Music

Taking a paired approach to PCG, Holtar et al. developed a system to couple the procedural generation of game content and music [21]. Such an approach inherently produces appropriate music, but naturally raises issues based on the complexity of music composition.

An approach even further in the direction of a game based on music was the work Hoover et al. did to procedurally generate game levels architecture based on the music in their game [22]. These works invert the approach of generating music for a game by placing it as the guiding development principle; it is important to take note as these perspectives afford insight that the traditional approaches do not immediately present.

3.8.5 Music Theory Driven

Pairing music theory concepts with game design, Gillespie et al.'s recent work in 2017 tackles the problem of smooth transitions that is a significant limiting factor of incorporating procedural music to a game [18]. Taking a composer-based perspective, their work is essential to solving the in-game music transitions as the means to do so in a musically coherent manner requires a music theory perspective coupled with game design.

3.9 Summary

This section summarized and analyzed the research and tool developments to illustrate the state of research in PMGG. Thoughtfully complex multi-modal systems that account for musical structure based on music theory concepts as well as immediate adaptivity to maintain player immersion present both promising and reliable solutions. Further refinement of these systems will present a powerful solution that avoids ear fatigue and evokes suitable affective states for developers [43].

Chapter 4

SYSTEM DEVELOPMENT

This system is a multi-agent expert system, which generates components of music by various means. Each algorithm is specifically developed to handle a specific musical composition aspect, all of which are synthesized to a single MIDI file that contains all the movements, sections, rhythms, notes, and instrumentation necessary to playback a full composition in-browser, in real-time for the game. This system draws inspiration from the interactive music generation concepts proposed in Scirea et al.'s work on MetaCompose [48], but includes various improvements and alterations that allow the music generation system to adapt specifically to NPCs in a more responsive and expressive manner.

An expert-system model is ideal for the rapidly evolving environment in video games. While machine learning-based music generation has shown promising results, it is neither fast nor clean enough to be applied to the dynamic environment of video games at this time.

4.1 Composition Algorithm

4.1.1 Data Flow

As input, the system receives four values that are resolved from the developer input: the first ranging from 0.0–1.0 representing arousal, second ranging from -1.0–1.0 representing valence, third representing the intended time period, and last being a randomly generated bit string serving as the seed. These values together resolve to various aspects of the final composition, each as will be discussed below. To have a better idea of how the system takes in this input and outputs the composition, see the following data flow diagrams: figures 4.1 and 4.2.

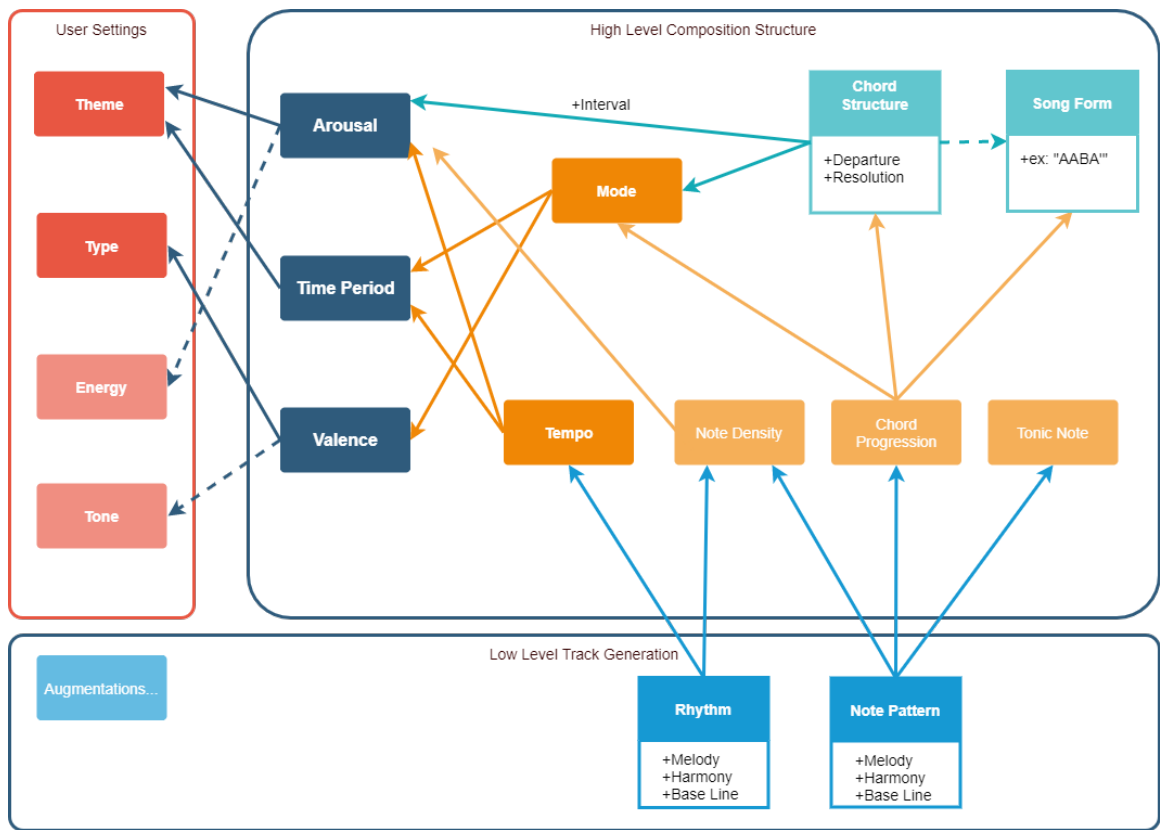


Figure 4.1: Illustrates the dependencies of the core musical constructs within this system

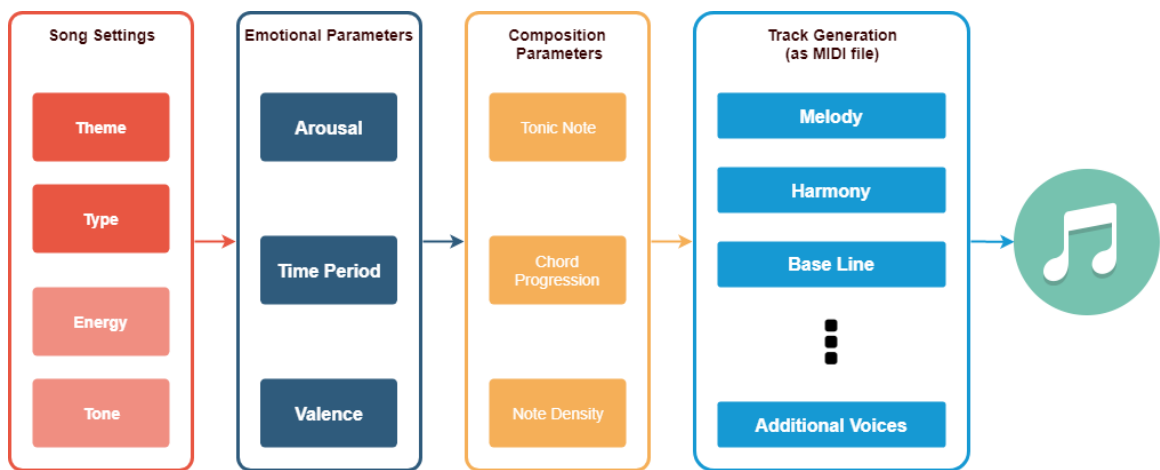


Figure 4.2: Simplification of musical composition feature dependencies

4.1.2 Harmonic Structure

First, the resolved **valence** value is manipulated to decide the level of *lightness* or *darkness*, which then resolves to the final mode the composition will be written in. Secondary and third modes are also selected to allow for *modal interchange* that allows the composition generation space to sound more gradated between the modes. The algorithm is as follows:

```
Input      : The resolved valence val
Output    : The first, second, and third most appropriate church mode
modes = [Phrygian, Aeolian, Dorian, Mixolydian, Ionian, Lydian];
first =  $\lfloor (val + 1) / 2 \rfloor * modes.length$ ;
stren =  $((val + 1) / 2) \bmod (1 / modes.length) * modes.length$ ;
if val  $\geq 0$  then
  | pos = true;
else
  | pos = false;
  | stren = 1 - stren;
end
if pos = true and stren > 0.5 || pos = false and stren  $\leq$  0.5 then
  | second = modes[i + 1];
  | third = modes[i - 1];
else
  | second = modes[i - 1];
  | third = modes[i + 1];
end
return first, second, third;
```

Algorithm 1: Resolving mode from given *valence* value

Note that on the fifth line, *stren* is a measure of the normalized value indicating the strength of choice of mode. This is used to calculate the second and third choice of mode. From here, the next compositional step includes setting up the harmonic skeleton of the piece: deciding the song form and the chord progressions within each section of the form.

4.1.3 Chord Sequence Generation

Given the resolved mode, we can apply diatonic music theory rules to a parameterized, random generation of chord progressions. Given the rule structure of chord progression theory, designing an algorithmic grammar made the most sense for this section of generation.

Each scale degree's chord in a given mode has an assignable quality: *tonic*, *tonic substitution*, *dominant*, *dominant substitution* *subdominant*. The classification mostly relies on whether or not the chord contains *fundamental* or *characteristic* tones: the former tends to sound mode-agnostic, while the later tends to establish to the ear that we are definitely in the given mode. (See the Music Theory section in the Background for review: 2.3). The corresponding chord number classifications are as follows for the Ionian (Major) mode:

Quality	Chords	Rules
T: Tonic	I	Home base; only the root chord
T2: Tonic Sub.	III, VI	Contains root note, no characteristic tone
S: Subdominant	II, IV	Non-Cadential: no root or characteristic tone
D: Dominant	V	Strong Cadential: defining character tone
D2: Dom. Sub.	V, VII	Secondary Cadential: contains character tone

See appendix A.1 for the associated classifications of each mode's chord numbers.

For the purposes of this user study, **A B A B** was the only song form used. The system is capable of producing more forms, but due to the short user study format, this was left out.

With these classifications, we can construct a larger chordal structure. Within chord progressions, there are sections known as *closed* = C and *open* = O passages. This is where grammars lend well; each of these consist of a specific ordered collection of the above abstract chord quality classifications. Using string replaceall, we can string together a variety of these progressions that are both varied and harmonically correct. Each **A B** section consisted of an *origin*, which was either: C C, C O, C O C, or C O O C. Following are a few examples of what a closed or open passage could chordally consist of:

1. Closed

- T S D T2
- T D S T2
- T D2 D S T2
- T S D T2 S T

2. Open

- D T S D
- D2 T2 S D
- D T D S D2
- D T2 D T2 S D

Notice that *closed* passages generally start and end on a **tonic** chord. This is due to the general feeling of “resolution” that this affords listeners. Alternatively, *open* passages generally start and end on a **dominant** chord; this chord passage leads listeners on a departure from the “home base” of the music, and leaves them wanting to resolve to tonic.

