Application and evaluation of affective adaptive generative music for video games

by

Cale Plut

M.F.A., Simon Fraser University, 2017
B.F.A., Simon Fraser University, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Interactive Arts and Technology
Faculty of Communication, Art and Technology

SIMON FRASER UNIVERSITY
Spring 2022

Declaration of Committee

Name:	**Cale Plut**
Degree:	**Doctor of Philosophy**
Thesis title:	**Application and evaluation of affective adaptive generative music for video games**
Committee:	**Chair:** William Odom Assistant Professor, Interactive Arts and Technology
	Philippe Pasquier Supervisor Associate Professor, Interactive Arts and Technology
	Steve DiPaola Committee Member Professor, Interactive Arts and Technology
	Marek Hatala Examiner Professor, Interactive Arts and Technology
	Matthew Guzdial External Examiner Assistant Professor, Computing Science University of Alberta

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Music is an element in almost every video game. Because games are interactive and music is generally linear, writing music that matches the actions and events of gameplay presents a unique challenge. Adaptive music partially addresses this, but creating adaptive music requires extra labour and restricts elements of the composition.

Generative music, created with some degree of autonomy from it's input, presents a possible tool for addressing these drawbacks. Depending on the particulars of the system, these systems are capable of creating large amounts of music, very quickly, that fit a given set of constraints. Additionally, while training and creating generative models can be an expensive process that requires large amounts of computing power, the generation of music is generally computationally lightweight. Theoretically, generative music systems may be capable of creating highly adaptive music with far less labour cost than manual composition.

We design, implement, and evaluate an application of affective adaptive generative video game music. We first survey uses of generative music in games. While academic approaches generally create novel algorithms for real-time affective adaptive music composition, there is a large gap between the integration of academic systems in games and common industry approaches to using music in games. We therefore focus on the application of generative music in common game music frameworks.

Academic approaches to generative music generally use a model of emotion to control the affective expression of the accompanying score. We investigate this approach, and find that audience members perceive affective musical adaptivity. To use emotion as an intermediary between gameplay and music, we split this mediation into two tasks: We describe the perceived emotion of gameplay in an emotional model, and we control the perceived emotion of music to match the emotional model.

To describe the perceived emotion of gameplay, we create the Predictive Gameplay-based Layered Affect Model (PreGLAM). PreGLAM is inspired by research in affective non-player-characters, and uses a cognitive appraisal model to respond to gameplay events. PreGLAM essentially acts as an audience member, watching the gameplay, and modeling a perceived valence, arousal, and tension value. We empirically evaluate PreGLAM, and find that it significantly outperforms a random walk time series in matching ground-truth annotations of perceived gameplay emotion.

To create our generative score, we use the Multi-track Music Machine (MMM) [62] transformer model to generate variation stems from a composed adaptive musical score. Because MMM generates variations based on an input clip of music, we control the emotional expression of MMMs output by controlling the emotional expression of the input. To do so, we create a parametric composition guide to compose an adaptive score that expresses three levels of affective perception in a three-dimensional Valence-Arousal-Tension (VAT) model of emotion, titled the "IsoVAT" guide. The IsoVAT guide is based on a collation of multiple cross-discipline surveys of empirical music-emotion research (MER), and describes how alterations in musical features affect the listener's perceived affect in a VAT model. We empirically evaluate the IsoVAT guide by following it to compose a corpus of 90 clips which are evaluated across 3 different study designs.

We expand our adaptive score using MMM. We adaptively re-sequence individual tracks from our generative variations, creating almost 14 trillion unique musical arrangements in our generative adaptive score. We also write a linear score to serve as a baseline, which is produced using the same synthesis and performance techniques as the adaptive and generative scores. We empirically evaluate our musical scores using real-time annotations of perceived emotion, as well as with a post-hoc questionnaire. Our findings indicate that our application of generative music in games comparably maintains perceived emotional congruency of previous applications, while outperforming previous applications in perceptions of immersion.

Keywords: Generative Music, games, game audio, game music, emotion

Dedication

For my loving family

Acknowledgements

I first would like to thank my supervisor Philippe Pasquier, without whom none of this would have been possible. I would like to thank my colleagues in the Metacreation lab, particularly my co-authors Jeff Ens and Renaud Tchemeube. Also, I would like to thank Steve DiPaola for his assistance with this work. I would additionally like to thank Tiffany Taylor, Lisa DaSilva, and the other SIAT administrative staff for creating the environment that allows such work to be accomplished.

Finally, thanks to my loving wife and children, without whom my brain would have crumbled into oblivion long ago.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction and motivation

1.1.1 Thesis format

This thesis takes the form of a cumulative thesis. Cumulative theses are a collection of scholarly peer-reviewed articles, completed in lieu of a monograph dissertation. This thesis consists of a total of 3 journal papers and 2 conference papers: one published journal paper, two journal papers currently undergoing peer review, one published conference article, and one conference article currently undergoing peer review. These articles outline the background review, design, implementation, iteration, and evaluation of an affective generative adaptive music system in an Action-RPG (ARPG) video game. More information about a cumulative dissertation can be found in Appendix A.

This thesis investigates the use of adaptive generative music in video games. More particularly, this thesis investigates the use of music that adapts to match the perceived emotion of the moment-to-moment gameplay of an ARPG genre video game. The design of our approach is informed by a survey of previous applications of generative music in games in both academia and the games industry, as well as an empirical study that we conducted on affective game music. We create the Predictive Gameplay-Based Layered Affect Model (PreGLAM) to control the adaptivity of our musical score, based on previous research in affective Non-Player Character (NPC) design. We create and use the "IsoVAT" composition guide to compose a score that adaptively expresses 3 levels of perceived affect in each dimension of our Valence-Arousal-Tension (VAT) affect model, as well as a linear score that expresses varying levels of VAT without adapting to the gameplay. We use the Multi-track Music Machine (MMM) [62] to generate variations on our adaptive score. We empirically evaluate PreGLAM, the IsoVAT guide, and all three musical scores. Figure 1.1 shows an overview of the research contained in this thesis. Chapter 2 examines previous applications of generative music in games from academia and the games industry, and examines the general trends. This informs the overall design of the rest of the thesis work. One finding in Chapter 2 is the commonality in academic approaches to match the musical emotion

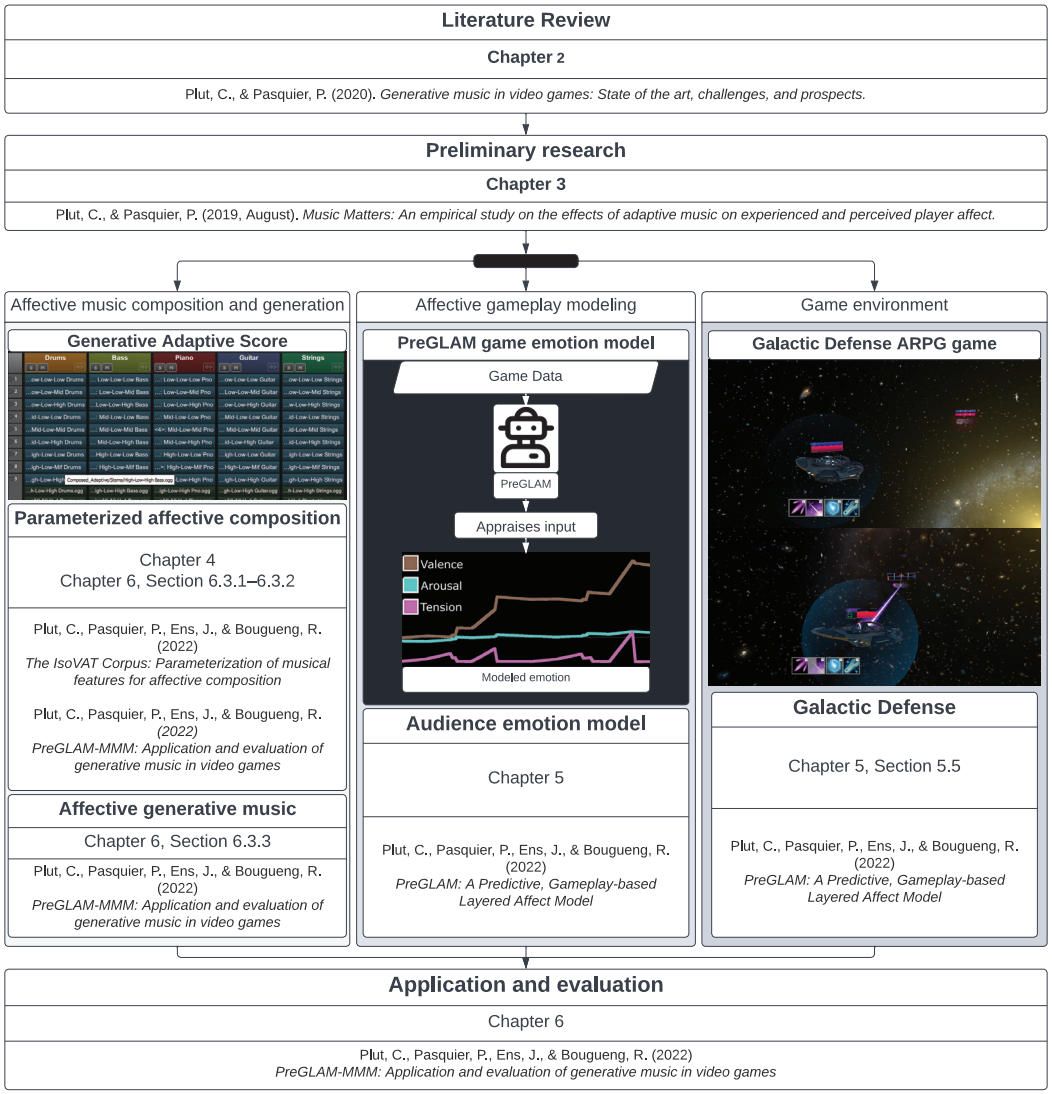


Figure 1.1: Graphical structure of thesis.

to some model of player emotion. We evaluate the effects of music on player emotion in Chapter 3. Following the findings of these two papers, we design, implement, and evaluate our application of generative music in games as described in Chapters 4, 5, and 6.

Chapter 4 describes our work on parameterically composing music to express a given level of emotion in our VAT model. Chapter 5 presents PreGLAM, a cognitive agent that models the real-time perceived emotion of a passive gameplay spectator with a provided bias. Chapter 5 also presents *Galactic Defense* (GalDef), a video game that we create as a platform and environment to implement and evaluate PreGLAM and our musical scores. Chapter 6 effectively combines these elements into a single application and evaluates this application, in addition to introducing our use of MMM to create our generative score.

Chapter 6 describes the creation and evaluation of our linear, adaptive, and generative musical scores for GalDef. All composed material follows the IsoVAT guide, and all musical adaptivity is controlled via the output of PreGLAM.

1.1.2 Motivation

The video game industry is one of the largest entertainment industries in the world. In 2020, the collected market value of the games industry was estimated at $155.89 billion USD [37]. Approximately 2.7 billion people play games as a hobby, and an average of 25 games are released on "Steam", the largest digital game storefront, each day [86].

Music is in almost every video game, but there is an inherent mismatch between the linearity of music and the interactivity of games. Most music is linear - it plays from beginning to end in mostly the same way each time it is performed or played. Games are interactive - the player's actions influence the game world, and the game world's responses influence the player's future actions.

This mismatch is addressed by game composers in several ways. Music serves multiple roles in games, and some roles do not require synchronicity between gameplay and music, bypassing the problem altogether. Table 1.1 describes some functions that music fills in games, as described in two separate game music texts [177, 227]. We note that while there is a fair degree of overlap between the conceptual functions of music in both sources, there is no standard delineation.

Linear music can be used to fill some game music functions, but others highlight the temporal mismatch between linear music and interactive gameplay. When acting as "an audience" music is expected to react and comment on the action of the game [177]. In order to do this, a technique called "Adaptive music" can be used. Adaptive music, sometimes called "interactive music", is music that responds to a control input [227]. In most cases, adaptive music attempts to match the perceived emotion of gameplay [227, 177].

Adaptive music is a powerful tool for using music as an audience, and we demonstrate that adaptive music can have an emotional impact on the player's self-reported experienced emotions [189]. However, adaptive music has several drawbacks. Care must be taken to construct stems that are musically pleasing regardless of how they're combined, and each stem must smoothly loop and transition to other stems (or stingers) smoothly. This means that elements such as harmonic structure and instrumentation must be complimentary across different stems. This restricts elements of the musical composition, and requires additional skills and labour to create compared to composing a linear piece of music [177, 227]. This effectively produces an economical trade-off when using adaptive music — resources spent creating additional adaptivity are not put towards other elements of the composition.

This economic trade-off is exacerbated by the length of video games compared to other media. The average length of a theatrical play or musical is approximately 120-160 minutes [161], the average runtime of a major movie is approximately 80-130 minutes [114], and

Table 1.1: Functions of music, according to game composition books.

Author	Use	Description
Phillips	State of mind	Help player achieve specific mindset - "the zone"
Phillips	World builder	Provide aural details about the nature of the setting
Phillips	Pace setter	Reflect and augment the pace and energy of gameplay
Phillips	Audience	Attempt to create the impression that the music is watching gameplay and commenting on successes and failures of player
Phillips	Branding	Create association with game and music
Phillips	Demarcation	Musically outline different gameplay type or location
Sweet	Setting the scene	Help define time and place with instrumentation and/or harmonic relationships
Sweet	Introduce characters	Leitmotif. Character themes provide characterization and organization
Sweet	Signal a change in game state	Often used briefly for transitions between other states
Sweet	Increase or decrease dramatic tension	No description given. Increase tempo and/or add layers to increase tension.
Sweet	Communicate an event	Stingers. 3-12 seconds, enhance a particular event
Sweet	Emotionally connect a player to the game	"Iconic theme" can establish overall tone and feel
Sweet	Enhance narrative and dramatic story arcs	"Enhance the emotional high and low points in your game"

the average length of a pop song is about 3-5 minutes [145]. Meanwhile, the average game length is approximately 15 hours, but can take as little as <1 minute or as many as >100 hours to complete [132].

While a film composer may write music to accompany every minute of the movie's runtime, this is not a tenable solution for a 100 hour long video game even if the music is entirely linear. To manually compose an adaptive score that acts as an audience for the duration of a video game would be economically unfeasible.

Generative music, created with some form of systemic autonomy, presents a potential solution to these drawbacks. Depending on the system, generative systems are capable of composing large amounts of music very quickly, potentially in real-time. Additionally, generative systems can be capable of producing music that follows parameters of a given input [190]. Despite these potential advantages, generative music is not widespread in the games industry, and there is almost no interaction between academic research and the game industry in this area as far as we are aware.

There are several advantages to using linear music in games. Composers are, by and large, familiar with writing linear music, and there are almost no additional considerations for using linear music in games compared to other media, beyond smooth looping being preferred. Because linear music requires far less labour than adaptive music, composers can spend more of their time considering how the music will interact with other game elements like narrative, structure, and overall musical quality. Additionally, some degree of linearity is necessary to fill musical roles such as "branding".

Final Fantasy XIII (FFXII) [221], a Japanese Role Playing Game (JRPG) released in 2009, with an average runtime of 50-60 hours [109], demonstrates the efficacy of linear music in supporting a game's structure. Overall, the musical score makes heavy use of leitmotifs (recurring themes commonly associated with characters or environments [21]) to support an epic fantasy drama, acting as a "world builder".

As with other games in the *Final Fantasy* series, battles in FFXIII take place in a different game environment than other gameplay. In battles, movement is altered or removed entirely, and the player selects actions from a menu, controlling a team. Typically, there is a particular linear piece of music that plays during battles, and occasionally different battle themes for bosses and important enemies. FFXIII's music is entirely linear, and there is a linear battle theme as in other *Final Fantasy* games.

During different moments in FFXIII's narrative, the use of the battle theme changes. In most parts of the game, the battle theme is used as is standard in JRPGs - there is an environmental piece of music that plays, and when a battle starts, the music changes to the battle theme. In some parts of the game, there is no musical change when starting a battle, and the environmental music plays throughout. This use of the theme also avoids repetition fatigue [177]. In other parts of the game, the battle theme also acts as the environmental music, acting as a "pace setter" for the game segment.

An adaptive battle theme in FFXIII may have been able act as an audience for each battle, which the linear theme does not. However, the additional labour required to create the adaptivity would have likely reduced the focus on composing and utilizing the music to support the other longitudinal aspects of the game such as narrative, character development, and changes to the in-game world.

In theory, generative music could reduce the labour cost of adaptive music so that composers can focus on the composition and use of music in a game, gaining both the benefits of linear music and adaptive music. This use of generative music follows the suggestions of Casini et al. that generative music models are often best applied as assistive tools for composers [29].

There is an additional benefit to using generative music to support composers by extending adaptive music. Herremans, Chuan, and Chew note that a remaining over-arching challenge for generative music systems is long-term structure [104]. In using generative music to address the moment-to-moment reactions to gameplay, a human composer may be able to focus more on the long-term structures of the music. Essentially, games present an opportunity to exploit the strengths of both generative and human-composed music.

While the theoretical benefits of generative music are supported by findings of academic research, it has not achieved widespread use. In our survey of academic and industry approaches to generative music in games, we found applications of generative music mostly following one of two approaches. Academic approaches generally focus on creating generative algorithms that can compose in real-time, based on an input of an emotional model of the gameplay. Industry approaches generally use stochastic methods to arrange composed pre-recorded stems. There are strengths and weaknesses to both approaches.

Academic approaches use more generic emotion models to respond to gameplay, which can be applied across multiple settings. Academic approaches tend to focus on creating new generative algorithms that compose music in real-time, based on an input of emotion. These models have a high degree of granularity, and can quickly adapt to the input emotion. However, these approaches tend to only generate simple music such as a chord progression, or a melody over an arpeggiated chord. Academic approaches tend to only generate for one or two instruments, which are generally synthesized using General MIDI sounds. Additionally, when academic approaches are evaluated, they are often evaluated in very simple game environments without gameplay mechanics. Examples include adapting tension based on a player navigating a maze with mobile obstacles [195], or adapting valence based on the number result of a dice roll [35].

Industry approaches are generally much simpler in architecture, and rarely develop new models or methods for generating music. These approaches are integrated into real-world gameplay mechanics, but mostly adapt to a single gameplay measure, such as the number of remaining enemies in a fight, or to small sets of game states such as “exploration” or “combat”. These approaches often cross-fade between levels of adaptivity, which can create

obvious seams in the musical adaptivity. However, industry soundtracks use recordings of live musicians, or virtual instruments that are synthesized offline, such as "Virtual Studio Technology" (VST) instruments. Ensembles in industry scores can range from a few instruments to full orchestras. This produces music that aesthetically consistent with music in other media such as film or theatre, and superior to General MIDI.

Differences in perceptions between these approaches are illustrated in a study by Willaims et al. [254]. Williams et al. create a generative system that is evaluated empirically in *World of Warcraft* [64], and compare it to the game's original score. The original score is written for an orchestra, and has a relatively even emotional expression throughout each piece. Williams et al.'s generative system creates music for solo piano, and adapts based on an input emotion. The input emotion is based on researcher-assigned valence and arousal values, linked to specific game states. Williams et al. give an example of the system generating "angry" music while in combat, and "content" music when the player is victorious.

When evaluated on a 9-point Likert scale, Williams et al.'s generative score's average rating is significantly higher than *World of Warcraft*'s original soundtrack by one point in emotional congruency. However, the generative score's average rating is significantly lower than the original soundtrack by 1.75 points in ratings of immersion. We believe that this indicates that generative models are successful at real-time adaptation to game emotion, but that other elements of the score, such as the musical quality and synthesis, may be a limiting factor in the widespread applicability of generative music in games.

We also believe that the results of Williams et al.'s study demonstrate potential confounds in previous research in this area. In evaluating *Escape Point*'s generative system, Prechtl compared the generative system's output with changing tension, to the generative system's output with static tension [195]. Scirea compares the output of *Metacompose* to the output of *Metacompose* with parts of the model replaced by random generation [215]. While these approaches evaluate whether the adaptivity affects players affective perception, they do not resemble real-world video game scores.

Williams et al. compare an affective adaptive generative music system to a real-world game score, but the evaluated scores differ in genre, instrumentation, timbre, emotional expression, and production quality [254]. Additionally, the score in *World of Warcraft* can most easily be interpreted as acting as a "world builder", and so these scores also differ in game music function.

Our goal with this research is not to create another new, faster, more emotionally adaptive generative music system. Instead, we target the application of a generative music system into gameplay. By focusing on how generative music may be used within existing game music frameworks, we look to bridge the gap between the technologically impressive generative models of academic research, and the high degree of musical quality and production in the games industry. In other words, we examine the practical applicability of the theoretical applications of generative music in games.

An additional goal for this research is to evaluate our generative score in comparison to adaptive and linear music that are similar in musical and production quality to both real-world game scores and the generative score. Essentially, we isolate the composition source of the music from the previously identified potential confounds, and investigate the generative nature of the score.

Generative music is a potentially and theoretically powerful tool for assisting game composers in creating highly adaptive music without sacrificing musical quality elsewhere. Previous research primarily focuses on solving the technological limitations of generative music, while we explore the application of generative music into gameplay. Previous research often has a large gap between the generated music and music that is used in real-world video games, while we attempt to bridge this gap by using industry techniques in the creation of all of our musical scores.

1.2 Thesis structure

We divide our application of generative music into two primary elements:

1. A model of an audience member's emotional perception of input gameplay from a video game.

2. A musical score that expresses a particular input emotion.

To implement these elements, we also create a video game. Our game is a scaled-down Action-RPG, and integrates our emotional model, using the model to control the adaptivity of the accompanying music. By combining these elements, we produce a generative musical score that aims to satisfy Phillip's description of music acting as an audience: By adapting the music based on perceived emotion, and basing perceived emotion on an appraisal of the gameplay by an audience member, we aim to produce a score that feels like it is "watching the gameplay and commenting periodically on the successes and failures of the player" [177].

Chapters 2 and 3 detail our background research. We investigate previous uses of generative music in games, and empirically demonstrate that adaptive music affects players emotional perceptions in a user study.

Chapter 5 focuses on our Predictive, Gameplay-based Layered Affect Model, (PreGLAM), and our game *Galactic Defense*. As mentioned, because we are targeting the use of music as "an audience", PreGLAM attempts to simulate the emotional perception of an audience member who is watching the gameplay.

Chapter 4 describes our "IsoVAT" guide. The IsoVAT guide is based on surveys investigating perceived emotion of musical features, and maps changes in musical features to perceived changes in valence, arousal, and tension. We ground-truth this mapping by using it to compose 90 4-bar clips of music, which are empirically evaluated with a listener study.

Chapter 6 details how we interpret the IsoVAT guide to create our three musical stimuli: A composed linear score, a composed adaptive score, and a generative adaptive score. To create our generative adaptive score, we extend our composed adaptive score using the "Multi-track Music Machine" (MMM) [63] Transformer model.

We evaluate PreGLAM and our application of generative music in Chapters 5 and 6. We perform a user study that evaluates PreGLAM's real-time emotional modeling, the effects of differing musical scores on the real-time emotional modeling, and post-hoc user perceptions of the differing musical scores.

Overall, we present an application of generative music in games that is primarily based on previous research in the area, with a focus on exploring the use of generative music within current game music and game design frameworks and processes. We evaluate our generative score in comparison with similarly produced musical scores that also resemble real-world game music. This isolates the effects of the generative nature of the score, while also providing a more complete view of how generative music may be applied in video games.

We find that generative music can be used to extend a composed adaptive score. When we empirically evaluate this use of generative music, our generative adaptive score heavily outperforms our original composed adaptive score in rankings of emotional congruency, immersion, and preference.

1.2.1 Research questions and contributions

Our research questions are as follows, and Table 1.2 describes the papers included in this thesis and the research question(s) that they address. In Section 1.2.2, we provide an overview of the contributions for each constituent chapter of this thesis.

RQ1 How can we use and evaluate generative music in video games?

RQ2 Does adaptive music influence player emotion?

RQ3 How can we model gameplay for music adaptivity?

RQ4 How can we parameterize and control affective music composition?

RQ5 How can we control affective generative music?

RQ6 How can we evaluate generative music in comparison to current game music approaches?

1.2.2 Outline

Chapter 1 — Introduction

We provide an introduction and overview to this cumulative thesis in this chapter. We introduce the over-arching motivation for the research, the components used for the research,

Table 1.2: Papers and addressed research questions.

Cpt.	Paper	Contributions	RQs
2	Generative music in video games	Survey of academic and industry applications of generative music in games	1, 3, 5,6
3	Music matters	Empirical evaluation of effects of music on perceived and subjectively experienced emotion	2, 3, 4
4	The IsoVAT Corpus	Creation of guide for affective music composition. Composition of dataset following guide, and empirical evaluation of dataset and guide	5
5	PreGLAM	Creation and empirical evaluation of the PreGLAM, which models the real-time perceived gameplay emotion of a passive biased spectator	3
6	PreGLAM-MMM **Sections 6.3-end**	Builds on Chapters 4 and 5 to implement and evaluate the use of MMM [62] to generatively expand a composed adaptive score.	5, 6

and describe how these components fit together. We present the research questions and conclusions, and provide an outline for the thesis.

Chapter 2 — Generative music in video games: State of the art, challenges, and prospects

Research questions addressed:

RQ1 How can we use and evaluate generative music in video games?

RQ3 How can we model gameplay for music adaptivity?

RQ5 How can we control affective generative music?

RQ6 How can we evaluate generative music in comparison to current game music approaches?

In Chapter 2, we describe our survey of generative music in video games. While generative music is used in academia and the games industry, approaches from the game industry and academic research bear little resemblance to each other. Academic uses of generative music generally focus on real-time generation of music, while industry approaches generally use generative music to extend an otherwise composed score. Because of their focus on real-time music generation, academic approaches are often rudimentary in musical adaptivity and synthesis as well as gameplay modeling. In contrast, approaches from the industry tend to focus on fidelity of performance — the music is performed by real musicians or uses VST instruments with performance data, and the musical adaptivity to gameplay is given greater focus than in academic systems. Consequently, industry approaches generally use rudimentary approaches to generating the music itself.

Approaches from academic research primarily focused on applying novel generative algorithms for real-time composition. Emotion is often used as a link between game and generative music. An overview of academic systems in the paper, and more recent systems, is given in Table 1.3. Approaches to generative music in games from the games industry

tend to follow simpler, more constrained, rule-based approaches. Most of the individual differences between industry applications of generative music involve the treatment of musical content itself, rather than the algorithm used. The most simple industry application is using stochastic chance to extend a standard adaptive score, as seen in *Red Dead Redemption* [201] or *DOOM (2016)* [197].

Table 1.3: Generative music systems in academic research for games.

Author	Year	Emotion model	Music source
Lopes	2015	Tension - provided tension curve w. level generation	Placement of audio cues
Prechtl	2016	Tension - proximity to enemy	Markov
Williams et al.	2017	Valence-Arousal - affective tag of game state (e.g. combat)	Markov
Scirea	2017	Valence-Arousal - fitness evaluation of game state	Multiple - Evolutionary and rule-based
Plut and Pasquier	2017	Tension - proximity to enemy	Composed
Hutchings and McCormack	2019	Discrete - objects are associated with emotion, spreading activation	Multi-agent rule-based
Washburn and Khosmood	2020	Valence-Arousal - NPC attributes	Multi-agent rule-based

Another difference between academic and industrial uses of generative music is the musical representation. Academic systems generally use symbolic notation to represent their scores, which are synthesized in real-time, most often by General MIDI. Industry approaches mostly use audio recordings or VST instruments that are synthesized offline. Williams et al. compare a generative system that uses General MIDI directly to a commercially recorded soundtrack, and find that participants report reduced immersion when listening to the general MIDI score [254]

Chapter 3 — Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect

Research questions addressed:

RQ2 Does adaptive music influence player emotion?

RQ3 How can we model gameplay for music adaptivity?

RQ4 How can we parameterize and control affective music composition?

As we note in Chapter 2, the most common source of musical adaptivity for academic generative music systems for games is affect — researchers create some model of player affect that controls the adaptivity and/or generation of the musical score. Chapter 3 describes our empirical study on affective adaptive music in games. We explore 4 different uses of adaptive music in games, adapting the music based on an single tension value.

Players reported subjectively experiencing and perceiving increased tension when the music is adaptive at all, and further increased when the adaptivity is congruous with the estimated game tension. This demonstrates the viability of affective adaptive music, and shows support for a positive answer to RQ3. Music Matters also presents *Galactic Escape*, our initial prototype for what would become *Galactic Defense*, our game environment for evaluating generative music. Galactic Escape begins an exploration into RQs 3 and 4,

modeling gameplay for musical adaptivity, and in attempting to control affective musical composition. We model game emotion following previous uses of generative adaptive music in academic approaches, and assign a value directly to the distance between the player and a pursuing NPC. We model musical emotion by following previous music-emotion guidelines from Western music theory while attempting to keep a natural sound, consistent with previous music emotion research.

Chapter 4 — The IsoVAT Corpus: Parameterization of musical features for affective composition

This chapter addresses RQ5: How can we parameterize and control affective music composition?

In Chapter 4, we describe the creation of the IsoVAT composition guide and the IsoVAT corpus. The IsoVAT composition guide provides a set of musical features, and the ordinal emotional perception associated with changes in the features in Western tonal music. The IsoVAT guide is derived from multiple meta-reviews of music-emotion research in musical features and affect, as well as meta-reviews on translating between different affect representations.

Previous affective generative music systems primarily manipulate musical affect via a set of composition rules [136, 195, 215, 254]. These rules are often based on interpreting music theory, as well as previous MER research that examines particular musical features and their associated affective expression. Surveys of MER research note that given the extreme breadth of emotion models, stimulus, and experimental designs in MER, there is a lack of consensus and generality in the literature [53, 251].

We use the IsoVAT composition guide to create a corpus of 90 musical clips. Clips are organized into sets of three and organized by manipulated affective dimension. Each sets consists of a clip that expresses the lowest level of an affective dimension, a clip that expresses the highest level of the dimension, and a clip that expresses a level between the other two. We empirically ground-truth the order of our IsoVAT corpus using three different study designs, and evaluate the composition guide based on our empirical findings.

Our corpus is evaluated across three study designs, and we find support for the general construction of the IsoVAT composition guide. The primary differentiation between the study designs is the amount of musical context present — participants rank the clips within a full set of 3, with subsets of 2 clips drawn from the set of 3, or rate a clip in isolation using a Likert scale.

Overall, results are consistent with results from similar MER studies, with inter-rater agreement ranging between 56–76%, depending on the affective dimension. We compile and collate ground-truth orders from across these study designs, and create a ground-truth order of the clips in the IsoVAT corpus. We perform a musical analysis of clips that are ground-truth ordered differently than composed. We identify musical features that are common

across these clips and not found elsewhere in the dataset. We believe these features present confounds for the emotional perception of Western music.

Chapter 5 — PreGLAM: A Predictive, Gameplay-based Layered Affect Model

This chapter addresses RQ3: How can we model gameplay for music adaptivity?, while also answering a more general research question of "how can we model a passive spectator's emotional perception of gameplay?" We note that while PreGLAM itself does not control musical adaptivity, it is designed to allow for the adaptivity of music acting as an audience. Therefore, PreGLAM simulates a passive spectator, mimicking the emotional perception of the audience.

This chapter describes the Predictive Gameplay-based Layered Affect Model (PreGLAM). PreGLAM is an artificial cognitive agent that models the emotional perception of a passive spectator who has a provided bias. PreGLAM uses an appraisal-based emotion model that takes a mood value and set of Emotionally Evocative Game Events (EEGEs) as input and outputs one real-time value per emotional dimension, based on the events of gameplay.

All EEGEs are assigned a base emotion value for each affective dimension. For our implementation, we base all values on an arbitrary unit of 1, which represents the base intensity of the emotional response to EEGEs. EEGEs are further modified by a set of context variables that describe the gameplay context of the event, as well as a time ramp value that represents emotions rising and fading in time.

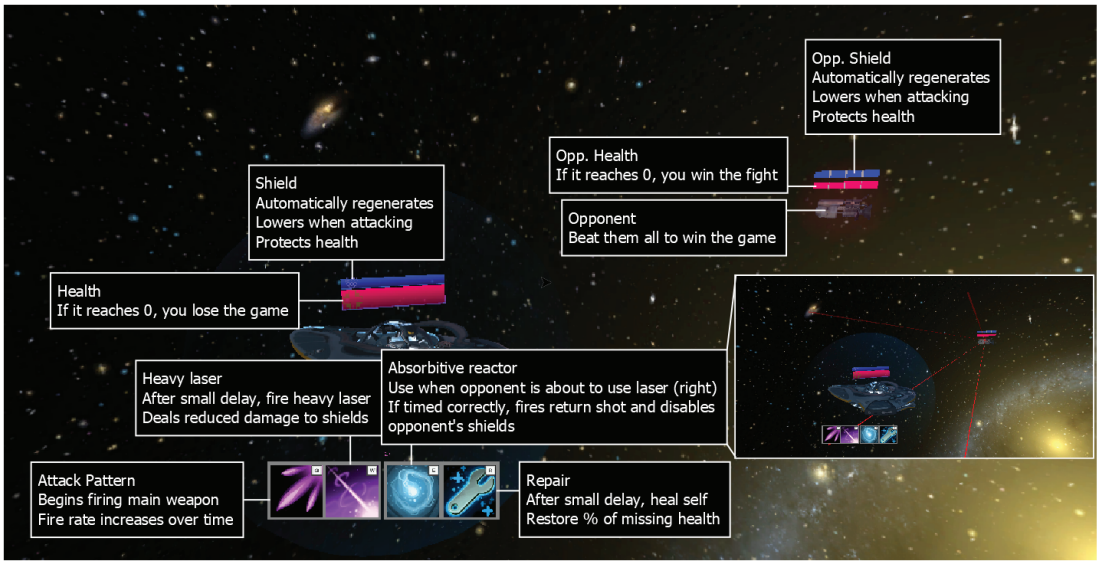


Figure 1.2: Screenshot and visual tutorial of "Galactic Defense".

Chapter 5 also introduces and describes *Galactic Defense* (GalDef), a game that we created to act as our experimental environment to evaluate PreGLAM and our application of generative music. GalDef is a minimalist Action-RPG, that uses a small array of

standard game mechanics to remain approachable within a study context. The player has four abilities, which have tactical strengths and weaknesses — a single ability may be very powerful or completely useless depending on the gameplay context. This design creates fluid and contextual gameplay within a common game framework. We assign EEGEs in GalDef manually, via iterative experiential playtesting during the development of the game.

We evaluate PreGLAM by collecting real-time ground-truth annotations of perceived emotion from spectators. We compare distance measures between PreGLAM and ground-truth annotations to the distance between a random walk time series and ground-truth annotations, and find that PreGLAM is significantly closer to ground-truth annotations than the random walk is. That is to say, PreGLAM significantly outperforms a random time series in accurately modeling a spectator's perceived emotion.

Chapter 6 — PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games

Research questions addressed:

- RQ5: How can we control affective generative music?
- RQ6: How can we evaluate generative music in comparison to current game music approaches?

This chapter combines and extends findings from Chapters 2, 4, and 5. We note that this chapter re-states information that is present in these other chapters. Information that is new to this paper can be found beginning in Section 6.3.

This chapter describes the application and evaluation of generative music in Galactic Defense. We use the Multi-track Music Machine (MMM) [62]'s *bar inpainting* capability to create our generative score. Bar inpainting involves resampling a subset of the bars present in one or more input tracks, or altering a subset of musical material based on the remaining unaltered musical material. To create an adaptive generative score, we first compose an adaptive score, which is used to condition MMM's generation. Our adaptive score's composition is informed by the IsoVAT guide. We also compose a linear score, which is primarily based on the composed adaptive score arranged into a linear format. By differentiating our scores only by musical adaptivity and source, we focus our evaluation on these aspects. In other words, while stylistic elements of our score may affect listener perception, we expect that any bias will be identical between various scores, as the stylistic elements are consistently present.

We empirically evaluate our application of generative music simultaneously to our evaluation of PreGLAM. In addition to the real-time ground-truth annotations, we collect post-hoc questionnaire responses concerning perceived gameplay congruence, emotional congruence, immersion, and musical preference. While responses to previous applications of generative music find generative adaptive music as comparably or marginally more emotionally

congruent to composed linear scores, previous applications generally decrease the player's reported immersion when compared to a linear score. In our evaluation, we find that the generative score is comparable to the linear score in emotional congruence as well as player immersion and preference. This indicates that our approach to using generative music in games is generally successful.

Chapter 7 — Conclusion

We conclude this thesis by summarizing and outlining the main contributions of the component research, and suggest future areas for investigation.

Appendix A — Guidelines for Cumulative Thesis

This appendix contains the SIAT guidelines for the completion of a cumulative thesis.

Appendix B — LazyVoice: A multi-agent approach to fluid voice leading

This appendix includes a paper discussing *LazyVoice*, a generative music system we designed during the course of this research. LazyVoice is a multi-agent system that targets the rendering of symbolic chord progressions into smooth voice leading for between 1 and 8 voices. LazyVoice introduces a flexible approach to representing chords, based on techniques from choral music improvisation and jazz harmonic theory.

1.2.3 Publications

Generative music in video games: State of the art, challenges, and prospects [190]

Plut, C., & Pasquier, P. (2020). *Generative music in video games: State of the art, challenges, and prospects.* Entertainment Computing, 33, 100337.

Note about authorship: Cale Plut is the first and lead author on this paper, and was responsible for performing the literature and game reviews, writing the content, creating the taxonomy and all supporting materials, and revising the content for publication.

Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect [189]

Plut, C., & Pasquier, P. (2019, August). *Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect.* In 2019 IEEE Conference on Games (CoG) (pp. 1-8). IEEE.

Note about authorship: Cale Plut is the first author on this paper, and created the research stimulus game and music, designed and ran the study for empirical evaluation, analyzed the data, and wrote and revised the paper content for publication.

The IsoVAT Corpus: Parameterization of musical features for affective composition [193]

Plut, C., Pasquier, P., Ens, J., &; Tchemeube, R. (2022). *The IsoVAT Corpus: Parameterization of musical features for affective composition.* Submitted to Transactions of the International Society for Music Information Retrieval (TISMIR).

Note about authorship: Cale Plut is the first author for this paper as well. Cale reviewed the previous MER literature and surveys, collated findings, constructed the IsoVAT guide, composed the IsoVAT dataset, designed and ran the empirical evaluation study, and wrote and revised the paper content for publication.

PreGLAM: A Predictive, Gameplay-based Layered Affect Model [191]

Plut, C., Pasquier, P., Ens, J., & Tchemeube, R. (2022). *PreGLAM: A Predictive, Gameplay-based Layered Affect Model.* Submitted to Entertainment Computing.

Note about authorship: Cale Plut is the first author for this paper. Cale designed and programmed the Galactic Defense game that serves as research stimulus and environment, designed and programmed PreGLAM, designed and ran the empirical evaluation study, created stimulus for the research study, analyzed the data, and wrote and revised the paper.

PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games [192]

Plut, C., Pasquier, P., Ens, J., & Tchemeube, R. (2022). *PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games.* To be submitted to Foundations of Digital Games (FDG).

Note about authorship: Cale Plut is the first author of this paper. Cale composed the two composed scores that are compared to the generative score, and generated the generative adaptive score. Cale also performed and produced the composed scores, and performed light production on the generative score. Cale designed and implemented the musical adaptivity based on PreGLAM's output. Finally, Cale designed, ran, and analyzed data from the empirical evaluation experiment, and wrote and revised the paper.

Appendix B: LazyVoice: A multi-agent approach to fluid voice leading

Plut, C and Pasquier, P. (2022) *LazyVoice : A multi-agent approach to fluid voice leading.* International Computer Music Conference (ICMC)

Note about authorship: Cale Plut is the first author of this paper. Cale designed and programmed LazyVoice, and performed all musical analyses.

Chapter 2

Generative Music in Video Games: State of the Art, Challenges, and Prospects

As published in Plut, C., & Pasquier, P. (2020). *Generative music in video games: State of the art, challenges, and prospects.* Entertainment Computing, 33, 100337.

Abstract

Music is a common element in most video games. Most music in games is written by a human composer, and played as a linear piece behind gameplay. Adaptive and/or Generative music systems can be used to extend the musical content or create new musical content using algorithms and AI. While there is research into these systems, there has yet to be an organized examination of their architecture and use. We present a taxonomy of generative music for games, to allow for examination and discussion of generative music systems. In doing so, we also present a survey of the current state of the art of generative systems in games, and discuss challenges and prospects of generative music for games.

2.1 Introduction and Motivation

2.1.1 Game Audio and Music

The video game industry is one of the largest media industries in the world, with 65% of American adults reporting playing video games [21]. As the games industry becomes larger, more and more attention is being paid to the rigorous study and examination of games.

While much of this study centers around the design of interaction, or the visual aspects of games, one of the most key components of games is audio. Even before the advent of digital games, audio has been a key component in even the pure foundations of play. Audio is so key to play that it transcends human designed play - young animals vocalize their play with yips and growls [35, 68]. Audio is so fundamental to games that in 1958, when the first video game *Tennis for Two* was developed, its gameplay was accompanied by audio.