With this, an example of a fully resolved **A** section could look like this:

$$\mathbf{A} \Rightarrow \mathbf{C} \ \mathbf{O} \ \mathbf{C} \Rightarrow [\mathbf{T} \ \mathbf{S} \ \mathbf{D} \ \mathbf{T}2] \ [\mathbf{D} \ \mathbf{T} \ \mathbf{S} \ \mathbf{D}] \ [\mathbf{T} \ \mathbf{D}2 \ \mathbf{D} \ \mathbf{S} \ \mathbf{T}2] \ :||$$

4.1.4 Melodic Algorithm

The melody generation algorithm had the most complex rules applied in terms of rhythm and pitch decisions. After an appropriate rhythm was generated for a given musical measure, the pitches were assigned to the rhythmic subdivision of the measure according to standard melodic line rules as represented by a combination of distributions.

```

Input      : Empty base rhythm sequence rhythm
Output    : Full rhythmic sequence rhythm
rhythm = 1/4 note subdivision of measure;
tries = 0;
while tries++ < 15 and not thresholdMet do
  | density = rhythm / measureSubDivision;
  | if density ≤ densityThreshold then
  | | subdivide first largest section in rhythm;
  | end
end
return rhythm;

```

Algorithm 2: Assigning subdivisions within certain threshold for melodic rhythm

Rhythm Generation

There is also a case in which the first two chord progression's rhythms are repeated throughout if the *arousal* value is sufficiently high.

Pitch Contour Generation

The randomly generated input binary string serves as the generative seed for the base pitch number string. This base pitch number string is then adjusted by passing through the following rules, each weighted against one another as distributions. The included subset of composition rules for melodic line are as follows:

- Remove third+ repeating note
- Force a 2nd interval jump if a FARJUMP has occurred (more than a 3rd interval)
- Added dissonance for interest
- Biased range: notes pass if within a certain range from one another
- Close Jumps interspersed among Far Jumps

Each rule was given a distribution and later normalized and combined, weighted as follows:

- CLOSEJUMP = $2 * (1 - arousal) + 1$
- FARJUMP = $2 * arousal$
- DISSONANCE = $3 * valence$
- BIASRANGE = $2 * arousal + 1$

4.1.5 Rhythmic Criteria

Time Signature

Common time, $\frac{4}{4}$ was the default time signature for all valence values above 0.5. For a lighter, bouncier feel, waltz time, $\frac{3}{4}$, was used instead. Some responses in the results section even correctly noted a “bouncier” feel when this time signature was used.

The accompaniment and bass line rhythms were kept simple for this user study; the accompaniment could either be split with the first lowest note on the first quarter and the rest of the notes played as a chord on the second quarter of the measure, or an arpeggiation variation with the bass note played on the first quarter, and the rest of the notes played on the second and third quarters in a randomly chosen pattern.

4.2 Generation Process in Practice

4.2.1 Non-Player Character Attribute Selection

Given the system inputs, developers would decide which NPC attributes they would like reflected in the music. See the this section in Experiment Design for how attributes were selected for this use case: 5.2.1.

These numerical values would then be aggregated influence whichever musical aspect the developer decides would be the best to represent the given attribute. They also are able to select a general *theme* and *time period* if they rather generate a suite of generalized tracks that don't necessarily relate directly to a character's attributes.

4.2.2 Dynamic Attributes on Repeat Encounters

As the *energy* and *valence* values are calculated from a JSON file that is dynamically updated by the game, the generated music will adapt to player input upon return to the active locale for the piece (the NPC bosses in this case). Given the multi-variable input to the generation algorithm, the music will still stay thematically consistent despite drastic changes to the NPC attributes. To maintain thematic consistency in this user study, the same generative seed for each respective boss was used in the final boss scenarios players teleported to after "eating" the one of the four random cakes.

4.2.3 Mapping of Attribute to Musical Feature

Figure 4.3 represents the most appropriate set of static and dynamic NPC attributes for the game used in this user study, and illustrates the flow of how and where these attributes affect the generated music.

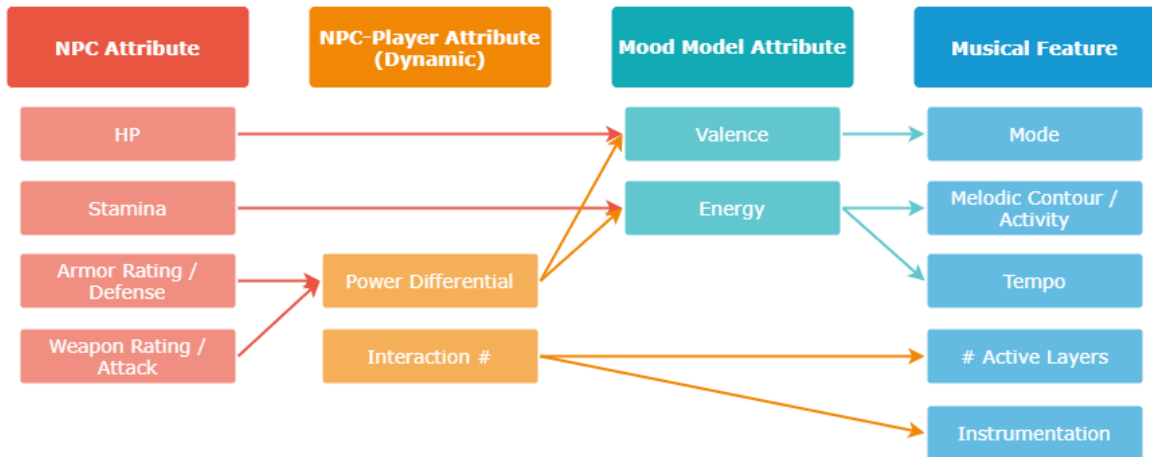


Figure 4.3: Illustrates the dynamic relationships of game features to musical features within the generation system

4.2.4 Building and Running the Program

To reflect any changes made to the ‘music-builder’ module in the game, run the npm script to webpack the TypeScript files for use with JavaScript and Phaser.

NPM RUN BUILD

This should compile the necessary functionality all into the file `JS/THING.JS`. In this case, the actual compositions being played are being accessed via the browser’s `WINDOW` attribute. The compositions are built then made accessible in the `MUSIC-BUILDER/MAIN.TS` file, then called from the appropriate locations in `JS/CLIENT/GAME.JS` file.

Phaser Quest is a reproduction of Mozilla’s Browserquest using the following tools:

- The Phaser framework for the client
- Socket.io and Node.js for the server and client-server communication

Chapter 5

EXPERIMENTAL DESIGN FRAMEWORK

5.1 Case Study Hypotheses

For the user study, participants used the previously described Phaserquest game with a Webpacked version of my system to test the dynamic music generation system as if it was used as a tool in development. The hypotheses to be tested are as follows:

- Does this system produce relevant compositions for the corresponding NPCs?
- Does the generated composition sound musically expressive?
- Did the music appear to dynamically adapt to player-instantiated changes or to relationship changes to NPCs?

5.2 Game Environment Design

To allow for a relatively streamlined playthrough of an open-world exploration game, I kept the map design vertical to allow for progression that generally scales up in intensity as the player progresses north (see figure 5.1). To keep the participants focused on the generated music and how their interactions could be affecting it, keeping the map simple to eliminate potential distractions was reasonable.

The end goal was to re-encounter one of the four bosses the players visited as they progressed through the four environments; at the end of the map, I placed four identical cakes for players to select one. Each cake randomly spawned the player in one of four final boss rooms (see figure 5.2) after being “eaten”; the intention was for players to recognize solely based on the music which boss lay in wait for them at the end of the hallway. From here, the experience

would have afforded participants enough information to give feedback on whether or not the music successfully indicated any of the boss attributes as they relate to their own avatar.

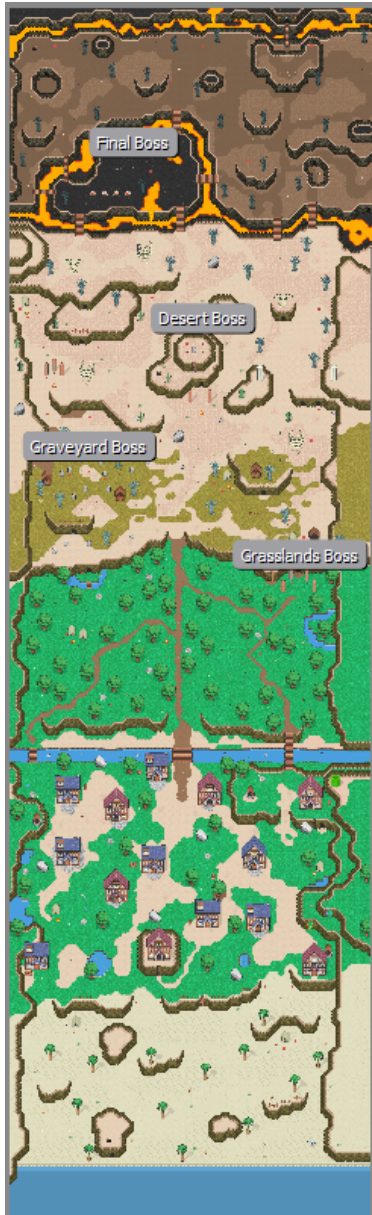


Figure 5.1: Full Phaserquest map. Note the general vertical progression structure.



Figure 5.2: Final boss scenarios. View frustum of player is zoomed in, so they cannot initially see which boss they are about to face.

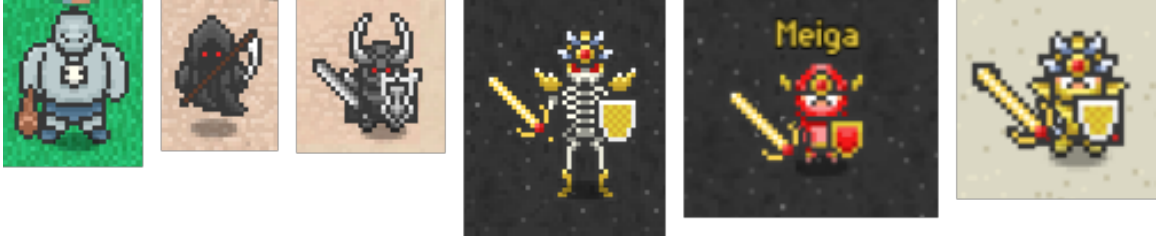


Figure 5.3: Left to right: Ogre, Spectre, Deathknight, and Lavaboss; second-to-best gear player, maxed-out player.

5.2.1 NPC Attribute Selection

Four differing configurations of boss character attributes were developed to exemplify the potential range of music this system can generate. In preparation for further development, the algorithm for generating the music references all relevant NPC and player attributes from a single exposed JSON file from the game’s base code, which adjusts these attributes in real-time. As the system responds dynamically to the NPC attributes as they specifically relate to the player’s progression in the game, each example character was placed in four separate areas of the map: this creates the opportunity for the player to return to the boss NPC after progressing their equipment loadout further, therefore affecting the dynamic parameters of the music generation system. See the following figure 5.3 for the collection of boss NPCs, along with two examples of a player’s sprite:

5.2.2 Player Progression Influence

One of the user study goals was to noticeably align a given aspect of the generated music with the power differential between the player and the regional boss they were closest to. To minimize confusion for my participants, power differential between the player and given NPC was the only player-related attribute that affected the music generation, as many music generation parameters had already been tied to the region and boss-only characteristics.

5.3 User Study Structure

After briefing players on how to play the game, they are allowed to switch between the survey questions and exploring the open world game. This is to allow for deeper consideration of how the music may be interacting with the game and vice versa, as the nuances of gameplay and audio feedback that are important to keep in mind while answering the survey may be easy to forget. The entire experience can take anywhere between 10-35 minutes to complete a full playthrough and to complete the survey.

5.3.1 Basis Questions

This section of the study aims to establish a baseline understanding of whether or not the hypotheses previously described achieve the expected result. Here, participants are first asked via free response if they are able to identify if *tempo*, *melody*, *harmony* / *accompaniment*, and *musical mode* represent anything about the game state, and if so, to describe how so. Following this, participants are given more guidance with a multiple choice question that includes concrete attributes that affect the music generation, of which they can select multiple and write-in additional thoughts.

5.3.2 Musical Quality Questions

The framework used to evaluate musical quality is adapted from Scirea et al.’s evaluation work for their own generative music system, *MetaCompose*, which contains a highly suitable set of qualifiers for this system that cover a broad range of ways to rate music.

Pleasantness at its core evaluates how easy a piece is to listen to, or how pleasing to the listener’s ear it is. Alone, it does not account for what a listener could consider to be “good” music, in which certain harmonic dissonances that are considered “incorrect” or unconventional in diatonic music theory may be considered *interesting* to the listener. [45]

Randomness accounts for how unstructured or dissonant a piece appears to the listener. It gathers how much the listener “perceives there to be a cohesive quality and internal structure” [45]. It is intended to measure how much the listener can discern the underlying musical structure, or whether it is clouded by an over-weighted randomness.

Harmoniousness Not to be confused with *pleasantness*, this is intended to be a more objective measure: “less of a personal preference and more of an ability to recognize the presence of dissonances and harmonic passages.” [45].

Interestingness is intended to extend the *pleasantness* measure and overcome the limitations inherent in its definition. This accounts for when a piece is somehow interesting to the user, even when it is lacking in *pleasantness* or *harmoniousness* [45]. On the scale of subjectivity to objectivity, *interestingness* is the most subjective, followed by *pleasantness*, and lastly *harmoniousness* is the most objective measure.

Dynamism accounts for whether or not a listener was able to detect significant change in a generated piece of music upon returning to its emitter. For example, in the user study, a player would first visit a boss NPC with a large power differential between themselves and the NPC; later, they would return to a slightly varied but thematically similar piece of generated music once they have obtained better gear, therefore reducing the power differential between themselves and the boss they confronted for the second time.

5.3.3 Gameplay Questions

This section was included to catch any discrepancies that may have arose with the game itself while participating in the study, and to allow for feedback that may have otherwise clouded the relevant responses. As the goal was to gather insight to the generated music and interaction with game progression and understanding, this helps to partition all the feedback into higher and lower importance. To improve upon future studies and experiences with this system, this feedback was helpful, though not as pertinent to the study.

Chapter 6

RESULTS AND EVALUATION

6.1 Survey Questions

The survey designed for the user study was reviewed and verified by the Institutional Review Board (IRB) representatives at Cal Poly, SLO. The questions are grouped based on the topic they primarily concern, and each is followed by their responses. Note that in figure 6.3, *CSC* stands for Computer Science, *CPE* for Computer Engineering, *LAES* for Liberal Arts and Engineering Studies, *MATH* as its autological, *CM* for Construction Management, *BMED* for Biomedical Engineering, and *COMS* as Communication Studies.

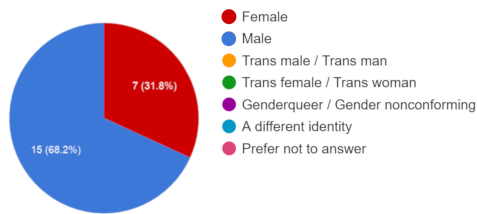


Figure 6.1: Gender distribution

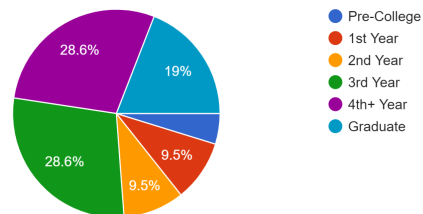


Figure 6.2: Academic status

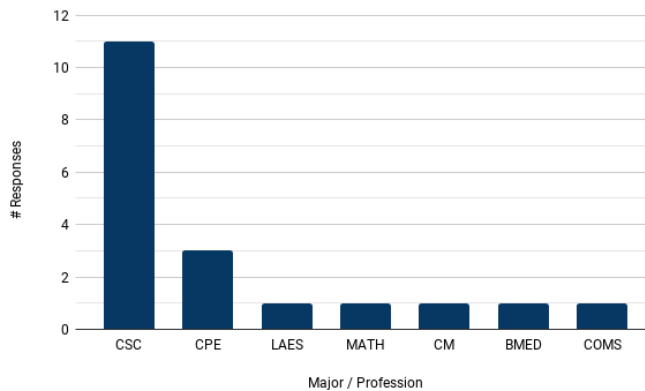


Figure 6.3: Major and / or profession distribution

6.1.1 Basis Questions

These questions largely focused on participants' general interpretation of the generated music as it related to gameplay. To gauge whether or not participants were able to pick up on the independent musical aspects as related to in-game parameters, they were split into tempo, melody, harmony / accompaniment, musical mode, and instrumentation. The last question was a baseline to see whether or not participants were able to distinguish any relationship between the music and the NPC characteristics.

I: *Tempo*: do you think it represented anything?

1. Risk-Based Responses

- “Faster = more danger or close to boss area, slower = more passive or overworld”
- “The danger level in an area - higher tempo = greater danger”

2. Environment-Based Responses

- “Faster tempo for areas with enemies, slower for village, etc”
- “Yes, tempo felt very explorer-like, making the character feel like a wanderer”
- “Represents how eventful the area is. However, can't tell what triggers change”

3. Gameplay and Feel-Based Responses

- “The closer I got to a goal, tempo changed”
- “How intense the game is; it felt slower earlier on”

Two participants of 23 were unable to tell a difference in tempo as they played through. For this category, the ideal response would have related to *boss speed*. All responses generally fell into one of three categories: risk, environment, or gameplay and feel.

For a full list of responses, see the tempo appendix B.1.1.

II: *Melody*: do you think it represented anything?

1. NPC-Based Responses (*Ideal*)

- “Changed depending on which environment you were in, or which boss/character you were encountering”
- “The melody related to either the characters or the various levels when entering a new environment”
- “Feels a little forlorn, like the player character is fighting monsters after the monsters have already won”

2. Risk-Based Responses

- “Peaceful melodies for safe spots, more menacing ones for dangerous areas”
- “Amount of action and area in world”
- “Most of the songs were very soft, despite the game showcasing an action adventure, it felt more relaxing than anything to explore and hack away at monsters”

Four participants of 23 were unable to discern whether or not the melody changed according to in-game events. This was the musical aspect that most participants noticed was tied to the NPC attributes, noting the most success in indicating the appropriate boss characteristic in their thematic music to the players. Many attributed the melodic changes, again, to environmental characteristics and location-based changes. For this category, the ideal response would have related to *NPC friendliness and / or disposition*. All responses generally fell into one of three categories: NPC-related, environment, progression, or risk.

For a full list of responses to this question, see the melody appendix B.1.2.

III: *Harmony / Accompaniment*: do you think it represented anything?

1. NPC-Bases Responses

- “Harmony represented the enemies”

2. Progression-Based Responses

- “Encouraged me to keep going, because it was continuous and pleasant”
- “Rising tension near the final boss battle and comfort when in the starting town”
- “It conveyed a happy tone, like nothing’s wrong with the world. An escape”

3. Environmental-Based Responses

- “I think it represented the mood of each environment; western for the desert areas, fresh and reviving for the forest area”

4. Risk-Based Responses

- “It seemed to diminish / disappear after defeating a boss, but that may just have been timing”

5. Gameplay and Feel-Based Responses

- “Nothing disquieting, but let you know when the game is intensifying or not”
- “Calm game phases without conflict / fighting”

Seven participants of 23 were either unable to distinguish what the harmony / accompaniment was, or were unable to discern whether or not it represented anything. For this category, the ideal response would have related to *boss life, or HP / strength*. All responses generally fell into one of three categories: NPC-related, environment, risk, gameplay and feel, and non-game related (music only).

For a full list of responses to this question, see the harmony appendix B.1.3.

IV: *Musical Mode*: do you think it represented anything?

1. “It was represented when you encountered an enemy; the tempo increased and it felt as if there was an attacker. When in the village, the beat was very calm, kind, and comfortable, like the village; there were no skeletons or enemies.”

2. “Helped to make environments feel more distinct, adding to the progression of the game, as quick and limited as it was”

3. Ideal Response:

- NPC tendency

Half of my participants (11 out of 23) were unable to discern the musical mode, which is fair; many that have experience with music analysis still have a difficult time identifying a piece of music’s exact musical mode. The responses of those who were able to distinguish a mood as derived from the mode had a difficult time discerning what aspects of the game the mode actually related to according to the generative algorithm. For this category, the ideal response would have related to the respective *NPC’s tendency*. All responses generally fell into one of three categories: NPC-related, environment, risk, gameplay and feel, and non-game related (music only).