Game audio is most commonly classified into the categories of *speech*, *sound/ effect*, and *music* [81]. *Speech* refers to voiced game audio that emanates from a character in a game. This includes the Player Character (PC) and Non-Player Characters (NPCs). Recordings of human voices are the most common source for speech, but speech may also be synthesized [92]. *Sound/Effect* generally refers to audio elements that are nonperiodic and nonmusical, often inspired by real-world sound effects such as a gun firing or wind rustling through trees. Sound/Effect elements may also be abstract and artificial, such as Pac-man's "wakka-wakka", the sound of Mario jumping, or a simple beep when a button is pressed [12].

The final classification of game audio is *music*, and is our focus. While defining "music" is a contentious issue, we use Luciano Berio's definition of music as anything that intends to be music [4]. Generally this is pitched, and has some regular division of time, but these features are not necessary. It is important to note that while the speech/sound/music divisions are useful to describe audio content in games, the divisions are descriptive rather than prescriptive, and the lines between these classifications are not set in stone. In short, while we restrict our scope to game music, we take an inclusive perspective on what constitutes game music.

2.1.2 Generative and Adaptive music

The most common use of music in games is to play a linear, composed piece of music during the gameplay. In many games, music is directly tied to the current level[1] and/or game state. With linear composed music, the music begins playing through a musical piece when the associated level is loaded. If the music reaches the end of the piece, the music loops. When a new level is loaded, the music either abruptly changes or quickly fades out, and is replaced with the new level's associated music. This use of music is often associated with older games such as 1985's *Super Mario Bros.* [52], 1986's *Castlevania* [38], and 1987's *Mega Man* [8]. However, lienar composed music is still in use in games such as 2009's *Final Fantasy XIII* [77], 2014's *Shovel Knight* [90], and 2017's *Pyre* [80].

[1] "Level" refers to a section or area of a game that is delineated from other sections or areas of the game. There are many terms for "level" across game genre and development tool. For instance, in fighting games, each fight takes place on a "stage". In the Unity engine, assets are grouped and loaded by "scene". These terms are interchangeable. When discussing levels, we will use the internally agreed upon term from the genre, tool, or community

Linear composed music can also be used without clear level delineations. As computer memory has become more plentiful and programming tricks allow for seamless loading of content, games have relied less on clear level delineations between game states. The "open-world" genre type exemplifies this, with open-world games minimizing or eliminating level delineations all together. Such games still often have clear delineations between game activities. The most common activity change is one between "combat" and "non-combat" gameplay. 2017's *Hellblade: Senua's Sacrifice* [50] is an example of this activity change. In *Hellblade*, there is a clear change between combat and non-combat – Senua draws her sword, the environment changes to create a small inescapable arena, the camera slightly changes its angle, and the players controls change to allow new actions. The music in *Hellblade* also changes with this state change. The "non-combat" music quickly fades out and the "combat" music quickly fades in. Games such as 1998's *Baldur's Gate* [6], 2009's *Batman: Arkham Asylum* [65], and 2016's *XCOM 2* [23] use this technique as well. It is important to note that this technique is not purely a more advanced game design or use of music, but an alternative structure. In *Final Fantasy X*, there is a level change between non-combat and combat. In *Final Fantasy XII*, there is simply an activity change. In *Final Fantasy XIII*, the level change returns, and in *Final Fantasy XV*, there is no level change.

There are two primary techniques to extend linear, composed music in games. The first technique - *adaptive* music - addresses the *linear* use of music. Adaptive music is sometimes called "interactive music", and is music that reacts to a game's state [11]. Adaptive music can provide large amounts of unique music from limited musical content. Adaptive music directly connects musical features to game variables. These features can include adding or removing instrumental layers, changing the tempo, adding or removing processing, changing the pitch content, etc. These changes in adaptive music are directly linked to gameplay variables. The adaptivity of music can be understood as a dimension. Low levels of adaptivity may only adapt to a small set of in-game variables, while higher levels of adaptivity may adapt to tens or hundreds of in-game variables.

One use of adaptive music can be seen in *Luftrausers* [91]. In *Luftrausers*, composer Julio "Kozilek" Kallio wrote a single 120 second musical piece. This piece of music is split into 3 groupings of instruments, each of which has 5 different arrangements, for a total of 125 different arrangements that can provide 4 unique hours of music. These arrangements are linked directly to the player's selection of parts that makes up their avatar ship, as seen in Figure 2.1. Adaptive music has been shown to increase a players' perceived and experienced tension during gameplay [59].

The other technique - *generative* music - addresses the creation of music. Most music is composed by an individual or team of human composers. Computational creativity is a field that explores the automation of creative tasks, and Musical metacreation (MuMe) is a subfield of computational creativity that addresses automating the creation of music [54]. *Generative* music [27] is music that is created via systemic automation, and is sometimes

called procedural music, musical metacreation, or algorithmic music. These terms are mostly synonymous and can be used interchangeably, but we will use "generative music" for simplicity.

Generative music can provide endless unique music in a game, and can be adaptive on a much deeper level than composed adaptive music, providing music that is individually tailored to the player's actions in a game [89]. Despite these potential benefits, generative music has not yet achieved widespread use in video games.

There is debate as to whether all game music can be considered generative [10, 95]. Because games are interactive, the exact timings of events is different for each player [68], which means that the musical timings will also be unique to each player. Mozart's *Musikalisches würfelspiel*, or musical dice game, is a well-known piece of generative music [48] in which the score is made up of multiple musical sections, each of which can transition to any other section. To play the piece, a performer/player rolls dice to determine which sections to play in which order. If we consider Mozart's dice game as a *game* rather than a music performance, we can consider each dice roll to be a game state change. As this does not present systemic autonomy from the game state, the dice game could be described as having *composed adaptive* music, not generative music.

One problem with this understanding of the *Würfelspiel* is that the music and the gameplay of the game are inseparable - The game has no gameplay outside of constructing a musical piece. While there are video games in which the music is a core component of the gameplay loop, these games generally provide gameplay that is not purely musical. In "musical exploration game" *Fract OSC* [55], the player solves puzzles by moving and interacting with abstracted physics objects, each of which directly controls parameters on a virtual synthesizer. While the music is reactive, and is directly changed based on the gameplay, the music is not the only component of the gameplay. In the *Musikalisches würfelspiel*, there is no gameplay other than the arrangement of the music. This differentiation of gameplay and musical construction is key to our understanding of generative music in games.

For our purposes, music can be considered generative within a video game if the music is produced by a systemic automation that is partially or completely independent of the gameplay. This independence can have a large range of possibilities. A generative linear system may be almost completely independent of the gameplay - a piece of music can be requested, and is then linearly played through regardless of the gameplay. A highly adaptive generative system may use a large array of game variables to inform its generation.

Figure 2.1 gives examples of games that use either generative, adaptive, or both techniques in their music. The two most common uses of music in games are *composed linear* and *composed adaptive* music. We focus this survey on uses of both *generative linear* and *generative adaptive* music.

Table 2.1: Examples of games with linear, adaptive, composed, and generative music.

Source	Linear	Adaptive
Composed	Mega Man	Final Fantasy XV
	Shovel Knight	Luftrausers
Generative	Spore	DOOM (2016)

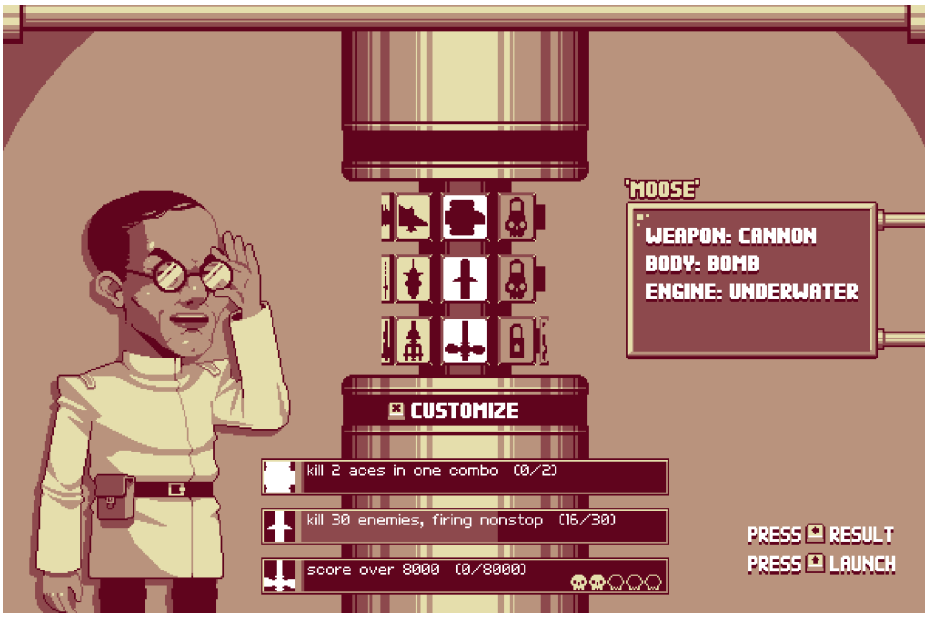


Figure 2.1: *Luftrausers* selection of gameplay parts that influence music [91].

2.1.3 Motivation

Most games have a soundtrack with between 1 and 4 hours of music, but gameplay time can range from 6 to over 100 hours [15]. This often leads to the player hearing musical tracks repeated many times. While repetition is important in music, too much repetition can break immersion and become grating to the player [72, 81]. One contributing factor to the repeated music in games is cost. As an example, *Pillars of Eternity* [69] takes an average of 60 hours to complete. If all 60 hours of gameplay were to be filled with unique composed music, the budget of the game would roughly double [7].

Although generative and adaptive music is becoming more popular in the industry, linear composed music represents a majority of games music, and generative music remains a niche field in both the industry and academia.

Generative systems can also be resource-intensive, and games are often pushing the limits of computing power without generative music [57]. While independent "indie" games often focuses on non-technical artistry, AAA (large-scale, big-budget) games continue to often push the limits of technology, with computational resources primarily directed towards graphical fidelity and fidelity of computational simulations [14]. While historically the technology for games music has kept pace with hardware, the relative resources allocated for music in games remains slim [81].

Another reason that generative music may not have received widespread attention in the games industry is that it is often unpredictable and can be difficult to control. The audio director of *No Man's Sky* (26), Paul Weir, notes that generative music was used in the game with an acknowledgment that it could produce "worse" music than composed music [92]. Generative music can also display the "10,000 bowls of oatmeal" problem [9, 32], where the music is acceptable, but monotonous.

Games are expensive to make, with some game budgets reaching into the tens and hundreds of millions of dollars [81], and emerging technology also requires investment. It is understandable that games companies are not investing in a more expensive, less performative technology that may result in worse music, compared to simply using linear composed music. This lack of industry investment also means that while there is academic research in this area producing advanced systems, the research often takes place without industry collaboration, which limits the academic systems.

The challenges for generative music also present opportunities, however. While the cost of developing a generic generative music system may be high, a generative music system can also provide an amount of content well beyond what a composer is capable of at a much lower cost-per-minute of music. This would allow a game like the previously mentioned *Pillars of Eternity* to fill all 60 hours of gameplay with unique music at a cost well below the cost of 60 hours of composed music.

Generative music is also capable of more personalization of music for the player. Adaptive music allows for music to more closely align with the actions of an interactive game, but cannot necessarily match the extreme breadth of gameplay possibilities. Generative music can be composed in real-time as the player interacts with the game, allowing it to adapt more completely to the player actions.

Finally, generative music can empower and assist human composers. While constraints on creative freedom often paradoxically allow for greater creativity [79, 81], too many creative constraints can be frustrating for human composers, and can limit their expressive range. Generative music would allow composers to focus their attention on the artistic aspects of composing music, and not on the technical details of preparing music for adaptivity. This will allow for greater creative agency for game designers, audio designers, and composers.

While there is some academic research into generative music for games, the research is at a very early stage. Often, systems with a stated goal of integrating into games will not have any integration into games [15]. Additionally, the design decisions for many systems are based entirely on practice and theory [19, 57]. Finally, the evaluation of generative systems often takes place with either very limited video games [60] or without any integration into a video game at all [73]. Without clear and informed design goals or formal in-game evaluation, the benefits of generative music in games have not been demonstrated yet. This in turn results in less opportunity for academic collaboration with industry to advance the field.

One challenge in discussing generative music in games is that there is a wide range of poorly defined terminology in use in the field, and this terminology is often used incorrectly [92, 81]. We present a typology of generative music in games, in the interest of allowing for more structured discussion of the state of the art. We use this taxonomy to present the state of the art as we know it, including peer-reviewed and published papers, public presentations and interviews, and industry uses. We survey both the research and the industrial implementations of generative music in games, and discuss the challenges and prospects of using generative music in games. When discussing a game that uses a system, or a system itself in the text, we refer to Table 2.2 using the convention *system name* (#).

2.2 Typology of generative game music systems

For our typology, we adapt the language used in the MuMe community [54]. The alterations that we make are driven by the interactive nature of games, and the unique requirements that games have for music. Our identified dimensions are shown in Figure 2.2, and as before are to be understood as descriptive rather than prescriptive. We note that many dimensions are not mutually exclusive, and a single system may address multiple aspects of many dimensions.

We have examined 34 generative musical systems from games, and have identified 10 dimensions that form a typology. These dimensions can be grouped into three dimensional types:

- **Sections 2.3 and 2.4: Musical Dimensions**: Due to the complex interactions between musical dimensions, we will first define the terms in Section 2.3, and then apply them to hierarchical dimensions in Section 2.4. These dimensions describe how a system manipulates music.
- **Section 2.5: Gameplay dimensions**, which describe how a system interacts with a game
- **Section 2.6: Architecture dimensions**, which describe the structure and algorithm of a system

2.3 Musical Definitions

The musical dimensions address a system's relation to its musical output. Our four identified musical dimensions are *generative task*, *directionality*, *granularity*, and *grid/groove*. Music is multifaceted, and examining any dimension out of its musical context provides only a partial understanding. We begin our descriptions of musical dimensions by introducing and defining common musical terminology. We then discuss the hierarchical structures that constitute the musical dimensions of our taxonomy.

While there are many ways to analyze music, we use western music theory for our musical dimensions, as we believe that it provides a description of the musical output of the examined systems that can best be taxonomized. Most music for games falls within the western tradition of music, with non-western instruments or harmonies primarily used as a special musical effect [81].

2.3.1 Generative Task

The first musical dimension to consider for a generative system is the generative task that the system addresses. This dimension describes what the system generates. While there are many tasks that a system can address, there are three over-arching families of tasks that generative music systems for games can address.

Composition

The composition task addresses the creation of new music entirely through some process.

Arrangement

The arrangement task addresses the recombination of extant musical elements in new ways.

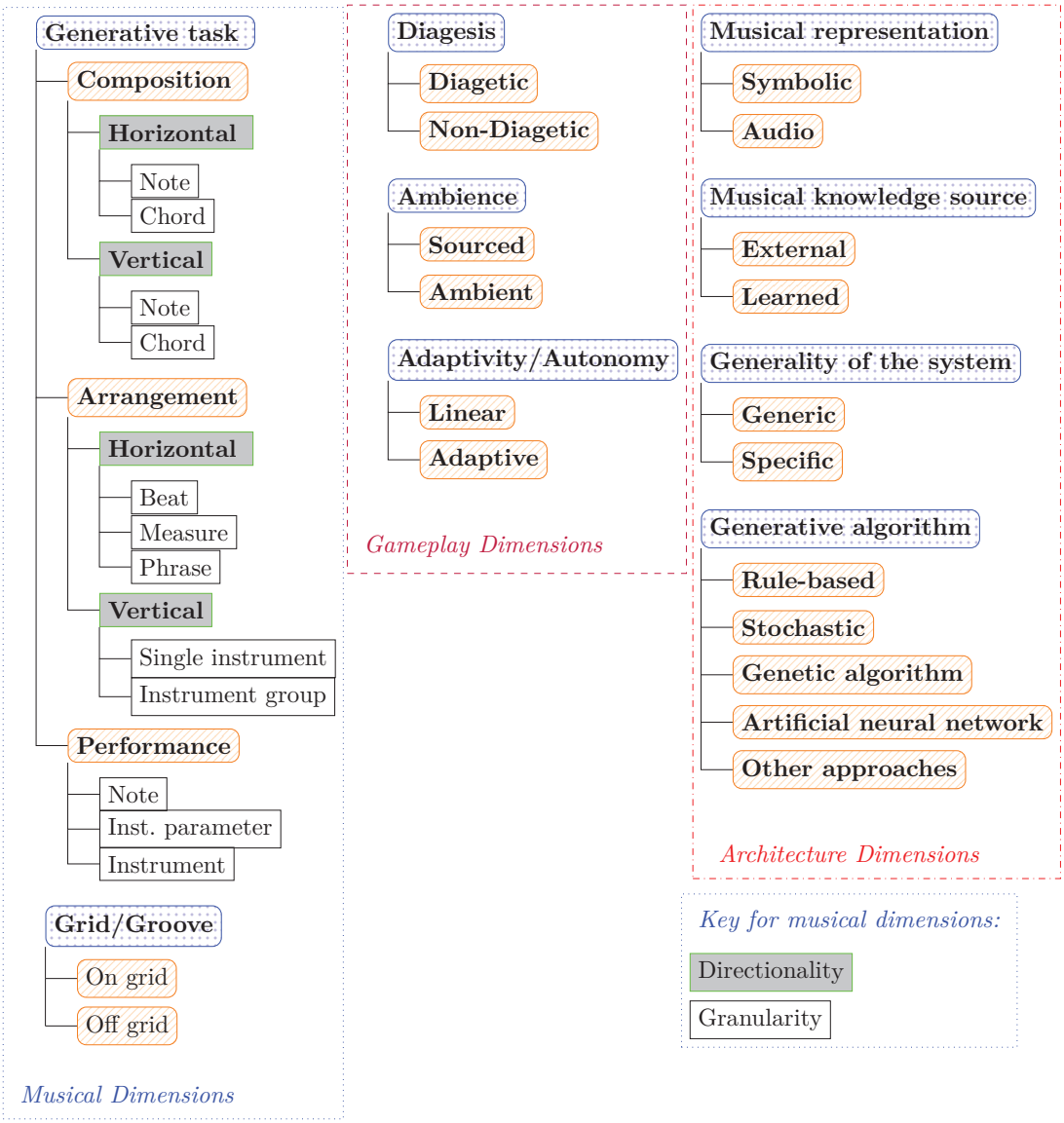


Figure 2.2: Typology of generative music systems for games.

Performance/Interpretation

The performance task addresses the interpretation and synthesis/playback of a composition.

These tasks are not mutually exclusive, and a single system may address any combination of tasks. Most of the systems that we have surveyed address a single task, though *Ballblazer* ①, *Agate/AGMS* ⑫, *Spore* ⑬, *AudiOverdrive* ㉒, *Anthony Prechtl's system* ㉘, *MetaCompose* ㉚, and *Melodrive* ㉛ address multiple tasks.

2.3.2 Directionality

The second musical dimension to consider for a generative system is the directionality of the systems manipulation. Sheet music is arranged with time progressing from left to right in each musical system. Musical events that happen at the same time are represented in sheet music by appearing at the same horizontal position within the musical system. Because of this arrangement on the page, music can be examined horizontally or vertically. The horizontal dimension of music represents the way that the music unfolds over time. The vertical dimension of music represents the way that the music fits together with itself at any single point in time. Systems can also act in mixed or hybrid directionality, manipulating both the horizontal and vertical dimensions of music.

2.3.3 Granularity

In music, individual elements combine to create complex groupings of features. These groupings also combine to form more complex meta-groupings, which can continue to combine in increasing complexity. The granularity dimension addresses what level of groupings the system manipulates. The exact manipulation of different granularity levels depends on both the task and directionality of a system, though there are a few commonalities across music.

2.3.4 Horizontal properties:

Note

A note is a single musical event. Notes usually have a pitch, but this is not necessary. Notes usually have a duration between 0.125 to 2.0 seconds, though may be longer or shorter. While other terminology exists for musical events, the differences are not semantically relevant. We continue to use western music theory inclusively in this case and will refer to any single musical event as a "note".

Beat

A beat is a regular division of time in music. Beats generally have a length of between 0.3 to 1.0 seconds. This is measured in music by the number of beats in a minute. While this range falls within the range of individual notes, a key differentiation between beats and notes is that beats generally maintain a similar duration for longer periods of time, while

note are much more variable. A beat may also be understood as any regular tick, and the terms may be used interchangeably.

Measure

A measure is an organized collection of beats. Most commonly, a measure has either three or four beats.

Phrase

A musical phrase is a collection of notes that combine to form a musical idea. Phrases are most often between four and eight measures long, and can also be combined together to create longer phrases.

Chord

A chord is any selection of two or more notes that are meant to sound together. These notes may either play at the same time or may *arpeggiate*, playing the notes of the chord across time with the intention that they are heard as a single unit. While the creation of a chord is within the vertical dimension, chords may also be treated as a musical unit - the notes C-E-G can be described as simply a C Major chord. When chords are arranged horizontally in series, this is called a *chord progression*.

2.3.5 Vertical properties:

Instrument parameter

In acoustic music performance, a performer of an instrument has a variety of ways to alter the timbre, dynamics, envelope, and more aspects of the sound of their instrument. Digital instruments also have parameters that may be similarly altered.

Instrument

An instrument is a single, internally consistent source of sound, that can be heard as a single musical entity across time. In most cases, a musical phrase will not change instruments midway through, though there are exceptions.

Instrument group

Multiple instruments are often grouped together in music. Instruments are often grouped together due to having similar sonic qualities or similar musical function [81], but may be grouped in any combination.

Chord

As mentioned, a chord is any selection of two or more notes that are meant to sound together. When chords are manipulated along the vertical dimension, individual notes are combined to create or manipulate chords, rather than using common, pre-defined chords.

2.3.6 Grid/Groove

While the previous dimensions describe the way that systems generate music through time, the Grid dimension describes the way that systems understand time. One of the most common interfaces for building drum tracks is a step sequencer, which divides each measure into 16 equal steps, equivalent to one 16th note each. Trackers, another common interface for music composition for video games, also arrange the music as a grid of equal steps. The grid dimension describes whether a system plays musical events at even divisions of time, or whether musical events can happen at any time.

On Grid

Systems that are on grid restrict the timings of their musical events to some regular division of time. Systems that are on grid are sometimes described as having a "groove", though the term is poorly defined. *Rez* ⑧ organizes its musical events on a grid - when the game triggers a chord cluster from player events, the exact timing is changed to fit within an 8th note groove.

Off Grid

Systems that are off grid do not restrict the timing of musical events to any regular division of time. Such systems generally do not restrict the timing of musical events at all. The generative system used in *Spore* ⑬ does not have any restriction on when notes may be played, resulting in aryhthmic music.

2.4 Hierarchical musical dimensions

When the dimensions of Generative Task, Directionality, Granularity, and Grid/Groove are taken out of musical context, they provide an insufficient and incomplete understanding of the musical structures that a system manipulates. To understand the relationships and dependencies of these dimensions, we must also examine several common hierarchical structures that place these dimensions in the proper context.

Our examined hierarchical structures are the most common combinations of the dimensions of task, directionality, and granularity. As before, these common combinations are not mutually exclusive. It is possible for a single system to automate both horizontal and vertical composition, or to automate horizontal composition on a note level of granularity,

vertical arrangement on an instrument level of granularity, and the instrument parameters of performance.

2.4.1 Horizontal Composition

Horizontal composition systems automate the creation of music across time. This may be on one of two levels of granularity:

Note

Horizontal composition systems that function at the note level of granularity automate the creation of new music by selecting a series of individual notes as they will be heard through time. Note that some note-level systems may also create larger groupings of notes that can later be used by an arrangement system, which is similar to the use of *leitmotifs* in composed music. The system by *Cullimore, Hamilton, and Gerhard* (23) addresses the composition task and functions at the horizontal note level.

Chord

Horizontal composition systems that function at the chord level of granularity automate the creation of chord progressions. These chords may either be selected from common chords (e.g. C Major), or may be built procedurally with vertical composition. *MetaCompose* (30) generates chord progressions by choosing common chords from a tree, while addressing the composition task.

2.4.2 Vertical Composition

Vertical composition systems automate the construction of music at points in time. Such a system exclusively functions on a note level of granularity, combining notes to create chords. We have found no systems that fit our scope of generative music for games that exclusively address the composition task in the vertical direction. However, there are systems that include a chord-building element. *The Audience of the Singular* (29) considers the vertical note dimension while composing music.

2.4.3 Horizontal Arrangement

Horizontal arrangement systems automate the combination and recombination of composed musical beats, measures, and phrases in new ways.

Beat

Horizontal arrangement systems that function on a beat level of granularity may either combine individual composed beats together to form larger collections of beats. They may also instead combine larger musical groupings together, but have the option to alter the

music at each beat. Therefore, a beat-granularity horizontal arrangement system may play through phrases completely if left unattended, but choose to change which phrase is playing mid-way through the phrase at a specific beat. In *No One Lives Forever* ⑦, the system plays through phrases entirely while in a single game state, but can transition to different musical phrases on any beat.

Measure

Horizontal arrangement systems that function on a measure level of granularity are almost identical to those that function on a beat level of granularity. These systems can either arrange composed measures together, or have the option to alter longer phrases at any measure. The former is more common at the measure level of granularity than it is at the beat level of granularity. Mozart's previously mentioned *Musikalisches würfelspiel* is an example of a horizontal arrangement system at the measure level of granularity.

Phrase

Horizontal arrangement systems that function on a phrase level arrange composed phrases together across time to create more completed musical pieces. Figures 2.3 and 2.4 provide an example of how horizontal phrase arrangement functions. In Figure 2.3, various musical phrases are provided. In Figure 2.4, these phrases are combined across time to create a novel musical piece. Complete musical phrases generally have a duration of 8-32 seconds. Because changes in game states may occur at any time, if a horizontal arrangement system can only change between musical cues at the end of a phrase, there is a possibility that the music may feel disconnected from the gameplay [81]. Because of this, systems that exclusively address horizontal arrangement in the phrase granularity are rare. For all of the music in Figures 2.3 - 2.5, a recording can be heard at `https://bit.ly/2NKI7rk`. These examples were composed by the first author for the purposes of demonstrating these concepts.

2.4.4 Vertical Arrangement

Instrument

Vertical arrangement systems that function in the instrument level of granularity arrange single instrumental lines into and out of a fuller orchestration. Figure 2.5 demonstrates how instrumental vertical arrangement may work. In the first system, all musical parts (Melody (M, blue), Harmony (H, red), and Bass (B, green)) are playing. In the second and third systems, individual instruments are removed and re-introduced to the total arrangement, creating a new combination of musical elements.

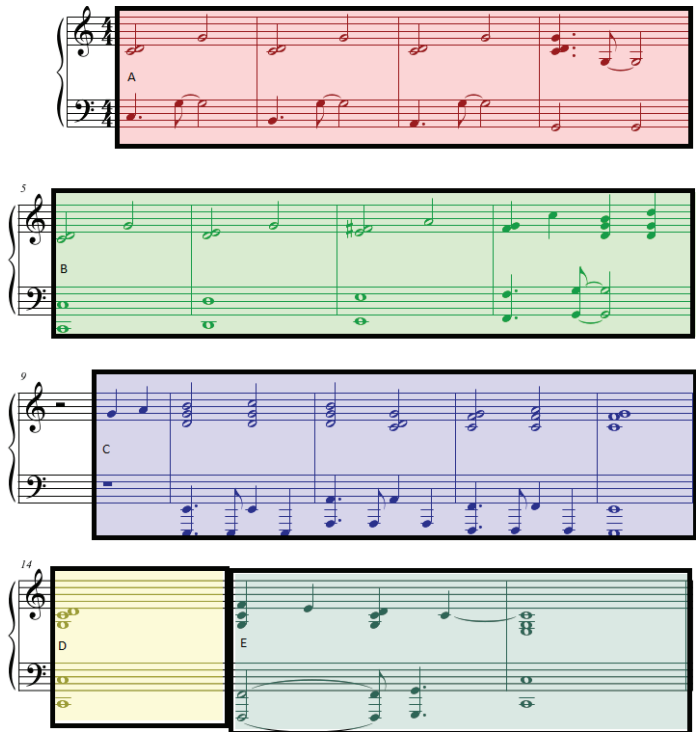


Figure 2.3: Individual phrases for use in horizontal arrangement.

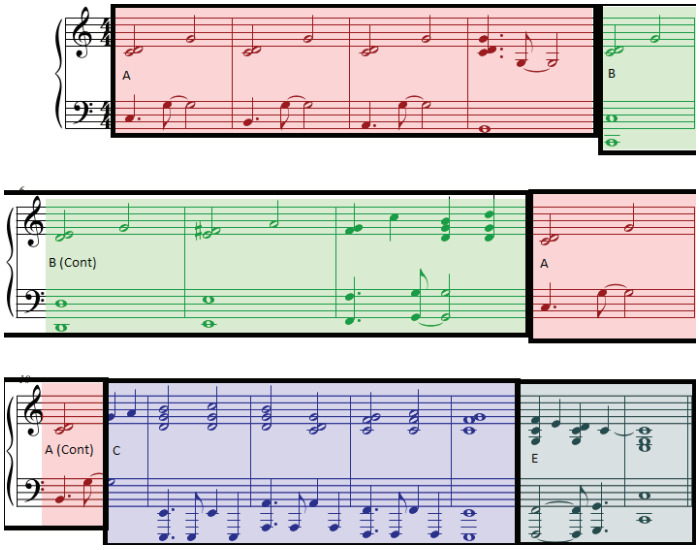


Figure 2.4: Sample horizontal arrangement.

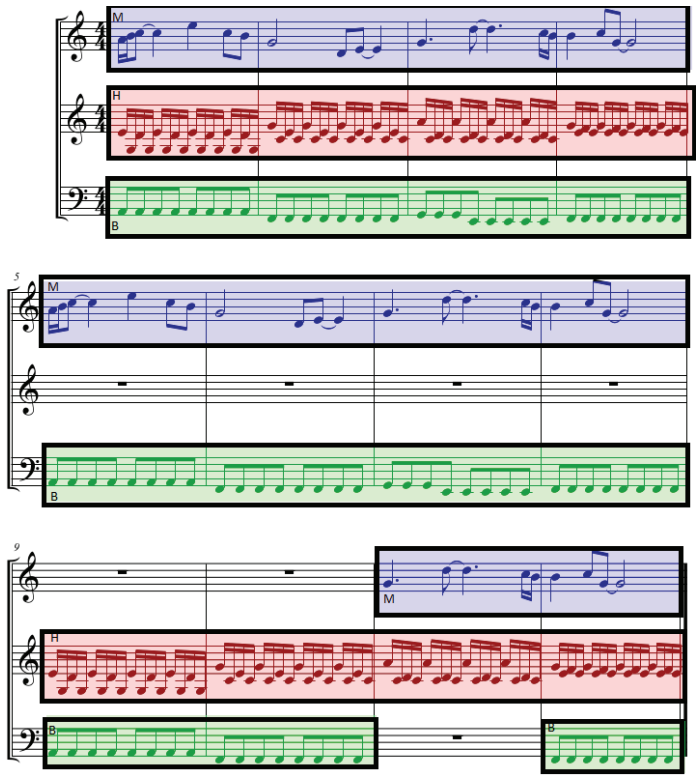


Figure 2.5: Example of vertical arrangement.

Instrument Group

Vertical arrangement systems that function in the instrument group level of granularity are almost identical in function to those that function on the instrument level of granularity. However, instead of removing or adding individual instrumental lines from an orchestration, such systems instead remove or add instrumental families from the mix. The system in *Dark Void* (17) functions on an instrument group level of granularity - the generative system cannot add or remove a single insturment, but instead adds or removes combinations of instruments together.

2.4.5 Performance

Performance systems automate the interpretation of music. Performance systems are not differentiated by their directionality. This is because performance systems alter properties that describe moments of time, but alter the dimensions across time. That is to say, performance systems alter vertical properties horizontally. Performance systems also often alter several properties at once. For simplicity and clarity, we will describe the smallest granular unit that a system manipulates when discussing performance systems.

Note Performance systems may alter the pitch of notes during gameplay. Pitch alterations are generally limited to only one or two semitones. The pinball game *Black Knight 2000* [62] alters the pitch and timing of musical sound effects based on the surrounding musical context [81]. In *Chuchel* (32), most of the sound effects and voice sounds are pitched or semi-pitched. The system alters the pitch of the PCs audio to fit within the surrounding musical context. Performance systems may also alter the timing and duration of notes, or may choose to omit or add notes during gameplay. This is sometimes part of a larger gesture such as a tempo change, or a smaller gesture such as a fermata. Performance systems that add or omit notes often to so to develop or simplify a phrase, or may add musical ornaments. *MetaCompose* (30) can both alter the duration of notes, as well as adding or removing notes from a generated melody.

Instrument parameter Performance systems may automate the parameters of a single or multiple instruments. Some examples of parameters that performance systems may alter are dynamics (velocity/volume), ADSR envelope, and spectrum (using filters or changing the waveform of the sound).

Instrument Effect Performance systems may automate the presence and parameters of audio effects on instruments. Effects are commonplace in composed music, but are generally underused in generative systems. Effects are often used together. We borrow and modify Michael Sweet's taxonomy of audio effects [81], which are separated into three categories:

- **Time-based effects** generally add some form of echo to shape the way that a sound evolves over time. These effects include reverb, delay, chorus, flange, and phase effects.
- **Frequency-based effects** alter the spectrum of the frequency, with effects such as filters, equalizers, or resonators. Vibrato is also an example of a frequency-based effect, where a low-frequency oscillator alters the frequency of a note subtly.
- **Volume-based effect** change the dynamics of music. These effects include a Tremolo, which uses a low-frequency oscillator similar to vibrato, but alters the dynamics instead of the frequency. Other volume-based effects include limiter, compressor, gate, and expander.

Instrument Performance systems may automate which instrument is playing a line. This is distinct from arrangement systems selecting instruments, as a performance system will select *which* instrument to use when playing through a composed line, not *whether* the instrument will play. *Otocky* (2) adaptively changes instruments during gameplay.

These dimensions and common structures describe the musical construction and output of generative systems for games. The next set of dimensions describe the interaction between the music system and the gameplay itself.

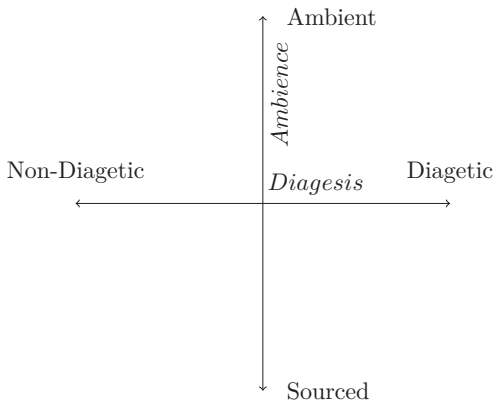


Figure 2.6: Adapted IEZA model of game audio.

2.5 Gameplay dimensions

Our first gameplay dimension is the adaptivity, or autonomy, that the system has. This dimension describes how much the system adapts to the gameplay. Adaptivity can be understood as the dimension that describes how the system deals with input from the game.

We borrow Richard van Tol and Sander Huibert's "IEZA" framework for classifying game audio [83], which describes how the system's output interacts with the game. The IEZA framework is intended to describe all aspects of game audio, and we adapt the framework to focus on music. Figure 2.6 shows our adapted framework. The two dimensions of our framework are *diagesis* and *ambience*.

2.5.1 Diagesis

The dimension of diagesis describes whether the music is diagetic or non-diagetic. Diagetic music is music that exists within the game world, while non-diagetic music exists outside and along the game world.

Diagetic

Diagetic music originates from within the game. This may take the form of audio eminating from an in-game object such as the radio stations on a "pip-boy" in *Fallout 3* [75], it may take the form of an in-game musical instrument such as in *The Legend of Zelda: The Ocarina of Time* [51], and it may even take the form of in-game sound effects having musical pitches, as in the case of *Pole Riders* [17]

Non-diagetic

Non-diagetic music does not originate from within the game. Most games music is non-diagetic, where a musical score simply plays during gameplay. The in-game characters are not aware of non-diagetic music, it is provided exclusively for the player's benefit.

2.5.2 Ambience

The dimension of ambience describe whether the music is connected to a source, or is ambient. Most music in games is ambient and not connected to a source.

Sourced

Music that is sourced is linked to a specific in-game source. A simple example of non-diagetic sourced music is a musical response when the player clicks on a UI button. As an example, in *Chuchel* (32), the only sounds that the player's character makes are abstract non-language vocal sounds, and are pitched to fit in with the musical surroundings. This is an example of a diagetic sourced music.

Ambient

Ambient music is not sourced, and instead emanates from the environment. Most music in games is non-diagetic and ambient. We do not use "ambient" to refer to a musical style, but rather to the use of the music. A non-musical example of a diagetic ambient sound is ambient weather sounds. A musical example of the common non-diagetic ambient sound is in *Halo 2* (11), in which the music is not connected to any source, and does not emanate from within the game world.

2.5.3 Adaptivity/Autonomy

The level of autonomy that a system has from a game describes how the system reacts and responds to the events and state of a game. As we have mentioned, to qualify under our definition of generative, a system must have some degree of autonomy from the gameplay. Because generative music systems for games exclusively output audio of their compositions in real-time with the gameplay, this dimension addresses the adaptivity of the music in the game. The amount of autonomy that a system has is inversely related to how adaptive the system is. The autonomy dimension is also distinct from many other dimensions, because it is a continuous dimension, while other dimensions are more categorical. There are two main divisions of autonomy for game generative music systems:

Linear

Linear systems have a high degree of autonomy from the gameplay. Such systems generate music with little input from the gameplay. This use of music can involve generating a single piece of music for a level/game state as the level loads, that is then used as a linear piece of music. This can also involve creating a musical composition in real-time as the gameplay unfolds, but only using variables that were set at the beginning of the generation. *Spore* (13) presents an example of linear generative music. In *Spore* (13), a piece of music is generated at the same time that the game environment loads. Once the music is generated, it is used

as though it is any other linear piece of music - looping through the music from beginning to end.

Adaptive

Adaptive systems have a lower degree of autonomy from the gameplay. Such systems generally generate their music in real-time, continuously updating the features of their generation to match with the constantly updating game state. This theoretically allows for a generative system to be adaptive on a much deeper level than with composed adaptive music, because the music can be altered more completely than with composed adaptive music. *Melodrive* (31) describes this use of music as "Deep adaptive music".

Reactive

Reactive systems have no autonomy from the gameplay. In *Fract OSC*[55], the gameplay acts as an interactive synthesizer. The music in *Fract OSC* reacts directly to the position and properties of in-game objects, that are directly controlled by the player. While reactive systems are considered generative by some definitions [10], a purely reactive system does not fall within our scope because the system has no autonomy from the gameplay.

2.6 Architecture Dimensions

The musical and gameplay dimensions describe the way that a generative game system interacts with a game, and how its musical output is structured. While these dimensions describe the practical dimensions of a system, there are also architectural dimensions that describe the inner workings and knowledge of a system.

Architecture dimensions describe how a system is internally structured. These dimensions describe the way that a system organizes and understands its data, in contrast to the previous dimensions that describe the way the system manipulates its data.

2.6.1 Generality of the system

As mentioned, generative music does not yet have widespread use in the games industry. Most industrial systems that use generative techniques do so to extend and expand a game's composed music to fit within the game. This also means that industry systems generally are designed specifically for the music and the game that they are integrated into. In contrast, academic systems are generally designed to provide a generic platform that can be integrated into many different games without fundamental changes. The generality dimension measures whether the generative system is designed as a generic platform, or whether the system is designed specifically for a single game.

Generic

Generic systems attempt to create a framework that is game-agnostic. Generic systems are independent systems, that may be integrated into multiple different games without requiring major systemic overhaul. Generalist systems are most common in academic systems, but there are systems used in the game industry as well. *DirectMusic* ④ and *Agate/AGMS* (12) are both examples of generic music generation systems.

Specific

Specific systems are designed explicitly for the game that they provide music for. Such a system is intrinsically linked into the game that it provides music for. These systems generally are designed based around the surrounding musical context and gameplay events and variables for the game that they provide music for. Game-specific systems are most common in industrial uses. *Red Dead Redemption* (16) provides an example of a game-specific system. The musical system in *Red Dead Redemption* (16) takes important game variables as input. Some of the tracked variables and musical reactions are generic and applicable to many games, such as playing faster and more active music during combat sequences. However, many of the tracked variables and reactions are specific to *Red Dead Redemption* (16), such as changing the music when the player mounts a horse. While the core idea of an adaptive system that reacts to game events could be used elsewhere, removing the game context from this system would fundamentally change the workings of the system, and as such it is not generic.

2.6.2 Generative Algorithm used by the system

While the generality of a system describes how it fits into the larger world of the game, the algorithm of a system describes how it creates music. Almost any AI algorithm can and has been used for computational creativity purposes [54]. Listing all of the possible generative musical algorithms far exceeds the scope of this survey. Instead, we focus on the four most common algorithms for use in MuMe. Of these four algorithms, only two are common in generative systems for games.

Rule-based

A rule-based algorithm uses a set of rules, either learned or programmed in, to generate its output. A simple example of a rule based system in the symbolic music domain is one that uses species counterpoint[2] to create harmony lines. *Rez* ⑧ uses a rule-based system where the exact timings of musical stings depend on a set of rules determined by the surrounding

[2]"Species counterpoint" is a strict adherence to certain musical relationships in melody and harmony. It is generally used as a pedagogical method to teach melodic and harmonic writing [44]

musical context. Some rule-based systems are purely deterministic - given identical input they will produce identical output. Some rule-based systems instead are non-deterministic, often using elements of stochasticity.

Stochastic

A stochastic system uses a pseudorandom process to generate its output. This randomness may be evenly distributed or be the result of a statistical model providing weighted changes. A Variable-order Markov model (VOMM) is an example of a statistical model that uses stochastic methods to generate music. In a VOMM, the generated music is determined by a probability that is based on the previous music. The VOMM then selects which note to choose next by a random or weighted random chance. *The Audience of the Singular* (29) uses a varaible-order Markov chain with shifting probabilities to generate its music.

Genetic Algorithm

A Genetic Algorithm (GA) is modeled after the natural selection process as seen in nature. A GA begins with a set of randomly generated states. Each state is then evaluated by a fitness function. The fittest states are then combined using some process. This process can include random mutations as well [67]. Interactive Genetic Algorithms, which use a human subject to determine fitness, have been used in generative music systems [5]. We have identified one system, *MetaCompose* (30), which partially uses a GA to address the composition task.

Artificial Neural Network

An Artificial Neural Network (ANN) is an algorithm that is modeled after the human brain. ANNs are composed of neurons, connected by links. These neurons may be in a single layer, or may have hidden layers of perception and activation [67] ANNs are capable of forming complex statistical distribution models, and are differentiated from other stochastic methods because they form *connections* rather than *rule-based distributions* [47]. While ANNs are gaining popularity in MuMe, we have only identified one generative system for game music that uses ANN algorithms.

Other Approaches

There are many other algorithms that can be used for generating music, and describing every possible generative algorithm for music is far outside of the scope of this survey. The listed algorithms are the most common algorithms in generative music systems, with almost all generative music systems for games using rule-based or stochastic algorithms. Further information on generative music algorithms can be found in an online class on *Kadenze* [53].

2.6.3 Musical Representation

As mentioned, almost any AI algorithm can be used for generative music. While the algorithm that is used describes how the system manipulates its musical knowledge, the musical representation dimension describes how a system stores its knowledge of music. Because a generative system for game music will at some point produce audio, there are limits to the range of possibilities for the knowledge representation of systems. Surveyed systems either store their knowledge *symbolically*, and contain the ability to synthesize audio, or they contain *audio clips*, which are streamed from the storage media.

Symbolic

Systems that represent their knowledge symbolically can use any symbolic notation to represent the music. The most common notation in symbolic music representation for computing systems is MIDI, in which each musical event is represented by a series of variables such as pitch, velocity, channel, on/off, etc. *The Audience of the Singular* (29) represents music using MIDI values. *AOTS* (29) uses symbolic representation both in the corpus that supplies its knowledge, and in the representation of its output, which is then synthesized in real-time.

Audio

Systems that represent their knowledge with audio combine pre-existing samples of sound together into musical pieces. These systems are more common in industry uses of generative arrangement of music, where recording composed samples of music and arranging them adaptively has a similar work-flow to using composed adaptive music.

2.6.4 Musical knowledge source

While the algorithm dimension describes how a system manipulates its musical knowledge, and the musical representation dimension describes how a system stores its knowledge, the musical knowledge source describes where the knowledge originates. This knowledge can come from one of two sources:

External

Systems with external knowledge have their features, parameters, and values input by either their user or creator. The "user" in this case may describe the game's player, but often describes the composer or audio designer, if they are not the creator of the system as well. In generative systems for games, external knowledge is exclusively provided by the system's creator. One example of a system that uses external knowledge is seen in *Anarchy Online* ⑨. In *Anarchy Online* ⑨, each musical transition was hand-coded by the composers.

Learned

Generative systems may also take their knowledge from analyzing a corpus of musical input, and building a statistical model based on the input. Such systems may analyze either audio or symbolic data, though in the examined systems, only symbolic corpora have been used. *The Audience of the Singular* (29) builds a Variable-order multiple-viewpoint [13] Markov chain by analyzing a corpus of MIDI files. The MIDI files for *The Audience of the Singular* (29) were arrangements of music from the Super Nintendo Entertainment system and the Nintendo Entertainment system. The Markov chain for *The Audience of the Singular* (29) is learned a single time, before gameplay. During gameplay the system uses the existing, pre-trained Markov chain.

This concludes our taxonomy of generative music systems for games. In the next section, we will now examine the systems that fit within the scope of generative systems for games along this taxonomy.

2.7 Examination of musical systems

Table 2.2 shows the examined extant systems for generative music and audio in games. These games are listed in chronological order of release. It is important to note that while we have done our best to organize and collect data on these systems, in many cases the systems are used in commercial products. Because of this, the information concerning the systems is not always complete. Finally, for clarity we will simply be referring to the games that use a generative music system by the title of the game.

Table 2.2: Extant game and academic systems examined within taxonomy.

Identification			Musical dimensions				Gameplay dimensions			Architecture dimensions			
Number	Game/System	Year	Generative Task	Directionality	Granularity	Grid	Diagesis	Ambience	Adaptivity	Generality	Gen. Algorithm	Musical Rep.	Musical KS
1	Ballblazer	1984	All	Horizontal	Phrase, Note	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Symbolic	External
2	Otocky	1987	Performance	N/A	Note	On	Diagetic	Sourced	Adaptive	Specific	Rule-based	Symbolic	External
3	iMuse	1991	Arrangement	Horizontal	Measure	On	N/A	N/A	Linear	Generic	Rule-based	Symbolic	External
4	DirectMusic	1996	Arrangement	Mixed	Measure	On, Off	N/A	N/A	Adaptive	Generic	Rule-based	Symbolic, Audio	External
5	GhostWriter	1998	Composition	Horizontal	Note	On	Non-Diagetic	Ambient	Adaptive	Specific	Rule-based	Symbolic	External
6	Munge/MNG	1998	Arrangement	Horizontal	N/A	Off	Non-Diagetic	Ambient	Adaptive	Generic	Rule-based	Audio	External
7	No One Lives Forever	2000	Arrangement	Horizontal	Beat	On	Non-Diagetic	Ambient	Adaptive	Specific	Rule-based	Symbolic	External
8	Rez	2001	Performance	N/A	Note	On	Non-Diagetic	Sourced	Linear	Specific	Rule-based	Audio	External
9	Anarchy Online	2001	Arrangement	Mixed	Phrase, Instrument	On	Non-Diagetic	Ambient	Adaptive	Specific	Stochastic	Audio	External
10	Diner Dash	2004	Arrangement	Horizontal	Phrase	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
11	Halo Series	2004-2015	Arrangement	Vertical	Instrument	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
12	Agate/AGMS	2008	Composition, Arrangement	Mixed	Note, Inst.	Off	N/A	N/A	Linear	Generic	Rule-based	Mixed	External
13	Spore	2008	Composition, Performance	Horizontal	Note	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Symbolic	External
14	Tom Clancy's EndWar	2008	Arrangement	Mixed	Phrase	On	Non-Diagetic	Ambient	Adaptive	Specific	Rule-based	Audio	External
15	Uncharted 2: Among Thieves	2009	Arrangement	Horizontal	Phrase	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
16	Red Dead Redemption	2010	Arrangement	Mixed	Phrase, Instrument	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
17	Dark Void	2010	Arrangement	Vertical	Inst. Group	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
18	The NLN Player	2011	Arrangement	Horizontal	Phrase	On	Non-Diagetic	Ambient	Linear	Generic	Stochastic	Audio	External
19	Child of Eden	2011	Performance	N/A	Note	On	Non-Diagetic	Sourced	Linear	Specific	Rule-Based	Audio	External
20	Bit.Trip Runner	2011	Performance	N/A	Note	On	Non-Diagetic	Sourced	Adaptive	Specific	Rule-Based	Audio	External
21	Remember Me	2013	Arrangement	Vertical	Inst. Group	Off	Non-Diagetic	Ambient	Adaptive	Specific	Stochastic	Audio	External
22	AudiOverdrive	2013	Arrangement, Performance	Mixed	Phrase, Instrument	Off	Diagetic	Sourced	Adaptive	Specific	N/A	Audio	External
23	Cullimore, Hamilton, and Gerhard	2014	Composition	Horizontal	Chord	On	N/A	N/A	Linear	Generic	Stochastic	Symbolic	Learned
24	Engels et al.	2015	Composition	Horizontal	Note, Chord	On	N/A	N/A	Linear	Generic	Stochastic	Symbolic	Learned
25	Sonancia	2015	Arrangement	Horizontal	Phrase	On	Non-Diagetic	Ambient	Linear	Specfic	Rule-based	Audio	Learned
26	No Man's Sky	2016	Arrangement	Mixed	N/A	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
27	DOOM (2016)	2016	Arrangement	Horizontal	Phrase	On	Non-Diagetic	Ambient	Linear	Specific	Rule-based	Audio	External
28	Prechtl	2016	Composition, Performance	Horizontal	Chord	On	Non-Diagetic	Ambient	Adaptive	Generic	Stochastic	Symbolic	External
29	The Audience of the Singular	2017	Composition	Horizontal	Note, Chord	On	Diagetic	Sourced	Adaptive	Generic	Stochastic	Symbolic	Learned
30	MetaCompose	2017	Composition, Performance	Mixed	Note, Chord, Inst. Parameters	On	N/A	N/A	Adaptive	Generic	Stochastic, GA, Rule-based	Symbolic	External
31	Melodrive	2018	Composition, Arrangement	Mixed	N/A	N/A	N/A	Ambient	Adaptive	Generic	N/A	Symbolic	External
32	Chuchel	2018	Performance	N/A	Note	Off	Diagetic	Sourced	Linear	Specific	Rule-based	Audio	External
33	Ape Out	2019	Performance	N/A	Instrument	On	Non-Diagetic	Ambient	Adaptive	Specific	Rule-based	Audio	External
34	Adaptive Music System/AMS	2019	Arrangement	Horizontal	Note	On	Non-Diagetic	Ambient	Adaptive	Generic	GA, RNN (Stochastic)	Symbolic	Mixed

2.7.1 Composition Systems

Horizontal composition

GhostWriter (1998) [63] The proposed academic education-oriented game *GhostWriter* ⑤ uses a musical system that maps in-game tension to musical tension using a rule-based system with random pitch selection. As far as we can determine, neither the musical system nor the proposed VR game were developed beyond the proposal phase.

As with many other academic musical systems, *GhostWriter* ⑤ attempts to map in-game activities to a dimensional model of affect [18], and specifically targets the dimension of tension. *GhostWriter* ⑤ does not attempt to use any automatic recognition of affect during gameplay, and instead uses a human director, who observes the gameplay and attempts to match the level of tension to the gameplay events. Because *GhostWriter* ⑤ has a stated design goal of acting as a classroom activity, Robertson assumes the presence of a teacher or facilitator who can mediate and navigate the experience and music.

GhostWriter ⑤ uses a three step process to generate it's music. The first step is the creation of a high-level form, though the authors do not provide further information on this step. Once a form is created, the system generates rhythmic data by selecting rhythmic "feet", based on the rhythmic feet of poetry. These feet are provided tension levels by the creators of the system. Once rhythms are generated, the system uses a version of Arnold Schoenberg's Theory of Harmony to create first a chord type (major, minor augmented, diminished). The system then creates a melody by choosing random pitches that fall within a set of constraints as decided by the harmony and rhythm. The system then creates an accompaniment using random generation within a simpler, more harmonic-focused set of constraints. Finally, the system assigns instruments to all parts, based on the provided tension level. The rules and constraints for each of these generative steps were hand-crafted by the system authors.

Spore (2008) [94] The generative system in *Spore* ⑬ automates both the composition task and performance task, using a modified version of PureData, called "EA-PD". Theoretically, the music system in *Spore* could be used as a generic system, as it does not require game-specific messages or information, but publisher Electronic Arts has not published the system for external use. The system in *Spore* creates its music primarily by generating multiple independent lines randomly, and the randomness is controlled via seed manipulation. This is a multi-agent approach to music generation [82], where simple individual generative agents combine to create complex combinations.

The music system for *Spore* is both reactive and linear. The generation parameters are directly controlled by game state changes, but the music that is generated is the played back as linear music. *Spore* represents the only linear generative system that we are aware

of. The mappings of the game state and generative parameters are externally provided by the composers and audio designers for *Spore*.

Little information is available about how *Spore*'s system addresses the performance task, though the creators describe the system's capability to apply DSP effects in real-time [37].

One of the reasons that generative music is uncommon in video games is that generative systems are often CPU-intensive. While this presented a possible problem in *Spore* due to the large amount of other PCG that can have large CPU draw, the designers of the system solved this issue by mainly using generative music during parts of *Spore* where there is limited game logic, and therefore the CPU is more free to be used on music generation [37].

Cullimore, Hamilton, and Gerhard (2014) [15] The next system that we examine is an unnamed system from *Cullimore, Hamilton, and Gerhard* (23). While this system does address the composition task, it does not generate complete musical pieces. Instead, this system targets a weakness that is common in horizontal arrangement systems by composing short, chord-based transitions between musical cues.

One challenge in creating horizontal arrangement systems is that musical transitions can sound jarring if not properly handled [81]. Hand-creating transitions for each musical possibility requires a large amount of labor. Music may also be written for a horizontal arrangement such that any transition will sound acceptable, as in the case of *Red Dead Redemption* (16)'s musical system. Cullimore, Hamilton, and Gerhard's system attempts to computationally create chord progressions that can transition between any two bichords. The system encodes chords in a 2-dimensional space, with each state consisting of a pair of notes. The space is organized such that horizontal movement changes transposition of notes, and vertical movement changes the intervalic distance between the notes. The system is capable of creating chord progressions that link two chords together, though it is limited. The system exclusively generates bichords, and the authors do not mention whether transitions can be altered in case the game state rapidly changes. Also, this system seems to generate chord progressions without any sort of rhythm - in order to use this system in a game setting, some additional rhythm logic is required.

Engels et al. (2015) [19] Another unnamed generic system is presented by *Engels et al.* (24). This system represents the first use of a Markov chain to address the composition task in games. Engels et al.

The first and most basic model that Engels et al.'s system uses is a Markov chain. This Markov chain encodes musical events that occur at the same time together into a state. This has the effect of adding flexibility to the number of simultaneous voices that the system can play, as a state with a single note may lead to a state with a fully voiced chord.

Engels et al.'s system separates musical sections into different models. The Engels et al. system automatically segments music by using a Support Vector Machine (SVM)[3] to group similar musical sections, based on pitch, duration, timbre, and volume. In addition to the Markov chain and the segmentation, Engels et al.'s system uses a Hidden Markov Model, with chords as hidden states. These chords can be provided externally, or if not provided the system will attempt to automatically tag the chords. The hidden chord layer of the Markov chain is used to restrict individual voices so as to avoid clashes.

Prechtl (2016) [60] Anthony Prechtl created both a generative system and a game, integrating the system into the game for evaluation purposes. The game that Prechtl created is titled *Escape Point*, and is a abstract horror game.

Prechtl's system uses a Markov chain to compose a chord progression that loosely adapts to the level of tension during gameplay. As with *Ghostwriter* ⑤, this design is based off of Schimmack and Grob's 3-dimensional model of affect [70]. During gameplay, the system adjusts the probabilities for each chord based on the game state and tension level in the game. The tension in the game is represented by how close the player is to an enemy non-player character.

Prechtl's system has 12 parameter sets that control the generation of music. Seven parameters adjust the probabilities of each chord transition within the Markov chain. The other four parameters alter the performance of the music, altering the volume, velocity, timbral intensity, and the presence of a pulsing tone. The system uses two sets of parameter presets - one for low tension and one for high tension. The higher tension preset trends towards less major chords, more diminished chords, less tonal, and less diatonic than the lower tension presets. For the performance values, the higher tension presets are at a higher volume, velocity, timbral intensity, and pulse volume.

Prechtl's system is the only system from academia that we are aware of that has been evaluated in a video game setting. Prechtl found that for players with experience with games similar to *Escape Point* preferred the generative music to linear composed music. Prechtl also found that all players found the generative music more tense and exciting compared to the linear composed music. Finally, skin conductance responses were consistent with these findings, though Prechtl describes difficulties in analyzing the data. This provides both subjective and objective support of the strengths of generative and adaptive music.

The Audience of the Singular (2017) [58] *The Audience of the Singular*, or "*AOTS*", is similar to *Prechtl* ㉘'s system in that it is a generative composition system that uses modified Markov chains to generate music. It is also similar to *Prechtl* ㉘'s work as the

[3] A support vector machine is an algorithm that learns classifications from a pre-classified input dataset. The SVM then can use this data to classify new, unlabeled input data.

developer also built a game around the system. *AOTS* has a symbolic representation of music that is learned from a corpus of 30 pieces of video game music from the late 1980s and early 1990s. The game for *AOTS* was built to interact with the music system specifically, but the music system itself is generic, and could function without the surrounding game.

AOTS uses a variable-order Markov for its core horizontal generation, which is learned from the corpus offline. The VOMM creates four versions of five musical lines, one phrase at a time, starting with the bass line. *AOTS* uses four different VOMM models, which are independently learned by analyzing selected musical phrases from the corpus. For all lines, a rhythm line is first constructed. The VOMM organizes rhythms into beats, with 1-4 16th notes per beat. The probabilities for beats are based only on the previous chosen beats. Each beat also may end with a tie, allowing for syncopation.

The bass line is also the most simple version of the markov chain - the bass line probabilities are based exclusively on the previous notes and the length of the phrase. As the phrase approaches a length of 16 measures, the bass line begins to weigh more heavily towards functional cadence relationships. The system then composes a primary melody whose probabilities are based on the previous notes, a learned melodic contour based on phrase length, and the notes of the bass line. A secondary melody is then constructed similarly to the primary melody, though the probability spread is heavily changed by the distance to the melody. Finally, the system composes two harmony lines whose probabilities are based on the previous notes, bass line, and melody lines. The secondary harmony line also considers the primary harmony line. Once the generation is complete, *AOTS* selects a drum part from several composed grooves. To create the variation lines, *AOTS* removes notes and extends durations of notes that occur on weaker beats, with the final variation containing long notes that begin on primarily strong beats. All of the probabilities for the VOMM are learned from the corpus.

2.7.2 Arrangement systems

Horizontal arrangement

Ballblazer (1984) [42] The earliest use of generative music in video games is 1984's *Ballblazer* ①. Ballblazer uses a generative technique that the creator calls "Riffology" [41], which is a catch-all term that can be applied to any system that uses constrained random selection of notes to generate music. *Ballblazer* ①'s generative system is unique in that it addresses all three generative tasks on grid. The system arranges accompanimental music horizontally at a phrase granularity, and it composes and performs musical melodies at a note level of granularity.

The system is *Ballblazer* provides music between games and in menus, but does not provide music during gameplay. The system has a corpus of 16 measure melodic fragments (called "riffs" by designer Peter Langston), and 4 measure accompaniment sections containing a bass line, drum line, and chords. These corpora of melodic fragments and accompani-

ment sections are provided by the system's designer, and contain notes exclusively drawn from an aeolian scale based on a (a natural minor), with an added ♭.

Details on the generation of the 4 bar accompaniment sections is limited, though Langston describes the logic as a "simpler form" of the melodic system. The melody composition generates several possible "riffs", or collections of randomly-selected notes within the provided scale. The system then chooses the riff that begins on a note that it closest to the note that ended the previous riff. Once the riff is selected, the system performs the riff by choosing to omit notes depending on variables, as well as determining the speed and volume of the performance.

iMuse (1991) [93] Surprisingly, the generative system with the largest effect on the state of the art for generative arrangement systems in games was created in 1991. *iMuse* and systems that extend its design by and large represent the state of the art in the games industry [57]. Because *iMuse* is a generic system that has been integrated into several games, it cannot be evaluated as a system along gameplay dimensions, though it is most commonly used to provide linear, ambient, non-diagetic music.

At its most basic functionality, *iMuse* plays a piece of music that is stored symbolically. If there are no game changes, *iMuse* will play the music as written, linearly. When there is a game state change, *iMuse* attempts to seamlessly transition the music to fit the new game state. This can be done by ending the music, or by transitioning between two musical pieces. Peter Silk created a short video that demonstrates *iMuse* transitioning between two musical pieces seamlessly, available at `https://bit.ly/1R39FPY` [74].

iMuse is, importantly, an arrangement system. This means that it does not compose new music, but instead resequences and alters composed music. For *iMuse* to create seamless transitions, musical content must be composed to enable the seamless transitions. This means that the music composed for *iMuse* must be in complimentary keys, at similar or complimentary tempi, and cannot use extended harmonies such as modal borrowing[4]. Also, the musical library, or the music used in *iMuse* must be annotated by hand to provide the system with information on where the music may transition.

Munge/MNG (1998) [10] *MNG* ⑥, pronounced and sometimes written as "Munge" is a filetype that stores generative music instructions for the games *Creatures 2*, *Creatures 3*, and *Docking Station*. These files contain a variety of rules and game states that correspond with an external corpus of audio music. The system in these games (which we will refer to as *Munge* for simplicity) interprets this data to address the arrangement task in a horizontal directionality. We are unable to determine the granularity of the system, though it is capable

[4]Modal borrowing is a compositional technique where notes or chords from a parallel mode are used. A common example is the use of a borrowed flat VI chord from the parallel minor of a major key

of both on grid and off grid generation. *Munge* ⑥ generates music that is non-diagetic, ambient, and adaptive. The system is generic, and uses a rule-based algorithm with external knowledge source and an audio representation of music.

The *MNG* filetype allows a programmer to set individual voices with musical variables, based on in-game variables, as well as randomness. Each *MNG* file is associated with a specific level of a game. As the game state changes, *Munge* receives as input a "mood" variable. The system then arranges the music based on a combination of the mood variable and randomness, according to the ranges and variables set up in the associated *MNG* file for the current level.

No One Lives Forever (2000) [34] *No One Lives Forever (NOLF)* ⑦ uses a system that refines the *iMuse* design. While *iMuse* ③ functions at a measure level of granularity, *NOLF* can transition between musical cues on any beat.

The primary refinement from *iMuse* to *NOLF*, beyond the granularity change, is that the system in *NOLF* can also alter the pitch and tempo of its musical library. This reduces the restrictions on the music, as the system can alter the tempi and keys of musical transitions to avoid jarring transitions. One weakness of this approach is that the musical library requires additional annotations to work within the system. While the system is capable of matching tempi and keys, it cannot automatically detect either. Essentially, the system in *NOLF* provides better flexibility and more adaptivity in the music, but at the cost of increased data entry labour.

Diner Dash (2004) [28] *Diner Dash* ⑩ uses a simple system to generate music with large amounts of variety from a small amount of composed music.. In *Diner Dash*, each musical cue is separated into phrases, any of which can lead to another phrase within the same cue. During gameplay, the system plays the phrases in a random order, based on the cue, which in turn is based directly on the game state.

Uncharted 2: Among Thieves (2009) [49] *Uncharted 2: Among Thieves* ⑮ uses a very simple generative arrangement system, that targets one specific problem in game music. In most games, when the player fails a gameplay segment or challenge, they return to a previous point in the game and attempt the challenge again. In games with linear composed music, this often re-starts the musical piece that is associated with the current game state. The music in *Uncharted 2* has multiple possible starting points, provided by the composer. When the player fails a gameplay segment, they are returned to the checkpoint, and the musical system selects a random starting point in the music. This avoids the player hearing the exact same music in the exact same way multiple times.

Nln-player (2011) [85] The *Nln-(Non-LiNear)-player*, which is demonstrated in the game *Shortburst*, uses a simplified version of a Markov chain to address the arrangement task. Most uses of Markov chains for generative music store individual notes or chords as single states to address the composition task. However, the *nln-player* uses a "cell-based" design inspired by a Markov chain where each state is a composed musical phrase, and transition matrices are unweighted. Essentially, the *nln-player* uses a Markovian design to ensure that transitions between states will not have any jarring or unexpected transitions. Within any game state, this design essentially means that the music is pseudorandomly shuffled, similar to other horizontal resequencing techniques [11, 81].

The nln-player's design necessarily involves restrictions on the music that is composed for it. Phrases that can transition to each other must be consistent in key, tempi, and mode, as in *iMuse*. Additionally, the composer must hand-annotate a configuration file within a specific metadata format. Also, the system can only interpret specifically formatted filenames.

DOOM (2016) [78] As with many horizontal arrangement systems, the system in *DOOM (2016)* (27) presents another refinement on the architecture of *iMuse* ③, and arranges composed musical phrases together based on both the game state and the surrounding musical context.

DOOM's generative system is similar to the designs presented in the *nln-player* (18), *The Audience of the Singular* (29), and *Engels et al.* (24), in that *DOOM* draws from multiple corpora separated as musical phrases. *Doom*'s corpora divisions are based a standard song structure of intro-verse-chorus-bridge-outro. Each phrase type has approximately 30 potential composed phrases. During gameplay, the system determines which phrase type to play based on the surrounding gameplay. The system then choses a musical phrase at random from the corpus.

There are two notable aspects of the music system in *DOOM*. *DOOM*'s gameplay and music are built to be complimentary, with the flow of the game matching a standard song form, according to composer Mick Gordon [61]. This facilitates the design of the system, as the possibility space for the music is also contained.

The second notable aspect of *DOOM*'s system is that, similar to other horizontal arrangement systems, the music is heavily constrained for the composer. As seen elsewhere, because *DOOM* allows for any phrase to transition into any other phrase, and cannot alter the pitch content or rhythmic content of its corpus, all pieces must be written in complementary keys and tempi. *DOOM*'s system does not require files to be annotated as fully as other examined systems such as the *nln-player*.

Horizontal arrangement systems allow for music to more seamlessly transition between otherwise different musical states, as exemplified by *iMuse* ③ and *NOLF* ⑦. These systems also allow for greater variety in the musical content, as seen in *DOOM* (27) and the *nln-*

player (18). We will now focus on systems that use vertical arrangement, which is sometimes referred to as "adaptive stem mixing" or "vertical layering".

Vertical arrangement

We have identified two systems that exclusively target vertical arrangement. Both of these systems provide increased musical variety, and both do so with less compositional restrictions than horizontal arrangement systems require. Because these systems share many elements of function, we examine them together.

Halo series (2004-2018) [45], Dark Void (2010) [1] The *Halo* (11) series share a music system. While the system has evolved as the games evolve, the core design remain consistent. *Dark Void* (17) uses a similar system. Both of these systems add or remove groups of instruments in real-time, which lets the player hear a single piece of music with many different arrangements, reducing the fatigue from repetition.

The key difference between the systems of *Halo* (11) and *Dark Void* (17) is their granularity. While *Halo*'s system can create much more variety due to the number of controllable instruments, *Dark Void*'s system provides more control to the composer, as the composer can decide which instrument groups will sound better together.

An advantage to using vertical arrangement is that the music can be written without any horizontal restrictions. The only restriction that vertical arrangement systems place on the musical corpus is that during any piece of music, any collection of parts must be able to play together. This is a minimal restriction, as most instrument parts are composed together in music by default. Also, unlike horizontal arrangement systems, there is only limited annotation required for the musical corpus, as different instrumental parts of a single composed musical piece can generally be assumed to play in the same tempo, with complementary notes playing at any given time.

Mixed arrangement

DirectMusic (1996) [46] *DirectMusic* ④ presents a major improvement on the *iMuse* ③ design. As with *iMuse* ③, because *DirectMusic* is a generic system, it cannot be evaluated along gameplay dimensions, but it is most often used to provide ambient, non-diagetic music. Unlike *iMuse* ③, *DirectMusic* is capable of adaptive music, and can generatively arrange in the vertical dimension.

DirectMusic can simultaneously play multiple musical sources, and reads from MIDI, WAV, and a proprietary format that contains both symbolic data and a wavetable for synthesis, similar to MOD files. *DirectMusic* can individually add or remove parts from any filetype, allowing for vertical arrangement. In a departure from the *iMuse* design, *DirectMusic* can also alter MIDI events, allowing it to alter pitch and tempo, and can apply DSP

effects such as reverb. *DirectMusic* can also spacialize audio, providing 3d sound. These capabilities provide adaptive audio e.g. a single symbolic musical cue may change between a major and minor mode based on the game state.

Anarchy Online (2001) [25] *Anarchy Online (AO)* ⑨ uses a system called the "Sample-based Interactive Music tool", or "SIM tool" [40]. The *SIM Tool* implements a Marovian algorithm similar to the *nln player* ⑱, where each state represents a short musical phrase.

Anarchy Online is a Massively Multi-player online game, a type of game where thousands of simultaneous users interact with a persistent world. These games generally have high amount of game content, with players regularly spending over 100 hours in the game world. This high gameplay length exacerbates a problem of linear composed music - the player may hear single musical cues multiple times, resulting in boredom [81, 40]. As far as we are aware, *Anarchy Online* is the only game of this genre to use generative music to address this concern.

Music composed for the *SIM tool* is split into short clips, and each clip is individually sampled to create consistent reverb trails. The clip must be annotated with tempo, meter, "layer" (an associated game state), and a transition matrix to other clips. Clips contain three to five transitions without switching layers, and "a fair number" [40] of transitions to other layers. Layers represent both horizontal and vertical responses to game states. Audio Designer Bjørn Lagim provides an example of both - as the player moves between environments, the system will transition between layers, arranging music horizontally to differentiate the environments. When the player engages in combat, the *SIM tool* adds or removes layers from the 14 available combat layers, based on the relative health points of the player and their opponent, as well as the size of the opponent, arranging music vertically to adapt to the gameplay.

Lagim describes several drawbacks to the *SIM tool*. He notes that the clips used in *AO* are a few seconds long, and that they are only able to transition at certain points. This can cause musical transitions to occur well after the associated gameplay state change. The *SIM tool* can cross-fade between clips during playback based on chord progression, though this was not implemented in the game. Another drawback to this tool is that the composer must provide the annotations for each clip by hand. If a clip is added, other clips that may transition to the new clip must be updated to include the new clip. If a clip is removed, clips that do transition to the removed clip must also be updated by hand.

Tom Clancy's EndWar (2008) [86] *Tom Clancy's EndWar* ⑭ uses a common mixed arrangement system design, one that is similar to the more recent systems used in *Red Dead Redemption* ⑯ and *No Man's Sky* ㉖. The music in *Endwar* is divided into individual short musical phrases, described internally as "cells". Each cell is placed into a corpus, described internally as a "pool". Within each corpus, any phrase may transition to any

other phrase upon completion. The system may also adjust the density of the music by adding a constrained random pause between each phrase. This pause can be individually set for each corpus during runtime.

The system generatively arranges music horizontally by adding or removing individual corpora from the total mix, based on game state and the musical context. When a layer is removed, the system allows the current musical phrase finish playing within that layer, rather than cutting off the music. The corpora that are playing at any point are determined by the game state, which allows the system to adaptively alter the arrangement of the music.

Designer Ben Houge describes several drawbacks and constraints of this system. Houge notes that the technology for reading music off of a DVD directly was too slow in 2008 to have multiple corpora of short phrases seamlessly transition and play. This required the system to load the music into RAM, which is normally allocated to game levels and textures. Also, Houge describes the workflow of the system, which involved numerous iterations and mockups to create music for the system, as well as what Houge calls "significant time" tweaking parameters in game context [33].

Red Dead Redemption (2010) [64] The system in *Red Dead Redemption (RDR)* (16) presents the most extreme form of systemic flexibility at the expense of restricting the musical possibility space. While *RDR* uses a rule-based algorithm, its architecture can also be understood as a Markov chain in which any state can lead to any other state. Music in *RDR* is divided vertically by instruments, and each instrument has an associated function e.g. "Melody" or "Bass". Within any game state, the system creates music by randomly selecting one musical phrase from each function's associated corpus. When the game state changes, the system crossfades to a new randomly selected group of phrases from the associated corpora.

This design presents an extremely flexible system, as there are no restrictions on how the corpora can be combined. However, this design presents severe restrictions on the composed library. All of the music in *RDR* is composed in the key of a minor, at a tempo of 130 beats per minute (bpm). The music does not contain extended harmonies or modal borrowing. In short, because the system can combine any collection of musical phrases together, all music composed for the system must combine well with all other music composed for the system. This severely limits the expressive range of the system.

No Man's Sky (2016) [31] *No Man's Sky* (26) uses multiple similar generative systems to create multifaceted generative audio. The systems in *No Man's Sky* address the tasks of world audio playback similar to *Sonancia* (25), the generation of alien-sounding speech, and the generative arrangement of music, though we narrow our focus to the music generation system.

The corpus for *No Man's Sky* is a composed score. The band "65daysofstatic" composed a linear score for the game absent the generative system. The composed music is divided into small clips of music for the generative system, and each musical piece acts as a single corpus. Because the elements of each corpus are sourced from composed music, the composed clips can be assumed to fit together musically. This does require the music to be composed without key or tempo changes within each piece, but it also reduces the need for hand annotations of music.

No Man's Sky's generative system differentiates music horizontally based on five associated game states: "Wanted", "Space", "Planet", "Map", and "AmbientMode". Each game state has associated musical pieces, which are pseudorandomly assigned to play together. Details of the rules and restrictions that govern the combinations of musical elements is unavailable. The system arranges music vertically based on a random procedural playback of each location in-game. The instrumentation depends both on the location of the player and where the player is looking [92].

Adaptive Music System/AMS (2019) [36] *The Adaptive Music System (AMS)* (34) has similar design elements to other academic generative systems, such as *MetaCompose* (30) and *The Audience of the Singular* (29). AMS alters pre-composed music based off of an affective mapping, that is taken from the game state. It does this using a combination of rule-based algorithms, genetic algorithms, and a Recurrent Neural Network (RNN). An RNN is a class of Neural Network that maintains past information while receiving new information.

AMS extends a categorical model of affect from music perception literature, with 6 affect categories: happiness, fear, anger, tenderness, sadness, and excitement. To connect the game state to the affective data, a model of "spreading activation" is used. This model represents affect and game concepts as weighted verticies in a 2-dimensional plane. When a vertex activates, it also activates nearby vertices. In *AMS*, because game concepts and affects are both on the same plane, when a game event activates, it will also activate the nearby affective vertices, which influence the music generation.

To generate music, *AMS* uses a multi-agent approach with three agent roles. The first agent role is the "harmony" role, which builds a chord progression using an RNN algorithm trained on a symbolic corpus. This agent does not generate notes, but as in *The Audience of the Singular* (29), the harmony information is used to constrain the output of multiple melody agents. The melody agents use a rule-based approach to alter pre-composed musical pieces. The rules are created offline using a supervised genetic algorithm, in this case trained by a single expert composer. Finally, *AMS* creates a percussive line with a single agent that uses a similar RNN approach to the harmony agent.

One element that sets *AMS* apart from other academic systems is that while it is generic, it has been evaluated using real-world games. As part of an evaluative study, *AMS* was

integrated into an open-source Zelda clone titled *Zelda: Mystery of Solarus* as well as the Real-time strategy game *StarCraft II*. A correlational analysis of the games with *AMS* and with their original score found that *AMS* significantly, if slightly, increases player immersion.

2.7.3 Performance systems

While composition and arrangement systems automate the creation of new music, performance systems automate the interpretation and playback of music.

Otocky (1987) [84] *Otocky* ② is one of the first examples of a game using generative music system. The gameplay of *Otocky* is a horizontal "shoot-em-up" or "shmup", similar to the *Gradius* games. As the player progresses, they collect various upgrades, each of which shares a name with an associated synthetic instrument. When the player fires their weapons, the associated instrument plays alongside the composed linear soundtrack, with pitch determined by the soundtrack and rhythm quantized to the nearest eighth note.

Rez (2001) [87] and Child of Eden (2011) [20] *Rez* ⑧ and it's prequel *Child of Eden* ⑲, have nearly identical gameplay, and use identical musical systems. While these systems are almost identical in design, they are not generic systems, but are nearly identical game-specific systems.

The systems in *Rez* and *Child of Eden* are very similar to the system used in *Otocky* ②. The gameplay of *Rez* and *Child of Eden* are third-person, 3d shmups that are on-rails (the player does not control the motion of their avatar). The player controls the location of a reticle on screen, and when they move the reticle over an enemy, they lock on to the enemy. When the player presses a button, their avatar fires its missile-like weapons, which automatically track and hit the locked-on enemies. When the missiles hit the enemies, a pitched cluster of notes is played. The exact timing of the notes is quantized to be on grid, and the pitches are based on the surrounding musical context provided by the composed linear score.

Bit.Trip Runner (2011) [26] *Bit.Trip Runner* ⑳ uses a system that is nearly identical to the system in *Rez* ⑧ and *Child of Eden* ⑲, though it is simplified. A difference in *Bit.Trip Runner*'s system is that, similar to *Otocky* ②, the system provides adaptive music.

Bit.Trip Runner ⑳ is an infinite runner game, where the player character moves at a constant speed, and the player must take action to avoid oncoming obstacles and to pick up power-ups. When the player jumps or slides, a note is randomly selected from a pentatonic scale that matches the surrounding musical context. This note plays at the next available beat. The instrument that plays the note is directly related to the number of power-ups that the player has collected.

Chuchel (2018) [56] *Chuchel* (32) is unique among the surveyed systems because the diagetic, sourced music that system performs also functions as sound effects and even as character voices. When the player interacts with objects in *Chuchel*, the object almost always produces a sound. There may also be an auditory reaction from the player character. In many cases, these sounds are pitched, and the pitch of the sound is determined by the surrounding musical context. In *Chuchel*, unlike *Rez* ⑧, *Child of Eden* (19), or *Otocky* ②, this system has a low degree of player control over the music. While actions in the previously mentioned performance systems always respond musically, in *Chuchel* the musical nature of sounds is unpredictable and inconsistent.

Ape Out (2010) [16] The music system in *Ape Out* (33) is unique in our surveyed systems as the game soundtrack and the musical output of the system contain no pitched music at all. Instead, the music for *Ape Out* is provided exclusively by a virtual drummer. The player can interact with the game world in only two ways - grabbing a non-player character/object, and throwing the NPC/Object. When the player throws an NPC into a surface, the NPC explodes in gratuitous violence. During gameplay, there is a constant drum groove that plays. When an NPC is killed, there is also a cymbal accent.

The drum groove is selected based on the current gameplay level, the amount of on-screen action, and random chance. The generative system contains a library of 1,000 short drum grooves and cymbal hits. Each level of the game has an associated library of drum groves. The drum grooves are also delineated by what the developers describe as "level of action". As the action in *Ape Out* becomes more intense, more active drum grooves play. The cymbal accents are chosen randomly from a library of cymbal hits.

2.7.4 Fringe Systems

While the previously examined systems have well-defined generative tasks, we have also identified seven systems that do not fit as cleanly into our taxonomy. These systems can be described using our taxonomy, though either they fill multiple roles, or they approach music in a unique way that is not fully captured by the taxonomy. We describe these systems as fringe systems as they exist on the fringes of our taxonomy.

Agate/AGMS (2008) [30] *Agate* (12), also called *AGMS*, is a system that simultaneously addresses the composition and arrangement tasks, with both a horizontal note granularity and a vertical instrument granularity. The system has not been integrated into a game and therefore cannot be described with gameplay dimensions, though it does play linear music. *Agate* is designed as a generic system that uses a rule-based algorithm. *Agate*'s musical representation is both symbolic and audio, though the knowledge source is exclusively external.

Agate organizes its music in libraries and rule sets that are associated with "moods". *Agate's* moods are provided by an external source, and are attached to a game state, rather than being based in a more general representation of affect. *Agate* combines two simultaneous rule-based algorithms to address both the arrangement and composition task. *Agate* generatively addresses the composition task with a set of rule-based constraints on otherwise random generation. The designer can select a collection of notes that may be used, the level of randomness, the beats that notes may play on, the possible durations of notes, and the available instruments. *Agate* then creates ambient soundscape music by randomly selecting pitches.

For the arrangement task, *Agate* also uses constrained random generation. The composer or designer provide either symbolic or audio representations of short musical phrases. The designer also sets parameters that constrain the activity level of the samples. *Agate* plays these phrases at random times over the ambient soundscape that is generated, constrained by the activity level.

Sonancia (2015) [43] As with *No Man's Sky* (26), *Sonancia* (25) uses multiple different generative algorithms to generate a level, populate the level, and add audio triggers to locations in the level. We focus our examination primarily on the music generation aspects of *Sonancia* (25).

Sonancia is a game that generates a horror-game level to match a provided or machine-generated tension curve. Once the level is generated, the musical system distributes musical cues throughout the game environment to match the generated level's tension curve. Each musical cue is annotated along Schimmack and Grob's 3-dimensional model of affect [71]. The musical cues are distributed into each room in the generated level via the previously mentioned rule-based algorithm. The selection method for the music can be chosen by the designer from one of four provided methods: "Hall of Fame", which selects the top n pieces that match along a single emotional dimension, "Equidistant", which selects n pieces based on their ranking within the chosen emotional dimension, "Granular", which selects the closest emotional match to the generated room, and "Random", which selects randomly.

A core difference of *Sonancia*'s generative system, compared with other systems that require musical annotation, is that annotation for *Sonancia* does not need to be provided by an external source. *Sonancia*'s initial corpus of music is annotated via crowdsourced ranking. Support Vector Machines are then trained on the user provided annotations and a selection of audio features. The SVMs are then used to annotate new musical files based on the same selection of audio features. This use of SVMs in the prediction task allows for the automatic annotation of music files, reducing the labour of the standard arrangement-oriented pipeline.

MetaCompose (2017) [73] *MetaCompose* (30) uses two different systems to address both the composition and performance tasks. The component sub-systems of *MetaCompose* differ not only in the task that they address, but also the generative algorithm that they use.