V: *Instrumentation*: do you think it represented anything?

1. NPC-Based Responses

- “Different instruments for each area due to different enemies in each area”
- “Different characters”

2. Gameplay and Feel-Based Responses

- “Felt as if the instruments were similar to cultures represented by the setting, like sitar in the sand area and something similar to lute in the starting town”

3. Environment-Based Responses

- “The instruments in a sense corresponded to the environments that they were being played in. The early songs playing used instruments to convey a mellow tone, while the later songs were not as cheery but still tried to match their respective environments”
- “I think the instruments sounded similar to the ”time period” of the game. Fantasy games usually take place in a medieval setting”
- “The instruments felt fitting for the different levels, regions, and intensity levels”

Five of my 23 participants were either unable to tell or had no extra insight as to what instrumentation may represent; many assumed that it was “standard” video game music instrumentation. Recall from figure 4.3 this related to encounter number and number of active voices or layers.

For the accompanying free response question verifying whether or not players detected any correlation between the layered voices in the music and the in-game attributes, see appendix B.1.6.

VI: Did the music help indicate any of the following?

- Boss HP
- Boss Stamina
- Boss Weapon Rating / Attack
- Boss Armor Rating / Defense
- Power Differential (between you and the boss)
- Interaction Number (first, second, third time you encountered the boss)

A about a quarter of participants correctly identified the *Power Differential* between themselves and the boss NPCs as the key in-game characteristic that controlled the generated

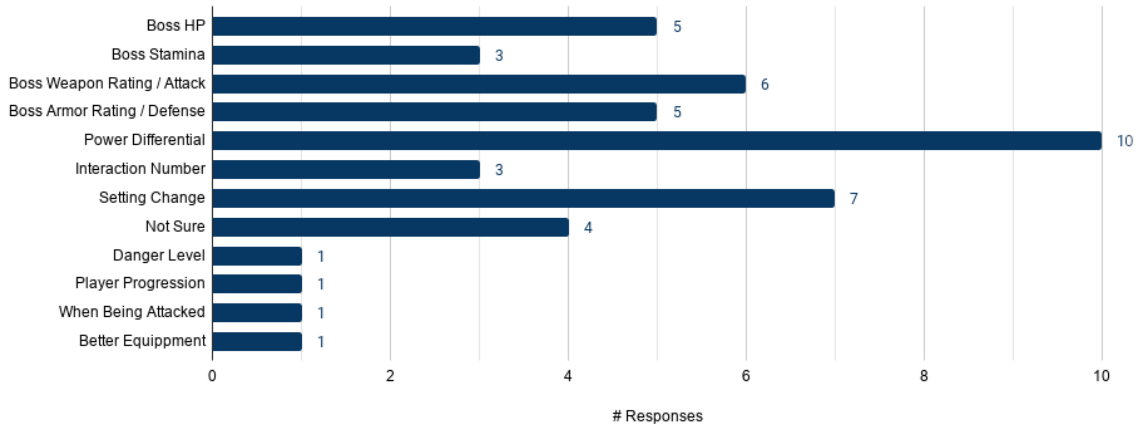


Figure 6.4: Total responses including write-in answers

music parameters. For further discussion of these results, see the following section: 7.1.1.

6.1.2 Musical Quality Questions

The following questions were solely about musical quality. As the fidelity and integrity of generated music was as equally important as indicating information to players about NPCs in the game through the music, these questions were pertinent. Graphical representations of these responses are included in appendix C.1.

Was the music:	Definition Summary	Average
Pleasing?	Easy to listen to, not necessarily good	7.81 / 10
Random?	How unstructured or dissonant music was	4.57 / 10
Harmonious?	Objective musical "correctness"	7.38 / 10
Interesting?	Subjective preference for generated music	7.76 / 10
Dynamic?	On return to NPC, noticeable variance	8.29 / 10

Recall that the breakdown of musical quality attributes as above are adapted from Scirea et al.'s evaluation work for their own generative music system, *MetaCompose* [45]. The average scores were higher than expected for musical quality. After inverting the score for randomness (as a lower score in this category is preferable, inversion is necessary to match the rest), the *overall score* for musical quality incorporating all attributes is **7.16 / 10** with

a *standard deviation* = $\sqrt{\sigma^2}$ of **4.40** using the following calculation for variance.

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n} \quad (6.1)$$

For a full list of all the free responses regarding the quality of the generated music, see appendix C.2. Most responses were very positive, and felt that certain attributes in-game were reflected in the generated compositions.

6.1.3 Gameplay Questions

For a full list of all the free responses regarding the quality of the generated music, see appendix D.1. The majority of responses were positive and helpful in terms of bug reports.

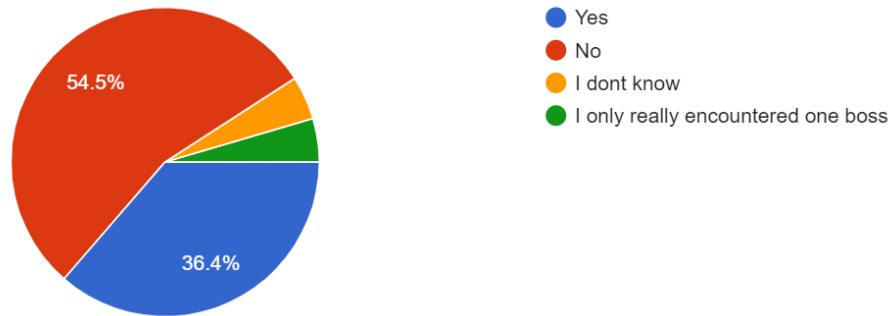


Figure 6.5:
Did the music save you time? As in, did it indicate anything to you that other feedback in the game didn't make immediately apparent to you?

For further discussion of these results, see the following section in the Conclusion: 7.2.3.

Which boss did you *think* you were facing in the final scenario, *versus* the boss you actually faced? Did the music help inform you?

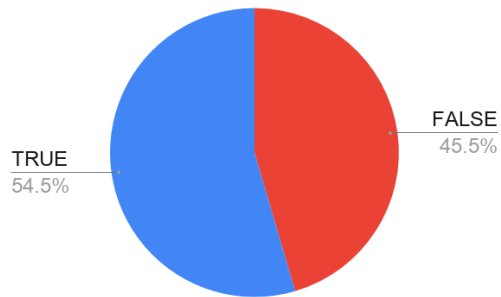


Figure 6.6:
 True: those who guessed the boss they were about to face correctly;
 False: those who didn't

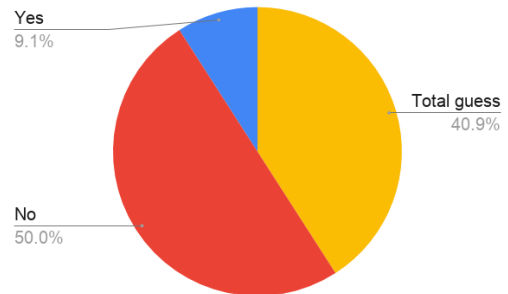


Figure 6.7:
 Did you guess which boss you were about to face based on the music you heard?

Chapter 7

CONCLUSION

7.1 User Study Analysis and Hypotheses Discussion

The first discussion will be on the core topics: whether or not the compositions were relevant to their respective NPCs, and whether or not the music assisted player immersion. The second discussion topic will be on significant trends that consistently presented themselves in the user study responses. With the context of analyzing why these trends were present, we will be better prepared to break down the case study hypotheses alongside the results from the user study, and ultimately discuss whether or not they were proven.

7.1.1 Hypothesis I: Relevant Compositions for NPC Attributes

Power differential was the primary factor that affected the **valence** parameter of a new instantiation of a generated music composition. Recall that the **valence** measure of a piece of music generally accounts for the perceived happiness or sadness of that piece. Conversely, the **energy** level accounts for the perceived arousal or sleepiness of a piece of music. Together, this dyad creates a two-dimensional emotional space that can account for a broad range of emotions [42], all of which can be broadly associated with musical mode, and dialed in with musical details such as tempo, rhythmic choices, instrumentation choices, dynamics, and more (such as was done in this experiment). A majority of my participants were able to identify this attribute as one of the main factors that affected the music generation.

The question that addresses the core of relating a composition to a given NPC's attribute set is question 6.1.1. The characteristics aside from power differential I listed in this multiple

choice question were also in-game attributes that affected the generated composition, but to lesser degrees than the power differential did. For example, a higher *Boss HP* raised the value of **valence**, but to a lesser degree than the *Power Differential*. *Boss Stamina*, or boss speed in this case, was the principal attribute that affected the **energy** value in the generative algorithm; higher speed = higher energy. *Boss Weapon Rating / Attack* and *Boss Armor Rating / Defense* also minimally weighed in on the valence value due to the fact both attributes were being compared to the player's own attack rating (as that is how *power differential* is calculated), but were primarily responsible for **instrumentation** choice. Changes in the music due to *Interaction Number* may have been more difficult to pick up on, due to the fact that this would only be predominant if the player had visited the boss with low attack or low defense (bad weapons and bad armor) first, then returned to visit the boss with better gear (higher attack or higher defense). Recall from figure 4.3, this attribute accounted for instrumentation choices and number of musical layers present.

Given this, I can conclude that the system did in fact generate relevant compositions given specific NPC attributes.

7.1.2 Hypothesis II: Musically Expressive

As the developer, I know that the musical expressivity is limited due to the generative space limitations (such as the limited mode selection, song form, dynamics, and rhythm structures). That being said, the following feed back indicated positive opinions on the expressive quality of the music, therefore showing promising prospects of the expressivity of this system.

- “I thought that the quality of the composition was great. I liked the different areas show all having its own atmosphere and vibe.”
- “The musical mode helped to make environments feel more distinct, adding to the progression of the game, as quick and limited as it was.”

- “Certain regions did feel more ”happy” or ”sad” based on the music”
- “The melody seemed to get more bouncy or intense during certain situations”

With an ample amount of feedback similar to this, I can conclude that the system was in fact musically expressive.

7.1.3 Hypothesis III: Dynamic Adaptation

The main question aimed at addressing this topic asked participants whether or not they were able to notice the music adapting to any changes in their return encounters with the boss NPCs. The average score in this category was **5.28 / 10** with a standard deviation of **37.12**; these results can be deemed inconclusive. For a graphical representation of responses to whether or not the music appeared to be dynamic, see figure C.5 in the appendix.