The first system uses a combination of a stochastic algorithm, a genetic algorithm, and a rule-based algorithm. The first system begins its generation by generating a chord progression using a random walk on a directed graph of possible chord transitions. This creates chord progressions which over time resolve to the I chord. Once a progression is created, a genetic algorithm evolves a melodic line. This genetic algorithm uses a fitness function that is provided externally. The fitness function's design mirrors species counterpoint, with restrictions on large leaps, and tendencies towards chord notes during leaps and after chord changes. Finally, this composition system creates a framework for accompaniment. The accompaniment system uses two rule-based components to create a rhythm and arpeggio for accompaniment. This system uses Euclidian rhythm to create even and repeating divisions of time, and selects from pre-composed arpeggios to play the chords through time.

The second system in *MetaCompose* targets the performance task at the instrument and note levels of granularity, using a rule-based system. This final performance system takes the previously generated music, and alters and synthesizes the music according to provided valence and arousal values. The system directly links arousal to volume, and valence to brightness of timbre. Additionally, this system chooses an accompanying drum line to accompany the music. This drum line is more prominent and faster as arousal increases, and more regular and steady as valence increases. Finally, this system alters notes of the provided composed music, adding dissonant tones from alternative musical modes as valence decreases.

Melodrive (2018) [89] *Melodrive* (31) has limited information available, as it is an in-development system from the games industry. Because *Melodrive* is a generic system, it cannot be discussed along gameplay dimensions, though it is intended to provide adaptive music. *Melodrive* uses a symbolic representation of music with an external knowledge source, though we cannot determine the algorithm for *Melodrive*. *Melodrive* is differentiated from the other surveyed generic music systems by being integrated into the *Unity* game engine. This means that *Melodrive* can be integrated into any game built using the *Unity* engine without requiring large amounts of labour to port the system to a new engine or project.

Melodrive (31) is available as a Unity plugin, and presents a simple interface for designers. To generate original music, the *Melodrive* script must be given a style and emotion. Both of these options are categorical, with the set of possibilities determined by *Melodrive*. *Melodrive* can also interpolate between different emotions, and Russell's 2-dimensional model of emotion [66] may also be used. *Melodrive* creates fully-featured music without requiring large amounts of additional labour or musical restrictions. However, *Melodrive* also does not

offer the customization possibilities in more open-ended generic systems such as *DirectMusic* ④

2.8 Tools for adaptive and generative music

The two most popular publicly available video game engines in the game industry are the *Unreal* engine and the *Unity* engine. Another less popular publicly available game engine is Amazon's *Lumberyard*, based on Crytek's *Cryengine*. Large game companies often use an internally-developed game engine to create their games, such as Electronic Arts' *Frostbite*, Ubisoft's *AnvilNext*, Square-Enix's *Luminous* engine, Bethesda's *Creation* engine, and Rockstar Game's *Rockstar Advanced Game Engine*. The engines that are in use by large companies generally do not publish their specifications. In both *Unity* and *Unreal*, each audio asset is attached to at least one object in the game. To trigger an audio asset, the object must call the audio asset from code. This design does not easily allow for generative music, as the number of managed audio assets would easily become unfeasible. *Lumberyard*, as far as we are aware, does not have any audio rendering capabilities built in. As far as we are aware, most internal industry game engines have similar audio capabilities to *Unreal* and *Unity*. External audio solutions and tools can be used to facilitate generative music in games.

2.8.1 Audio Middleware

Audio Middleware engines provide a solid base for interactive and adaptive audio. Middleware can be used to fill any of the musical and gameplay dimensions of our taxonomy. However, middleware engines are limited in architecture dimensions - currently they are only capable of creating systems specific to a game, that use a rule-based algorithm with an external knowledge source, and audio representation of music.

Audio middleware engines seamlessly integrate into game engines, and facilitate adaptive music and audio. The two most popular middleware engines are Firelight Technologies' *FMOD Studio* [24], and Audiokinetic's *Wave Works Interactive Sound Engine (Wwise)* [3]. Both middleware engines share similar functionality.

FMOD and *Wwise* act like traditional DAWs for audio editing, but with additional controllable parameters. These additional parameters can be any numeric or boolean game data, which is passed via an API call. These parameters can be used to add, remove, or alter audio effects, including volume, DSP effects, and spacialization. Middleware can also loop sections of music until game parameters are changed.

FMOD and *Wwise* can both also use indeterminacy for selecting clips. A standard implementation of this feature in non-musical audio is to provide variety in commonly-heard sounds. A single footstep that is repeated every time the PC takes a step will get grating. By creating a corpus of possible footstep sounds and randomly selecting one each

time the PC takes a step, the resulting sounds are less repetitive and more believable. A musical example of this can be seen in *Tom Clancy's Endwar* (14), where the system shuffles musical phrases based on randomness, taken from a corpus of possibilities.

These capabilities, when used together, facilitate generative systems that address the arrangement task. Generative arrangement systems that use middleware generally require the same musical restrictions as seen in many of the surveyed systems: Any layers that intend to play together must be composed at identical tempi and keys as each other. Additionally, any clips that may randomly play alongside each other must fit together musically.

2.8.2 iMuse

iMuse ③ has been discussed previously due to the consistent use of the system in games. We now describe *iMuse* as a tool that can facilitate generative music. *iMuse* is most easily used for horizontal arrangement on-grid, and can fill any gameplay dimensions. *iMuse* can only create music from external sources, with symbolic representation of audio, using a rule-based algorithm. While *iMuse* could theoretically be used for vertical arrangement, but this would require considerable work and the software is not designed for vertical arrangement. *iMuse* could also theoretically create adaptive music, but it's design is most suited to linear music.

2.8.3 DirectMusic

As with *iMuse* ③, *DirectMusic* ④ also functions as both a system in games and a tool to facilitate generative music. *DirectMusic* extends the *iMuse* possibilities, and can be used to address the arrangement task in both horizontal and vertical directionalities. *DirectMusic* is most commonly used on grid, but this is not a necessary part of its design. Systems that use *DirectMusic* can fill any gameplay dimensions, though must use a rule-based algorithm with an external knowledge source. *DirectMusic* can use both symbolic and audio representation of music.

2.8.4 PureData

Miller Puckette's PureData (PD) can be used as a generative music tool in games, in a limited capacity. PD is a visual programming language that targets real-time audio generation. PD can be used in a system that fills any musical, gameplay, and architecture dimensions of the taxonomy. Electronic Arts used a modified version of PD called "EAPD" for the generative soundtrack in *Spore* [94]. PD was also used to synthesize the music of *The Audience of the Singular* (29). Unfortunately, despite the potential strengths of PD, there is no official support for PD implementation in any game engine that we are aware of, and external libraries are often outdated and unstable.

2.8.5 Other languages

Real-time coded generation of music, such as *csound* and *Max 8*, can also be used to synthesize and perform generated music. *Chunity* [2] and *uRTcmix* [29] are examples of real-time audio/music languages that can be used as plugins for *Unity*. These plugins allow for easy use of audio generation functions for audio playback and synthesis of primitive waves. We have found no instances of these languages being used for generative music in games, though as with the other synthesizers, these languages could be used to synthesize music from any symbolic representation, or to sequence any audio representation of music.

2.8.6 Custom Synthesizers

Both the *Unreal* engine [22] and *Unity* [88] allow programmers to access the audio data directly. This allows a designer to programmatically build synthesizers directly into the engine. These synthesizers may then interpret symbolic music data into musical sounds. Because these synthesizers act only as a playback device, a system that uses custom synthesizers may fill any dimension of the taxonomy. Perhaps because of the complexity of programming synthesizers from scratch, we find no examples of this being used in either the academic research or industry use. The *Unreal* engine contains very basic waveform synthesizer blueprints as samples, though we are not aware of any industrial or research game that uses these synthesizers.

2.8.7 Open Sound Control

Open Sound Control (OSC) is a protocol for computer communication. OSC can also be used locally, sending data from different programs on the same machine. Because of this, OSC can allow for programs such as Ableton Live or Max/MSP to provide audio for games. As before, because OSC is primarily a tool for communication, rather than containing any algorithm itself, a system that uses OSC may fill any role in our taxonomy. *Audioverdrive* (22) uses OSC to allow for interaction between level generation and music. We are not aware of any examples of commercial use of OSC. Most likely, this is because OSC cannot be integrated seamlessly as middleware engines can. During gameplay, a system using OSC must still be running an external program as well. Games generally run from a single executable, and players are not expected to follow long setup processes to play a game. This may be why OSC is not used in the games industry.

2.9 Discussion

2.9.1 Analysis of trends

We identified 34 systems that fit the scope of generative music in games. We acknowledge that this list may be incomplete, as public information on industrial games is limited. We

also draw attention to our narrow scope of generative music in games. We do not discuss games with highly adaptive non-generative music, such as *Final Fantasy XV* [76]. We also do not discuss games where user-selected music is used to procedurally generate a game level, such as in *Audiosurf* or *Beat Hazard*. Finally, we do not discuss the "music game" genre of games, including games like *Fract OSC* or *Rock Band*. While these games all allow for interaction with music, our scope is limited to games that use a generative music system with some level of systemic autonomy.

The systems from the games industry have many commonalities with each other. These systems generally address the arrangement task on grid. They provide non-diagetic, ambient music. These systems generally are specific to their game, and use a rule-based algorithm with an external knowledge source and audio representation of music. We believe that there are several reasons for these common trends. As discussed, audio middleware is very common in the games industry, and are capable of a rule-based algorithm with external knowledge source and audio representation. Most game music is non-diagetic and ambient, and the source of the generation does not necessarily have an impact on the gameplay dimensions. Finally, creating on-grid arrangement systems most closely matches the workflow of using composed music in games.

The systems from academic research generally address the composition task with mixed directionality. These systems also primarily provide non-diagetic ambient music, and generally are adaptive. Academic systems are far more varied than industry systems in the architecture dimensions, with systems using stochastic, rule-based, and genetic algorithms, with both symbolic and audio representation, and both learned and external knowledge sources. While this does indicate that academic systems are more technologically advanced, it is important to recognize that many of these systems have not been integrated into or evaluated within actual gameplay. The academic systems also do not target a commercial release, which means that they can produce music that does not sound as "good" as a human composer, without affecting commercial success.

Generative music systems for games have trended towards audio representation of music, especially in the industrial applications. This is most likely due to an assumed dislike of MIDI sounds in the audience [39], and the higher fidelity and quality of audio representation. However, we do note that award-winning games such as *Shovel Knight*, *Celeste*, and *Luftrausers* make heavy use of synthetic instruments.

2.9.2 Conclusion and suggestions for future work

Generative music for games is becoming increasingly commonplace, and is advancing quickly. Of our 34 surveyed systems, 19 of them are from the 10 years prior to this writing, while the remaining 15 are from the preceeding 26 years. However, there is still much room for advancement in the area. Current systems tend to fall into two main categories: Simple and

effective systems, which are more common in industrial applications of generative music, and more advanced but untested systems, which are more common in academia.

We believe that the future of games music will involve increasing use of generative techniques. Generative systems can provide a greater variety and adaptivity to music than is possible with composed music, and with less required labour. Generative systems can also create endless amounts of music, which is well suited to longer-duration games. Generative systems can also provide large amounts of variety, which is particularly useful in run-based games.

There is a valid concern that generative music may cause harm to video game composers by rendering their work unnecessary. We note that this concern is not reflected in the current implementations of generative music, which require either a library of curated or provided music, or musical expertise in the design of the system. Current AI techniques for broader, non-game generative music also often require corpora of composed music to be effective. We believe that human-composed music is still capable of greater expression than computer generated music, especially when the game activity is predictable, and suggest that future work in this area continue to leverage the strengths of both human-composed and generative music together.

As we have discussed, generative music systems from the game industry generally address the arrangement task. We have identified two main weaknesses in the current state of the art for these systems. These weaknesses are linked together, and any attempt to rectify one weakness within current paradigms exacerbates the other. The first weakness is that the generative music systems are often highly restricted in expressive range. *Red Dead Redemption* (16) demonstrates this weakness - while the system is capable of producing huge amounts of music due to the large corpus that it draws from, the system is incapable of producing music that is not at a tempo of 130 beats per minute, or producing music that is in any key other than a strict diatonic a minor.

The other weakness of industrial systems is that the current architecture requires large investments of labour. *Anarchy Online* (9) demonstrates this weakness. In *AO*, each piece of music needs to be annotated with transitions in and out. Additionally, changes to any music cue requires all other cues that transition to the altered cue. This requires far more labour than composed music does, as composed music can be directly assigned to states, assuring that horizontal transitions and vertical layers will smoothly transition to each other. These weaknesses are exacerbated by each other - any attempt to increase the expressive range of an arrangement system will require increased amounts of musical variety, which will increase the amount of metadata required to ensure that the resulting music does not clash with itself. Any attempt to reduce the labour cost of these systems will restrict the composed musical library, which reduces the expressive range of the system.

Academic systems in contrast tend towards addressing the composition task. This removes some of the inherent weaknesses of arrangement systems, but these systems also have

shared weaknesses. The biggest weakness of generative composition systems in the current state of the art is that they are limited by their isolation from the larger game industry context. This isolation often results in music composition systems that could theoretically provide music for a game, but do not engage with the interaction that differentiates games from linear media.

It is clear that generative, adaptive music systems have the potential to provide not only greater variety of game music, but also more compelling and powerful music. The academic evaluation of adaptive and generative systems demonstrates support for this - adaptive music has a greater effect on a player's subjective experienced emotion [59] than linear music, and the generative systems that have been evaluated also demonstrate that generative systems have a similar effect, and can cause objective affective responses as well [60, 73].

There is interest in generative music in the games industry, but it is thus far akin to dipping a single toe into the pool of possibilities. We suggest continued cooperation between academia and the games industry, with the intent of developing systems that can address more generative tasks with more expressive range, and that can allow composers to focus on crafting musical worlds, rather than on the data-entry labour required by many current industrial systems. We believe this cooperation will also lead to these systems having access to more evaluative and design resources to smooth out the rough edges that are common in current academic systems. Ultimately, we believe that future cooperation between academia and industry in the field of generative music for games will lead to better games with better music.

Bibliography

[1] Airtight Games, Jose Perez III, and Jason Lamparty. (Game) Dark Void, 2010.

[2] Jack Atherton and Ge Wang. Chunity: Integrated audiovisual programming in unity. *New Interfaces for Musical Expression*, 2018.

[3] Audiokinetic. (Software) Wwise, 2017.

[4] Luciano Berio. *Two interviews*. New York : M. Boyars, 1985.

[5] John A. Biles. GenJam: A genetic algorithm for generating jazz solos. *Proceedings of the International Computer Music Conference*, pages 131–137, 1994.

[6] BioWare, Black Isle Studios, and James Ohlen. (Game) Baldur's Gate, 1998.

[7] Canadian League of Composers. Commissioning Rates. URL: `https://www.composition.org/commissioning-rates/`, 2015. Last accessed: 04-25-2022.

[8] Capcom and Akira Kitamura. (Game) Mega Man, 1987.

[9] Tim Challies. "No Man's Sky" and 10,000 Bowls of Plain Oatmeal. URL: `https://www.challies.com/articles/no-mans-sky-and-10000-bowls-of-plain-oatmeal/`, Oct 2016. Last accessed: 04-25-2022.

[10] Karen Collins. An introduction to procedural music in video games. *Contemporary Music Review*, 28(1):5–15, 2009.

[11] Karen Collins. *From Pac-Man to Pop Music Interactive Audio in Games and New Media*. Farnham : Ashgate Publishing Ltd, Farnham, 2011.

[12] Karen Collins. *Playing with sound: a theory of interacting with sound and music in video games*. The MIT Press, Cambridge, MA, 2013.

[13] Darrell Conklin and Ian H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[14] Chris Crawford. *Chris Crawford on game design*. New Riders, 2003.

[15] Jason Cullimore, Howard Hamilton, and David Gerhard. Directed transitional composition for gaming and adaptive music using q-learning. In *ICMC*, pages 332–338, 2014.

[16] Gabe Cuzzillo. (Game) Ape Out, 2019.

[17] Die Gute Fabrik. (Game) Sportsfriends, 2014.

[18] Tuomas Eerola and Jonna K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, jan 2011.

[19] Steve Engels, Fabian Chan, and Tiffany Tong. Automatic real-time music generation for games. In *11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 220–222, Palo Alto, CA, 2015. AAAI Press.

[20] Q Entertainment and Tetsuya Mizuguchi. (Game) Child of Eden, 2011.

[21] Entertainment Software Association. 2019 essential facts about the computer and video game industry. URL: `https://www.theesa.com/resource/essential-facts-about-the-computer-and-video-game-industry-2019/`, 2019. Last accessed: 04-25-2022.

[22] Epic Games and Tim Sweeney. (Game Engine) Unreal Engine. URL: `https://www.unrealengine.com/`, 1998. Last accessed: 04-25-2022.

[23] Firaxis Games and Jake Solomon. (Game) XCOM 2, 2016.

[24] FMOD. (Software) FMOD Studio, 2016.

[25] Funcom. (Game) Anarchy Online. URL: `https://www.anarchy-online.com/`, 2001. Last accessed: 04-25-2022.

[26] Gaijin Games and Alex Neuse. (Game) Bit.Trip Runner, 2011.

[27] Philip Galanter. What is Generative Art? Complexity theory as a context for art theory. *In GA2003–6th Generative Art Conference*, 2003.

[28] Gamelab and Nicholas Fortugno. (Game) Diner Dash, 2004.

[29] Brad Garton. (Software) RTcmix. URL: `http://rtcmix.org/`, 2019. Last accessed: 04-25-2022.

[30] Jim Hedges, Kurt Larson, and Christopher Mayer. An Adaptive, Generative Music System for Games. URL: `https://www.gdcvault.com/play/1012710/An-Adaptive-Generative-Music-System`, 2010. Last accessed: 04-25-2022.

[31] Hello Games. (Game) No Man's Sky, 2016.

[32] Pierre Hoegi. *A Tabular System: Whereby the Art of Composing Minuets is Made So Easy that Any Person, Without the Least Knowledge of Musick, May Compose Ten Thousand, All Different, and in the Most Pleasing and Correct Manner. Invented by Sigr. Piere Hoegi.* Printed at Welcjer's musick shop, 1763.

[33] Ben Houge. Cell-based music organization in Tom Clancy's EndWar, 2012.

[34] Craig Hubbard, Kevin Stephens, Wes Saulsberry, Guy Whitemore, Chris Miller, Samantha Ryan, and Monolith Interactive. (Game) The Operative: No One Lives Forever, 2000.

[35] Johan Huizinga. *Homo Ludens: A Study of the Play-Element in Culture.* Beacon Press, 1971.

[36] Patrick Hutchings and Jon McCormack. Adaptive music composition for games. *IEEE Transactions on Games*, 2019.

[37] Kent Jolly and Aaron McLeran. Procedural Music in Spore. URL: `https://www.gdcvault.com/play/323/Procedural-Music-in`, 2008. Last accessed: 04-25-2022.

[38] Konami and Hitoshi Akamatsu. (Game) Castlevania, 1986.

[39] Grace Kramer and Derek Alexander. (Video) Why the Music in Dragon Quest XI is so Terrible. URL: `https://www.youtube.com/watch?time{_}continue=200{&}v=xfdFU3O3nf8`, 2018. Last accessed: 04-25-2022.

[40] Bjorn Arve Lagim. The Music of Anarchy Online: Creating Music for MMOGs. URL: `https://www.gamasutra.com/view/feature/131361/the_music_of_anarchy_online_.php`, Sep 2002. Last accessed: 04-25-2022.

[41] Peter S. Langston. Six Techniques for Algorithmic Music Composition. *International Computer Music Conference. San Francisco: Computer Music Conference Association*, pages 164–167., 2005.

[42] David Levine, Peter Langston, David Riordan, and Garry Hare. (Game) Ballblazer, 1984.

[43] Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis. Sonancia: Sonification of Procedurally Generated Game Levels. *Proceedings of the 1st Computational Creativity and Games Workshop.*, 2015.

[44] Phillip Magnuson. Basic rules for species counterpoint. URL: `http://academic.udayton.edu/PhillipMagnuson/soundpatterns/speciesctpt/`, 2008. Last accessed: 04-25-2022.

[45] Microsoft. (Game) Halo 2. Game, November 2004.

[46] Microsoft. (Software) DirectMusic, Apr 2009.

[47] Eduardo R. Miranda and Duncan Williams. Artificial intelligence in organised sound. *Organised Sound*, 20(1):76–81, 2015.

[48] Wolfgang Amadeus Mozart. Musikalisches Würfelspiel, 1787.

[49] Naughty Dog, Bruce Straley, and Hennig Amy. (Game) Uncharted 2: Among Thieves, 2009.

[50] Ninja Theory and Tameem Antoniades. (Game) Hellblade: Senua's Sacrifice, 2017.

[51] Nintendo, Yoichi Yamada, Eiji Aonuma, and Yoshiaki Koizumi. (Game) The Legend of Zelda: Ocarina of Time, November 1998.

[52] Nintendo Creative Department and Shigeru Miyamoto. (Game) Super Mario Bros., 1985.

[53] Philippe Pasquier. (Kadenze Class) Generative Art and Computational Creativity. URL: `https://www.kadenze.com/courses/generative-art-and-computational-creativity-i`. Last accessed: 04-25-2022.

[54] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to Musical Metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[55] Phosfiend Systems. (Game) Fract OSC, April 2014.

[56] Jaromir Plachy and Amanita Design. (Game) Chuchel, 2018.

[57] 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.

[58] Cale Plut. (Game) The Audience of the Singular, 2017.

[59] Cale Plut and Philippe Pasquier. Music matters: An empirical study on the effects of adaptive music on experienced and perceived player affect, 2019.

[60] Anthony Prechtl. *Adaptive music generation for computer games.* PhD thesis, Open University (United Kingdom), 2016.

[61] Lucas Reycevick. (Video) The Brilliance of DOOM's Soundtrack. URL: `https://www.youtube.com/watch?v=7X3LbZAxRPE`, 2016. Last accessed: 04-25-2022.

[62] Steve Ritchie, Ed Boon, Doug Watson, and Joe Joos Jr. (Game) Black Knight 2000, April 1989.

[63] Judy Robertson, Andrew de Quincey, Tom Stapleford, and Geraint Wiggins. Real-time music generation for a virtual environment. In *Proceedings of ECAI-98 Workshop on AI/Alife and Entertainment*. Citeseer, 1998.

[64] Rockstar Games. (Game) Red Dead Redemption. Game, May 2010.

[65] Rocksteady Sudios and Sefton Hill. (Game) Batman: Arkham Asylum, 2009.

[66] James A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[67] Stuart J. Russell. *Artificial Intelligence: A modern approach*. Pearson series in artificial intelligence. Pearson, fourth edition. edition, 2021.

[68] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA, 2004.

[69] Josh Sawyer, Bobby Null, and Eric Fenstermaker. (Game) Pillars of Eternity, 2015.

[70] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345, 2000.

[71] Ulrich Schimmack and Rainer Reisenzein. Experiencing activation: Energetic arousal and tense arousal are not mixtures of valence and activation. *Emotion*, 2(4):412–417, 2002.

[72] Marco Scirea. *Affective Music Generation and its effect on player experience*. PhD thesis, IT University of Copenhagen, 2017.

[73] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines*, 18(4):433–465, 2017.

[74] Peter Silk. (Video) iMUSE Demonstration 2 - Seamless Transitions. URL: `https://bit.ly/1R39FPY`, May 2010. Last accessed: 04-25-2022.

[75] Bethesda Softworks. (Game) Fallout 3, October 2008.

[76] Square Enix and Hajima Tabata. (Game) Final Fantasy XV, 2016.

[77] Square Enix and Motomu Toriyama. (Game) Final Fantasy XIII, 2009.

[78] Marty Stratton, Hugo Martin, Timothy Bell, Jason O'Connell, Billy Ethan Khan, Hugo Martin, Adam Gascoine, Mick Gordon, and id Software. (Game) DOOM (2016), 2016.

[79] Igor Stravinsky. *Poetics of music in the form of six lessons*. Harvard University Press, 1970.

[80] Supergiant Games. (Game) Pyre, 2017.

[81] Michael Sweet. *Writing interactive music for video games: A composer's guide*. Addison-Wesley, Upper Saddle River, NJ, 2015.

[82] Kıvanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, 48(1):56–105, 2019.

[83] Richard van Tol and Sander Huiberts. IEZA: A Framework For Game Audio. URL: `https://www.gamasutra.com/view/feature/3509/ieza_a_framework_for_game_audio`, Jan 2008. Last accessed: 04-25-2022.

[84] Iwai Toshio. (Game) Otocky, 1987.

[85] Than van Nispen tot Pannerden, Sander Huiberts, Sebastiaan Donders, and Stan Koch. The NLN-Player: A system for nonlinear music in games. In *ICMC*, 2011.

[86] Ubisoft Shanghai and Michael de Plater. (Game) Tom Clancy's EndWar, 2008.

[87] United Game Artists, Jun Kobayashi, Tetsuya Mizuguchi, Hiroyuki Abe, Katsuhiko Yamada, and Katsumi Yokata. (Game) Rez, 2001.

[88] Unity3d. (Game Engine) Unity. URL: `https://unity.com/`, 2019. Last accessed: 04-25-2022.

[89] Valerio Velardo. Melodrive: Adaptive music generation. URL: `https://melodrive.com/index.php`, 2018. Last accessed: 01-02-2019.

[90] Sean Velasco and Yacht Club Games. (Game) Shovel Knight, 2014.

[91] Vlambeer. (Game) Luftrausers, 2014.

[92] Paul Weir. (Video) The Sound of "No Man's Sky". URL: `https://bit.ly/2UKShXS`, 2017. Last accessed: 04-25-2022.

[93] Kristina Winbladh, Hadar Ziv, and Debra J. Richardson. (Software) iMuse. *ACM SIGSOFT*, page 383, 2010.

[94] Will Wright. (Game) Spore, 2008.

[95] Georgios N. Yannakakis and Julian Togelius. *Artificial intelligence and games*, volume 2. Springer, 2018.

Chapter 3

Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect

As published in Plut, C., & Pasquier, P. (2019, August). *Music Matters: An empirical study on the effects of adaptive music on experienced and perceived player affect.* In 2019 IEEE Conference on Games (CoG) (pp. 1-8). IEEE.

Abstract

Music is an important affective aspect of video games. We present the findings of an empirical study on the affective effects of adaptive uses of music in games. We find that adaptive music can significantly increase a players reported experienced feeling of tension, that players recognize and value music, and that player recognize and value adaptive music over linear music.

3.1 Introduction and Motivation

3.1.1 Music for video games and other media

Music is an integral part of video games, and almost every video game has music [46]. Most music for games is linear, and is not affected by the actions of the game [32]. Adaptive music is music that changes based on the state of the game, and has many theoretical benefits [39]. In film, music that more closely aligns with the actions of a movie significantly

increase the viewer's emotional response and enjoyment of the media [19]. This phenomenon is previously assumed to exist for video game music as well [39].

Despite the potential advantages of adaptive music, most game music plays linear music during an associated level or game state [32]. One reason that adaptive music is not used across all games is that it entails a higher production cost and can reduce the expressive range of the music [44]. Another reason for the lack of adoption of adaptive music in the industry may be that it has uncertain benefit without empirical support.

The research that targets music in games is almost all concerned primarily with immersion [46], player performance [6], or other non-affective measures [5]. Research into affective effects of music in games is often overly broad or narrow. Previous studies have only manipulated the presence of music, not the content [15]. Studies done in Virtual Reality have not also investigated non-VR interaction [10]. We found two studies that found significant affective effects from adaptive music in games [37, 32]. However, neither study engages with game design literature, and both studies test the output of multifaceted music generation systems rather than isolating the adaptivity of music.

Understanding the affective impact of adaptive game music has a wide array of benefits for game development. In linear media, music is a powerful tool for manipulating an audience's experienced affect, in part due to empirical study of linear multimedia music [1]. Greater understanding of adaptive music in games leads to more impactful and better games.

In addition to improving games, this knowledge can influence the design of generative music systems. Currently, research in this area has many objectives, including musical style imitation [11], and generation of musical ambience [18]. As far as we are aware, only three systems follows an affect-based approach: *MetaCompose* [37], Prechtl's unnamed system [32], and *Melodrive* [43]. Our research helps the design and evaluation of generative music systems for games.

We present a study on the effect that adaptive music has on a player's experienced and perceived affect. Specifically, we isolate the affective dimension of tension. Our hypothesis is that music that adapts to and matches the tension curve of a game will strengthen a player's experienced tension. We created a game, titled *Galactic Escape (GE)* to study adaptive music. *GE* is detailed in Section 3.4. Our Independant Variable (IV) is the music that accompanies the game. There are four levels of the IV: No music, neutral music, music that adapts in inverse of game tension, and adaptive music that matches game tension. Our Dependent Variable (DV) is the player's reported affective response to the game.

We find that adaptive music has a significant effect on the player's experienced and perceived affect. We find that in a game with a rising tension curve, the player's experienced tension is significantly increased when adaptive music is added over linear music, even if the adaptive music is mapped inversely to game tension. We also find that players are aware

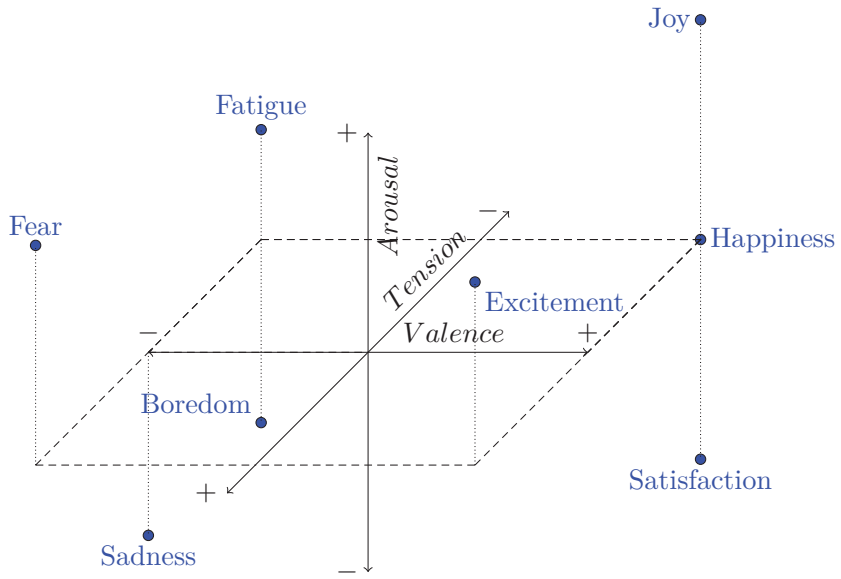


Figure 3.1: 3-Dimensional Model of Affect.

of the presence of adaptive music and feel that adaptive music significantly adds to the experience of playing a game.

3.2 Background

3.2.1 Affect

We use a 3-dimensional model of affect shown in Figure 3.1, with dimensions of valence, tension, and arousal. This is based on a model with dimensions of valence, tense arousal, and energy arousal [35], modified for simplicity, parity, and to bring the language into line with common terminology from the field of ludology [39]. This model is useful in both music and games, due to the importance of tension in both [19, 3].

Tension can be understood through the lens of cognitive dissonance [7], and is a strained emotional state resulting from conflict between contradictory elements [27]. While tension is often associated with increased arousal and decreased valence, it is a distinct dimension [36]. The opposite of tension is a feeling of resolution [4], or consonance. Mental consonance, or cognitive coherence [27] occurs when cognitive elements are not in conflict with each other. Tension is a temporal emotion that arises from a lack of resolution over time. We focus on tension as our IV due to its importance across media disciplines [3, 40].

3.2.2 Affect in Music

The relationship between music and emotion is complicated [19], though it is agreed that music has an affective impact on its listener. While some contend that listeners may only

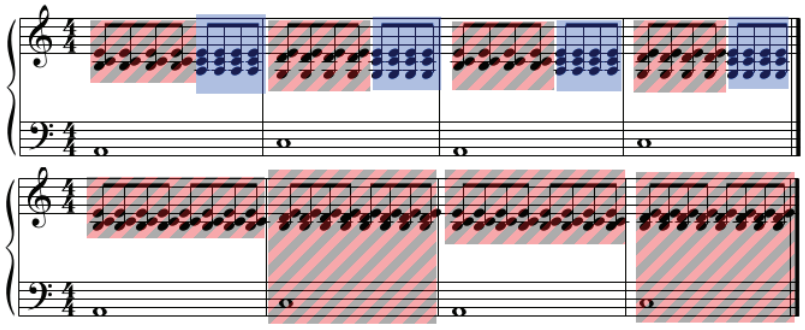


Figure 3.2: Top (a): Resolved dissonances. Bottom (b): Non-resolved dissonances.

perceive emotion in music [21], biofeedback technologies show a neurological and physiological affective response to music [2, 22]. This demonstrates that the affective impact of music has both physiological and psychological effects — it is both physically felt and perceptually experienced.

A listener's physiological response to music is stronger when paired with film [9], and, affective responses to film are stronger when paired with music [26, 41]. In addition to affective considerations, viewer ratings of films increase when the music is emotionally congruent with the film [16].

3.3 Generating tension in music and games

3.3.1 Tension in Music

The rise and fall of tension, or "tension curve", is a feature across many media types. Musical tension emerges when the listener has an expectation of musical movement that does not occur, or that occurs in a way that creates a new expectation of movement [4].

In musical harmony, dissonances can create tension. While there are many ways to create dissonance [38], we use the standard western 12-tone theory definition of dissonance to prevent confounds arising from unfamiliarity or novelty of harmonics and timbre. A dissonance is a chord, interval, or note that implies a future resolution to consonance [20]. The implication of future resolution is what causes tension in dissonances.

Figure 3.2 shows two similar musical excerpts. 3.2a resolves its dissonance, but 3.2b does not, creating tension. The dashed red highlights indicate dissonant intervals, and the solid blue highlights indicate consonant chords. Audio of these examples, and the examples in Figure 3.3, are available at *https://bit.ly/2QBwV1a* [29].

Rhythm is temporal in nature, and rhythmic tension may be created by introducing unbalanced and changing time signatures. The time signature of music indicates how the music is organized in time. Most western music occurs in time signatures with an even number of eighth notes in each measure. Disrupting this even division of time creates a dissonance between the expected timing of a note and its actual timing [42]. However, this

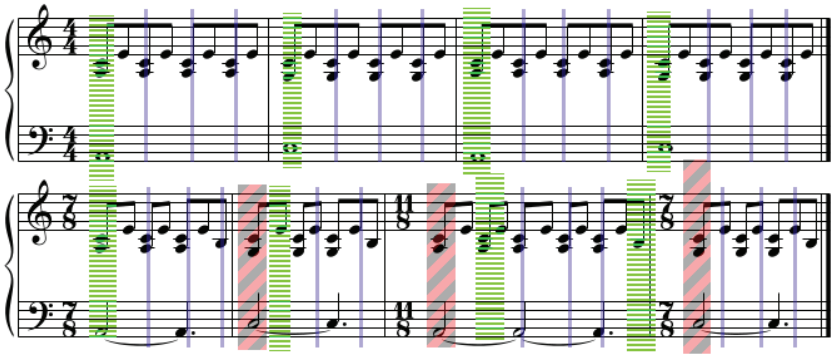


Figure 3.3: Top (a): Balanced resolved rhythm. Bottom (b): Unbalanced tense rhythm.

effect is unstable, and if the unbalance continues long enough for the listener to adjust their expectations, the tension can be lost or resolved. Figure 3.3 shows two similar musical excerpts. In 3.3a, the rhythm is evenly distributed. In 3.3b, rhythms are unbalanced and changing, creating tension. The striped green highlights indicate expected strong beats. The solid blue lines indicate the expected weak beats. The dashed red highlights indicate unexpected strong beats.

3.3.2 Tension in Games

In games, tension is generated through conflict: the placing of two opposing objectives against each other. Conflict is a necessary component part of games [34]. The most common form of conflict in games is a violent conflict - Chess abstracts war. Conflict can also be non-violent, as in the card-scoring mechanics of Poker. Games can also have both violent and non-violent conflict. The opposition of objectives that defines conflict is a dissonance. The interactivity of games means that the player is responsible for resolving the conflict, which increases tension. This is similar to non-game interaction, where cognitive dissonance can occur when a person acts in opposition to a private opinion [13].

Conflict in games can almost universally be abstracted to a conflict of the player against timers. A timer is any game mechanic whose expiry causes the player to experience some loss [3, 40]. This potential loss, and the player attempting to avoid it, is a core source of tension in games [28]. Importantly, a timer does not need to display a number on a screen, but may take many forms. A fixed timer depletes at a set rate over time. A variable timer drains with game conditions. A health bar is an example of a variable timer, as it decreases with actions instead of time. Variable timers still function as timers because the average player can be expected to deplete the timer at a consistent rate.

Atari's *Centipede* is an often given example of a variable timer used to create a constantly rising tension curve [34, 40]. In Centipede, the player avatar is at the bottom of the screen. The eponymous centipede begins at the top of the screen. It moves horizontally until it hits

either a mushroom or the edge of the screen, which causes it to move down one level and reverse direction. Mushrooms will appear both from player actions and enemy actions [34].

There are three timers in *Centipede*. As mushrooms appear, the player loses control of the environment and the centipede moves faster. As the centipede nears the bottom, the player is closer to losing a life. As the player loses lives, they are closer to losing all progress. These are all examples of timers creating an escalating tension curve [33].

This abstraction of game resources as timers can be applied to any game. Another example can be seen in a rising stack of tetrominos in *Tetris* [25]. Tension is reduced in games when timers complete and the player receives a loss, the player completes their objectives before the timer expires, or the timer is removed by some other means.

3.4 Galactic Escape

We created a game titled *Galactic Escape (GE)* for this study. *GE* has a rising tension curve caused by a variable timer. *GE* is designed to be easy to play without in-depth knowledge or familiarity with games. A video that fully explains the gameplay and mechanics of *Galactic Escape* is available at *https://youtu.be/3vxXbMeJGkw* [31].

The mechanics in *Galactic Escape* are similar to wager-based games of chance such as craps or roulette [8, 3], and are inspired by *Blades in the Dark* [17]. The player does not have direct control over whether they succeed or fail each challenge, but instead controls the ramifications of their success or failure.

At the beginning of the game, the player is given a very light text introduction, inspired by *House of the Dying Sun* [24], that uses common tropes to quickly let the player know that they are being pursued and must escape. The player then begins gameplay, navigating space in a small spaceship. The gameplay loop of Galactic Escape is shown in Figure 3.4. To win the game, the player must navigate the map shown in Figure 3.5 and complete a challenge at each point before a pursuing ship catches them.

The player begins by selecting a destination as seen in Figure 3.6. At each destination, the player must overcome a challenge. Before the challenge is resolved, the player places their wager by selecting one of three colour-coded levels of risk/reward: Green (low), Yellow (medium), or Red (high).

To determine success or failure, the player character has four attributes. Each challenge addresses one of the attributes, and can result in an outcome of failure, partial success, or success. The chance of each outcome is shown in Table 3.1. Attributes start at level 2, and can be modified based on the consequences of the roll.

Once the game has rolled the dice, it implements the rewards and/or consequences. Regardless of the outcome, the player must also wait a short time that is determined by their wager. The rewards and consequences, based on the roll result and player wager, are shown in Table 3.2.

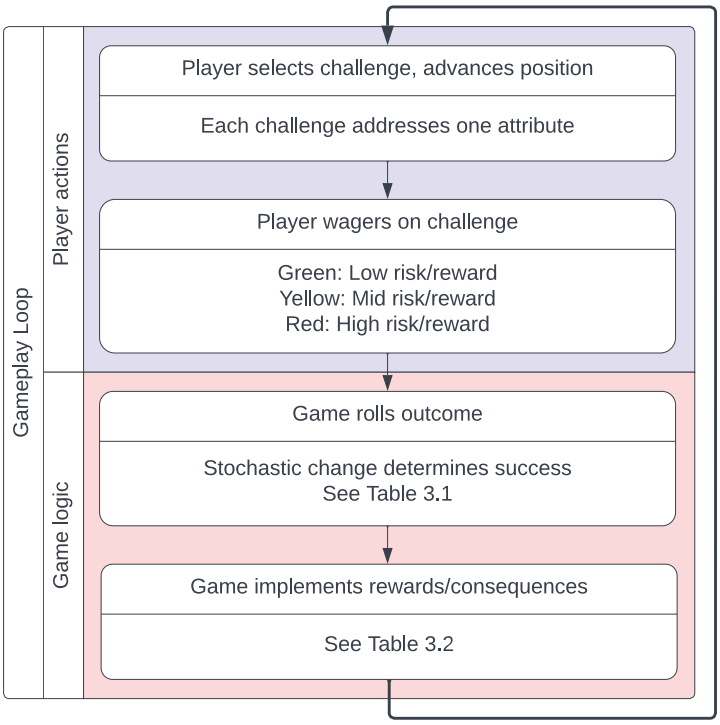


Figure 3.4: Gameplay loop of *Galactic Escape*.

The risk/reward levels of the player wagers are balanced by a pursuing enemy. If the player does not take risky actions the enemy will catch them, but the consequences for risky failures negatively affect future rolls as well. This creates a negative feedback loop that further incentivizes further risky actions [34].

3.4.1 Gameplay Tension in *Galactic Escape*

Tension in *GE* is created primarily with two timers: one fixed and one variable. The first timer is fixed and lasts 30 seconds, during which the player is alone on the map and can begin completing challenges. This timer is presented to the player as an on-screen number with the remaining time. When the first timer expires, the second timer begins. The second

Table 3.1: Percent chance of successes, partial successes, and failures for challenges, based on attribute level.