While this is the case, the binary question of whether or not participants thought that the music for any of their return encounters to the bosses sounded thematically the same, 56.5% of participants said yes. This shows that the parameters affecting the dynamic changes in thematic music generation still require adjustment to reliably be perceived as an adaptive generative system.

7.2 Additional Analysis

7.2.1 Successful Immersion

Many users gave feedback that indicated a positive experience and a fair amount of immersion in the game due to the music.

- “Layering helped the experience to make clear several things are happening simultaneously for example, or a transition from one to another theme.”

- “I was eager to explore different areas and enemies to hear the different music.”
- “Great Job! Very interesting music, helped indicate where I was, who I was around.”
- “I love the composition. It’s so pleasing to listen to and I feel like it fits the game well and the energy I feel while going through the game.”

Others gave highly valuable feedback that directly addressed many of the major issues present with the system, such as ear fatigue from too much repetition and a lack of smooth transition states. Many make a good case for implementing a smaller radius of influence for NPC themes, and incorporating this into a musical transition framework. Others highlighted the need for multi-section song form generation, which does exist in the system, but was not balanced and tuned to vary enough.

- “Repeated very quickly, but there was a lot of variation between levels.”
- “While I might be able to tell that the music is a bit ”off”, it felt very close to music I’d hear in a game and was very impressive. If I didn’t know beforehand that you were working on generated music, I might not have jumped to that conclusion. I thought the overall quality was fairly high even if pieces felt a bit off.”
- “I found most of the generated music unpleasant to listen to not based on melody but because of its robotic nature and tinny instruments. However I found the tune for the area directly after the grasslands (I think it was directly after the grasslands) to be pretty nice actually.”
- “At times the music was very pleasing, other times a little disorienting.”

7.2.2 Setting-Based Confusion

A few participants noted that the change in music was not drastic enough beyond the biome / region changes to notice a difference in the music. Much of the environmental-based responses can be attributed to the fact that the areas in which players had their first interactions the bosses were very expansive. This may have caused an overload in

uncontrolled variables for the players, which may have clouded their own insight as to what aspects of the game were actually responsible for the changes in the generated music.

7.2.3 Beyond Hypotheses: Saved Time

An additional goal was to test whether or not the music was conducive to gameplay in a deeper manner than just immersion and providing NPC attribute information. If players were able not only to discern certain characteristics of a given boss NPC, but to be able to determine whether or not they are prepared to face that boss, this could be considered an exciting step in a new direction for the capabilities of PMGG. The responses to this question largely indicated that this was not necessarily the case (possibly due to a high volume of things to pay attention to during the experience): 35% of participants believed that the music did serve as a mechanic that saved them time, whereas about half did not. That being said, the following feedback indicated positive results towards this mechanic:

- “Based on the music, I had a feeling I would be facing other bosses because of the previous boss’s music”

Chapter 8

FUTURE WORK

Potential future avenues for this system include honing in certain details of the existing system, and others include using this system as a base to extend research into the field of PMGG. This music generator is separable as a module to be integrated to other JavaScript games or interactive projects; it is also extensible in the way it allows for custom music theory structure that musical developers may be able to apply another paradigm to generation, making it a great addition to the PMGG field. Some notable ideas are as follows:

- Music becomes the mechanic; player uses the generated music as the feedback system of what is happening in-game.
- Music is generated beforehand alongside sound effects (SFX); this system can be used to generate music almost immediately responsive to player input (SFX) and dynamic as well.
- As mentioned in the conclusion, an additional compatible musical transition system would be greatly beneficial to this system. It would expand its potential applications, and potentially address the issue in which users confused boss themes for area themes.

8.1 Improvement: Musical Transitions

An issue I immediately identified during the user study was the lack of musical transitions. Musical transitions in a swiftly-changing environment (such as a video game) is an entire issue unto itself. Many elegant solutions currently exist, each appropriate for the specific in-game use case and atmosphere they were created for. Unfortunately, these solutions were not immediately compatible with this system's design at the time of creation, and

therefore were left out for the time being. That being said, it is a major potential area of improvement for this system: not only would it allow for a more immersive experience, but it may address some of the issues that arose during user testing, namely confusing a boss's theme song for the area theme song. Following are comments that highlight the need for this framework:

- “I enjoyed the music - it sounded like what you'd typically here in a video game like this. I don't know if my session differed from the others, but I found the music didn't correlate with the gameplay in a discerning way other than changing "songs" when I changed regions.”
- “I liked it. I thought it was similar to the music of Pokémon where each city had a soundtrack.”
- “Pretty good overall, would have sounded better if it flowed together more.”

Given a musical transition algorithm, overworld music can seamlessly be tied to a boss's confrontation theme and vice versa. With this, I would reduce the radius of influence around the bosses and label all other areas as the "base" musical context, and use the player's location as compared to these labels as the hook into the musical transition framework.

8.1.1 Generative Space Considerations

As is the case with many procedural generation systems, a major factor in determining unique success is how well the generative space is pruned and presented to the user. Some users noted they experienced ear fatigue from hearing the same track too often, or similar motifs without enough variation; I believe more noticed this as well (even if only sub-consciously) but did not say so in their feedback. These make a case for improving the multi-section generation system, as well as adding rules for more modes and composition formats found around the world.

1. “The music sounds like background music, but it loops a lot, so it would be irritating to listen to for long.”

8.2 Modes to Include

Modes that were a part of the system, but were not included in the user study due to nuances of composition.

- Church (Gregorian) [37]:
 - **Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian**
 - *Note:* **Locrian** not used due to overly-dissonant sounding results
- Non-Church Modes:
 - **Pelog, Phrygian Dominant, Hungarian, Persian**
- 5-note scales in order of apparent brightness, less chromaticism [49]:
 - **Yo, Major Pentatonic, Minor Pentatonic, Insen, Hirajoshi, Iwato**

The following figure 8.1 is what the ideal system data flow would look like if the previous modes are to be included.

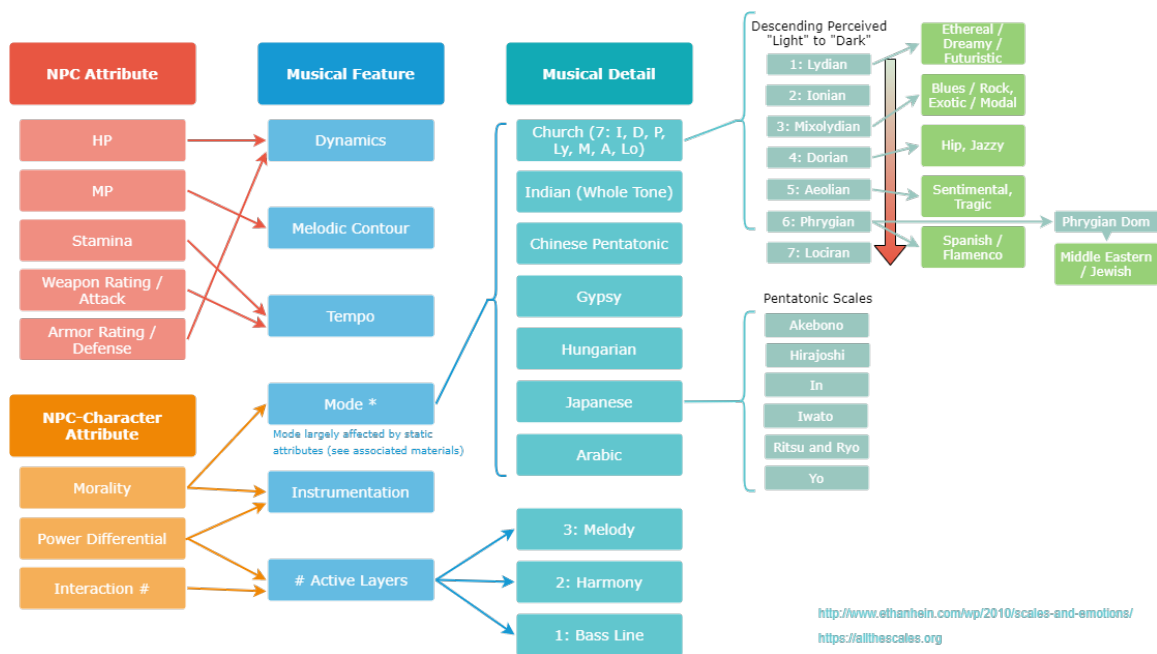


Figure 8.1: Illustrates potential dynamic relationships of game features to musical features within the generation system

BIBLIOGRAPHY

- [1] E. Aarseth. A narrative theory of games. *Foundations of Digital Games Conference Proceedings*, pages 129–133, 2012. Foundations of Digital Play, Raleigh.
- [2] T. Adam, M. Haungs, and F. Khosmood. Procedurally generated, adaptive music for rapid game development. In *FDG 2014 Global Game Jam Workshop, Foundation of Digital Games*, 2014.
- [3] C. Agon, G. Assayag, and J. Bresson. Openmusic, Apr. 1998-2013.
- [4] D. Birchfield. Generative model for the creation of musical emotion, meaning, and form. In *Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence, ETP '03*, pages 99–104, New York, NY, USA, 2003. ACM.
- [5] J. Brantingham. Unlocking the mysteries of diatonic harmony, June 2019.
- [6] I. C. Straehley and J. Loebach. The influence of mode and musical experience on the attribution of emotions to melodic sequences. *Psychomusicology: Music, Mind, and Brain*, 24:21, 3 2014.
- [7] J. Chan, J. J. Daza, W. Kwan, and A. Basu. Facilitating player progression by implementing procedural music in videogames. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2328–2333, Oct. 2017.
- [8] K. Collins. An introduction to procedural music in video games. *Contemporary Music Review*, 28(1):5–15, 2009.
- [9] D. Cope. Recombinant music: Using the computer to explore musical style. *Computer*, 24(7):22–28, July 1991.
- [10] D. Cope. The well-programmed clavier: Style in computer music composition. *XRDS*, 19(4):16–20, June 2013.