Attribute level	Success	Partial Success	Failure
0	3	22	75
1	17	33	50
2	31	44	25
3	42	45	13
4	52	42	6

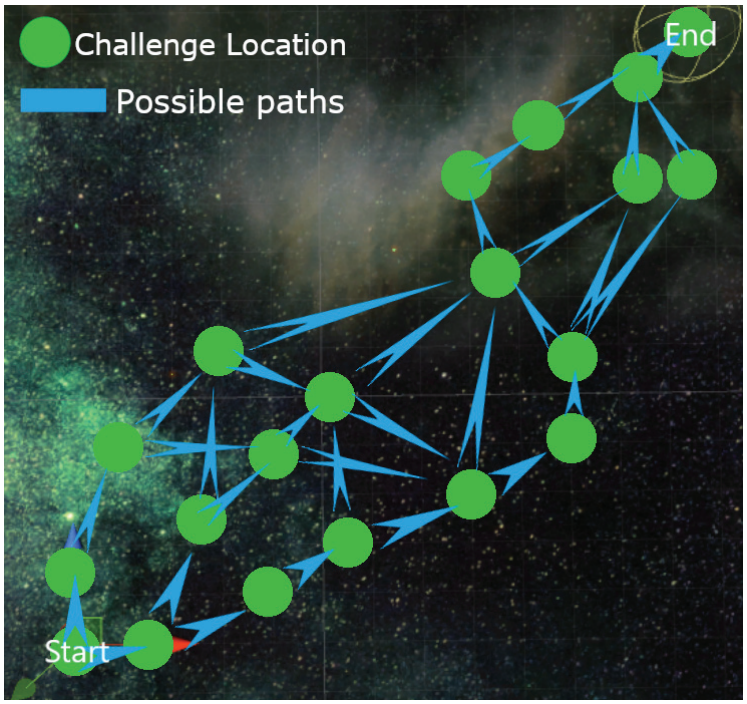


Figure 3.5: Overhead shot of game map with challenges and paths. The player does not see this map.

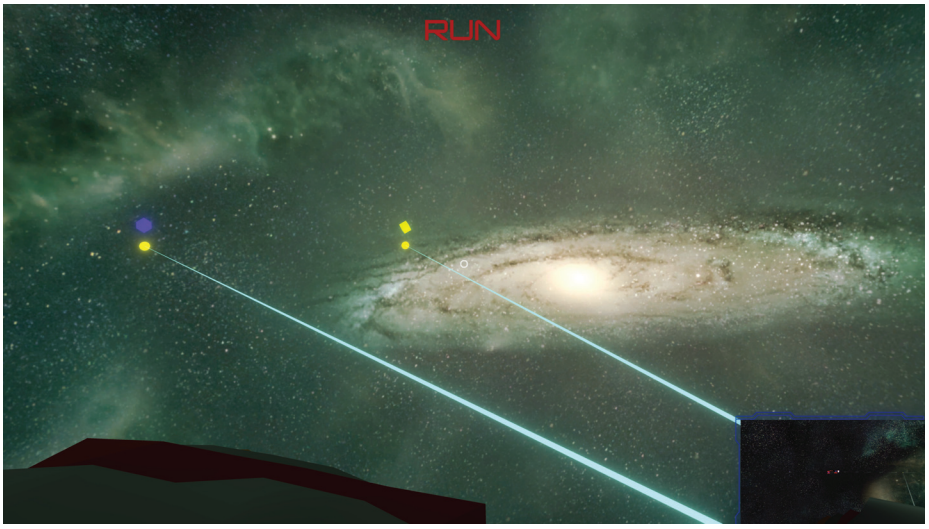


Figure 3.6: The player selects a destination by clicking on one of the two glowing points.

timer is a variable timer that is represented by a pursuing enemy ship. The player may look around during gameplay, and can see the pursuing ship. The pursuing ship appears at the player's initial position, and follows the same path that the player takes. The pursuing ship moves slightly slower than the player's ship, but does not need to clear challenges. This timer is represented by the distance between the player ship and pursuing ship. If the

Table 3.2: Consequences of die rolls based on roll result.

	Success (6)	Partial Success (4-5)	Failure (1-3)
Green 5 seconds	Consequences affect *current challenge* None	Time Penalty (2.5 Seconds)	Repeat Challenge
Yellow 3.5 seconds	Consequences affect *next roll* +1 Die on Next Roll	-1 Die on Next Roll	-1 Die on Next Roll Repeat Challenge
Red 2.5 seconds	Consequences affect attribute permanently +1 Die to Attribute	-1 Die to Attribute	-1 Die to Attribute Repeat Challenge

Table 3.3: Musical adaptivity based on experimental condition.

Condition	Musical response to game
None	No Music
Neutral	Neutral tension music, doesn't change with game tension
Inverse tension	Music decreases in tension as game tension increases
Tension	Music increases in tension as game tension increases

pursuing ship catches the player, the player loses. The player attempts to reach the final challenge before this timer expires.

Gameplay tension is measured by a variable *tense*, which has a value between 0-100 and represents how close the timers are to expiry. During the first timer, *tense* rises at a fixed linear rate from 0-25 over the 30 second timer. Once the second timer begins, *tense* takes its value from the Euclidian distance between the pursuing ship and the player ship. As the distance between the two approaches 0, *tense* approaches 100.

3.4.2 Musical Tension in *Galactic Escape*

Musical tension is the independent variable for this study. There are four different musical conditions, which are summarized in the Table 3.3.

FMOD Studio [14] is used to adaptively map the musical tension to the gameplay tension. The music changes based on the value of *tense*. The mappings of the ranges of gameplay tension across time as represented by *tense*, as well as the mappings of the ranges of musical tension across time, based on the ranges of *tense*, are shown in Figure 3.7. A short clip of each piece with *tense* rising from 0-100 over one minute is available at *https://bit.ly/2FomQMF* [30].

The music for the Neutral condition does not change based on *tense*. It is ambiguous, and uses harmonies and notes which are shared between multiple modes. These notes could

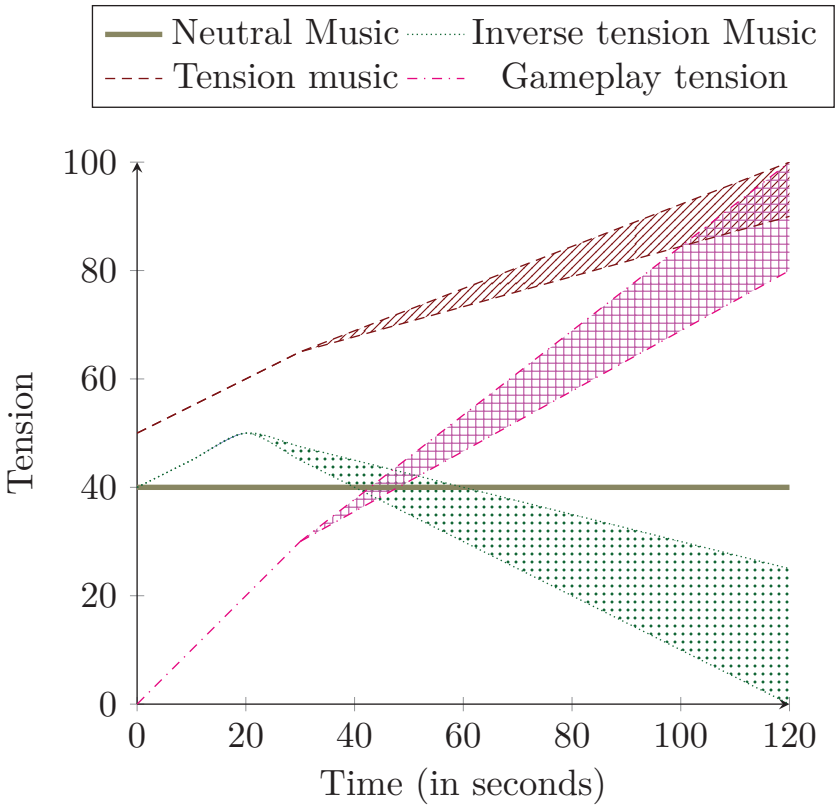


Figure 3.7: Approximate mapping of gameplay and musical tension for each condition, based on time. Note that the specific levels will depend on player actions.

sound consonant or dissonant if given more musical context. Rhythmically, the neutral music generally avoids strong accents.

The exact musical behaviour as it responds to the value of *tense* is described in Table 3.4. The music for the inverse tension condition is the most complex in its relationship to *tense*, because tension must first be present to decrease. The inverse tension music builds unstable and harsh sounds as *tense* is low, and resolves these instabilities as *tense* increases.

The music for the tension condition adaptively matches musical tension to *tense*, adding dissonant tones, harsh timbres, and crowded harmonies as tense increases. Also the music is rhythmically unbalanced, which becomes increasingly pronounced as *tense* increases.

3.5 Method

3.5.1 Design

Our experiment follows a within-subject design. Our Independent Variable is the musical tension as it relates to game tension. Our Dependent Variable is the participant's affective response to the game. Data is gathered from a multi-question survey that participants complete after each condition.

Table 3.4: Inverse tension music behaviours as system reacts to *tense*. Note that in inverse behaviour, musical tension increases as *tense* decreases.

tense	Musical behaviour
0-25	7ths and 2nds act as dissonances
(Highest music tension)	Polytonal fifth stacking on 2nd scale degree
	Higher spectra in timbre
	High-timbre bells
	More polytonal fifths
	Timbral shifts in bells
25-50	Bells resolve polytonality
	7ths/2nds resolve to octaves/3rds
50-75	Timbral shifts in bass + synth
	Polytonal stacking resolves to one tonal centre
75-100	Bass line simplifies and resolves
(Lowest music tension)	Drums enter with strong beats

3.5.2 Apparatus

GE was created using Unity 2017.3, with models made in 3ds Max. Participant responses are automatically uploaded to an online database after obtaining consent for storing anonymous data. In-person participants play the game on an Apple iMac, with Monoprice-branded over-the-ear headphones. Remote participants are free to use any setup they are comfortable with, as long as they are able to hear the audio. While lack of environment control may seem to be a weakness of the design, it more closely simulates the actual audio conditions of playing a game, increasing external validity.

3.5.3 Participants

35 participants took part in this study: 8 in-person and 27 online/remote. 5 remote participants were removed from the data after failing to complete all four conditions, leaving a final participant pool of 30. Participants were recruited from mailing lists for games music, online gaming boards, students from an undergraduate 3rd year sound design class, and Amazon's Mechanical Turk. The course material in the sound design class is not directly related to the research. The data is consistent between participant groups. Participant age ranges from 17-54, with an average age of 28 (SD = 10.08). 12 participants are female. Participants spend an average of 90 seconds in each game scenario, and 90 seconds answering each survey. Participants report a variety of gaming experience, with 10 participants who report playing 0-2 hours of games per week, and two participants who report playing over 10 hours per week.

3.5.4 Procedure

Each of the four conditions consists of a single play of the game, with the associated musical condition. The order of the conditions is randomized by Unity to prevent order effects. Before the first condition, demographic data about the participant's age, gender, and gaming experience is taken. The participant is then shown a brief tutorial that explains the gameplay of *GE*, and then plays a training game without loss, music, or timers, to familiarize themselves with the game. The player indicates when they are ready to begin the first condition by pressing a button.

After each condition, the participant fills out a survey with 13 quantitative, and 2 qualitative questions. The questions are a modified version of an instrument designed to measure game enjoyment [12]. Questions from the original instrument that are not relevant to the study are removed, and questions are added to tailor the instrument to the affective nature of the study. Fang et al. verified the original instrument with a Cronbach's alpha of 0.73, indicating consistency and reliability. After filling out the survey, the participant begins the next condition. This results in one value per question for each condition. Participants indicate their response between 0 (Disagree) and 1 (Agree), using a continuous slider with a granularity of $<.001$. The questions are grouped together for analysis. These questions and groups are shown in Table 3.5. The "code" refers to the internal representation for analysis and charts, and will be used to refer to the questions moving forwards. These questions are also split into 3 primary groups:

Experienced enjoyment questions

These questions were derived from previous research on player engagement and enjoyment [12], and measure positive and negative emotions for gauging overall player enjoyment. These questions are not intended to map to affective dimensions, but they provide a non-dimensional overview of player enjoyment. As player enjoyment increases, positive responses will rise in reported value, and negative responses will fall in reported value.

Affect/Emotion questions

These questions are split into two groups:

Experienced affect questions These questions measure the player's self-reported experienced feelings of the major affective dimensions, and directly address the 3-dimensional model of affect [35].

Perceived emotion questions These questions measure the player's perception of the soundtracks emotional congruency with the gameplay.

Table 3.5: Questions from modified instrument.

Category	Code	Question
Enjoyment		
Positive	Happy	I feel happy when playing this game
	Calm	I feel calm when playing this game
	Immersion	I am aware of my surroundings when playing this game
	Enjoyment	I enjoy playing this game
Negative	unhappy	I feel unhappy when playing this game
	Worried	I feel worried when playing this game
	Exhausted	I feel exhausted when playing this game
	Miserable	I feel miserable when playing this game
Affect/Emotion		
Affect	Valence	This game was pleasant to play
	Arousal	I feel energetic when playing this game
	Tension	I feel tense when playing this game
Perceived motion	Soundtrack experience	The soundtrack added to the experience of playing this game
	Soundtrack matching	The soundtrack matched my progress in the game
Other		
Qualitative	N/A	Please describe what you think of the soundtrack in this game
	N/A	Please describe any additional thoughts about this game
Play data	N/A	Order of tests
	N/A	Num. of challenges reached
	N/A	Time taken from start-finish
	N/A	Number of success rolls
	N/A	Number of partial success rolls
	N/A	Number of failure rolls

Other

These questions are also split into two groups. Qualitative data is collected, but did not contain any discussion of the adaptivity of the music or affective response to the game. Play data was collected to check for any potential confounding effects from player differences.

We are aware of critiques of rating-based models in HCI research [45]. However, these critiques do not apply to this study. Individual differences are accounted for by using a within-subjects approach. The participants give their ratings as a numeric slider value, rather than by interpreting language or categories. Because each playthrough of *GE* takes ˜120 seconds to complete and has an affective impact on the player, using a ranking system is not feasible as the player may not remember their affective state from over two minutes prior. We acknowledge that self-report data collection methods can only give subjective experiential data rather than the objective data that biofeedback technologies can give, self-report methods are common in studies of musical affect and are valid for understanding experienced and perceived emotion [19].

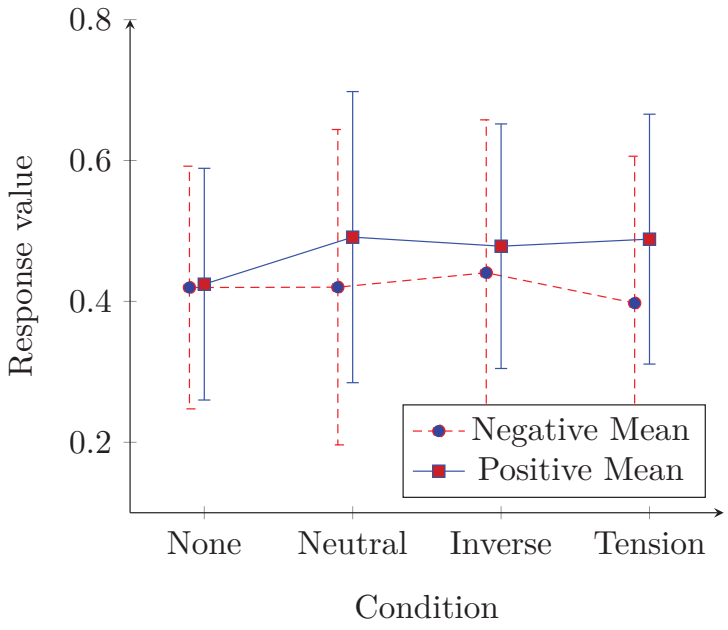


Figure 3.8: Means and Standard Deviations for experienced enjoyment, grouped by positive and negative categories in Table 3.5.

3.6 Results

Aggregate surveyed results show a response to the IV manipulation, as shown in Figures 3.8-3.10. A surprising result is that there appears to be a stronger impact with the introduction of adaptive music than when the music adaptively matches game tension. The statistical significance of these results will be discussed in Section 3.6.2.

3.6.1 Descriptive statistics

The experienced enjoyment emotion responses indicate that players enjoy the game more with the introduction of music. Enjoyment slightly decreases as the music tension adapts inversely to the game tension, and increases more as the music tension matches the game tension. Figure 3.8 shows the averages and standard deviations of questions as grouped in Table 3.5. For example, the "positive" mean demonstrates the mean and standard deviation for the responses to questions of happiness, calmness, immersion, and enjoyment.

The perceived emotion responses show a more consistent trend. Players report that they feel the music adds to the experience. This effect increases as music tension adapts to game tension, and further increases as music tension matches game tension. This effect is shown in Figure 3.9.

The experienced affective dimensions of valence and arousal follow a similar pattern to the perceived dimensions, and can be seen in the dotted blue and dashed green lines in Figure 3.10. Reported values for valence and arousal rise when music is introduced. These

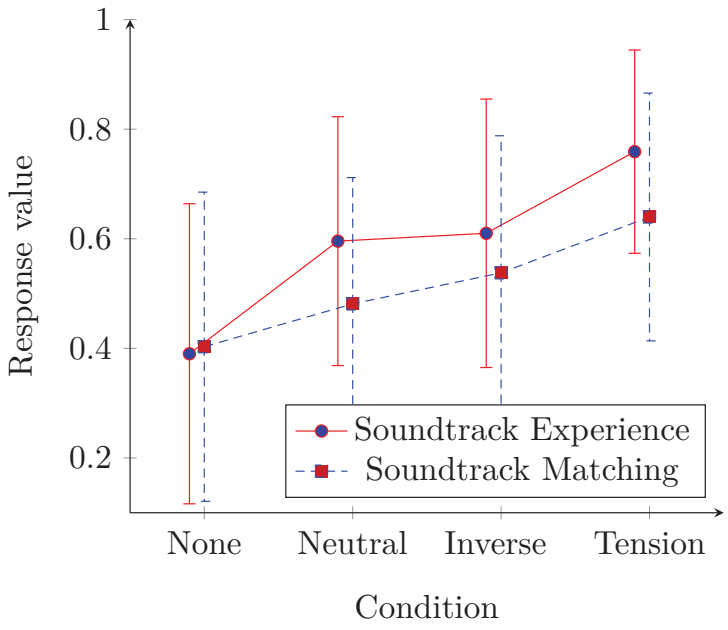


Figure 3.9: Means and Standard Deviations for ratings of soundtrack matching the emotions and adding to the experience of the game.

values further rise when the music is adaptive. These values rise again when the music adapts to match the gameplay tension.

The player's reported feeling of tension differs from this path, and can be seen in the solid red line of Figure 3.10. As music is introduced, player tension is reduced. As the music tension adapts to the game tension, player tension increases. As the music tension matches game tension, player tension rises further.

3.6.2 Inferential statistics

Our hypothesis is that tension-adaptive music will strengthen the player's experienced tension. To test this hypothesis, we perform a repeated measures multivariate ANOVA. Before running the ANOVA, we test the assumptions. A majority of the affective responses for each scenario are normally distributed. The violations are shown in Table 3.6. These violations contain 25% of all responses. Because of the robustness of the ANOVA, and because the violations are small and in only 25% of all responses, a normal ANOVA is performed. Mauchly's assumption of Sphericity is not violated in any of the responses, and no corrections are necessary.

The ANOVA shows significant change across multivariate responses to the conditions $F(39, 231) = 1.521$, p=.032. Separate post-hoc univariate repeated measures ANOVAs are then performed on each of the dependent variables. Three responses from participants are individually significant.

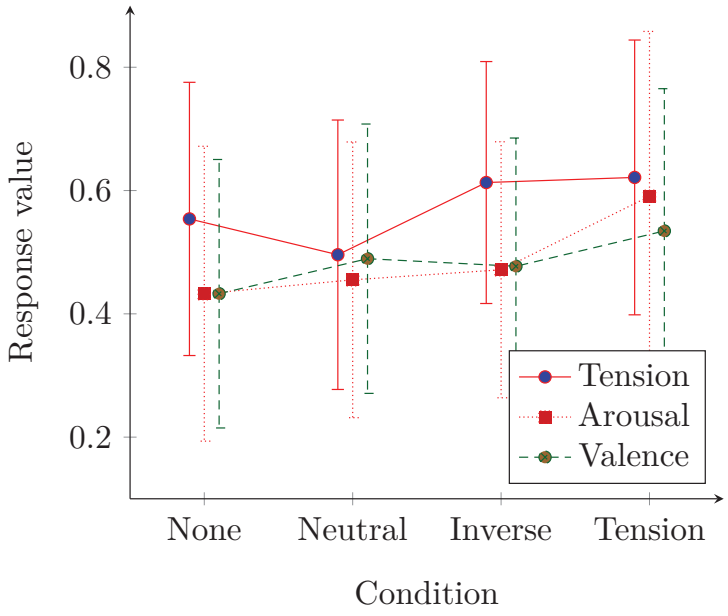


Figure 3.10: Means and Standard Deviations for ratings of experienced valence, arousal, and tension.

Table 3.6: Questions with responses containing violations of normality.

Condition	Violations
None	exhausted, arousal, soundtrack matching, soundtrack experience
Neutral	miserable
Inverse tension	immersion
Tension	worried, exhausted, miserable, enjoyment, valence, tense soundtrack matching

Table 3.7: Means and Standard Deivations for individually significant univariate responses by condition.

	None		Neutral		Inverse		Tension	
	M	SD	M	SD	M	SD	M	SD
Tense	0.55	0.22	0.49	0.21	0.61	0.19	0.62	0.22
Soundtrack Matching	0.40	0.28	0.48	0.23	0.53	0.25	0.63	0.22
Soundtrack Experience	0.39	0.27	0.59	0.22	0.61	0.24	0.75	0.18

Both perceived emotion responses — "soundtrack experience", $F(3, 87) = 11.227, p < .001, \eta_p^2 = .279$, and "soundtrack matching", F(3, 87) = 6.967, p<.001, η_p^2=.194, are significant with large effect sizes. Tension is also individually significant F(3, 87) = 3.662, p=.015 η_p^2=.112. The values for these means and differences are shown in Table 3.7.

Individual post-hoc t-tests further clarify these results. For tension, there is a significant change between the Neutral and Inverse tension conditions p=.048 and Neutral–Tension

conditions p=.036. For "soundtrack matching", there is significance between None-Tension p<.001, Neutral-Tension p=*.022*, and None-Inverse tension p=.045. For "soundtrack experience", individual t-tests show significance between all conditions with the exception of Neutral-Resolution p=.824.

3.7 Discussion

We demonstrate significant support for the hypothesis that as musical tension adapts to game tension, the player's experienced tension is increased. However, this relationship is not a linear one. The player's experienced tension is reduced with the introduction of static music, and increases with the introduction of adaptive music, even if the musical tension is inverse of game tension. This may be the result of tension's nature as a temporal affect, as tension must be created before it can resolve.

We also show that players perceive the emotional congruency of the music that is playing, during gameplay, and report that they feel that tension-adaptive music adds more to the experience of playing a game. This indicates that players both perceive and value music, and perceive and value adaptive music more.

Viewers rate films higher when the music is emotionally congruent [16], though film music is sometimes intentionally emotionally incongruent [23]. While players in this study report increased tension when adaptive music is emotionally congruent, the more significant change occurs with the introduction of adaptive music. This suggests that the adaptivity of the music is more important of a factor than the emotional congruency of the music.

3.8 Conclusion

We present an empirical study on the affective ludology of adaptive music, focused on the affective dimension of tension. We show that tension-adaptive music amplifies the player's experienced tension when compared to linear or no music. We also show that players are aware of the affective congruency of music and gameplay, and that their experienced tension increases with affectively-congruent adaptive music.

While this study provides an important step in understanding the relationship between music and video games, it is only a step. Our results align with Prechtl and Scirea's previously mentioned research. While both the musical and game tension are grounded in literature and theory, they have not been independently confirmed. It is unknown if the measured effects will change if interaction speed changes, or whether different camera or avatar representations may disrupt or change these effects. Other currently unstudied facets of this interaction are the potential roles of listening environment, previous gaming experience, input device, and narrative elements. Finally, while we measure subjective experienced and perceived emotion with self-report methods, biofeedback technology would provide objective data on felt emotion. We do note that both Prechtl and Scirea described difficulties

in comparing biofeedback in their studies, but that the biofeedback data agreed with the subjective data.

We present support for our hypothesis. We also present support for the statement that for affective impact, music matters in video games, and that adaptive music matters more. While only some of our measured data is significant, the trends are consistent: adaptive music increases player enjoyment, and strengthens the affective impact of a game.

Acknowledgments

We would like to thank Kivanc Tatar for assisting with participants and data-gathering. We would also like to thank Bernhard Riecke and Alexandra Kitson for their assistance with the design of this study, and we would like to thank Jianyu Fan for his help in proofreading and reviewing this paper.

Bibliography

[1] Fernando Arroyo Garcia Lascurain. *Affect and Feelings: The Persuasive Power of Film Music*. PhD thesis, University of California, University of California, 2016.

[2] Tonio Ball, Benjamin Rahm, Simon B. Eickhoff, Andreas Schulze-Bonhage, Oliver Speck, and Isabella Mutschler. Response properties of human amygdala subregions. *PLoS ONE*, 2(3):e307, 2007.

[3] Richard Bartle and Chris Bateman, editors. *Beyond game design: Nine steps towards creating better videogames*. Charles River Media/Course Technology, Boston, Mass., 2009.

[4] David Bashwiner. Tension. In *Music in the Social and Behavioral Sciences: An Encyclopedia*, volume 2, pages 1113–1115, Thousand Oaks, 2014. SAGE Publications, Inc.

[5] JaeHwan Byun and Christian Loh. Audial engagement: Effects of game sound on learner engagement in digital game-based learning environments. *Computers in Human Behavior*, 05 2015.

[6] G.G. Cassidy and R.A.R. Macdonald. The effects of music on time perception and performance of a driving game. *Scandinavian Journal of Psychology*, 51(6):455–464, 2010.

[7] Joel Cooper and Kevin M. Carlsmith. Cognitive dissonance. In *International Encyclopedia of the Social and Behavioral Sciences*, pages 76–78. Elsevier ltd, Amsterdam, 2015.

[8] Greg Costikyan. *Uncertainty in games*. Mit Press, 2013.

[9] Eran Eldar, Ori Ganor, Roee Admon, Avraham Bleich, and Talma Hendler. Feeling the real world: Limbic response to music depends on related content. *Cerebral Cortex*, 17(12):2828–2840, 2007.

[10] Andrew Elmsley, Ryan Groves, and Valerio Velardo. Deep Adaptation: How Generative Music Affects Engagement and Immersion in Interactive Experiences. *Digital Music Research Network One-day workshop*, 12:7, 2017.

[11] Steve Engels, Fabian Chan, and Tiffany Tong. Automatic real-time music generation for games. In *11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 220–222, Palo Alto, CA, 2015. AAAI Press.

[12] Xiaowen Fang, Susy Chan, Jacek Brzezinski, and Chitra Nair. Development of an instrument to measure enjoyment of computer game play. *Intl. Journal of Human–Computer Interaction*, 26(9):868–886, 2010.

[13] L. Festinger and J. M. Carlsmith. Cognitive consequences of forced compliance. *Journal of abnormal psychology*, 58(2):203–10, 1959.

[14] FMOD. (Software) FMOD Studio, 2016.

[15] Hans-Peter Gasselseder. Dynamic music and immersion in the action-adventure an empirical investigation. *ACM International Conference Proceeding Series*, 2014, 10 2014.

[16] Waldie E. Hanser and Ruth E. Mark. Music influences ratings of the affect of visual stimuli. *Psychological Topics*, 22(2):305–324, 2013.

[17] John Harper. *(Game Rulebook) Blades in the Dark*. Evil Hat Productions, Maryland, USA, 2017.

[18] Jim Hedges, Kurt Larson, and Christopher Mayer. An Adaptive, Generative Music System for Games. URL: `https://www.gdcvault.com/play/1012710/An-Adaptive-Generative-Music-System`, 2010. Last accessed: 04-25-2022.

[19] Patrik N. Juslin and John Sloboda. *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[20] Michael Kennedy and Joyce Bourne. *The Oxford Dictionary of Music*. Oxford University Press, Oxford, 6 edition, 2012.

[21] Peter Kivy. *The corded shell*. Princeton essays on the arts. Princeton University Press, Princeton, 1980.

[22] Lars-Olov Lundqvist, Fredrik Carlsson, Per Hilmersson, and Patrik N. Juslin. Emotional responses to music: Experience, expression, and physiology. *Psychology of Music*, 37(1):61–90, 2009.

[23] Henry Mancini. *Did they mention the Music? The autobiography of Henry Mancini*. Cooper Square Press, New York City, New York, 2001.

[24] Marauder Interactive. (Game) House of the Dying Sun, November 2016.

[25] Alexey Pajitnov and Vladimir Pokhilko. (Game) Tetris, June 1984.

[26] Rob Parke, Elaine Chew, and Chris Kyriakakis. Quantitative and visual analysis of the impact of music on perceived emotion of film. *Computers in Entertainment (CIE)*, 5(3):5, 2007.

[27] Philippe Pasquier and Brahim Chaib-Draa. Agent communication pragmatics: the cognitive coherence approach. *Cognitive Systems Research*, 6(4):364–395, 2005.

[28] Bernard Perron, Mark J.P. Wolf, and Thomas H. Apperley. *The video game theory reader 2*, volume 2. Routledge New York, 2009.

[29] Cale Plut. (Soundcloud) Examples of Harmonic and Rhythmic Tension. URL: `https://bit.ly/2QBwV1a`, 2018. Last accessed: 04-25-2022.

[30] Cale Plut. (Soundcloud) Music Matters Musical Examples. URL: `https://soundcloud.com/khavall/sets/music-matters-musical-examples/s-5fBfr`, March 2019. Last accessed: 04-25-2022.

[31] Cale Plut. (Video) Galactic Escape description. URL: `https://youtu.be/3vxXbMeJGkw`, 2019. Last accessed: 04-25-2022.

[32] Anthony Prechtl. *Adaptive music generation for computer games*. PhD thesis, Open University (United Kingdom), 2016.

[33] Richard Rouse. *Game Design: Theory and Practice*, volume 2nd ed of *Wordware Game Developer's Library*. Jones & Bartlett Learning, Plano, Tex, 2005.

[34] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA, 2004.

[35] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345, 2000.

[36] Ulrich Schimmack and Rainer Reisenzein. Experiencing activation: Energetic arousal and tense arousal are not mixtures of valence and activation. *Emotion*, 2(4):412–417, 2002.

[37] Marco Scirea. *Affective music generation and its effect on player experience*. PhD thesis, IT University of Copenhagen, Digital Design, 2017.

[38] William A. Sethares. *Tuning, timbre, spectrum, scale*. Springer, London, 2nd ed. edition, 2005.

[39] Michael Sweet. *Writing interactive music for video games: A composer's guide*. Addison-Wesley, Upper Saddle River, NJ, 2015.

[40] Richard Terell. Tension: Threats and Timers. URL: `https://bit.ly/2Ff6cPm`, 2009. Last accessed: 01-02-2019.

[41] Julian F. Thayer and Robert W. Levenson. Effects of music on psychophysiological responses to a stressful film. *Psychomusicology: A Journal of Research in Music Cognition*, 3(1):44–52, 1983.

[42] William Forde Thompson. *Music, thought, and feeling*. Oxford University Press, New York, second edition. edition, 2015.

[43] Valerio Velardo. Melodrive: Adaptive music generation. URL: `https://melodrive.com/index.php`, 2018. Last accessed: 01-02-2019.

[44] Paul Weir. (Video) The Sound of "No Man's Sky". URL: `https://bit.ly/2UKShXS`, 2017. Last accessed: 04-25-2022.

[45] Georgios N. Yannakakis and Héctor P. Martínez. Ratings are overrated. *Frontiers in ICT*, 2:13, 2015.

[46] Jiulin Zhang Xiaoqing Fu. The influence of background music of video games on immersion. *Journal of Psychology & Psychotherapy*, 05, January 2015.

Chapter 4

The IsoVAT corpus: Parameterization of musical features for affective composition

As submitted to Plut, C., Pasquier, P., Ens, J., & Tchemeube, R. (2022). *The IsoVAT corpus: Parameterization of musical features for affective composition.* Transactions of the International Society for Music Information Retrieval.

Abstract

While there is a breadth of research in mapping Western musical features to perceived emotion within research in music and emotion, a critique of the field is that this breadth of methodologies lacks in inter-communication, which may reduce the generalizability of findings across the field. We consolidate previous research in this area to construct a parameterized composition guide that maps musical features to their associated emotional expression. We then use this guide to compose the "IsoVAT" dataset, a collection of symbolic MIDI clips in a variety of popular Western styles. This dataset contains a total of 90 clips of music, with 30 clips per affective dimension, organized into 10 sets of 3 clips. Each clip within a set is composed to express a low, medium, or high level of an affective dimension when compared to the other clips within the same set. We perform an empirical experiment to evaluate the validity of our affective composition guide, and to establish the ground-truthed emotional expression of the IsoVAT Dataset. The ground-truthing reveals 19 sets that match the composed labels, 10 sets that have ground-truthed labels that disagree with composed labels, and 1 clip that does not have clear agreement across the three study designs.

4.1 Introduction

Music-emotion research (MER) is a broad interdisciplinary field that uses numerous approaches, models, and methodologies. Criticisms of MER concern a lack of internal coherence in terms of stimulus selection, emotion model, and definitions of musical features [7, 35]. Within Western music, Eerola and Vuoskoski surveyed 251 MER studies, describing broad trends [7]. Warrenburg surveyed 306 studies to build a database of previously used musical stimulus [35].

These two surveys identify musical stimuli as a potential confound in MER, noting that most studies use commercial recordings of existing music, and are chosen primarily based on inclusion in previous studies. We discuss these surveys in Section 4.2 While using "real-world" music keeps external validity high, control over musical features is lost. To ensure that only desired features and parameters are altered, and to verify the emotional expression of musical stimulus, surveys suggest composing parameterically controlled music, as well as empirically ground-truthing the emotional expression of a musical dataset [7, 35].

The semantic gap between human perception of music and low-level features extracted from audio is another possible confound in MER [39]. Panda, Malheiro, and Paiva suggest that audio features are insufficient for determining emotional expression [22]. We address this confound by using symbolic representation of our corpus in MIDI, allowing for direct control over composition features.

One issue in identifying a set of musical parameters to control for emotional expression is the lack of internal coherence in MER literature. In Section 4.2.5, we collate survey results from across MER [16, 19] to create a set of musical features and their relationship to emotional expression. We further delineate these features by whether they are primarily in the domain of musical composition, or expressive performance. Section 4.3 describes and details our collated composition guide that describes a set of musical features and their associated emotional perception.

As mentioned, one goal in creating this guide is to have a set of musical parameters to control for emotional expression in music composition. Therefore, to evaluate the validity of the guide, we compose a set of music based on the guide, and empirically evaluate the emotional perception of the music. Because the guide is intended for use across a range of popular Western musical styles, we compose our music in a variety popular Western styles.

We compose the "IsoVAT" corpus, manipulating the composition-related features as described in the guide to manipulate the intended emotional expression. This corpus contains 90 4-bar musical clips, and is described in Section 4.4. The IsoVAT corpus is divided into three sets, expressing the isolated emotional dimensions of valence, arousal, and tension. These clips are further divided into 10 sets of three - each set of three clips contains one clip that expresses a lower level of the associated affective dimension than the other two clips in the set, one clip that expresses a higher level of the associated affective dimension than

the other two clips in the set, and one clip that expresses a level of the associated affective dimension that is between the other two clips. In other words, each set contains a clip expressing a comparatively low, medium, and high level of the associated dimension. Each set of three shares instrumentation, genre, and tempo, to control for the possible effects of these features. The genres in the corpus are primarily a mix of popular, classical, and jazz styles.

We ground truth our musical set across three study designs in Section 4.5. The first study design, "2-rank", is discussed in Section 4.5.1. This design evaluates the clips as composed, with participants selecting the clips that they perceive the lowest and highest level of the perceived affective dimension. The "1-rank" design is discussed in Section 4.5.1, and asks participants to rank 2-clip subsets of each set, selecting the clip that expresses a higher level of the associated affective dimension. Finally, the "Likert-type" design is discussed in Section 4.6.3, and asks participants to rate each clip from 1-7, based on the level of perceived affect. These study designs show a surprising amount of variance, particularly the 1-rank comparison design. The most stable evaluation of the corpus occur in the 2-rank design. The dimension of valence exhibits the most variance, and arousal exhibits the least.

We discuss the results of our empirical evaluations in Section 4.6. While our results exhibit substantial variance, the corpus itself is also composed with several constraints that limit its emotional expression. The 2-rank and Likert-style results demonstrate trends that support the composition guide overall, though the 1-rank results are more varied. We combine all results to produce 29 ground-truthed ranked sets of 3 clips, with 1 set exhibiting too much variance to accurately ground truth. In Section 4.7, we musically analyze sets from the corpus whose ground-truth order is different than the composed order. We discuss common themes and elements that occur in these sets.

Overall, we investigate whether the collected and centralized findings from previous MER literature concerning the relationship between musical features and affect can be used to control a compositional process, to expresses a desired affect. In other words, we explore whether the affective study and analysis of musical features can be applied to guide the creation of new music. In doing so, we find support for both findings and critiques of previous MER. We collate and generalize mappings of musical features and expressed emotion, and create a corpus that evaluates this mapping. Our corpus evaluation supports the general trends in our guide, though there is a high degree of variance.

4.2 Background and Motivation

4.2.1 Affect model and representation

Though the mechanisms are not fully understood, listeners are capable of perceiving emotion in music, and music is commonly believed to be capable of evoking and inducing emotions in the listener. There are numerous affect models used in MER. Categorical models describe

a set of basic universal emotions, from which all other emotions derive. Dimensional models describe emotions with two or three bipolar dimensions. The number of dimensions in a model is often derived from the application of the model, and the 3-dimensional model often contains some correlation between dimensions [16, 30].

Eerola and Vuoskoski describe a potential drawback to discrete emotion models is that they may produce Type 1 "false positive" errors and overconfidence [7]. In a study with both categorical and dimensional models, Vieillard et al. find support for this, with dimensional responses showing higher variability than categorical responses [34].

We use a 3-dimensional Valence-Arousal-Tension (VAT) model, similar to other 3-dimensional models [38, 30, 27]. Table 4.1 provides an overview of common 2- and 3-dimensional emotion models used in previous MER. Tension is often discussed in music [16], and therefore include tension in our model. The most common other emotional models use valence/pleasure and arousal/activity [39, 6, 35].

Table 4.1: Summary of common dimensional emotion models.

Model	# of Dimensions	Dimensions	Source
Wundt	3	Pleasure/Displeasure Arousal/Calmness Tension/Relaxation	[38]
Circumplex (Russell)	2	Valence Arousal	[29]
2DES (2-Dimensional Emotion Space)	2	Valence Arousal	[32]
PAD	3	Pleasure Arousal Dominance	[20]
Schimmack and Grob	3	Valence Energy Arousal Tension Arousal	[30]

Valence

Valence, sometimes called the "hedonic tone", is associated with the pleasantness or attractiveness of stimulus [6, 30]. Stimulus that is pleasant or attractive has a high, positive valence. Stimulus that is unattractive or unpleasant has low, negative valence. Examples of high-valence emotions include joy, excitement, and triumph. Examples of low-valence emotions include sadness, fear, and disgust.

In music, positive valence is generally associated with major modes and consonant harmonies [16]. Harmonic consonance is not always well-defined, as it often depends on contextual elements like genre or historical context. An example of a high-valence piece used in MER stimulus is Vivaldi's *La Primavera*. A similar low-valence example is Barber's *Adagio for Strings*.

Arousal

Arousal is an emotional dimension associated with the energy or activity of stimulus [6], and is occasionally called "energy arousal" [30]. Arousal is a state of heightened activity, which may be positive or negative. Examples of high-arousal emotions include excitement, anger, and triumph. Examples of low-arousal emotions include satisfaction, depression, exhaustion, and relaxation.

In music, arousal is often associated with tempo and note density, increased volume, pitch level, and melodic direction [16]. For example, many loud pitches moving with an upwards contour will likely express positive arousal. Barber's *Adagio for Strings* expresses a low arousal and low valence. Mussorgsky's *Night on Bald Mountain* is a high-arousal piece used in previous MER studies.

Tension

Tension is a prospect-based emotional dimension associated with future or prospective events [21]. Tension can occur with both positive and negative valence. For example, Excitement is a positively valenced tension - a subject believes that a future event is coming that will have a desirable effect. Fear is a negatively valenced tension - a subject believes a future event is coming that will have an undesirable effect. Examples of high-tension emotions include fear, excitement, and unease. Examples of low-tension emotions include satisfaction, sadness, and joy.

In music, tension is most often associated with harmonic instability, often described as dissonance [16]. Dissonances are a "clash" between notes that imply future resolution into consonance. Tension will generally increase as the implied resolution does not occur. Tension is also associated with melodic range and interval pitch level (size of intervals). *Night on Bald Mountain* expresses a high level of tension and arousal. Mozart's *Eine Kleine Nachtmusik* expresses a low level of tension.

4.2.2 Previous Music-emotion datasets

Warrenburg notes in 2020 that 37% of musical stimuli used in her "Previously Used Musical Stimulus" (PUMS) database was selected for a study due to inclusion in a previous study. In 2013, Eerola and Vuoskoski mention that the most common method for selecting musical stimulus was for the researchers to hand-select the stimulus, appearing in 33% of surveyed studies. Eerola and Vuoskoski advocate for creating parameterized musical stimulus, while maintaining ecological validity. Warrenburg highlights the importance of empirically ground-truthing datasets before their use [7, 35].

Most datasets in MER utilize commercial audio recordings of musical stimulus. Examples of audio-based datasets include *PMEmo*, drawn from Billboard Top 100 lists [40], and *Emotify*, separated into four genres, and drawn randomly from a selection provided by the

Magnatune company [1]. As mentioned in Section 4.1, using audio recordings reduces the extractable compositional features compared to symbolic music. Panda et al. describe multi-model approaches, such as utilizing both audio and lyrical information, as a solution to the limits of audio feature extraction, and note that there is little examination into multi-model datasets with symbolic representation of the music [23].

Thompson and Robitaille ask composers to write "short" melodies that express one of six categorical emotions, with no other specific musical instructions [33]. These pieces are evaluated by listeners with some musical training, rating each piece with a 7-point likert scale. Overall, composers were successful in writing music that was perceived to express the intended musical category.

Vieillard et al. compose sets of "film"-genre symbolic music for solo piano that expressed four discrete emotions: Happy, Sad, Peaceful, and Scary [34]. They composed these pieces using "the rules of the Western tonal system". They evaluate these pieces using categorical and dimensional affect representations, and the results of the empirical evaluation indicate support for the approach of composing custom music for MER tasks.

Panda et al. create a multi-model dataset that matches 193 of 903 audio clips in the AllMusic database, using the MIREX classifications that include categories such as "rollicking", "poignant", and "visceral" [23]. These clips are classified via machine learning (ML) approaches. Hung et al. create EMOPIA, a multi-model dataset with 1,087 clips annotated by the researchers, drawn from 387 solo piano performances. EMOPIA targets pop styles of "Japanese anime", "Korean and Western pop", "Movie soundtracks", and "personal compositions"[15].

4.2.3 Parametric co-creative composition

Generative music is music that is partially or completely created with some automated process [24]. One possible application of generative music systems is the co-creation of music with a human composer. Gerhard and Hepting note that one issue in current co-creative generative music systems is that they often require a composer to use tools and techniques that may be highly technical and unfamiliar to the composer [10], which may lead to frustration. Another difficulty in co-creation, as with MER research in general, is the lack of agreed-upon musical parameters and terminology. Paz et al. present one possible solution to this issue in automatically deriving sets of parameters from an input musical corpus [26].

Some generative music systems take a small, potentially single-piece corpus as input, and generate additional musical content (e.g. continuation of a partial musical phrase) or similar music [13]. *MidiMe* fine-tunes a VAE that is initially trained on Google's MusicVAE, and can be tuned based on a small corpus [5]. Two "inpainting" models take an input of two partial music clips, and output music that maintains the stylistic elements of the clips while musically transitioning between them [25, 11].

Ens and Pasquier's *Multi-track Music Machine (MMM)* [8] has several inpainting capabilities. MMM optionally takes as input a single MIDI clip, which may be single-track or multi-track. Depending on the user's interaction, MMM can create music without an input, add additional instrumental lines to an input clip, and replace user-selected musical content with similar musical content, that musically fits into the input piece's musical context.

We believe that one possible future application of providing human-interpretable musical parameters that can integrate into a composition process is to allow for some degree of parametric, affective control over the output of a co-creative generative music system. This approach could allow for affective control over a generative model without requiring a large, affectively tagged corpus of input music or formulaic definitions of musical parameters.

4.2.4 Musical features and associated emotional expression

As mentioned in Section 4.1, to produce a set of parameterically controlled musical clips, we first create a set of musical parameters to manipulate, which we call the *IsoVAT* guide. This guide is intended to be flexible enough to apply to a broad range of Western musical styles, while providing enough detail to be consistently applied when interpreted. Additionally, this guide is intended to be scalable with any instrumentation or degree of harmonic complexity. Section 4.3 further discusses this guide. To create the guide, we collate the results of surveys on musical features and their associated emotional expressions, and the results of cross-model surveys and studies of emotions.

We find two meta-reviews of research into Western musical features and associated affect [19, 17]. To achieve consensus, we include results only that are strongly present in both surveys. Gabrielsson and Lindström collect over 100 studies, differentiated by whether they are early studies using open-ended responses, multivariate listening studies, or post-2000 experimental studies. This survey does not translate affective models or terminology between each other, which means that an increase in tempo may correspond to both increased arousal and "excitement", a high-arousal emotion.

Livingstone and Brown survey 102 studies, translating results to the "2 Dimensional Emotion Space" or "2DES" model, developed by Schubert [31], which contains both categorical and dimensional emotion descriptors. This study uses data from surveys with different emotional models and descriptions of musical features, and translates them into a collated set of features and their associated expression within the 2DES model.

Importantly, these surveys directly use the musical terminology from their surveyed sources. As previously mentioned, one criticism of the broader MER field is the lack of internal consistency and coherence in the identification of musical features. To address this, we combine semantically related musical concepts into a single, unified vocabulary.

4.2.5 Collating results from various emotion models

As with musical terminology, previous MER feature-emotion surveys generally report emotional perception using the terminology and models of the surveyed source, which results in a lack of internal consistency and coherence in the literature. For example, three different musical features may have three different emotional perception associations in three different studies, while all referring to similar changes in music and similar emotional perception associations. As with musical vocabulary, we translate these emotional models into a single, dimensional VAT model.

To accomplish this, we examine studies that translate between affect models, both within and outside of musical contexts [6, 34, 14]. We identify 5 common categorical emotions with related dimensional mappings. While various studies use various scales (e.g. 1-5, 1-7, -5-5, 1-10), We normalize the data from these studies to a scale from -5-5, and average the normalized data to create a single value for each emotion, as shown in Figure 4.1. In our scale, a value of 0 indicates a neutral level of an affective dimension. As an example, we see that "Happiness", represented by a light-green diamond has a high valence value (>4), moderate arousal value (≈ 2.5), and moderately low tension value (≈ -2.5).

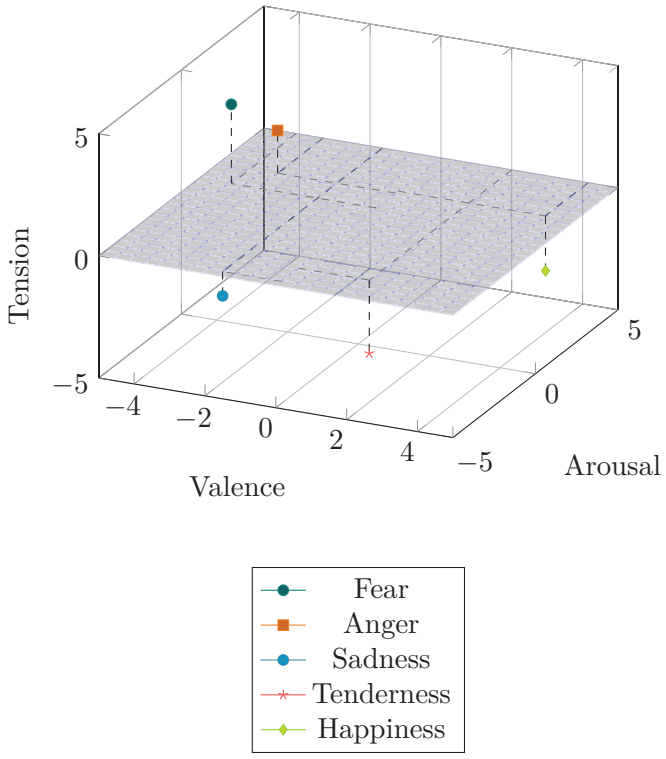


Figure 4.1: Discrete emotions placed in VAT space.

Gabrielsson and Lindström delineate the sources of their surveyed studies based on the experimental methodology, as described in Section 4.2.4. We include studies that use multivariate analysis or empirical experiments. Livingstone and Brown indicate which feature-

affect associations are found in at least 3 independent studies — we consider only the features that meet this quantity. We adjust for small differences in language, e.g. while Gabrielsson and Lindström may describe "Articulation connectedness", Livingstone and Brown instead differentiate "Articulation staccato" and "Articulation legato" as separate features. In such cases, we simplify to a single feature when possible, and therefore use "articulation connectedness". Other than combining feature descriptions, we attempt to stay as close as possible to the terminology used in the original surveys.

Figure 4.2a shows the dimensional mappings of composition-related features, and Figure 4.2b shows the mappings of performance-related features. We draw attention to the common trend in these results that all musical features have some correlative relationship with all three affective dimensions, indicating that musical features often produce multiple emotional correlations and expressions.

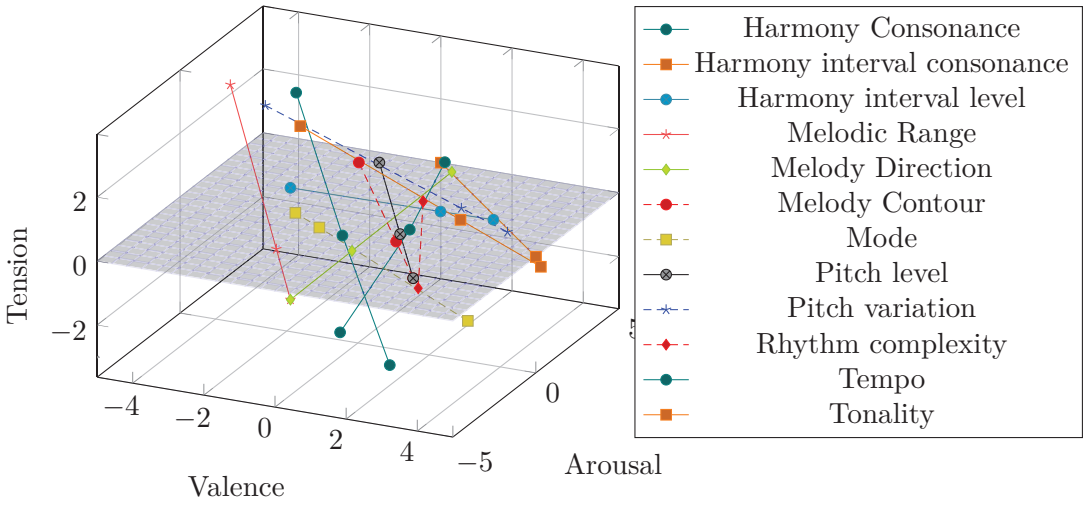
Figure 4.2 shows multiple directions to musically navigate the emotional space. To produce a composition guide for parameterically controlled emotional expression, we translate the data from Figure 4.1 one final time to an ordinal scale seen in Table 4.2.

4.3 The IsoVAT composition guide

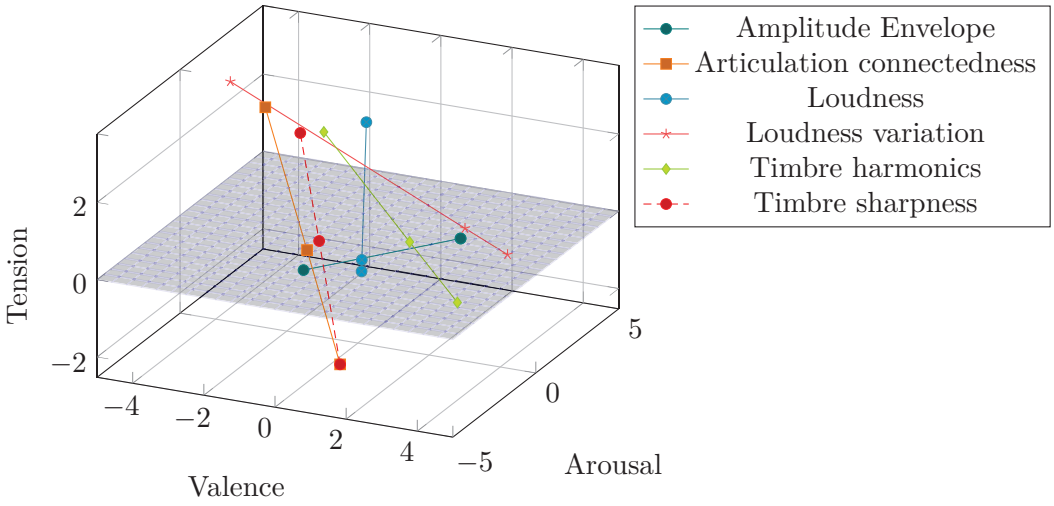
We collate the various MER sources into a central, unified model in both musical and emotional definitions, and present a set of feature-emotion mappings that is grounded in previous MER literature as much as possible, specific enough that various interpretations will result in relatively consistent emotional perceptions, and interpretable by a human composer during the composition task. The IsoVAT guide presents one of our contributions, as it centralizes findings from a wide range of MER topics and approaches, including controlled manipulations of individual musical features and multivariate analyses of full musical pieces.

Essentially the IsoVAT guide presents a method for applying MER research into human composition in a relatively controlled way. While there is interpretation required to realize the IsoVAT guide into music, it provides a higher degree of musical specificity and control than previous similar approaches [34, 33]. Because most MER research uses Western music, and primarily uses tonal music, the IsoVAT guide is primarily useful when composing tonal music. We note that the IsoVAT guide can be applied to hierarchical functional tonal music as well as non-functional tonal music.

The IsoVAT guide can be understood as a set of compositional *constraints*, to be used based on the intention of the composer to express particular emotions. Hasegawa describes how composers often integrate constraints into their compositional process [12], and gives several types of musical constraints. Of these, the IsoVAT is most well classified as a set of "relative material constraints". Hasegawa notes that composers are already often familiar with including relative material constraints in their composition process, as many of the "rules" of tonal music can be classified as such [12].



(a) Composition features



(b) Performance features

Figure 4.2: Musical features mapped to expressed affect.

Table 4.2 shows our composition guide, consisting of the identified feature-affect pairings. Table 4.2 reduces the data from Figure 4.2 into 3 ordinal relationships: Feature associated with a decrease in perceived affect (-), feature not associated with change in perceived affect (0), or feature associated with increase in perceived affect (+). We annotate the dimension that has the strongest affective association with an asterisk (*). This chart is based on findings from broad Western genres including popular and dance styles, classical, and folk, with popular styles being the most represented. We note that while a composer may include all musical features when modifying affect, they may also pick and choose individual features to manipulate, as the surrounding musical context or genre conventions may reduce the composers freedom to modify all features. While we collate data on musical features that are associated with performance, we do not manipulate these features in the IsoVAT corpus described in Section 4.4.

As an example of how this guide might be used, if the composer wishes to express an increasing amount of musical tension, decreasing the harmonic consonance, decreasing the interval pitch level (distance between pitches) and broadening the melodic range will express increasing tension. We use this guide to create sets of three clips that express differing levels of emotion, by manipulating these features in comparison to the other clips within the set. For example, a low-arousal clip may have a narrower melodic range, with a narrower melodic contour (moving in smaller intervals horizontally), that moves in a downward direction with lower pitch levels (tessitura), and narrower intervals in the accompanying harmony, compared to the moderate and high arousal clips within the set.

Table 4.2: Composition Guide for affective Western music.

Domain	Feature	Valence	Arousal	Tension	Description
Performance	Amplitude envelope roundness	0	0*		Increases as the amplitude envelope is more round/smooth
Performance	Articulation connectedness	+	-*	-	Increases as articulations are more connected/*legato*
Performance	Loudness	0	+*	0	Increases as overall volume increases
Performance	Loudness Variation	-*	0	+	Increases as volume level peaks have greater difference
Performance	Timbre Harmonics	+*	0	0	Increases with presence and strength of harmonics
Performance	Timbre sharpness	+	-*	-	Increases as higher harmonics are increasingly represented in timbre
Both	Tempo	0	+*	+	Increases as tempo increases
Composition	Harmonic consonance	+	-	-*	Increases as harmonies simplify — exact definition determined by genre
Composition	Interval consonance	+*	-*	0	Increases as melody uses simpler intervals — genre-dependent
Composition	Interval pitch level	0	+*	-	Increases as intervalic distance is increased
Composition	Melodic Range	-	+*	+	Increases as difference between high and low pitches in melody increases
Composition	Melodic direction	0	+*	0	Increases as melody motion is towards higher pitches
Composition	Melodic contour	0	+*	0	Increases as melody motion contains more internal distances between pitches
Composition	Mode	+*	0	0	Increases as mode is increasingly Major
Composition	Pitch level	0	+*	0	Increases as overall pitches are higher
Composition	Pitch variation	+*	0	0	Increases as distance of individual intervals increases
Composition	Rhythm complexity	0	0*	0	Increases as rhythms are moved away from standard “strong” beats such as 1 and 3
Composition	Tonality	+*	0	0	Increases as hierarchical tonal relationships are increasingly used

4.4 The IsoVAT corpus

4.4.1 Composition

Our composer and first author of this paper composes a total of 90 4-bar musical clips, which we call the *IsoVAT* corpus. Our composer has a background in music composition and live performance in an array of styles, ranging from classical, film, theatre, jazz, and rock. This background includes three years as a pianist and occasional band leader onboard luxury cruise ships, and 7+ years performing as a pianist and composer across the United States and Canada.

The duration of the clips was chosen to provide a single musical idea with a consistent emotional expression. Emotional perception of clips can be measured and modeled in two main ways: as a continuous time series, or as a classification [18]. When classifying individual clips of music with emotional perception, listeners are able to identify the expressed emotion with as little as 1 second of music [18, 7].

The *IsoVAT* corpus can be divided by emotional dimension, and further grouped into 10 sets of 3 per dimension. Each clip, within each set, is composed and labeled to express a low, middle, or high level of the expressed affective dimension, when compared to other two clips within the set [1].

We notate sets using the shorthand {Dimension}-{Number}, and clips using the shorthand {Dimension}-{Number}-{Clip}, where V-6 indicates valence set 6, and T-3-H describes the high-composed clip in tension set 3.

Each set of pieces shares an instrumentation and genre, drawing mostly from pop/rock, jazz, dance, and classical styles. For example, V-2 uses a single Disco ensemble for all 3 clips. We isolate a single affective dimension at a time in the IsoVAT corpus, and therefore do not require consistency of genre and instrumentation across dimensions. We include an example of well-known artists within each genre, and note that we do not attempt to mimic these artists, but they serve as examples of the target genre.

Each set is composed to primarily express affect by manipulating the composition-domain features identified in Table 4.2. We avoid manipulating features that are not strongly associated with the chosen dimension when possible. While we identify a set of music performance features, we only manipulate the composition features to produce our dataset. The genre, instrumentation, and examples of the target genre of each clip is provided in Table 4.3.

Figure 4.3 shows a score reduction for A-7, written for jazz ensemble in the swing/bebop genre, to provide an example of how our composition guide is used. All scores have been

[1]Both the original MIDI and rendered wav files are available on GitHub at https://github.com/CalePlut/IsoVAT_Dataset

Table 4.3: Genre and Instrumentation of IsoVAT corpus.

Valence

#	Genre	Instrumentation	Genre example
1	Classical	Solo piano	J.S. Bach
2	Disco	Disco ensemble	Earth, Wind, and Fire
3	Swing	Jazz combo	Glen Miller
4	Rock/Pop	Rock band	Rolling Stones
5	Piano Rock/Funk	Rock band (w. piano)	Stevie Wonder
6	Soft Rock	Rock band	Grover Wathington Jr.
7	60s Rock	Rock band	Creedence Clearwater Revival
8	Latin	Jazz combo	Guido Guidoboni
9	Ragtime	Solo piano	Scott Joplin
10	Film	Orchestra	John Williams

Arousal

#	Genre	Instrumentation	Genre example
1	Classical/Romantic	Woodwind quintet	Debussy
2	Rock/Pop	Rock band	AC/DC
3	Rock/Pop	Rock band	The Doors
4	Hard rock/Metal	Rock band	Metallica
5	Piano rock	Rock band(w. piano)	Billy Joel
6	Rock/Hard rock	Rock band	ZZ Top
7	Swing/Bebop	Jazz Combo	Dizzy Gillespie
8	Latin	Jazz combo	Bob Mintzer
9	Piano rock	Rock band(w. piano)	Elton John
10	Classical	Brass quintet	Malcom Arnold

Tension

#	Genre	Instrumentation	Genre example
1	Classical	Solo piano	Mozart
2	Classical	Brass quintet	Ligeti, Sousa
3	Surf rock	Rock band	The Surfaris
4	60s Rock	Rock band	Pete Townshend
5	Bluegrass	Bluegrass ens.	Foggy Mountain Boys
6	Europop	Electro/Synth	Haddaway
7	Rock/Pop	Rock band	Grateful Dead
8	Stadium rock	Rock band	Bon Jovi
9	Folk rock	Rock band	Dolly Parton
10	Choral	SATB+Piano	Eric Whitacre

written in the MuseScore 3 notation software[2]. The composition features where arousal is the strongest emotional association are: Interval consonance, interval pitch level, melodic range,

[2]MuseScore 3 is available at https://musescore.org

Table 4.4: Arousal-manipulating features as manipulated in arousal set 7.

Label	Int. consonance	Int. pitch level	Mel. range	Mel. dir.	Mel. contour	pitch level
Low	Mostly consonant	Steps, thirds	10 semitones	1 peak	Smooth, small	F3-E♭4
Mid	Generally consonant	Steps, thirds, fourths, fifths, octaves	12 semitones	2 peaks	Smooth, wide	B♭4-B♭5
High	Mix	Steps, thirds, fourths, tritones fifths, sixths, sevenths	16 semitones	6 peaks	Jagged, wide	E♭3-G4

melodic direction, melodic contour, and pitch level. We manipulate all of these features in A-7, as described in Table 4.4.

In Table 4.4, the high clip uses a mix of dissonant and consonant intervals, ranging from moving by half steps to moving by minor sevenths. The high clip melody has a total range of 16 semitones, from E♭3-G4. To measure the direction, we describe the number of melodic peaks, or the number of times the melody changes direction. In the case of the high clip, the melody has 6 melodic peaks. The contour of the high clip is jagged, with many direction changes involving large leaps.

4.4.2 Audio rendering and interpretation

MIDI represents music as data that must be synthesized to produce audio. When a MIDI file is played, the sounds are determined by a sample-based soundfont. Soundfonts can be used to replicate the synthesis of a MIDI file consistently between computers. We use the "Arachno" soundfont[3], to synthesize the IsoVAT corpus into an audio format that will be consistent across listeners.

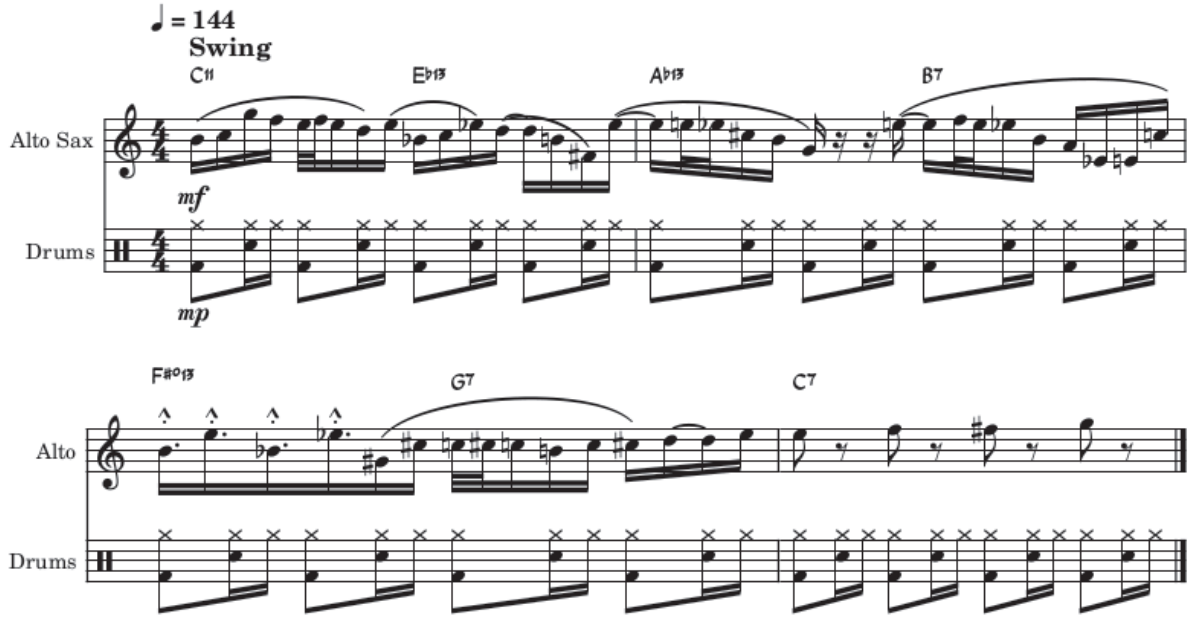
4.5 Ground truthing experiments

Each clip the IsoVAT corpus is labeled with the intended emotional expression compared to the other two clips in the set, e.g. "high", "medium", or "low". To evaluate the composition guide, we ground-truth order the dataset by empirically labeling the emotional perception that listeners report for each clip in varying degrees of musical context.

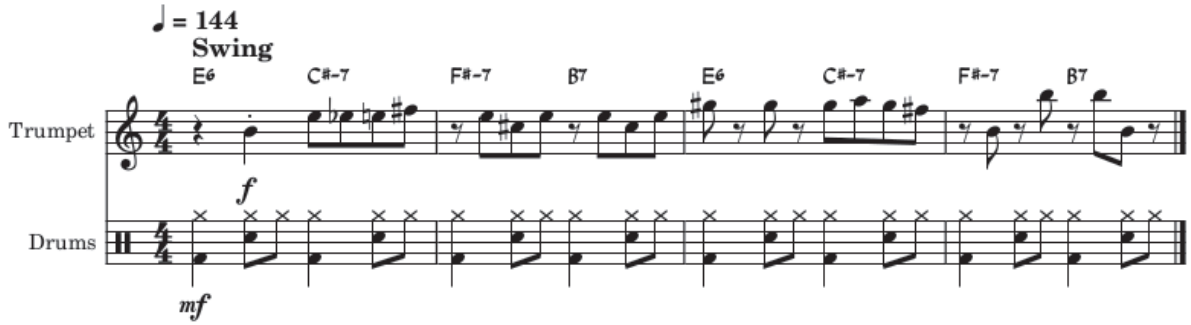
The primary difference between the three study designs that we use is the musical context for each clip. In the 2-rank design discussed in Section 4.5.1, clips are heard in their full composed context. In the 1-rank design in Section 4.5.1, clips are heard with one contextual clip, but not the other. In the Likert-type design in Section 4.5.1, clips are completely removed from their musical context, and instead rated on an absolute scale of emotion.

Across all empirical study designs, we collect 30 rankings per clip. Participants are recruited, and the study is performed, using Amazon's Mechanical Turk platform. MTurk does not provide, nor do we collect, additional demographic information. Consent is obtained

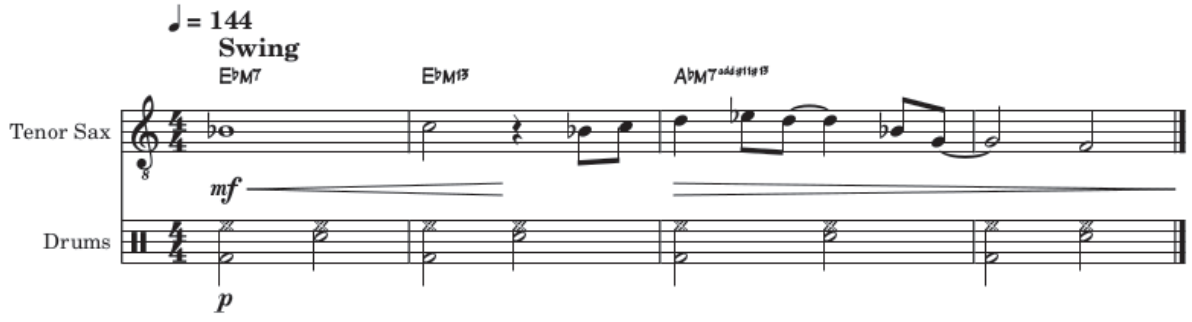
[3]Arachno soundfont available at http://www.arachnosoft.com/main/soundfont.php



(a) High



(b) Mid



(c) Low

Figure 4.3: Reduced scores for arousal set 7.

prior to participation. Participants may participate in each study design only once per dimension, but may participate in multiple study designs. Participants take an average of 20 minutes to complete all ranking tasks.

4.5.1 Empirical methodology

As composed: "2-rank"

Each participant answers a total of 10 questions relating to a single affective dimension per task. 90 participants are paid US$0.10/ranking ($1.00/task). Consent is obtained before each task. Participants are provided a description of the randomly selected affective dimension that they will evaluate. One set of 3 clips is randomly selected to provide an example of low, medium, and high levels of the assigned dimension.

Participants complete 9 ranking tasks, with the remaining sets. Each task involves listening to 3 musical clips, and identifying the clips that express the highest and lowest level of their dimension. To ensure participant accuracy, an additional audio file that consists of a voice instruction to select a particular response is included. Participants who fail to correctly follow the speech instructions are removed from the study.

Pairwise "1-rank" of 2 clips

In the "1-rank" study, 180 participants are asked to perform a single ranking between 2 clips for each question, rather than selecting a low and high. Participants listen to 15 pairs of music, including an example. For each question, participants listen to two clips, and select the clip that they believe expresses their assigned dimension more strongly. The clips are drawn from a random selection within 5 sets, with participants performing 3 rankings per set, for a total of 15 pairwise combinations of subsets. The order of the clips, subsets, and sets is randomized for each participant.

This 1-rank design provides some musical context for each ranking, as each clip is evaluated compared to a single other clip. However, it does remove part of the contextual musical information, as clips are composed in sets of 3.

Individual Likert scale rating

We evaluate the corpus via a single 7-point Likert scale. For this study, 180 participants listen to three example sets, arranged in sets of 3. Participants then listen to 14 individual clips, drawn randomly from the corpus, and provide a single rating from 0 (expresses a very low level of affective dimension) to 7 (expresses a very high level of affective dimension).

This design evaluates each clip in isolation, completely out of their composed context. While we are primarily investigating the parameterization as a contextual, ordinal guide, we expect that when removed entirely from context, participants will use a more absolute scale that will somewhat align with the contextual composition.

Table 4.5: Means and standard deviations in confusion matrices for 2-rank study results.

Ground truth order

(a) Valence

Label \ Response	Low	Mid	High
Low	0.59 ± 0.15	0.28	0.13 ± 0.07
Mid	0.27 ± 0.08	0.48	0.25 ± 0.07
High	0.16 ± 0.07	0.25	0.56 ± 0.15

(b) Arousal

Label \ Response	Low	Mid	High
Low	0.64 ± 0.10	0.29	0.07 ± 0.05
Mid	0.28 ± 0.10	0.57	0.15 ± 0.07
High	0.07 ± 0.05	0.17	0.76 ± 0.12

(c) Tension

Label \ Response	Low	Mid	High
Low	0.66 ± 0.13	0.27	0.07 ± 0.10
Mid	0.25 ± 0.10	0.56	0.19 ± 0.11
High	0.08 ± 0.07	0.19	0.73 ± 0.20

Composed Labels

(d) Valence

Label \ Response	Low	Mid	High
Low	0.44 ± 0.21	0.24	0.32 ± 0.23
Mid	0.22 ± 0.17	0.41	0.37 ± 0.21
High	0.21 ± 0.12	0.40	0.39 ± 0.23

(e) Arousal

Label \ Response	Low	Mid	High
Low	0.63 ± 0.11	0.30	0.07 ± 0.06
Mid	0.28 ± 0.12	0.62	0.10 ± 0.07
High	0.12 ± 0.08	0.12	0.76 ± 0.12

(f) Tension

Label \ Response	Low	Mid	High
Low	0.60 ± 0.20	0.32	0.08 ± 0.11
Mid	0.23 ± 0.15	0.54	0.23 ± 0.21
High	0.15 ± 0.19	0.17	0.68 ± 0.25

4.6 Empirical results

4.6.1 2-rank

Results are analyzed to view the inter-rater agreement and ranked order of each set. Participants agree with the derived ground-truth order of all three clips in a set 56–76% of the time. Agreement between composed labels and responses are between 39–76% compared to a random chance of 33%. 20 out of 30 sets are ground-truth ordered with the same labels as composed, with 6 valence sets, one arousal set, and 3 tension sets showing disagreement between composed labels and ground-truthed order. Shapiro-Wilk and Anderson-Darling tests are performed across participant data. In the 2-rank responses, participant responses demonstrate a normal distribution for all three dimensions.

Table 4.5 shows confusion matrices for responses as ground-truth ordered, and in their composed order, showing means and standard deviations for the low and high-selected clips. Because participants only select the low and high clip, the mid-level means are inferred. For example, we can see that for valence, clips labeled as the highest are ranked as the lowest clip 16% of the time, with a standard deviation of 7%. Figure 4.4 presents the same data without the inferred middle values.

4.6.2 1-rank

Participants agree an average of 58.5% of the time across all 30 sets and pairwise comparisons. Divided by dimension, these agreements are valence: 59.2%, arousal: 59.4%, and tension: 56.8%. Participants agree with the composed rankings 43.0% for valence, 48.3% for arousal, and 49.6% for tension. Table 4.6 shows the agreements by dimension and pairwise comparison. As in the 2-rank study, participant responses are normally distributed among answers.

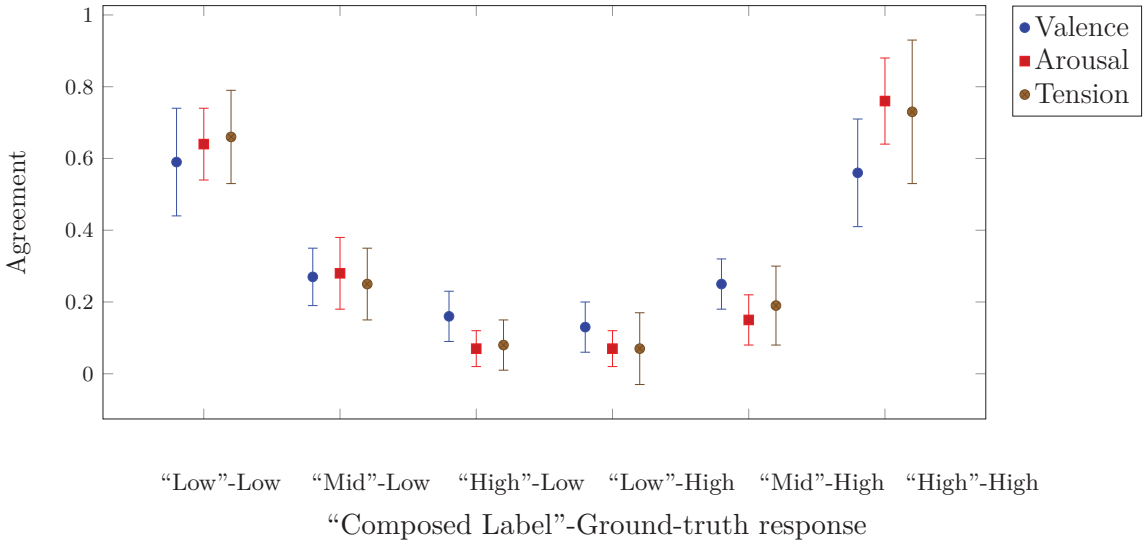


Figure 4.4: Means and standard error for each clip's ground-truth label-response pair.

Table 4.6: Agreement values from 1-rank study for ground-truthed order and composed labels.

Dimension	Labels	Ground truth		Composed labels	
		M	SD	M	SD
Valence	H-L	0.60	0.07	0.43	0.11
Valence	H-M	0.58	0.05	0.44	0.07
Valence	M-L	0.59	0.07	0.42	0.08
Arousal	H-L	0.58	0.52	0.49	0.10
Arousal	H-M	0.59	0.05	0.53	0.10
Arousal	M-L	0.61	0.07	0.43	0.11
Tension	H-L	0.55	0.05	0.51	0.07
Tension	H-M	0.56	0.04	0.49	0.07
Tension	M-L	0.59	0.06	0.49	0.11

We draw attention in Figure 4.6 to the lack of additional clarity in the comparisons of High-Low pairs compared to the intermediary comparisons. The high and low clips are expected to express the ends of an affective dimension, controlled for other musical factors, and we expect these end points to be more clearly differentiated than when one clip expresses a moderate level of the affective dimension.

4.6.3 Individual Likert-like rating

While Likert scales are commonly analyzed as interval data, we follow suggestions to treat Likert scales as ordinal data [37]. We compute the median absolute deviation for each clip. As with the other study designs, responses are normally distributed. We compute Cohen's kappa for each set of three clips to determine whether at least one clip within the set

expresses a significantly different emotional level than the other members of the set. These statistics are shown in Table 4.7 for each set.

In terms of agremeent with the composed labels, 1 valence set, all 10 arousal sets, and 5 of the tension sets agree with the composed labels when ties are broken towards the ground-truth order from the 2-rank study. When ties are broken towards composed order, 4 valence sets, 10 arousal sets, and 7 tension sets agree with the composed order. Overall, this means that when ties are broken towards the ground-truth order, Likert data agrees with composed order in 16 sets, and when ties are broken towards the composed order, Likert data agrees with the composed order in 21 sets. This data does not include the 3 excluded example valence sets, as discussed below.

In terms of showing significant differences within Likert rankings, 2 sets expressing valence levels, all 10 sets expressing arousal levels, and three sets expressing tension levels show significant differences. Of these sets, 8 arousal sets and one tension set show significant differences and agree with the composed set.

Our Likert data does not include 3 valence sets — we use data from previous evaluations of the corpus to select example sets that provided the most clear data. Due to only three sets of valence clips matching composed labels in the 2-rank design, we do not collect data on these example clips. For arousal and tension, example clips could be included without reducing the number of evaluated clips, as there are enough sets that match the composed label that we randomly sample from possible example clips, and gather data on the others. We note that this creates a further reduction in the accuracy and agreement of the valence clips in this design. Our Likert data also does not have information on the lowest expression of each dimension within set 1, due to a coding error.

4.6.4 Aggregating ground truth from multiple studies

Trends between studies are collated to produce a new ground-truthed order for each set of clips. Only one set for each dimension, coincidentally set 10, have a ground-truth order that is consistent across all three studies. For valence set 10, the ground-truth order is different than composed. Because there is only agreement between all three studies in $<3\%$ of the corpus, we apply a standard of requiring agreement between at least two ground-truth study designs. In the event of a tie within the Likert-like responses, the ground-truth 2-rank order breaks the tie. The valence sets that serve as an example for the Likert study are assumed to agree with the 2-rank ground-truth order for the purposes of deriving a aggregate ground-truth order.

Our collated results are presented in Table 4.8. Results are given with composed labels, the ground-truth orders derived from each study design, and the aggregate derived ground-truth order. Green $\star$ cells indicate agreement with the composed labels, blue $\diamondsuit$ cells represent ground-truth agreement in disagreement with composed labels, yellow $\square$ cells represent disagreement in both the ground-truth order and composed labels, and red cells with

Table 4.7: Medians, Abs. dev, and Chi Square tests for Likert-like responses.

Valence						
#		Low	Mid	High	ChiSquare	p
1	Med		3	3	0.06	0.79
	Abs err		1	1		
2	Med	5	4	5	1.11	0.57
	Abs err	1	1	1		
4	Med	4	5	5	17.17	<0.01*
	Abs err	1	1	1		
5	Med	4	4	3	0.26	0.87
	Abs err	1	1	1		
6	Med	4	4	4	0.65	0.72
	Abs err	1	1	1		
8	Med	5	5	4	13.19	<0.01*
	Abs err	1	1	1		
10	Med	5	5	6	3.29	0.19
	Abs err	1	1	1		
Arousal						
#		Low	Mid	High	ChiSquare	p
1	Med		2.5	4	4.80	0.03*
	Abs err		1	1		
2	Med	4	4	6	24.67	<0.01*
	Abs err	1	1	1		
3	Med	3	3.5	5	5.96	0.05*
	Abs err	1	0.5	1		
4	Med	4	4	6	30.92	<0.01*
	Abs err	1	1	1		
5	Med	3	3	4	9.94	<0.01*
	Abs err	1	1	1		
6	Med	4	4	6	17.39	<0.01*
	Abs err	1	1	1		
7	Med	4	4.5	7	13.21	<0.01*
	Abs err	1	0.5	1		
8	Med	4	4	5	28.70	<0.01*
	Abs err	1	1	1.5		
9	Med	3	3	4.5	12.33	<0.01*
	Abs err	1	1	1.5		
10	Med	3	4	4	8.05	0.02*
	Abs err	1	1	1		
Tension						
#		Low	Mid	High	ChiSquare	p
1	Med		3	5	3.55	0.06
	Abs err		1	2		
2	Med	4	4	4	0.54	0.76
	Abs err	1	1	1		
3	Med	3.5	4	5	6.29	0.04*
	Abs err	0.5	1	1		
4	Med	4	4	5	2.72	0.26
	Abs err	1	1	1		
5	Med	3	5	6	5.95	0.05*
	Abs err	1	1	1		
6	Med	5	4.5	4	0.35	0.83
	Abs err	1	1.5	1		
7	Med	4	5	4	2.17	0.33
	Abs err	1	1	1		
8	Med	4	4	4	1.31	0.51
	Abs err	1	1	1		
9	Med	4	6	5	8.89	0.01*
	Abs err	0.5	1	1		
10	Med	4	4	6.5	5.63	0.06
	Abs err	1	1	1		

either $\triangle$ or $\triangledown$ indicate an irreconcilable loop in the collected pairwise comparisons. Irreconcilable loops may occur as forwards loops with ground-truth ranking of H->M->L->H, indicated by $\triangle$, or reverse loops as L->M->H->L, indicated by $\triangledown$. While neither loop type can be turned into a ground-truth order, the reverse loop is the more serious change from the composed order, as only a single pairwise ranking is correct. In a loop, only a single pairwise ranking is incorrect.