- [11] C. Crawford. *The Art of Computer Game Design*. McGraw-Hill, Inc., New York, NY, USA, 1984.
- [12] Danigb. danigb/tonal, Apr. 2019.
- [13] G. Denis and P. Jouvelot. Motivation-driven educational game design: Applying best practices to music education. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Comp. Entertainment Technology*, ACE '05, pages 462–465, New York, NY, USA, 2005. ACM.
- [14] H. Dong, W. Hsiao, L. Yang, and Y. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 34–41, 2018.
- [15] H. Dong and Y. Yang. Convolutional generative adversarial networks with binary neurons for polyphonic music generation. *CoRR*, abs/1804.09399, 2018.
- [16] A. Farnell. An introduction to procedural audio and its application in computer games. In *Audio mostly conference*, volume 23, 2007.
- [17] H. A. García Salas, A. Gelbukh, and H. Calvo. Music composition based on linguistic approach. In G. Sidorov, A. Hernández Aguirre, and C. A. Reyes García, editors, *Advances in Artificial Intelligence*, pages 117–128, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [18] S. D. Gillespie and O. Bown. Solving adaptive game music transitions from a composer centred perspective, 2017.
- [19] J. Grekow. Music emotion maps in arousal-valence space. In K. Saeed and

- W. Homenda, editors, *Computer Information Systems and Industrial Management*, pages 697–706, Cham, 2016. Springer International Publishing.
- [20] M. Grimshaw, C. Lindley, and L. Nacke. *Sound and Immersion in the First-Person Shooter: Mixed Measurement of the Player’s Sonic Experience*.
www.audiomostly.com, 2008.
- [21] N. Holtar, M. Nelson, and J. Togelius. Audioverdrive: Exploring bidirectional communication between music and gameplay. In *Proceedings of the 2013 ICMC Conference: International Developments in Electroacoustics*, pages 124–131. International Computer Music Association, 2013.
- [22] A. K. Hoover, J. Togelius, and G. N. Yannakis. Composing video game levels with music metaphors through functional scaffolding, 2015.
- [23] C.-Z. A. Huang, T. Cooijmans, A. Roberts, A. Courville, and D. Eck. Counterpoint by convolution. In *Proceedings of ISMIR 2017*, 2017.
- [24] M. O. Jewell, L. Middleton, M. S. Nixon, A. Prügel-Bennett, and S. C. Wong. A distributed approach to musical composition. In R. Khosla, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 642–648, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [25] M. Land and P. McConnell. Sound engine: imuse, 2000.
- [26] P. S. Langston. 201 644-2332 or eddie & eddie on the wire: An experiment in music generation. *Peter Langston*, pages 13–27, 1986.
- [27] C. Liu and C. Ting. Computational intelligence in music composition: A survey. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(1):2–15, Feb. 2017.
- [28] J. McCartney. Supercollider, 1996-2019.
- [29] G. Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer Publishing Company, Incorporated, 1st edition, 2008.

- [30] C. o. Nuanáin. A brief history of machine-assisted music in video games, May 2018.
- [31] S. Oore, I. Simon, S. Dieleman, and D. Eck. Learning to create piano performances. In *NIPS 2017 Workshop on Machine Learning and Creativity*, 2017.
- [32] G. Papadopoulos and G. Wiggins. Ai methods for algorithmic composition: A survey, a critical view and future prospects. In *IN AISB SYMPOSIUM ON MUSICAL CREATIVITY*, pages 110–117, 1999.
- [33] C. Pedersen, J. Togelius, and G. Yannakakis. Modeling player experience for content creation. *Computational Intelligence and AI in Games, IEEE Trans. on*, 2:54–67, 4 2010.
- [34] W. Phillips. Video game music systems at gdc 2017: what are composers using?, July 2017.
- [35] D. Plans and D. Morelli. Experience-driven procedural music generation for games. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(3):192 – 198, 2012.
- [36] J. Posner, J. A. Russell, and B. S. Peterson. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology, 2005.
- [37] D. Ramos, J. Bueno, and E. Bigand. Manipulating Greek musical modes and tempo affects perceived musical emotion in musicians and nonmusicians. *Brazilian Journal of Medical and Biological Research*, 44:165–172, 2 2011.
- [38] A. C. Roads, P. Wieneke, and C. Roads. Grammars as representations for music. *Computer Music Journal*, pages 48–55, 1979.
- [39] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning (ICML)*, 2018.

- [40] J. D. F. Rodriguez and F. J. Vico. AI methods in algorithmic composition: A comprehensive survey. *CoRR*, abs/1402.0585, 2014.
- [41] T. Sanders and P. Cairns. Time perception, immersion and music in videogames. In *Proceedings of the 24th BCS Interaction Specialist Group Conf.*, BCS '10, pages 160–167, Swinton, UK, UK, 2010. British Computer Society.
- [42] E. Schubert. Measuring emotion continuously: Validity and reliability of the two-dimensional emotion-space. *Australian Journal of Psychology*, 51(3):154–165, 1999.
- [43] E. Schubert. Modeling perceived emotion with continuous musical features. *Music Perception: An Interdisciplinary Journal*, 21(4):561–585, 2004.
- [44] M. Scirea, Y.-G. Cheong, M. J. Nelson, and B.-C. Bae. Evaluating musical foreshadowing of videogame narrative experiences. In *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, AM '14, pages 8:1–8:7, New York, NY, USA, 2014. ACM.
- [45] M. Scirea, P. Eklund, J. Togelius, and S. Risi. Can you feel it?: Evaluation of affective expression in music generated by metacompose. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 211–218, New York, NY, USA, 2017. ACM.
- [46] M. Scirea, P. Eklund, J. Togelius, and S. Risi. Evolving in-game mood-expressive music with metacompose. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, AM'18, pages 8:1–8:8, New York, NY, USA, 2018. ACM.
- [47] M. Scirea, P. Eklund, J. Togelius, and S. Risi. Towards an experiment on perception of affective music generation using metacompose. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 131–132, New York, NY, USA, 2018. ACM.

- [48] M. Scirea, J. Togelius, P. Eklund, and S. Risi. Metacompose: A compositional evolutionary music composer. In C. Johnson, V. Ciesielski, J. Correia, and P. Machado, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 202–217, Cham, 2016. Springer International Publishing.
- [49] K. Shaffer and R. Wharton. Open music theory, June 2014.
- [50] M. Simoni. *Algorithmic composition: A gentle introduction to music composition using LISP and common music*. Michigan Publishing, University of Michigan Library, Ann Arbor, MI, 1 2003.
- [51] A. Sorensen and A. Brown, 1990-2009.
- [52] W. Strank and P. Moormann. *The Legacy of iMuse: Interactive Video Game Music in the 1990s*, pages 81–91. Springer Fachmedien Wiesbaden, Wiesbaden, 2013.
- [53] B. L. Sturm, O. Ben-Tal, Úna Monaghan, N. Collins, D. Herremans, E. Chew, G. Hadjeres, E. Deruty, and F. Pachet. Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, 48(1):36–55, 2019.
- [54] WalkThatBass. Modal interchange and borrowed chords.
- [55] H. Warpefelt. *The Non-Player Character : Exploring the believability of NPC presentation and behavior*. PhD thesis, Stockholm University, Department of Computer and Systems Sciences, 2016. At the time of the doctoral defense, the following papers were unpublished and had a status as follows: Paper 5: Manuscript. Paper 6: Manuscript.
- [56] G. Wiggins. Computer models of musical creativity: A review of computer models of musical creativity by david cope. *Literary and Linguistic Computing*, 23, 12 2007.
- [57] R. Wooller, A. R. Brown, E. Miranda, J. Diederich, and R. Berry. A framework for comparison of process in algorithmic music systems. *Creative Industries Faculty*, 2005.

APPENDICES

Appendix A

MUSIC THEORY

A.1 Harmonic Functions of Mode's Chords

1. Lydian

- **Tonic Substitute:** I, VI
- **Subdominant:** III
- **Dominant:** II
- **Dominant Substitute:** II, IV, V, VII

2. Ionian: Covered in System Design, section 4.1.3

3. Mixolydian

- **Tonic Substitute:** I, II, IV, VI
- **Subdominant:** II
- **Dominant:** VII
- **Dominant Substitute:** III, V, VII

4. Dorian

- **Tonic Substitute:** I, VI
- **Subdominant:** III, V, VII
- **Dominant:** IV
- **Dominant Substitute:** II, IV, VI

5. Aeolian (Minor)

- **Tonic Substitute:** I, III, VI
- **Subdominant:** II, IV
- **Dominant:** V
- **Dominant Substitute:** V, VII

6. Phrygian

- **Tonic Substitute:** I, IV, VI
- **Subdominant:** IV
- **Dominant:** III
- **Dominant Substitute:** II, III, V, VII

7. Locrian

- **Tonic Substitute:** I, IV
- **Subdominant:** II, VII
- **Dominant:** VI
- **Dominant Substitute:** III, V, VI

Appendix B

BASIS QUESTIONS

B.1 Free Response Questions

B.1.1 *Tempo*: do you think it represented anything?

1. Risk-Based Responses

- “Faster tempo gave a more dangerous vibe to the game”
- “I think it represented the safety level”
- “Faster = more danger or close to boss area, slower = more passive or overworld”
- “Helped get across the intensity for different regions of the game”
- “The tempo was slow and represented a feeling of calmness”
- “The danger level in an area - higher tempo = greater danger”

2. Environment-Based Responses

- “Faster tempo for areas with enemies, slower for village, etc”
- “Yes, tempo felt very explorer-like, making the character feel like a wanderer”
- “Might represent how eventful the is. However, can’t tell what triggers a change”
- “May have differentiated the different levels of the zones”
- “I think it signified the mood of the current setting you were in”
- “Vibe of the environment; associated each soundtrack with respective area”
- “Fit the area, preconceptions of grass/sand/cave area”

3. Gameplay and Feel-Based Responses

- “The closer I got to a goal, tempo changed”
- “How intense the game is; it felt slower earlier on”

- “I believe it represented the current state the character was in”
- “Helped set the pace, I felt that faster tempo songs made me want to go faster”
- “Adventure, it feels similar to a marching pace”
- “It feels standard. It wasn’t too fast paced, or slow either. The perfect balance between getting the player to move forward but also casually explore”

4. **Ideal Response:**

- Boss Speed

B.1.2 *Melody: do you think it represented anything?*

1. NPC-Based Responses (*Ideal*)

- “Changed depending on which environment you were in, or which boss/character you were encountering”
- “The melody related to either the characters or the various levels when entering a new environment”
- “Feels a little forlorn, like the player character is fighting monsters after the monsters have already won”
- “Maybe characters’ emotions?”

2. Environment-Based Responses (*Secondary Ideal*)

- “Depending upon where I was, melody changed”
- “The area I was in”
- “The melody feels like it fit the environment I am in. There’s definitely a change in the melody from the grassy area and the desert area”
- “Represented the scene you are in”
- “The general theme of the game environment, like in the desert or in the city in the game”

- “It helped set the tone for the scene, but I felt as though it was almost random; felt quite consistent”
- “Region”

3. Progression-Based Responses

- “How close you were to the end of the game”
- “Yes, seemed to get more bouncy or intense during certain situations”

4. Risk-Based Responses

- “Peaceful melodies for safe spots, more menacing ones for dangerous areas”
- “Amount of action and area in world”
- “Same as tempo, with preconceived notions of ‘beginning area’ music”
- “The melody was disjointed, but was still calming”
- “Most of the songs were very soft, despite the game showcasing an action adventure, it felt more relaxing than anything to explore and hack away at monsters”

5. **Ideal Response:**

- NPC Friendliness / Disposition

B.1.3 *Harmony / Accompaniment:* do you think it represented anything?

1. NPC-Bases Responses

- “Harmony represented the enemies”

2. Progression-Based Responses

- “Encouraged me to keep going, because it was continuous and pleasant”
- “Rising tension near the final boss battle and comfort when in the starting town”
- “I want to say it conveyed a happy tone, like nothing’s wrong with the world. An escape”

3. Environmental-Based Responses

- “I think it represented the mood of each environment; western for the desert areas, fresh and reviving for the forest area”
- “Yes, depending upon where I am harmony changes”
- “Area in world”
- “The area I was in”
- “Different areas were outfitted with different music”

4. Risk-Based Responses

- “It seemed to diminish/disappear after defeating a boss, but that may just have been timing”

5. Gameplay and Feel-Based Responses

- “The harmony added to the immersion of the game”
- “Nothing disquieting, but just to let you know when the game is intensifying or not”
- “Calm game phases without conflict/fighting”

6. Inconclusive Game Association

- “Helped create a richer sound but didn’t feel like it represented much”
- “I feel like the harmony represented jauntiness”

7. Ideal Response:

- Boss life, HP / strength

B.1.4 *Musical Mode*: do you think it represented anything?

1. “It was represented when you encountered an enemy; the tempo increased and it felt as if there was an attacker. When in the village, the beat was very calm, kind, and comfortable, like the village; there were no skeletons or enemies.”

2. “Set the mood or atmosphere?”
3. C major/A minor in grass area - peaceful
 B major in cave area - not sure
 A major/F# minor
 A major string ensemble
 - *Note: This is very close to true, but not quite the exact modes within the generative space for the respective NPC areas*
4. “Each different area had a different musical mode”
5. “Helped to make environments feel more distinct, adding to the progression of the game, as quick and limited as it was”
6. “ Not sure if I can tell, but certain regions did feel more ”happy” or ”sad” based on the music so I believe the mode was affecting it”
7. **Ideal Response:**
 - NPC tendency

B.1.5 *Instrumentation:* do you think it represented anything?

1. NPC-Based Responses
 - “Different instruments for each area due to different enemies in each area”
 - “Different characters”
2. Gameplay and Feel-Based Responses
 - “Felt as though the instruments were similar to cultures represented by the setting, like sitar in the sand area and something similar to lute in the starting town”

- “I think it represented a very explorer and energetic like feeling which matched the character and the minimal detail in the game. The music also felt very pixelated and short, just like the game and it’s visuals”
- “The instrumentation added to the feeling that the scene was suppose to give”

3. Environment-Based Responses

- “The instruments in a sense corresponded to the environments that they were being played in. The early songs playing used instruments to convey a mellow tone, while the later songs were not as cheery but still tried to match their respective environments”
- “Exotic level of a region? There were interesting instrument choices for the dessert (sounded like a Japanese instrument)”
- “Oboe: peaceful, grass biome
Banjo: excitement, desert biome”
- “The setting and the time period in the game”
- “Set the time and atmosphere”
- “I think the instruments sounded similar to the ”time period” of the game. Fantasy games usually take place in a medieval setting”
- “The instrumentation changed depending on the zone/environment”
- “The type of area you’re currently in”
- “The instruments felt fitting for the different levels, different regions, and intensity levels”
- “The area I was in”
- “Fit the area it was in”
- “The instrumentation represented the stage I was in”

B.1.6 Layering of Voices

Did you notice layering in the generated music (multiple lines and / or voices)? If so, do you think this correlated to anything, or helped and / or hindered the experience?

- “This helped to make the game play more interesting and it intrigued me, I was eager to explore different areas and enemies to hear the different music”
- “Maybe the number of opportunities available”
- “Yes, I heard the main melody phase in and out periodically. I couldn’t figure out what their significance was in the game - the transitions seemed to happen without any particular in-game cause”
- “Yes I definitely notice it. I’m sure they correlated with something but I don’t know what”
- “I didn’t particularly notice any layering; though if I missed it, it certainly didn’t hinder anything”
- “I think that multiple lines of music helped the experience because it added different affects to the atmosphere you were in”
- “No, seemed flat tone to me”
- “Layering in terms of different tracks being played at the same time (triggered to overlay), then yes. This helped the experience to make clear several things are happening simultaneously for example, or a transition from one to another theme”
- “Yes, I have noticed layered melodies. I think it depended on the number of enemies in the area and their strength”
- “I think it helped”
- “More enemies equaled more instruments, harder enemies had more instruments”
- “I did notice it, but I did not notice a correlation besides how recently I was in combat”
- “It felt like just more 16bit old-timey RPG music”
- “I think so, it helped build on the experience incrementally and made it more inter-

esting to see how it built up”

- “It helped set tone, I’m no music buff, but I believe the accompaniment changes the key of the main melody typically from major for comfort and home, to minor during danger or spooky skeletons”
- “Yes. I had a sense that more layers meant more intensity, but couldn’t discern exactly what that meant”
- “I did not notice the layering in the generated music”
- “I didn’t notice any voices, but thought generated music helped the experience”

B.2 Multiple Choice Question

B.2.1 *Boss Attribute*: did the music help indicate any of the following?

- Boss HP
- Boss Stamina
- Boss Weapon Rating / Attack
- Boss Armor Rating / Defense
- Power Differential (between you and the boss)
- Interaction Number (first, second, third time you encountered the boss)

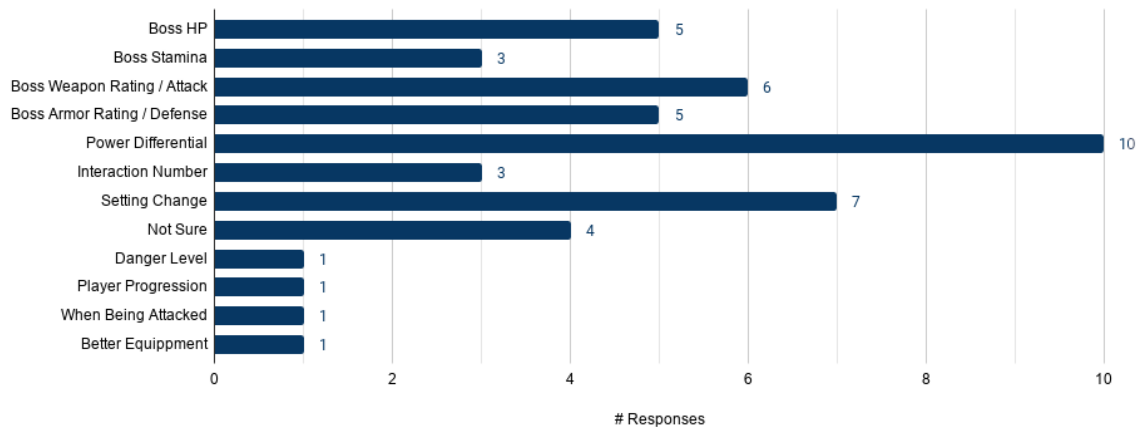


Figure B.1: Total responses including write-in answers

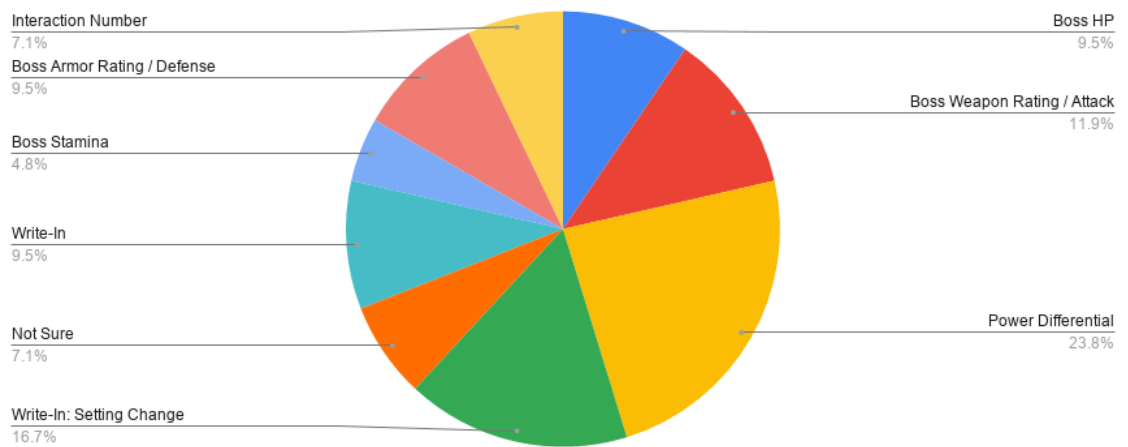


Figure B.2: Total responses consolidating write-in answers and *setting change*-based answers

Appendix C

MUSICAL QUALITY QUESTIONS

C.1 Quality Score Questions

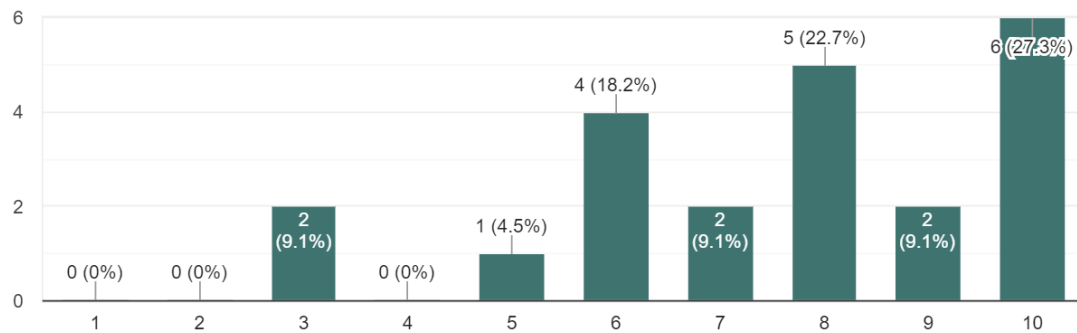


Figure C.1: Was the music pleasing?