As an example for reading Table 4.8, V-4 has an order of M->H->L in both the Likert and 2-rank responses, and an order of M->L->H in the 1-rank order. The Likert order shows a significant difference between at least one of the clips and the rest of the set, and contains a tie between the Medium and High clips. This tie is broken in favour of the 2-rank order, which aligns the Likert order with the 2-rank order, producing an overall ground-truth order of M->H->L.

V-2 is the only set that exhibits no agreement across at least two designs. In the remaining 29 sets, 27 of the ground-truth orders are derived from agreement between the 2-rank and Likert orders. Tension sets 6 and 9 derive their ground-truth order from agreement between the 1-rank and Likert orders. Both of these sets' 2-rank ground-truth order agrees with the composed labels.

Two sets, V-8 and T-6 are ground-truthed in the reverse order compared to the composed labels. Three sets demonstrate "major" differences in order, where the low or high clip is re-ordered to be on the opposite end of the order. Five sets demonstrate "minor" differences in order, where the low or high clip is swapped with the medium clip. In no sets is the composed "low" clip ranked as the highest without also reversing the "high" and "mid" order.

4.7 Musical analysis of potential confounds

We compose the IsoVAT corpus based on our composition guide, and therefore evaluate the guide based on the results of the ground-truth experiments for the corpus. We musically analyze the sets that do not agree with the composed labels, as well as the set that does not have a ground truth consensus. We identify four features that are found in pieces whose ground truth order disagrees with the composed labels.

In addition to the identified possible musical confounds, some of the variance in our results may be a result of participants being unfamiliar with Western music. As mentioned in Section 4.5, we do not screen participants for familiarity with Western music, and we do not collect demographic or location information. Participants may therefore be from regions or locations where Western music is not dominant. Because of the widespread, global reach of Western music [36], we expect that participants will have some familiarity with Western music, particularly in terms of exposure and experience hearing popular Western styles. While non-Western listeners generally bring local cultural readings to Western art and music [3],

Table 4.8: Central comparison of ground truth order by study design and final ground-truth order, see Section 4.6.4.

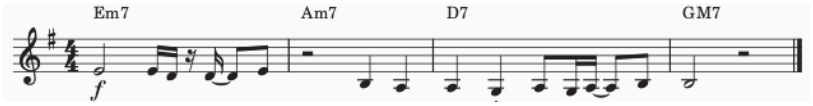
#	Composed labels	2-rank order	1-rank order	Likert Order	Aggregate G-T Order
			Valence		
1	H-M-L	(H-M)-L ⋆	L-M-H □	(H-M) ⋆	H-M-L⋆
2	H-M-L	M-L-H □	L-M-H □	(L-H)-M □	—
3	H-M-L	H-M-L ⋆	L-H-M □	Example —	H-M-L⋆
4	H-M-L	M-H-L ♢	M-L-H □	(M-H)-L* ♢	M-H-L♢
5	H-M-L	M-L-H ♢	Reverse Loop ▽	(M-L)-H ♢	M-L-H♢
6	H-M-L	M-L-H ♢	L-H-M □	(M-L-H) ♢	M-L-H ♢
7	H-M-L	H-M-L ⋆	M-L-H □	Example —	H-M-L⋆
8	H-M-L	L-M-H ♢	H-L-M □	(L-M)-H* ♢	L-M-H♢
9	H-M-L	H-M-L ⋆	M-L-H □	Example —	H-M-L⋆
10	H-M-L	H-L-M ♢	H-L-M ♢	H-(L-M) ♢	H-L-M♢
			Arousal		
1	H-M-L	H-M-L ⋆	L-H-M □	H>M* ⋆	H-M-L⋆
2	H-M-L	H-M-L ⋆	M-H-L □	H-(M-L)* ⋆	H-M-L⋆
3	H-M-L	H-M-L ⋆	Loop △	H-M-L ⋆	H-M-L⋆
4	H-M-L	H-M-L ⋆	H-L-M ♢	H-(M-L)* ⋆	H-M-L⋆
5	H-M-L	H-M-L ⋆	H-L-M ♢	H-(M-L)* ⋆	H-M-L⋆
6	H-M-L	H-M-L ⋆	Loop △	H-(M-L)* ⋆	H-M-L⋆
7	H-M-L	H-M-L ⋆	H-L-M □	H-M-L* ⋆	H-M-L⋆
8	H-M-L	H-L-M □	L-H-M □	H-(M-L)* ⋆	H-M-L⋆
9	H-M-L	H-M-L ⋆	R. Loop ▽	H-(M-L)* ⋆	H-M-L⋆
10	H-M-L	H-M-L ⋆	H-M-L ⋆	(H-M)-L ⋆	H-M-L⋆
			Tension		
1	H-M-L	H-M-L ⋆	H-L-M □	H-M ⋆	H-M-L⋆
2	H-M-L	H-L-M ♢	L-M-H □	(H-L-M) ♢	H-L-M♢
3	H-M-L	H-M-L ⋆	M-H-L □	H-M-L ⋆	H-M-L⋆
4	H-M-L	H-M-L ⋆	H-M-L ⋆	H-(M-L) ⋆	H-M-L⋆
5	H-M-L	H-M-L ⋆	L-M-H □	H-M-L* ⋆	H-M-L⋆
6	H-M-L	H-M-L ⋆	L-M-H ♢	L-M-H ♢	L-M-H♢
7	H-M-L	M-H-L ♢	R. Loop ▽	M-(H-L) ♢	M-H-L♢
8	H-M-L	(M-L)-H ♢	L-M-H □	(M-L-H) ♢	M-L-H ♢
9	H-M-L	H-M-L ⋆	M-H-L ♢	M-H-L* ♢	M-H-L ♢
10	H-M-L	H-M-L ⋆	H-M-L ⋆	H-(M-L) ⋆	H-M-L⋆

non-Western listeners are often able to identify expressed emotion in Western music [2, 9]. Additionally, Cespeded-Guevara and Eerola suggest that dimensional models of affect are well-suited to describing music emotion perception in cross-cultural environments [4].

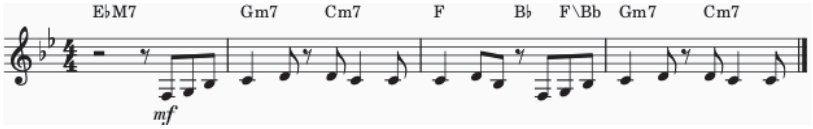
4.7.1 Sequences

V-5, V-6, and V8 are ground-truth ordered with the composed mid clip being moved to the high position. V-5-M and V-6-M are shown in Figure 4.5. These clips use a falling-fifths progression, starting in minor, changing the sonority outlined by each chord to fit in the mode. V-6-M begins on an EbM7 chord to allow for the melodic pickup. V-8-M uses a slightly more intricate i-vii^{o7}/VI-VI-VI^{+9} sequence.

Tonality, pitch variation, mode, and interval consonance are the features primarily associated with valence. In these sets, we begin our sequence on a minor chord, and assume that alternating the sonority expressed in each chord between minor and major would provide modal ambiguity, expressing a moderate level of valence. While these sequences could resolve to Major or minor end points, we believe that participants identified the tonal centre of each sequence as Major. This may indicate that the harmonic motion in 4th and 5th based sequences may be primarily perceived as an increase in tonal hierarchies, associated with positive valence.



(a) V-5-M.



(b) V-6-M.

Figure 4.5: Reduced score for V-5-M and V-6-M.

4.7.2 Harmonic complexity as dissonance

Pieces using complex Major harmonies such as Major 7ths, 9ths, and 13ths tend to be ground-truth ordered in disagreement with the composed labels. V-5-H, V-6-H, and V-8-H use Major 7ths and 9ths, and are ground-truth ordered in the lowest position. V-8-H is shown in Figure 4.6. V-4-H contains Major 7ths, but fewer other complex harmonies than sets 5, 6, and 8, and is ground-truth ordered in the middle position.

While these features mainly affect valence, harmonic complexity may also affect perception of tension. In T-7, the medium and high-composed clips are swapped in the ground truthing. The mid-composed clip uses a half-diminished 7th chord in place of an expected V chord, to disrupt an otherwise stable vi-iv-V progression, while the high-composed clip

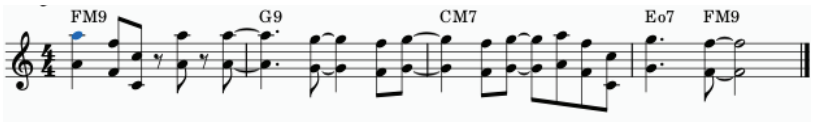


Figure 4.6: Reduced score for V-8-H.

uses unresolved suspended 4ths and dominant 7ths in an outlined V chord. The jarring chord substitution with a less stable chord may have overpowered the tension building from unresolved dominant-tonic motion. Harmonic complexity may also be more vulnerable to cross-cultural effects, as the most widespread and popular Western music is comparatively harmonically simple.

4.7.3 Density

T-7-H and T-8-H use short, uneven chords that move towards an unresolved dominant chord. In both sets, clips with longer, more sustained notes are ranked higher in perceived tension. While silence and uneven rhythms are often used to create tension in film scores, lowered density in pop music may be directly associated with lower tension. This relationship is not entirely consistent across ground truth designs, and this association may be weak.

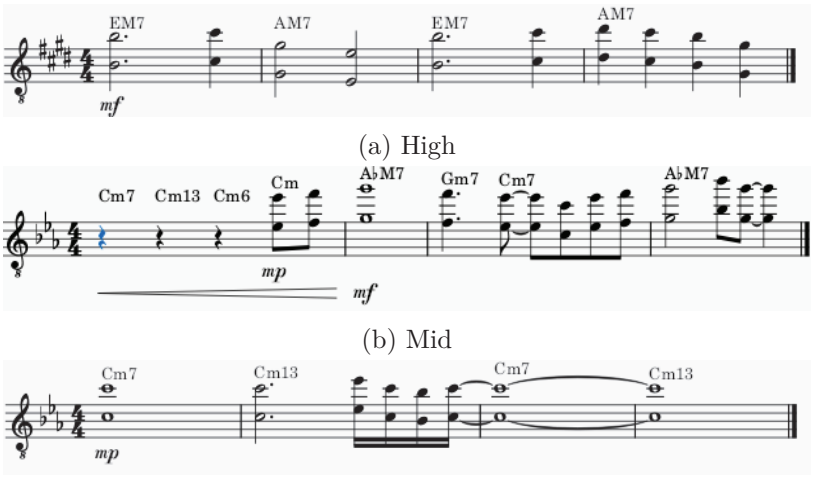
A similar effect occurs in V-10 between the low- and medium-composed pieces. The V-10-L is a fast, aggressive, minor piece, while V-10-M is slower and uses more ambiguous and shifting harmonies and orchestrations.

4.7.4 Genre

T-2 and T-6 are ground-truth ordrered in disagreement with the composed labels. In T-6, the ground truth ranking is based on an agreement between the 1-rank and likert order, with the 2-rank order agreeing with the composed labels. T-6 appears to be confounded by strict adherence to triad-based harmonies in "Europop". This genre mostly uses simple harmonies and consistent rhythms, which limits the dissonances that can be used without violating genre conventions. The lowered expressive range may have produced too little difference between the component pieces to produce a consistent ranking.

T-2 appears confounded for the opposite reason — the genre of "classical" is broad enough that sub-genre differences may create additional confounds. T-2-L is stylistically similar to a Sousa march, emphasizing I-V tonal relationships with triads, while T-2-M is much more harmonically complex, and uses more inversions to create rising lines with some dissonances (e.g. a $Vsus\frac{6}{4}$ chord), with a more contemporary, impressionistic style. These subgenre differences may confound the perception of tension.

Features discussed as previous possible confounds also often occur as part of genre conventions. For example, Disco and Latin music commonly makes heavy use of Major 7th and 9th chords. Bridge sections with instrumental breaks are common in rock/pop to



(a) High

(b) Mid

(c) Low

Figure 4.7: Reduced scores for valence set 2 High, Middle, and Low clips.

build excitement towards a final chorus. Genre conventions most commonly affect tension, possibly due to the shifting definition of "dissonance" in different genres.

4.7.5 Set without ground truth order

V-2, the first set to be composed, is the only set that receives a different ranking in all three of the listener evaluations, and is shown in entirety in Figure 4.7. V-2 is Disco-genre, and contains genre-specific complex Major harmonies as well as sequences.

4.8 Discussion

Unsurprisingly, the study design that most directly evaluates the composed order of the corpus produces the most agreement with the composed order, and between participants, with a maximum agreement of 76%. The study design that demonstrates the most variance in responses is the 1-rank design, achieving a maximum agreement of 61%. Also in the 1-rank design, the composed labels do not always outperform random chance. We believe this is primarily because of the musical and emotional context that is removed when evaluating clips pairwise, particularly given the compositional intent as a set of 3.

Trends are primarily shared between the 2-rank and Likert design. Clips in the arousal sets have the highest inter-rater agreement, and agreement with the composed order. Tension is generally less agreed upon than arousal, but more agreed upon than valence. Valence is the least agreed upon dimension. Surprisingly, these emotional dimension trends are reversed in parts of the 1-rank design, though we reiterate that the trends from the 1-rank design are far smaller and more varied than in the other two designs.

The dimension of valence shows the most variance in the 2-rank and likert designs. When the ground truthed order of clips disagrees with the composed labels, the features that may

most confound valence are the use of sequences that may be interpreted as moving towards a Major destination with tonal hierarchies, and the use of complex Major chords such as 7ths and 9ths.

Responses for arousal show the most agreement in the 2-rank and Likert designs. This is consistent with previous MER. In the Likert design, A-4, A-5, and A-8 show a clean differentiation between the high-composed clip and the medium/low-composed clips, but the medium and low-composed clips show nearly identical values.

Musical density, and the interaction between genre conventions and harmonic complexity may confound the perception of tension. While "dissonance" is often described as a clash between notes, the specific intervals or chords that are considered dissonant depends on features such as genre and historical context — in an extreme example, early organum choral music only considers octaves, fourths, and fifths as "consonant" [28].

Following the IsoVAT composition guide produced music with perceived emotional trends. However, there is still a fair amount of ambiguity in the descriptions and relationships between these features and their affective expressions. For example, as discussed in Section 4.7, sequences may be heard outlining a tonal centre rather than shifting chord sonorities — which musical feature is heard as more dominant, and therefore which emotional perception will be heard cannot necessarily be determined from the guide or our analysis of the dataset alone.

4.9 Conclusion

We present a composition guide in Section 4.3 for composing affective music using valence, arousal, and tension, based on previous MER. Our guide is based on collated consensus data involving the mapping of musical features and affective expression, as well as on the mappings of affective models to one another. This guide presents a set of musical features and the ordinal emotional expressions associated with changes in those features. This guide is intended to allow composers to exert some degree parameteric control over their composition, while integrating into common composition processes.

We use our guide to compose a corpus of 90 musical clips that express emotion based on this guide, as described in Section 4.4. Our corpus uses a variety of musical genres, allowing us to additionally evaluate the generalizability of our composition guide across styles. We empirically produce a ground-truth ranking of the emotional perception of 29 out of 30 sets, with 19 sets ground-truthed in the order as labeled.

Overall, we address several identified issues in previous MER, mainly related to the broad range of musical features and emotional descriptions in use. Previous music composed for research generally focuses on sounding natural, or informally manipulates musical features to express emotion. This composed music also generally is composed for only a melodic

line, or a single instrument, within a single musical genre. Previous MER corpora most commonly use audio representation, which limits the extractable compositional features.

We collate previous findings into ordinal relationships between a set of musical features and a 3-dimensional VAT affect model. We use this composition guide to compose sets of 4-bar MIDI clips that express low, medium, and high levels of a given affective dimension. These clips are ground-truthed by three empirical designs, and we find support for our collated composition guide. Overall, we create a musical corpus that is built following the collected findings from previous MER, and evaluate the perceived affect of the corpus. We find support for the practical applicability of MER in Western music.

4.10 Future work

While our composition guide provides a general set of guidelines for affective composition, the inclusion criteria was primarily determined by the source materials. This leads to ambiguity in the relationship between closely-related features such as "tonality" and "mode", or "melodic direction" and "melodic contour". As Eerola and Vuoskoski suggest, we suggest future research into isolating and controlling these particular relationships in future MER [7].

We do not analyze the emotional expressions in the corpus along dimensions that are not manipulated — we only evaluate the manipulated emotional perception. Given that all identified musical features show correlative relationships in multiple dimensions, cross-dimensional validation of emotional expression may produce increased clarity as to the effects of our musical manipulations. Furthermore, as the bounds of manipulation are based on our composition guide, cross-dimensional ground-truthing may serve to further evaluate the guide and corpus.

One element of the IsoVAT guide that could be further explored is its ability to describe dynamic emotion in music, as mentioned in 4.4.1. Because the IsoVAT guide provides the data in an ordinal form, it may be applied to compose discrete clips that express relative emotion levels to each other, as well as to compose music that expresses relative changes in emotion over time.

In addition to evaluating the IsoVAT composition guide, the IsoVAT corpus could be used to assist in affectively tagging larger, curated datasets. As discussed in Section 4.2, parameterically controlled composition and ground-truthing are recommended when selecting stimulus for MER. When training ML systems from user feedback, the use of clear examples and known "gold standards" is recommended to ensure accuracy in participant responses — the ISOVat corpus provides a set of parameterically controlled and ground-truthed clips that express particular emotional relationships.

As we mention in Section 4.2.3, one potential application of the IsoVAT guide is to be used to provide some control over an input corpus for small-batch co-creative generative mu-

sic. As the IsoVAT dataset is already composed based on the guide and has a ground-truth order, one immediate possible future project is to evaluate how much emotional expression is maintained when the IsoVAT dataset is used as an input for a generative system. Additionally, further examination of the IsoVAT guide could explore its use in parametric co-creative generative processes.

Bibliography

[1] Anna Aljanaki, Frans Wiering, and Remco Veltkamp. Collecting annotations for induced musical emotion via online game with a purpose emotify, 2014.

[2] Laura-Lee Balkwill and William Forde Thompson. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music perception*, 17(1):43–64, 1999.

[3] Russell Belk. Out of sight and out of our minds: What of those left behind by globalism? In *Does Marketing Need Reform?: Fresh Perspectives on the Future*, pages 217–224. Routledge, 2015.

[4] Julian Cespedes-Guevara and Tuomas Eerola. Music communicates affects, not basic emotions–a constructionist account of attribution of emotional meanings to music. *Frontiers in psychology*, 9:215, 2018.

[5] Monica Dinculescu, Jesse Engel, and Adam Roberts, editors. *MidiMe: Personalizing a MusicVAE model with user data*, 2019.

[6] Tuomas Eerola and Jonna K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, jan 2011.

[7] Tuomas Eerola and Jonna K. Vuoskoski. A review of music and emotion studies: Approaches, emotion models, and stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, 2012.

[8] Jeff Ens and Philippe Pasquier. MMM: exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*, 2020.

[9] Thomas Fritz, Sebastian Jentschke, Nathalie Gosselin, Daniela Sammler, Isabelle Peretz, Robert Turner, Angela D. Friederici, and Stefan Koelsch. Universal recognition of three basic emotions in music. *Current biology*, 19(7):573–576, 2009.

[10] David Gerhard and Daryl H. Hepting. Cross-modal parametric composition. In *ICMC*. Citeseer, 2004.

[11] Gaëtan Hadjeres and Léopold Crestel. The piano inpainting application. *arXiv preprint arXiv:2107.05944*, 2021.

[12] Robert Hasegawa. Creating with constraints. In *The Oxford Handbook of the Creative Process in Music*. Oxford University Press, 2020.

[13] Carlos Hernandez-Olivan and Jose R. Beltran. Music composition with deep learning: A review. *arXiv preprint arXiv:2108.12290*, 2021.

[14] Holger Hoffmann, Andreas Scheck, Timo Schuster, Steffen Walter, Kerstin Limbrecht, Harald C. Traue, and Henrik Kessler. Mapping discrete emotions into the dimensional space: An empirical approach. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3316–3320. IEEE, 2012.

[15] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. *arXiv preprint*, 2021.

[16] Patrik N. Juslin and John Sloboda. *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[17] Patrik N. Juslin, John Sloboda, Alf Gabrielsson, and Erik Lindström. *The role of structure in the musical expression of emotions*, page 367–400. Oxford University Press, 2012.

[18] Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *International Society for Music Information Retrieval (ISMIR)*, volume 86, pages 937–952, 2010.

[19] Steven R. Livingstone, Ralf Muhlberger, Andrew R. Brown, and William F. Thompson. Changing musical emotion: A computational rule system for modifying score and performance. *Computer Music Journal*, 34(1):41–64, 2010.

[20] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.

[21] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[22] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4):614–626, 2018.

[23] Renato Eduardo Silva Panda, Ricardo Malheiro, Bruno Rocha, António Pedro Oliveira, and Rui Pedro Paiva. Multi-modal music emotion recognition: A new dataset, methodology and comparative analysis. In *10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2013)*, pages 570–582, 2013.

[24] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to Musical Metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[25] Ashis Pati, Alexander Lerch, and Gaëtan Hadjeres. Learning to traverse latent spaces for musical score inpainting. *arXiv preprint arXiv:1907.01164*, 2019.

[26] Iván Paz, Àngela Nebot, Francisco Mugica, and Enrique Romero. Modeling perceptual categories of parametric musical systems. *Pattern Recognition Letters*, 105:217–225, 2018.

[27] Rainer Reisenzein. Wundt's three-dimensional theory of emotion. *Poznan Studies in the Philosophy of the Sciences and the Humanities*, 75:219–250, 2000.

[28] Alan Rich. Harmony before the common practice period. URL: `https://www.britannica.com/art/harmony-music/Harmony-before-the-common-practice-period`, Sep 1998. Last accessed: 04-25-2022.

[29] James A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[30] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345, 2000.

[31] Emery Schubert. Continuous response to music using a two dimensional emotion space. In *Proceedings of the 4th International Conference of Music Perception and Cognition*, pages 263–268. McGill University. Montreal, 1996.

[32] Emery Schubert. Measuring Emotion Continuously: Validity and Reliability of the Two-Dimensional Emotion-Space. *Australian Journal of Psychology*, 51(3):154–165, dec 1999.

[33] William Forde Thompson and Brent Robitaille. Can composers express emotions through music? *Empirical Studies of the Arts*, 10(1):79–89, 1992.

[34] Sandrine Vieillard, Isabelle Peretz, Nathalie Gosselin, Stephanie Khalfa, Lise Gagnon, and Bernard Bouchard. Happy, sad, scary and peaceful musical excerpts for research on emotions. *Cognition and Emotion*, 22(4):720–752, 2008.

[35] Lindsay A. Warrenburg. Choosing the right tune: A review of music stimuli used in emotion research. *Music Perception*, 37(3):240–258, 2020.

[36] Richard Wetzel. *The globalization of music in history*. Routledge, 2013.

[37] Huiping Wu and Shing-On Leung. Can Likert scales be treated as interval scales? A Simulation study. *Journal of Social Service Research*, 43(4):527–532, 2017.

[38] Wilhelm Max Wundt and Charles Hubbard Judd. *Outlines of psychology*. W. Engelmann, 1902.

[39] Yi-Hsuan Yang and Homer H. Chen. *Music emotion recognition*. CRC Press, 2011.

[40] Kejun Zhang, Hui Zhang, Simeng Li, Changyuan Yang, and Lingyun Sun. The pmemo dataset for music emotion recognition. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, page 135–142, New York, NY, USA, 2018. Association for Computing Machinery.

Chapter 5

PreGLAM: A Predictive, Gameplay-based Layered Affect Model

As submitted to Plut, C., Pasquier, P., Ens, J., & Tchemeube, R. (2022). *PreGLAM: A Predictive, Gameplay-based Layered Affect Model.* Entertainment Computing

Abstract

We present the Predictive Gameplay-based Layered Affect Model (PreGLAM), an affective game spectator model. PreGLAM extends affective NPC emotion models to model a passive, biased spectator of gameplay. We implement PreGLAM into a custom game *Galactic Defense*, which we also describe. We empirically evaluate PreGLAM's application in *Galactic Defense*, where we compare PreGLAM annotations with participant-provided ground-truth annotations. PreGLAM's significantly outperforms a random walk time series in how accurately it matches ground-truth annotations.

5.1 Introduction and Motivation

5.1.1 Motivation

Gaming is, among other things, an emotional experience. There are many potential benefits to modeling emotional responses to the gameplay of a video game. Most applications of modeling gameplay emotion focus on the player's perceived or induced emotions. Experience-driven Procedural Content Generation (EDPCG) uses affect models to inform the generation

of game content [65]. Affective music generation systems generate adaptive music based on a real-time emotion model [52, 49]. Adaptive serious games use real-time affect models to enhance the learning process and performance [38]. Non-human agents such as Non-Player Characters (NPC)s are modeled to create emotionally informed behaviour [18].

In addition to the agents who act in a game, gameplay can be watched by a passive agent, such as a spectator. Spectating is a popular way of consuming video game content — In 2021, a total of 8.8 billion hours of video game live streams were watched globally [8]. In 2020, the "League of Legends" Championship finals were watched by a peak of 4 million concurrent viewers, and a total of 45 million viewers [20].

Spectator emotion models may be used for diagetic audience reactions in games that simulate a real-world sport such as *FIFA* or *Madden*, or in fictionalized sports such as *Rocket League* or *Blitzball*. Spectator modeling in e-sports games may assist camera operators [42, 29], casters, and analysts in automatically recognizing highly emotive gameplay moments. Spectator emotion models may also be used to influence the adaptivity and/or generation of a musical score and/or audio design. While academic research often uses an affective model to create music that attempts to match the player's emotions [49, 51], Phillips describes one primary function of music as acting as "an audience" [43], and an affective spectator model is well-suited to influencing the adaptivity and/or generation of a musical score filling this function.

Spectating a game views an interactive experience in a linear fashion, and may be considered more similar to the experience of watching a film than of playing a game. Film music's capability to evoke emotions in its viewing audience is well-established [12], and it follows that watching a video of gameplay may have a similar effect.

Interestingly, sports spectators show emotional responses to games that are stronger than those from spectating non-game entertainment media, and more closely resemble emotional outcomes from personal successes or failures [26]. Holm et al. collect physiological responses from participants playing and watching a First-Person Shooter (FPS), comparing between participants who enjoy the FPS genre and those who don't [23]. Holm et al. find that among players who enjoy the genre, spectating and playing a game have similar physiological responses. This indicates that while the experiences differ in other ways, the emotional responses of a spectator and player are similar, particularly among those who enjoy and/or are familiar with the genre. Therefore, modeling spectator emotions may also provide insights into player emotions.

5.1.2 PreGLAM

We present the *Predictive Gameplay-based Layered Affect Model* (PreGLAM) — an artificial cognitive agent with privileged game information, that models the real-time perceived affect of a biased game spectator. PreGLAM's design is based on affective NPC models, such as *ALMA: A Layered Model of Affect* [17], GAMYGDALA [48], and *EMoBeT* [3], which we

discuss in Section 5.2. PreGLAM is a knowledge-based agent that does not directly impact the game world. PreGLAM is provided with a desire in the form of a game outcome. For our purposes, we model PreGLAM with a desire of the player winning the game, though other desires may include the player losing the game, a particular team of players winning the game, a player using a particular item or ability, or achieving some quantitative measure such as score. We note that PreGLAM could be simultaneously implemented with multiple desires, e.g. a separate PreGLAM instance may model a spectator biased towards each individual team in a competitive multiplayer game.

In terms of emotional model, PreGLAM uses a 3-dimensional Valence-Arousal-Tension (VAT) description of emotion as we will discuss in Section 5.2.3, and utilizes an appraisal model based on the "OCC" model proposed by Ortony, Collins, and Clore [40]. While NPC-focused implementations of the OCC model use a Pleasure-Arousal-Dominance (PAD) model, we believe that dominance is poorly suited as an emotional dimension for a passive agent. Tension is important in musical emotion [55], spectating film [54, 22], and is often described as important in gaming [28, 22]. Film and game composer Bear McCreary describes a primary role of music in games as creating tension [56].

PreGLAM is a predictive model, and models likely (prospective) future events. Some prospective events are provided with privileged and/or advanced information. As an example, in our implementation, opponents telegraph certain attacks with a visual indication shortly before the attack fires. For these attacks, PreGLAM knows of these attacks $\approx 3-5$ seconds before the visual indicator plays. Other prospective events are derived with assumed player strategy, e.g. in gameplay scenarios where there is a clear optimal move, we assume that the player will make the optimal move.

PreGLAM's knowledge base is provided as a table of **Emotionally Evocative Game Events** (EEGEs), which are events that impact the provided desire. EEGEs are assigned base VAT values, which represents the basic intensity of the perceived emotional response to the EEGE. EEGEs are also assigned one or more intensity modifiers, which describe any modifiers to the perceived emotional intensity. For example, if the player is about to receive a damaging attack, the emotional reaction is stronger if the player is also low on health. In this case, the EEGE of receiving an attack has a modifier of player health.

PreGLAM calculates its belief at each time step by aggregating past and predicted EEGEs, as discussed in Section 5.3. PreGLAM does not directly affect the game world, as spectators of games generally do not directly affect the actions of a game. Instead, PreGLAM outputs real-time unbounded VAT values. We use these values to inform the adaptivity of an accompanying musical score, implementing the description of adaptive game music acting as "an audience" [43]. Our musical score is further discussed in a separate paper [47].

In our implementation of PreGLAM, we derive a set of EEGEs via an informal experiential approach — we play the game that implements PreGLAM, and derive a set of EEGEs based on our perceived emotions. We note that EEGEs could be derived from any

source, e.g. an ML classifier trained on ground-truth emotional data, with full access to game information, could derive EEGEs. We discuss our implementation of PreGLAM in Section 5.5.2, and other theoretical implementations in Section 5.4.

To implement and evaluate PreGLAM, we create an action-RPG genre game titled "Galactic Defense" (GalDef). Lopes suggests that designing a game for experimental purposes is often preferable to modifying an existing game, due to the amount of personalization and control granted [32]. *GalDef*, is further described in Section 5.5, and our integration of PreGLAM is described in Section 5.5.2.

We empirically evaluate PreGLAM by comparing PreGLAM's output annotations with ground-truth annotations provided by human spectators of *GalDef* gameplay, which we describe in Section 5.6. After gaining familiarity with *GalDef*'s gameplay, 50 participants watch a total of 20 videos of *GalDef* gameplay. and provide annotations while watching video replays. We analyze results by comparing the Dynamic Time Warping (DTW) distance and the Root Mean Squared Error (RMSE) between ground-truth and PreGLAM annotations with the same metrics between ground-truth annotations and a random walk time series. PreGLAM outperforms the random walk across all experimental conditions, and in dimensions of arousal and tension, and insignificantly outperforms the random walk in the dimension of valence.

5.2 Background

5.2.1 Player experience models

Player Experience Models (PEMs) are commonly used in the field of Experience-Driven Procedural Content Generation (EDPCG), and are generally used to evaluate and generate game content to evoke a particular player experience [64]. There are two main approaches to real-time player affect models: Psychological and physiological. Psychological approaches estimate player emotion based on observable features [25]. Physiological emotion models may use biofeedback techniques to attempt to directly read a player's physiological reactions, such as increasing heart rate or EEG activity [27]. Occasionally, physiological measures are used to train a psychological ML-based model [37].

Physiological models of emotions for games primarily are used to create games that can respond in real-time to the physiological data, which alters or is used to create game content [25]. Visual-based facial or body emotion recognition may be used to gather physiological data, as well as sensors placed on the body to read autonomic responses such as electroencephalographic or galvanic skin response readings. These models generally use biofeedback as an additional input to the game.

Psychological models attempt to replicate the cognitive or neurological processes that lead to emotions by evaluating gameplay variables and events. Psychological models do not require additional equipment or sensors, and therefore can be integrated into a game without

any additional requirements on the player. However, this also means that psychological models do not have access to markers of player emotion outside of in-game behaviour. Because PreGLAM is unidirectionally serving to describe the gameplay emotion without influencing the game, we use a psychological model that estimates perceived emotion based on gameplay.

One use of player models is dynamic difficulty adjustment (DDA) — as the player plays the game, the model estimates how difficult the game is for the player, and adjusts the gameplay based on the model. *Left4Dead*'s "Director" is the most well-known DDA model [53], and there is research into applying deep learning to DDA [41]. PEMs are also used in live games, which use a continuous online environment, to identify sources of "churn" and increase player retention [5, 62, 36]. PEMs may also be used in game testing [1, 33], where models of players are used to evaluate elements of a game.

PEMs have also been used to influence the real-time generation of music, generally assigning associations between game features and affect, rather than deriving them from testing or ML techniques. Plans and Morelli use a PEM to attempt to influence the emotions of a player towards maximizing "fun" [44]. Separate systems by Prechtl and Plut use PEMs to estimate a tension value to adjust adaptive music [49, 46]. Systems by both Scirea and Williams et al. implement a PEM to control musical generation with emotional dimensions of valence and arousal [52, 63]. The "Adaptive Music System" (AMS) by Hutchings and McCormack [24] utilizes a spreading activation model to influence the affective expression of a generative score.

5.2.2 Affective Non-Player Character (NPC) models

NPCs may use an affective model to react to the player's actions in emotionally informed ways, with the intent to create believable characters. Cognitive appraisal models of emotion, such as the "OCC" model, named for its creators Ortony, Clore, and Collins [40], are common in modeling NPCs [25], and describe emotions as the results of an evaluation of how emotionally evocative conditions affect the subject.

In the OCC model, emotions arise from an evaluation of the outcome of events, and are distinguished by whether they concern the subject ("self") or an "other", as well as whether they involve prospective or known outcomes. Emotions related to the agency of actions are differentiated by whether the action is praiseworthy or blameworthy, and whether the actions are of the self or an other. Finally, emotions related to the attributes of objects are primarily related to whether the subject finds the objects appealing or unappealing.

We are aware of three systems for NPC design that implement the OCC model, appraising emotion based on the events and actions that arise during the interaction between player and game. *ALMA: A Layered model of affect*, attempts to provide emotional reactions in conversation [17]. *GAMYGDALA* presents a "black box" system for appraising emotion that focuses on providing a generalizable system across NPC interactions [48]. *EMoBeT*

builds on the architecture of *ALMA* by using the emotional reactions to control behaviour trees [3].

Across these implementations, game designers provide specific details on NPC goals or wants, and on the possible game events that can impact those goals/wants. ALMA appraises "relevant input" for all characters, using provided appraisal rules for each character. To use GAMYGDALA, designers provide explicit sets of goals and events that are relevant to those goals, and GAMYGDALA appraises game data based on these goals and events. For EMoBeT, developers provide a set of possible emotions and events that trigger these emotions, as well as behaviour trees that dictate the NPC responses to particular emotions.

In addition to applying the OCC model, both ALMA and EMoBeT use a layered representation of affect, providing a mood based on designed personality traits and values. The results of the appraisal process then layer an emotion value on top of the mood value, to reflect the longer-term affective states that can influence emotion.

5.2.3 Affect representation

Mood and Emotions

Affective experiences of mood and emotions are primarily differentiated by their duration and whether they are a reaction to a particular stimulus [11]. Emotions are triggered by an internal or external event [61], and last a short amount of time, generally seconds to minutes [60]. In contrast, moods are described as diffuse and global, generally last longer, and are not elicited by any particular event.

Figure 5.1 shows charts taken from Frijda [13, 14], and show examples of emotional curves produced when participants are asked to draw a graph of the course of their emotion. Frijda notes that the more linear chart, on the right in Figure 5.1 is more common than the chart on the left, and that participants often also chart multiple peaks.

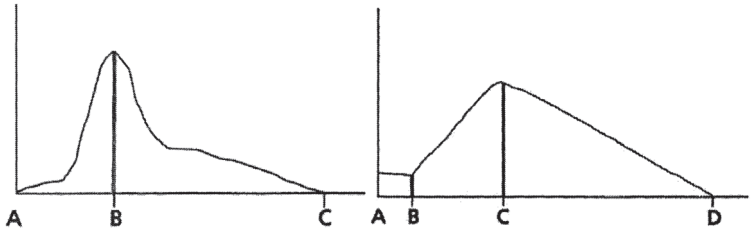


Figure 5.1: Charts of emotional intensity over time, from Frijda [14].

In addition to fading in time, emotions are contextual. Emotion classification on facial expressions is more accurate in video form than image form [14], and there is an increase in use of contextual data for automatic emotion recognition in both visual and audio formats [7, 2]. This indicates that emotions are often perceived as changes, rather than an absolute value.

VAT Model

We use a 3-dimensional "VAT" model of affect shown in Figure 5.2, with dimensions of valence, arousal, and tension. This is based on Schimmack and Grob's dimensional model of valence, tense arousal, and energy arousal [50], modified for simplicity, parity, and to bring the language into line with common terminology across both fields. Previous comparisons of affect models for multimedia content analysis demonstrate support for a three-dimensional VAT model [34]. As mentioned in Section 5.1, while other applications of the OCC model generally use a 3-dimensional PAD model of affect using, we believe that the dimension of "dominance" is not applicable when modeling an NPC that does not have any agency to affect the game.

Figure 5.2 places a subset of common categorical emotions in their approximate relative VAT location.

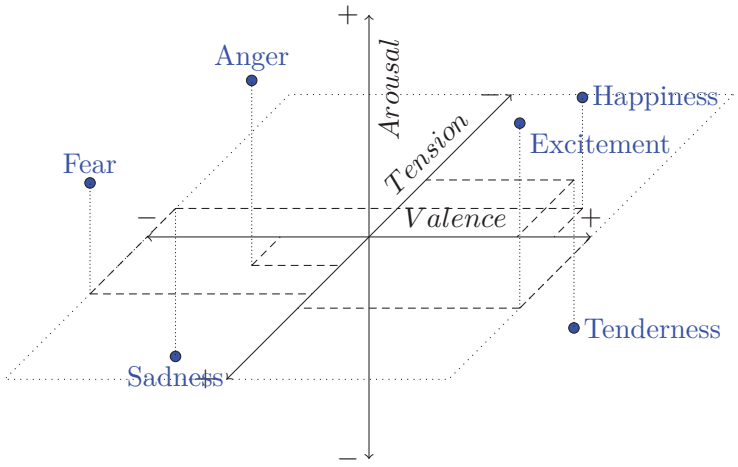


Figure 5.2: 3-Dimensional Model of Affect.

Valence

Valence is often paraphrased as "pleasantness", and describes whether an affect is "positive" or "negative". The OCC model primarily details the valenced reactions to emotionally evocative stimulus [40]. Positively valenced events are events that are desirable for the subject. In the case of games, participants identified positive valence events and actions in a racing game as passing another car, improving their position in the race, or making another car crash, while negative valence events were being passed, being hit by another car, or driving off the course [21]. We note that unpleasant emotions experienced or felt during gameplay may be later appraised as positive experiences [6].

Arousal

Arousal, sometimes paraphrased as "activity" or "energy", is the dimension of activation. Examples of high-arousal emotions are excitement and fear, while examples of low-arousal emotions are calmness and sadness. Emotions with high arousal are most associated with physiological changes such as increased heart rate, blood pressure, and increased electrical activity in the brain [39].

Tension

Tension is unique in that it necessarily involves the prospect of a future event. Tension is closely related to valence, but distinct — valence is often associated with the desirability of an associated event, while tension is associated with the prospect, or prediction of the event. Tension is distinct from valence, in that the prospect of an outcome can have both high valence and high tension, such as during the anticipation of good news, or may have low valence and high tension, such as in during the anticipation of bad news.

The OCC model describes tension as arising from prospective events — a subject believes that an event is likely to happen, and has an emotional response depending on the desirability of the prospective event for the subject. As an example of an in-game prospective event that an audience may react to, consider attack telegraphs in MMO games. Figure 5.3 shows a set of attack telegraphs from the MMO *Wildstar*. Highlighed areas on the ground indicate that an enemy attack will be incoming in soon, and will hit any players who are standing in the highlighted areas. This effectively communicates a prospective event of incoming attack, and a related increase in tension as to whether the player will avoid the incoming attack.

Figure 5.3: Attack telegraphs from *Wildstar* [58].

5.3 PreGLAM Framework

We present the Predictive Gameplay-based Layered Affect Model, or PreGLAM, a model of affect that simulates a biased spectator of a video game. *PreGLAM*'s design is inspired by *ALMA - A layered model of affect* [17], GAMYGDALA [48], and *EMoBeT* [3]. PreGLAM outputs values for affective dimensions of valence, arousal, and tension. PreGLAM models affect by combining a long-lasting, environmental mood value with an event-focused appraisal-based model of emotion, to produce a single affect value per dimension. PreGLAM calculates an emotion value from both events that have occured, and events that are predicted to occur. PreGLAM scales emotion values through time, to represent the rise and fall of emotions over time. Figure 5.4 shows how PreGLAM acts in relation to human players and spectators.