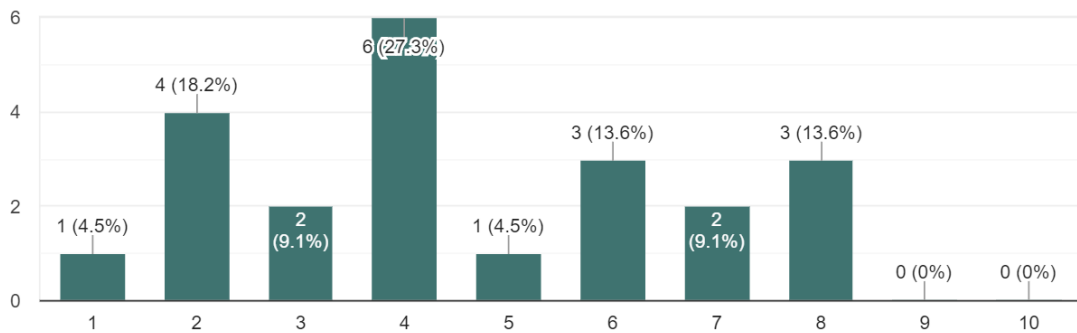


Figure C.2: Did the music sound random?

C.2 Free Response Question

1. "Great Job!! Very interesting music that help to indicate where I was and who was around me."
2. "I liked the variety of music throughout."

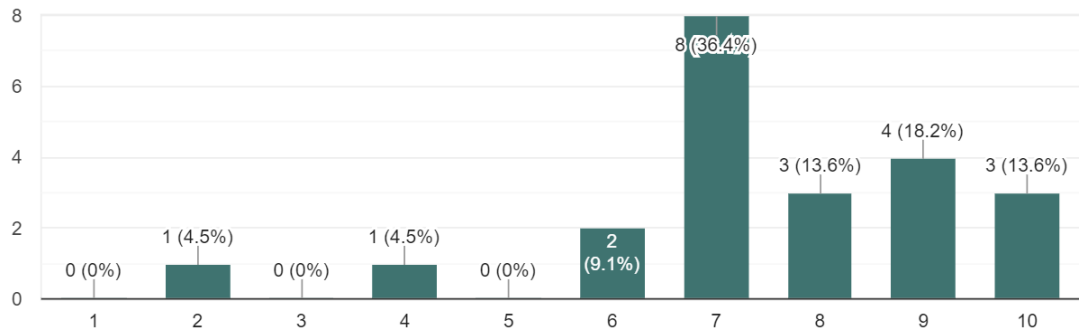


Figure C.3: Was the music harmonious?

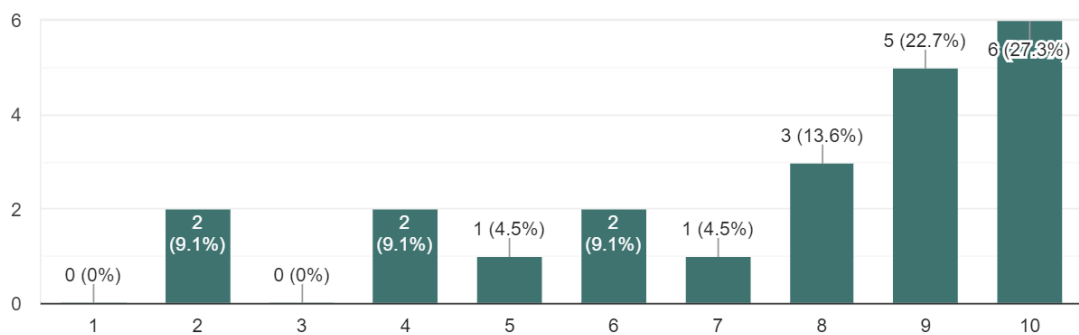


Figure C.4: Was the music interesting?

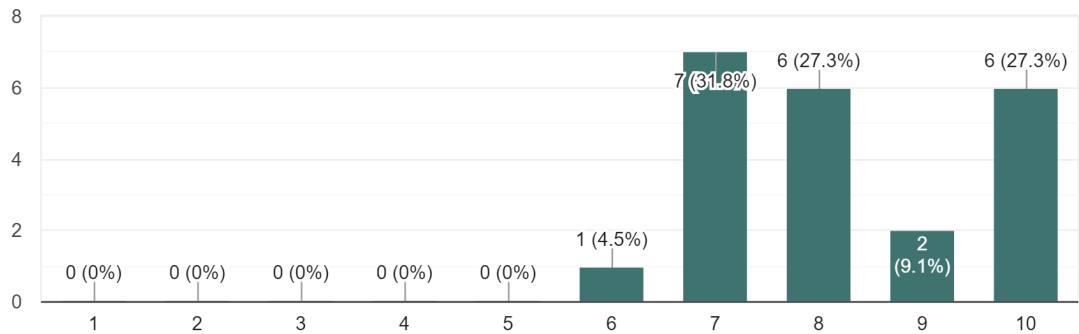


Figure C.5: Did the music appear to be dynamic? (As in, did it change as you progressed?)

3. “More variation in rhythms or chords throughout the piece would make it more interesting. Maybe add some percussion too?”
4. “I love the composition. It’s so pleasing to listen to and I feel like it fits the game well and the energy I feel while going through the game.”

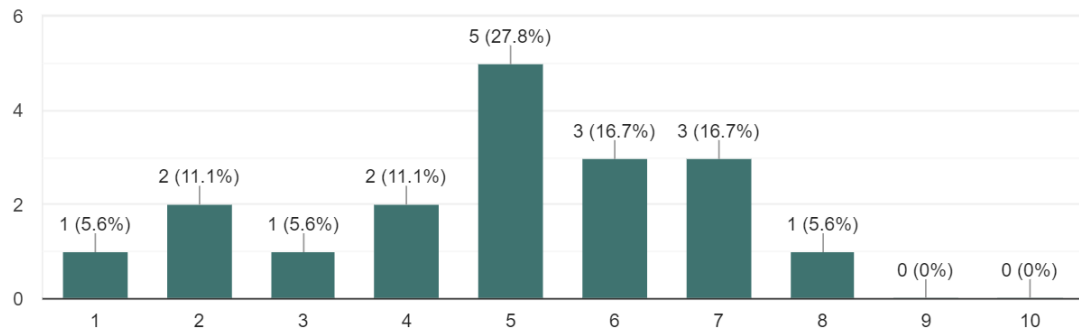


Figure C.6: Did the music appear to be change as you revisited areas and NPCs?

5. “I enjoyed the music quite a bit. I didn’t realize that it was the same melody throughout all the zones until the end; then when I went back through I could tell. I thought the music fit quite well with each zone, though I did notice that it sometimes seemed to take a bit of time to switch when moving between zones (though that may have been a limitation on the engine side of things?). It didn’t feel too repetitive (though I’m not the best judge of that), especially when actually playing through the game and changing zones (and the sound of the music) on a relatively regular basis.”
6. “I thought that the quality of the composition was great. I liked the different areas show all having its own atmosphere and vibe.”
7. “It was confusing that in the beginning when you clicked by chance on a rat underneath the intro text it would start hitting the rat with sound effects. First I thought that’s part of the music and that was confusing. Except I found it pretty pleasing and fitting overall. Pleasant soundtrack in general, fitting in almost all situations and places on the map.”
8. “Very nice.”
9. “Repeated very quickly, but there was a lot of variation between levels.”
10. “It was decent.”
11. “It felt like generic RPG music, it set the scene but it isn’t something I’d listen to in my free time.”

12. "Well I might be able to tell that the music is a bit "off", it felt very close to music I'd hear in a game and was very impressive. If I didn't know beforehand that you were working on generated music, I might not have jumped to that conclusion. I thought the overall quality was fairly high even if there were pieces that felt a bit off."
13. "I found most of the generated music unpleasant to listen to not based on melody but because of its robotic nature and tinny instruments. However I found the tune for the area directly after the grasslands (I think it was directly after the grasslands) to be pretty nice actually."
14. "Pretty good overall, would have sounded better if it flowed together more."
15. "I liked it. I thought it was similar to the music of Pokémon where each city had a soundtrack."
16. "If the goal is creating generated music in video games based on the area, which could apply to a lot of other aspects such as special events, this is pretty exciting."
17. "The music sounds like background music, but it loops a lot, so it would be irritating to listen to for long."
18. "The music was able to add to the experience in an efficient manner."
19. "I enjoyed the music - it sounded like what you'd typically hear in a video game like this. I don't know if my session differed from the others, but I found the music didn't correlate with the gameplay in a discerning way other than changing "songs" when I changed regions."
20. "At times the music was very pleasing, other times a little disorienting."

Appendix D

GAMEPLAY QUESTIONS

D.1 In-Browser Experience with Game

1. "Very nice, and easy to control"
2. "Overall the game worked pretty well. Maybe allow the character to move with the arrow keys in addition to point and click with the mouse"
3. "The in-browser experience was good. I had no problems with the game mechanics"
4. "I had no issues"
5. "I thought the experience was really interesting. I didn't pay attention to the subtle music changes but I did notice drastic changes like changing scenes or areas in the world"
6. "Very nice experience, smooth and pleasant, no issues encountered"
7. "I had to refresh since there was no boss in one of the cake scenes, and there were some times where the clicking lagged. Otherwise, it was a great experience!"
8. "Very fun and easy to play, was not expecting other people"
9. "It was neat. I didn't pay enough attention to the sound the first time. I wasn't aware it would indicate enemy conditions"
10. "It seemed alright but I would've preferred a full screen mode. Also, I think the boomer in me accidentally clicked on some bookmarks one too many times"
11. "It worked pretty well but there was some issues with lag / glitches but I don't think it hindered too much"

12. "Technical issues, otherwise fairly standard. Similar to early flash games in the early 2000's"
13. "That worked well. I liked that it saved your progress when you reloaded page"
14. "It was a little laggy, but otherwise it was fine"
15. "The clicking felt clunky and not responsive"
16. "The game was easy to play and intuitive"
17. "Fun flash back to when I was a wee-little trouble maker"