Conceptually, PreGLAM follows the design of GAMYGDALA, ALMA, and EMoBeT in terms of its input and appraisal. A desire is provided, from which all emotional appraisals are derived. During gameplay, PreGLAM uses a knowledge base of EEGEs that affect the desire, and outputs a value for valence, arousal, and tension based on an appraisal of the EEGEs. PreGLAM uses a flexible framework, and EEGEs may be derived from playtesting, design considerations, or from ML classification of gameplay with ground-truth emotional annotations.

We implement PreGLAM into our game *Galactic Defense*, as described in Section 5.5.2. For our implementation, we provide a baseline value of 1 as the minimum value for emotional impact, and scale all other values based on the baseline value. We implement intensity modifiers that scale these values between 100–200%, depending on the values of the intensity modifiers.

5.3.1 Mood

PreGLAM takes at any time a mood value for valence, arousal, and tension. As mentioned in Section 5.2, moods are longer-lasting and more general affective experiences than emotions, and are not connected to any particular source. Mood is therefore related to longer-lasting, environmental, macro-levels of gameplay. While there are no programmatic restrictions on how quickly PreGLAMs mood value can be changed, we recommend that mood values remain relatively stable during gameplay.

We implement mood directly based on the relative power between the player and their opponents at the beginning of a set of battles. Examples of other possible sources for mood values include environment e.g. "snow" levels have a lowered arousal than "fire" levels, particular enemy types e.g. bosses have elevated tension, or player resources, e.g. valence is lower when the player is out of healing items. The mood value provides a base perceived level of affect for PreGLAM. If there are no EEGEs being modeled, PreGLAM will output the mood value.

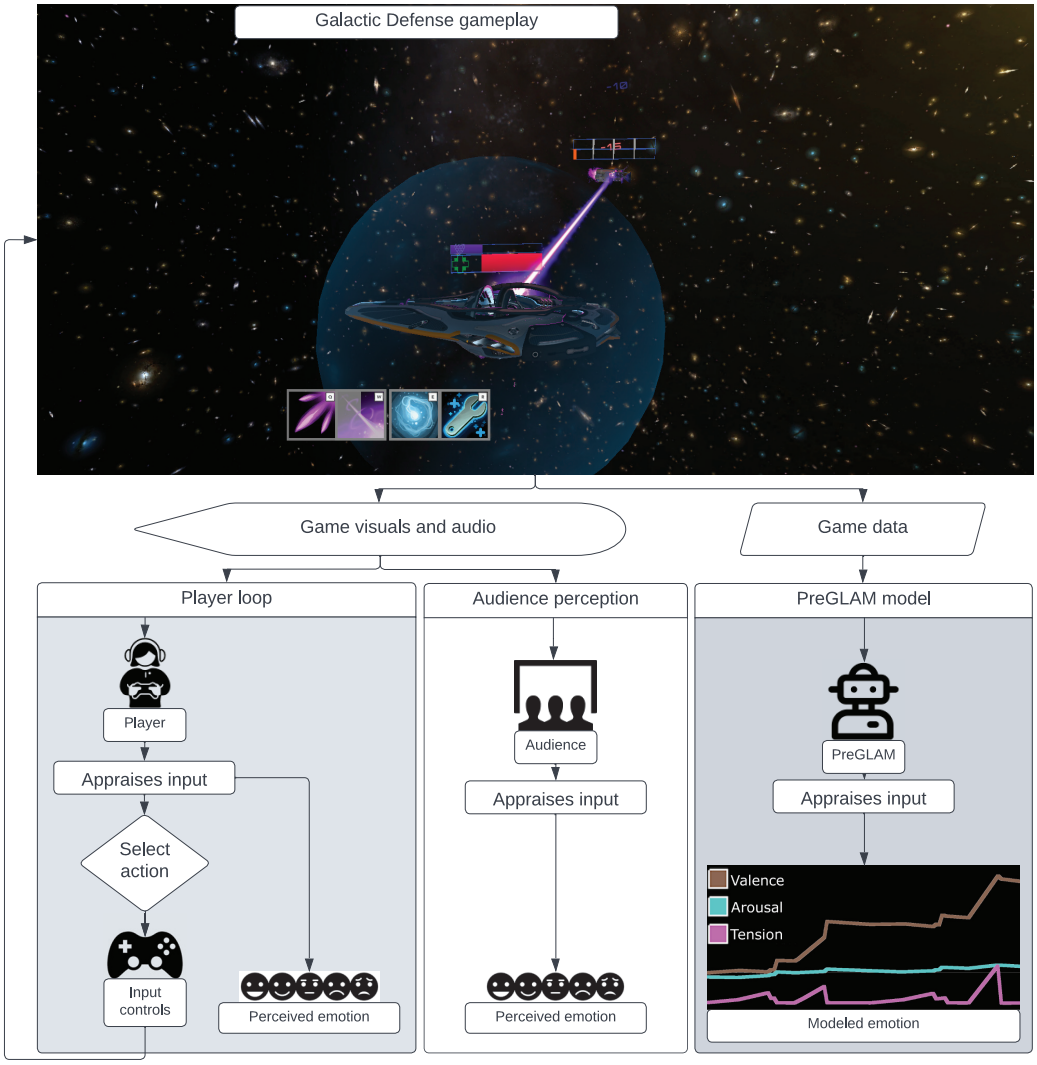


Figure 5.4: Diagram of PreGLAM's relation to player, audience, and game.

Table 5.1: EEGE variables.

Provided variables		
Variables	Symbol	Relevant Section
Name	N/A	N/A
Base emotion value	v_{dim}	Section 5.3.2
Context variables	$\{c_1, ..., c_n\}$	Section 5.3.2
Time ramp value	r	Section 5.3.2
Calculated variables		
Variables	Symbol	Equation
Intensity modifier	mod	Equation 5.1
Time scalar	s	Equation 5.2a
Emotion value	e	Equation 5.3

5.3.2 Emotionally Evocative Game Events

PreGLAM's appraisal of EEGEs drives the moment-to-moment reactions to gameplay. PreGLAM models two types of EEGEs: Past and Prospective. EEGEs that have occurred in the game are past EEGEs, while EEGEs that have not occurred yet are prospective EEGEs.

All EEGEs are provided a set of variables as shown in Table 5.1, which also shows the variables that PreGLAM calculates for each EEGE. EEGEs are provided a name, base emotion values for each dimension v_{dim}, a set of context variables $\{c_1, ..., c_n\}$, and a time ramp value r. In our implementation, we derive variables from playtesting — due to the lightweight nature of PreGLAM, creating and altering EEGEs is trivial, and we test our variables and ranges during the development of the implementation.

PreGLAM derives a single intensity modifier mod for each EEGE from the provided set of context variables $\{c_1, ..., c_n\}$ using Equation 5.1. At each time step t, PreGLAM calculates a time scalar s_t for each EEGE, based on the difference p_t between the EEGEs in-game occurrence and the current in-game time, and the provided time ramp value r, using Equations 5.2a or 5.2b, depending on whether the event is past or prospective. PreGLAM calculates an emotion value e_{dim_t} for each dimension, for each EEGE, using Equation 5.3.

$$mod = 1.0 + \frac{1}{n}\sum_{i=1}^{n} c_i \tag{5.1}$$

$$s_t = 1.0 - (p_t/r) \tag{5.2a}$$

$$s_t = p_t/r \tag{5.2b}$$

$$e_{dim_t} = v_{dim} * mod * s_t \tag{5.3}$$

Base emotion value

All EEGEs are provided a base emotional value v for dimensions dim of valence, arousal, and tension. We assign a tension value of 0 for all past events, as tension is associated only with the prospect of events in the OCC model. As mentioned in Section 5.1, we base all values on a unit of 1, though any consistent scale may be used. The base value represents the emotional perception of the event when it occurs, if no intensity modifiers are increasing the emotional intensity. We directly assign emotional values, but emotional values could also theoretically be derived from ground-truth annotations.

In our implementation, we assign all arousal values as 1. Given that arousal describes the overall activity level, we do not model different EEGEs with different arousal levels — all EEGEs equally contribute to the aggregate level of arousal. During development, we informally experimented with assigning varying levels of base arousal to events. However, in our playtesting, we found that utilizing a single, static arousal level more closely reflected our emotional perception.

Context variables and intensity modifier

EEGEs are provided a possibly empty set of context variables $\{c_1, ..., c_n\}$, which are given as a percentage of a given game variable. These variables describe the context of the EEGE, and modify the intensity of the emotion. For example, a player attack in the context of a full-health opponent may result in a less-intense emotional perception than a player attack in the context of a nearly-defeated opponent — lightly hitting an opponent is assumed to be less intense of an emotional experience than knocking them out. PreGLAM calculates a single intensity modifier value mod for each EEGE using Equation 5.1.

Time ramp and scalar

As discussed in Section 5.2.3, emotions rise and fall in intensity over time in a generally linear fashion. EEGEs are assigned an initial time ramp value r when created, which represents the duration over which the EEGE's emotion value will smoothly ramp through time. For past EEGEs, we assign a static ramp of 90 seconds. While the duration of emotions ranges from seconds to hours [61], we assign a 90 second ramp to past EEGEs based on playtesting.

At each time step t, PreGLAM calculates time scalar s_t for each EEGE using Equation 5.2a for past events, and Equation 5.2b for prospective events, where p_t is the time that has passed since the event's creation. In other words, as an event approaches the present, it's time scalar approaches 1.

5.3.3 Output

At each time step t, PreGLAM calculates a single emotion value per dimension e_{dim_t} for each EEGE using Equation 5.3. PreGLAM then calculates an output affective value for each

dimension a_{dim_t}, based on the current provided mood value m_{dim_t} and set of emotional values from all EEGEs $\{e_{1_{dim_t}}, ..., e_{n_{dim_t}}\}$, using Equation 5.4. We note that while PreGLAM's output is technically unbounded, values always trend towards the provided mood value over time.

$$a_{dim_t} = m_{dim_t} + \sum_{i=1}^{n} e_{i_{dim_t}} \tag{5.4}$$

PreGLAM uses the "Grapher" plugin for Unity [9] to create a graphical output and save all affect values to .csv format. While PreGLAM performs calculations on every frame, we sample the output every 250 ms. This sample rate is chosen due to Unity's inconsistent timing and to synchronize with annotation software for empirical evaluation, discussed in Section 5.6.

5.4 Use-case examples

To demonstrate the generalized applicability of the PreGLAM framework, we propose numerous uses in games across genres. We propose applications in three games: *Dark Souls*, a dark fantasy, mostly single-player action-RPG known for a high degree of difficulty, *The Sims*, a casual life simulation game, and *League of Legends*, a popular e-sport and competitive 5v5 multiplayer online battle arena with peak monthly player count of 180 million players in October 2021 [35].

As we discuss in Section 5.5, we assign a base arousal value of 1 to all EEGEs when implementing PreGLAM into a game. While the use of varying arousal values was explored, we found that a single arousal value as modified by context variables and time scalar best matched our perception of emotion. While other base arousal values may be used in other implementations, we assign an arousal value of 1 to all EEGEs in our examples.

5.4.1 Dark Souls

Dark Souls is a fantasy action-RPG game developed by From Software. The player plays as an undead human, tasked with the eventual goal of defeating a series of bosses and lighting a series of bonfires. Bonfires also act as checkpoints, and the player will revive at the most recently visited bonfire when they die. As the player progresses, they receive resources of "souls" and "humanity", which are used to increase their power. When the player dies (an extremely common occurrence in Dark Souls), they revive at the bonfire where they most recently rested, and the souls and humanity that the player has in their inventory when they die remain at the location of their death. If the player touches that location, they can re-gain their lost resources. If the player dies before re-gaining their lost resources, those resources are lost forever. For more information on the gameplay of Dark Souls, see the game mechanics guide page at IGN [57].

Table 5.2: Dark Souls EEGE table.

Event	Valence	Arousal	Tension	Context variables
P. Attack	1	1	1	Missing O. Health, P. Stats, P. Weapon Damage E. Weaknesses, P. Stamina, P. Spells remaining
O. Attack (P. Not blocking)	-2	1	2	Missing P. Health, Missing P. Estus, O. Damage Inv. P. Stats, Inv. P. Armour, Inv. P. Buffs
O. Attack (P. Blocking)	-1	1	1	Missing P. Stamina, Missing P. Health
P. Dodge	2	1	1	Missing P. Stamina, Missing P. Health

Most of the gameplay in *Dark Souls* is in combat, and the game and series are known for having a deliberate pace, and a high degree of combat difficulty. Attacks in Dark Souls have long animations, and the effect of an attack or ability may not occur until seconds after the player has pressed the corresponding button. This is true of enemy attacks as well, and thus enemy attacks in Dark Souls are often highly telegraphed. To succeed, the player must learn the animations and attack patterns, to respond appropriately and at the right time. These responses include blocking, dodging, moving away from the attack, or parrying. To parry, the player has a small timing window to attempt the parry — too soon or late compared to the attack and the player will receive full damage. Blocking and dodging both use a "stamina" resource, which is also used when attacking.

A source of a mood value in *Dark Souls* could involve elements of environment, such as assigning low levels of valence and moderate levels of tension to *Blighttown*, a notoriously unpleasant and difficult area of *Dark Souls 1*. Another source of mood could involve the player's held souls and humanity — while having resources to strengthen the player is good, the possibility of losing a large cache of resources is often a source of tension in *Dark Souls*.

The most direct source of EEGEs in *Dark Souls* is in combat, attacking and avoiding attacks. In addition to attacks themselves, the management of resources is a key part of the combat in *Dark Souls*. The player will be unable to take offensive or defense action without stamina. If the player integrates spellcasting into their build, they have a limited number of times that they may cast each spell. The player has a limited number of refillable Estus Flasks, and other healing items are rare and consumed on use.

Consider the situation shown in Figure 5.5, as appraised by PreGLAM with a desire of the player defeating the Taurus demon. The Taurus Demon is often the first boss that the player encounters after completing the tutorial area, and therefore we assign a mood value of somewhat elevated arousal and tension compared to the previous area.

We create a sample set of EEGEs in a Dark Souls boss fight, shown in Table 5.2, based on informal experiential evaluation. In all EEGE tables, we provide the name, values of v_{dim} for each dimension, and the set of context variables. We abbreviate "Player" to "P", and "Opponent" to "O". Therefore, the event "P. attack" indicates the player attacking, and a context variable of "Missing O. health" describes the health that the opponent has missing from their maximum. We also abbreviate "Inverse" as "Inv" — a marking of "Inv" indicates that the modifier value increases as the context variable decreases.

Figure 5.5: Screenshot from fight with Taurus Demon from Dark Souls, from "Nintendo Life" [30].

The Taurus demon has 5 moves: a jumping attack used at range, a vertical heavy attack with a long windup, a short-range horizontal swing with a short windup, a very short-range attack with a short windup, and a large jump used if the player stands in a particular spot. We do not know when the AI in *Dark Souls* determines which attack to use internally, but in Figure 5.5, the most likely next boss attack, given the positions of the player and boss, is the medium range jump attack.

The player has very low health, and they have used all of their healing potions. The boss, the Taurus Demon, has nearly full health. The player appears to be using an un-upgraded sword and shield. The player has 2 humanity, and 2441 souls. The player has recently received damage, but has full stamina.

Given the player's remaining health, any additional attack will likely result in the player's death. When the Taurus demon selects an attack, the game sends the event call to PreGLAM. To calculate $e_{valence}$, we multiply $v_{valence} = 2$ by $mod = \{c_1, ..., c_n\}$. As the modifier variables are all high in this example, *mod* will likely be near 2.0, giving an $e_{valence}$ value near -4.0 when the event occurs. This value will then be scaled by time scalar s_t.

In our example, because the player has full stamina, assuming that the player does not activate another ability, we additionally create a prospective likely event of P. Dodge with $v_{valence} = 2$ and context variables {Missing P. Stamina, Missing P. Health}. P. Health is low, but P. Stamina is high, and therefore the P Dodge event *mod* value is expected to be near 0.5, providing $e_{valence} = 2 * 0.5 = 1$. We thus calculate $a_{valence} = 0 + -4.0 + 1.0 = -3.0$ at the time of the event. Note that these values are the maximum — PreGLAM will begin to adjust values from 0.0 to these maximums linearly as the events approach. Also note that the values of all EEGEs are added to a provided mood value.

5.4.2 The Sims

Figure 5.6: Gameplay of the Sims 1, from "The Finked Films" [59].

The Sims series is a life simulation series developed by Maxis. The player controls one or more "sims", or virtual people, as the sims simulate an artificial life. Sims have wants and traits that control their behaviours, and also have a set of physical and emotional needs to attend to. The physical needs of sims are represented by a set of bars that fill or empty as the need is addressed or ignored. Some examples of physical needs are hunger, sleep, and bathroom.

Sims may get jobs to make in-game currency, which can be used to customize the furniture and architecture of the house that the player's sims live in. Additionally, in-game currency is used to pay for food and to pay regular bills that arrive for the player. Sims additionally have skills, which can be improved by interacting with certain types of furniture.

Consider the following hypthetical situation in *The Sims* as appraised by PreGLAM, with a desire of all player-controlled sims having their "needs" fulfilled. Table 5.3 shows several examples of EEGEs, given this desire.

For simplicity, in this example, the player controls only a single sim, who we call "Sim-p". In actual gameplay, the player may control multiple sims, in which case PreGLAM would perform appraisals for each sim and sum the collected affect values. It is 7:00PM in the game world. Sim-p has recently eaten, and their hunger gauge is full. Sim-p's energy, and

Table 5.3: The Sims EEGE table.

Event	Valence	Arousal	Tension	Context variables
Sim need depleted	-3	1	2	Missing sim need, distance to need
Sim skill point gained	1	1	1	Skill level, Relevance to Sim career
Sim career promotion	2	1	1	Skill levels, Avg. sim need fill
Income	2	1	1	Income amount

fun gauges are half-empty, and Sim-p's bladder gauge is nearly empty. It is a weeknight, and Sim-p has work the next day, which drains energy and fun. Sim-p has one skill that is nearly leveled-up, and Sim-p will likely be promoted if the skill is leveled up.

If the player chooses to increase Sim-p's skill, the likely event Sim skill point gained is created. Additionally, if the player successfully increases the skill, the likely event Sim career promotion is created. Additionally, three prospective Sim need depleted events exist, as the player's bladder, fun, and energy levels are low and emptying.

While the player may not know if they have enough time to train a skill before their bladder gauge empties, PreGLAM has perfect knowledge of the game mechanics. Therefore, PreGLAM may trend towards positive or negative valence in this situation, depending on whether the intensity modifiers on Sim need depleted or Sim skill point gained are more relevant — As sim needs are overall emptied, the associated negative valence is stronger, while as the sim is closer to gaining the skill point, the associated positive valence is stronger.

5.4.3 League of Legends

League of Legends (LoL) is a Multiplayer Online Battle Arena (MOBA) developed by Riot Games. Standard League of Legends games involve two teams of 5 human players. Each player controls a single champion, selected from a pool of 156 champions (as of 2021's release of "Ashkan"). Each champion has an attack and 4 abilities. Teams work together to ultimately destroy the opponent's "Nexus", which is at the back of each team's base. While at their own base, champions are healed and restore mana, and player's may buy items to strengthen elements of their champion, using in-game currency that is gained via gameplay. Figure 5.7 shows the map for *League of Legends*, and Riot games provides a more in-depth description of the mechanics [15].

Most of the gameplay in LoL revolves around the interactions between human players. Gameplay in LoL is very quick and reactive — unlike *Dark Souls*, attacks and abilities in LoL generally occur instantly, or nearly instantly after a player inputs a command. Dota 2, a MOBA that is very similar to LoL, is often used in evaluating game-playing AI, due to its complexity within a constrained space [4]. We believe that MOBAs such as LoL are also prime candidates for ML-derived implementations of PreGLAM.

Figure 5.7: Map in League of Legends, from Gao [16].

As with previous examples, we can derive sets of EEGEs in LoL that affect a provided desire of the player's team winning. We derive a set of events via analysis, shown in Table 5.4. Due to the complexity in LoL, we believe that ML prediction of events is suited for application in deriving and computing context variables. In this case, an ML classifier can be trained on raw game data, to predict the likelihood of prospective EEGEs — rather than explicitly defining a set of context variables, the ML model builds a prediction that incorporates all possible contextual variables.

Figure 5.8 shows a screenshot from a League Championship Series (LCS) game of LoL. A strategic analysis of the game state indicates that a dragon fight is imminent: Both teams have placed wards near the dragon's location, minions in the middle and bottom lane are in equilibrium and don't need attention from either team, and the dragon will spawn in 3 seconds. TSM will likely be able to win the dragon, as one member of TL does not respawn for 30 seconds. TL's likely future movements will be to attempt to steal the dragon (let TSM deal most of the damage and come in at the end), or to attempt to fight TSM while TSM is weakened from fighting the dragon.

Table 5.4: League of Legends EEGE table.

Event	Valence	Arousal	Tension
O. Tower destroyed	2	1	1
P. Tower destroyed	-2	1	1
P. Elemental Dragon	2	1	1
O. Elemental Dragon	-2	1	1
P. Elder dragon	3	1	2
O. Elder dragon	-3	1	2
P. Baron	2	1	2
O. Baron	-2	1	2
P. Kill	1	1	1
O. Kill	-1	1	1

Figure 5.8: Screenshot from League Championship Series (LCS) Summer Split 2021: Team Liquid (TL) vs Team Solo Mid (TSM).

The features that indicate a coming dragon fight are complex and contextual — the presence of wards in the bottom-side river does not itself indicate a dragon fight, nor do waves in equilibrium. While a situation can be strategically analyzed to conclude that a dragon fight is near, manually determining the set of features that indicates a dragon fight is not feasible. Therefore, we believe that an ML-based approach to predict likely events is best applied in this case.

5.5 Use-case application: Galactic Defense

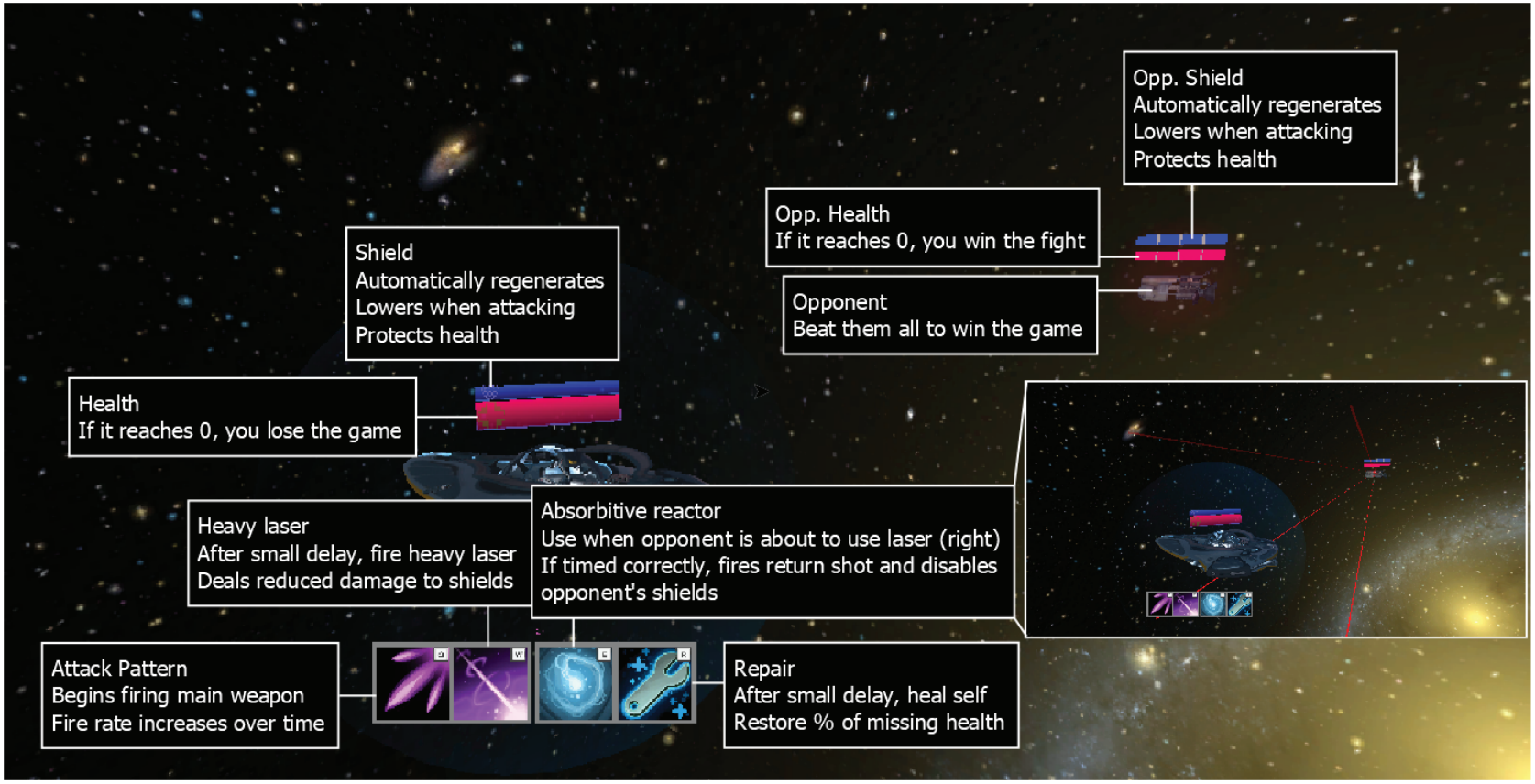


Figure 5.9: Visual tutorial for Galactic Defense.

5.5.1 Game description

To evaluate PreGLAM and provide an applied use-case, we integrate the model into a video game that we develop called *Galactic Defense* (GalDef). Figure 5.9 shows an annotated screenshot from GalDef. GalDef is an action-RPG game originally designed for research in game music and emotion. *GalDef* is open-source, and available on GitHub [1] [45]. In *GalDef*, the player controls a single spaceship, called the "Vatic Savate", and must win a series of 1-on-1 battles with opposing AI-controlled spaceships.

Each ship in *GalDef* has two resource bars: Health and Shield. Shields have low hit points, but constantly regenerate. If shields are completely depleted, they will not return until they have fully regenerated. Shields are temporarily lowered while a ship is taking action. Health does not regenerate, but can be partially restored with an ability. When a ship takes damage, it will take shield damage if the shields are up, and will otherwise take damage to health.

Each ship in *GalDef* has 3 abilities, and the player has an additional bonus ability. Figure 5.9 includes a description of each ability. Shields are removed while using an ability until a short time after the ability finishes or is canceled. The "Heavy Laser" and "Repair" abilities can be interrupted — there is a delay before they activate, and if any damage is taken during this time, the ability is canceled. When an ability finishes activating or is canceled, the acting ship will raise its shields after a short delay, if the shield has hit points remaining.

In *GalDef*, the player progresses through three stages of increasing difficulty. The player fights two opponents in the first stage, and three opponents for stages two and three. After defeating a stage, the player's ship is healed to full, and the player enters a non-combat section of the game. During this transitional segment, the player selects from a set of upgrades.

The full set of upgrades is shown in Table 5.5. From the 10 possible upgrades, 3 are randomly selected. The player selects 2 of these selected upgrades to apply to their ship. This design is inspired by games in the *roguelike* genre, in which player's must adapt their build based on available upgrades. This upgrade design provides additional strategic depth, as the player evaluates upgrades and may change their play to accommodate the stronger abilities. These upgrades are designed to be roughly equivalent in overall power, providing variety without sacrificing emotional consistency.

[1] https://github.com/CalePlut/Galactic_Defense

Table 5.5: Galactic Defense Upgrades.

Upgrade title	Effect
Laser Supercharger	Reduces charge time of laser attack by 50%
Repair Supercharger	Reduces charge time of self-repair by 50%
Cannon Supercharger	Reduces # of shots taken until both cannons fire by 50%
Laser focusing crystal	Laser attack deals 2x damage
Cannon focusing crystal	Light attacks deal 2x damage
Repair nanite swarm	Increases self-repair from 45→90% of missing HP
Absorbitive capacitor	Increases parry window by 2x
Ion supercharge	Increases riposte disable duration
Shield capacitor	Doubles shield points
Reinforced Hull	Doubles health points

5.5.2 PreGLAM Integration

Mood

The relative power between the player and their opponent in *Galactic Defense* is designed to create sections of alternating ease and challenge. The player begins the game with roughly equivalent attributes to their opponent, and is expected to win the first battle with moderate difficulty. After the first battle, the player upgrades an aspect of their ship, increasing their power. In the next battle, the player first encounters two opponents of equal strength to the opponent that they defeated in the previous stage, and then encounters an opponent of greater power, roughly equal to the player's upgraded power. This cycle then repeats for the final upgrade and stage, with the final boss tuned to be difficult for the player.

We provide mood values based on this power curve and general progress through the game. Figure 5.10 shows the relative power levels of the player and enemy as the player progresses through the game, and the corresponding mood levels. These mood values are derived from the design of the game — they serve to describe the general state of the game environment. We increase the mood value for valence as the player customizes and empowers the Savate with upgrades. We increase arousal's value as the player and opponent's powers increase — attacks are faster, and deal more damage. We increase the mood value for tension as the opponent's power increases, and as the player achieves more progress through the game.

This designed power curve informs the mood values for PreGLAM. Overall, valence, arousal, and tension rise as the player encounters more powerful opponents. Essentially, if PreGLAM understands that the un-upgraded player is expected to easily defeat a weak, early opponent, the mood level of valence, arousal, and tension is lower than if the upgraded player is fighting a difficult final boss.

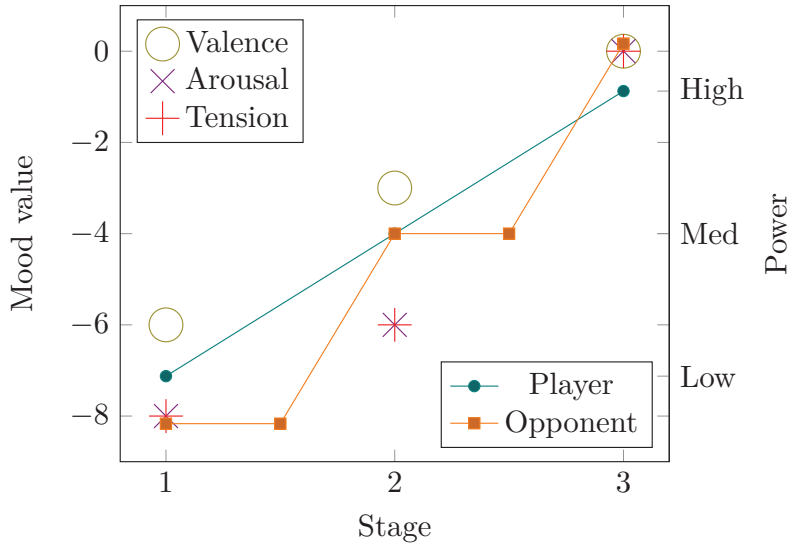


Figure 5.10: Mood values in Galactic Defense, based on Player and Opponent approximate power levels.

Emotion

We manually determine a set of EEGEs, with a desire of the player winning the game, shown in Table 5.6. As before, we notate the player as "P", and their opponent as "O". The selection and emotional value ranges of events are determined by experiential evaluation during playtesting of GalDef. We note that the tension values provided are only computed when the related event is prospective.

Table 5.6: EEGEs in *GalDef*.

Event	Valence	Arousal	Tension	Context variables
P. complete atk combo	1	1	1	Missing O. shield
P. heavy atk	1	1	1	Missing O. health
O. atk combo	-1	1	1	Missing P. shield
O. heavy atk	-2	1	2	Missing P. health, Parry active
P. shields down	-2	1	2	Missing P. health
O. shields down	2	1	2	Missing O. health
P. exploit O. disable	3	1	2	Missing O. health
P. death	-3	1	3	P. shield recharge time
O. death	3	1	3	O. shield recharge time
P. heal	2	1	2	Missing P. health
O. heal	-2	1	2	Missing O. health

5.5.3 Output

We use the output of PreGLAM to control the musical adaptivity of a score that adapta independently with 5 levels for valence, arousal, and tension. We use the "Multi-track Music Machine" transformer model to create a generative score, which expands on a composed adaptive score. These scores are further discussed in its own paper [47]. We assign thresholds for levels of low, low-medium, medium, high-medium, and high for PreGLAMs output, centered on a centre value of 0. These score thresholds are shown in Table 5.7.

Mood values in *GalDef* range from -6 to 0. Because the score's adaptive thresholds are based on a value of 0, a mood value of 0 results in a score that directly matches its emotional expression to the modeled emotions. By lowering the mood value, the thresholds for higher levels of affect in the adaptive score are effectively raised, producing a score that will trend towards the lowered central values.

As an example, consider a situation where the player is firing their heavy attack on a disabled opponent with full health. The associated EEGE will have a valence value of 3 when the attack is fired, as the context variable does not affect the outcome. With a mood value of -6, PreGLAMs output would be -3, within the range for a medium-low level of expression. With a mood value of -3, PreGLAMs output will be at 0, and therefore the medium level score will play. At a mood value of 0, PreGLAM's output will be 3, which will play a medium-high level of the score. Essentially, the accompanying score expands to its full dynamic range as the player progresses through the game.

Table 5.7: Adaptive score thresholds.

Level	Range
Low	<-10
Med-Low	(-10, -5)
Medium	(-5, 5)
Med-High	(5, 10)
High	>10

5.6 Empirical Evaluation

5.6.1 Empirical Methodology

To evaluate PreGLAMs accuracy in modeling an audiences perceived emotional response, we collect real-time user annotations from 48 participants. Each participant annotates a single affective dimension; Each participant watches a total of 4 videos of GalDef gameplay, and records one real-time annotation curve per video. We evaluate how closely these participant ground-truth annotations match the real-time annotations created by PreGLAM.

Figure 5.11: Screenshot of participant annotation interface. Note that chart re-sizes automatically to match unbounded participant range.

While we originally intended to use the *RankTrace* [31] function of *PAGAN* [36], due to cross-platform media issues, we implement a custom annotation software shown in Figure 5.11, based on *RankTrace* [2]. We attempt to exactly replicate the functionality of RankTrace.

While watching a video of gameplay, user can press the up or down arrows to indicate an emotional change. As with PAGAN/RankTrace, and in order to maintain consistency with PreGLAM's sample rate, button presses are collected every 250 ms, and the user is provided a visual graph of their annotation so far.

We create 20 videos of *Galactic Defense* gameplay. Each video is $\approx 3-4$ minutes in length, and we select clips that have clear changes to their emotional expression, particularly within a single affective dimension, based both on PreGLAMs output during the video and our informal evaluation. Each video has an accompanying annotation file for each dimension, generated by the output of PreGLAM.

Prior to annotating video, participants familiarize themselves with the gameplay of *GalDef*. Training is provided in an interactive gameplay tutorial, a graphical format as shown in Figure 5.9, and a video format is available for them to watch. Participants are

[2]Our software is available on Github at https://github.com/CalePlut/GalDef_Annotation

given 25 minutes to download and play the game, and may complete as many tutorials as they desire during this time. During this free play, an adaptive, generative score accompanies the gameplay, based on the output of PreGLAM. *GalDef* After 25 minutes, participants begin the annotation tasks.

Each participant completes one unbounded annotation curve per video, annotating along a single dimension. At each time step in the annotation software, participants use the up and down arrows on their keyboard to indicate perceived changes in emotion. As with RankTrace, participant annotations are unbounded, and participants are shown a graph with the history of their annotations.

5.6.2 Results

48 participants take part in our study. Of these, 23 use he/him pronouns, and 25 use she/her. 55% of participants report playing between 0-4 hours of games per week, and the average age of participants is 23.60 years old. 39 participants are recruited from undergraduate students at the School of Interactive Arts and Technology at Simon Fraser University, 4 participants are recruited via email and message boards, and 5 participants are recruited using Amazon's Mechanical Turk platform. For all participants, the study is identical.

We analyze our results using Dynamic Time Warping (DTW), with the dtw-python library [19], and calculate the Root Mean Squared Error (RMSE) based on z-score scaling. DTW is a measurement of similarity between two time series that may vary in speed. RMSE is a commonly used measure of the similarity between predicted and actual values. DTW provides a measure of similarity of contour between two time series, and RMSE provides an absolute measure of similarity. These values describe the similarity between the user annotations and PreGLAMs output.

For each participant, we calculate distance values between PreGLAM's output annotation and ground truth participant annotations. Because the input time series' are unbounded, based on an arbitrary unit value of 1, these distance values can only be interpreted in context. To provide an absolute frame of reference, we also compare the ground truth participant annotation with a random walk time series of equal length. We generate a new random walk for each participant. By comparing these distance measures, we evaluate whether PreGLAMs output annotations generally resemble the ground-truth annotations. This demonstrates whether PreGLAM's simulated perceived spectator emotions respond to the gameplay of *GalDef* similarly to the ground-truth perceived spectator emotions.

The distance measure between user annotations and the random walk time series have a mean of 26.08, with standard error mean (SEM) of 0.89. The RMSE mean for the random walk is 1.35, SEM=0.03. In comparison, the mean distance between PreGLAM output and user annotations is 18.24, SEM=0.66, and the average RMSE for PreGLAM is 1.04, SEM=0.03. Figure 5.12 shows the mean and 95% confidence intervals for overall DTW

Table 5.8: Results of t-test by dimension.

Measure	Overall	Valence	Arousal	Tension
Dtw-Distance	$p < 0.01$	$p = 0.08$	$p < 0.01$	$p < 0.01$
RMSE	$p < 0.01$	$p = 0.09$	$p < 0.01$	$p < 0.01$

distance measures for the comparisons of ground-truth annotations to PreGLAM and the random walk. Figure 5.12 also shows these values separated by affective dimension.

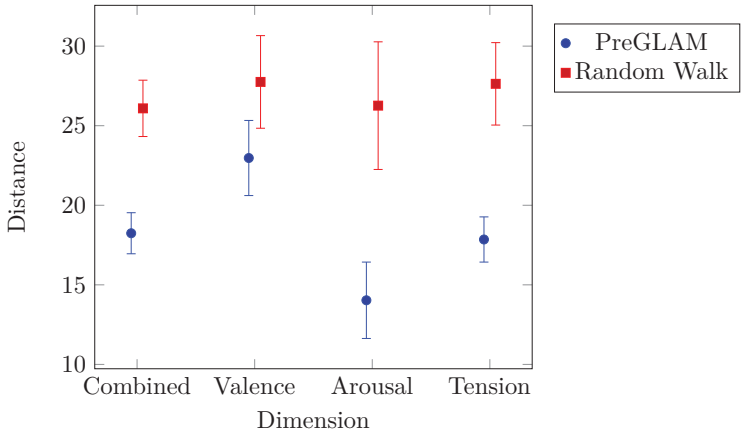


Figure 5.12: Dtw-distance between annotations and PreGLAM, annotations and Random Walk, separated by dimension.

We test the distribution, and find the data is normally distributed in all measures. A t-test finds significant difference between PreGLAM and random walk compared to user annotations, $p < 0.01$ for both metrics. We perform post-hoc two-way t-tests separated by dimension. Results of these t-tests are shown in Table 5.8.

PreGLAM significantly outperforms the random walk in both DTW-Distance and RMSE. Separated by dimension, PreGLAM significantly outperforms the random walk for arousal and tension, not for valence. We perform an ANOVA across all dimensions, and find that the three dimensions are significantly differentiated from another. Post-hoc Tukey tests show that all pairwise comparisons of dimensions are also significantly different — modeled arousal is significantly more accurate than modeled tension, which is significantly more accurate than modeled valence.

As mentioned in Section 5.5.1, we use the output of PreGLAM to inform the adaptivity of a musical score. During our empirical evaluation, multiple different musical scores were played during the videos — each participant annotated one video with only sound effects, and three other musical scores. More information regarding the scores is available in our paper on the matter [47]. Previous research indicates that players report perceiving music as affecting the emotions they experience [46], and the presence of game music that adapts based on the may be expected to influence audience's perceived emotions. The addition of

music does not significantly effect PreGLAM's accuracy, though ground-truth annotations most closely match PreGLAMs annotations when no music is playing.

5.6.3 Discussion

PreGLAM is generally successful at modeling an audience member's perceived emotion. Annotations from PreGLAM are significantly closer than a random walk series to ground-truth annotations. PreGLAM outperforms the random walk series with three different musical scores, including one score that provides potentially confounding emotional arcs. Additionally, participant annotations are closest to PreGLAM's annotations when no music is present — when judging gameplay without any additional emotional stimulus, ground-truth annotations are closer to PreGLAM than when additional emotional stimulus is added.

Previous affect models are mostly used to control either a single affective dimension such as tension [31, 49], or a 2-dimensional Valence-Arousal or Arousal-Tension model [63, 52]. PreGLAM uses 3-dimensional VAT model, corresponding to musical adaptation in all three dimensions. PreGLAM significantly outperforms the random walk for dimensions of arousal and tension, but not valence. Higher degrees of variance when measuring valence compared to other affective dimensions is consistent with previous research in music and emotion, where valence often exhibits the most amount of variance across listeners.

5.7 Future work

As an affect model, PreGLAM has several strengths. PreGLAM uses a flexible framework that can be integrated into the game design process. EEGEs may be derived from design considerations, or may be derived from ML techniques. PreGLAM does not make any assumptions about game mechanics, and therefore can theoretically be applied across a wide range of game genres and interactions.

PreGLAMs design is relatively robust to inaccuracy. PreGLAM sums together the collected emotion values from all local EEGEs to build an aggregate emotion value, and any individual event has a limited ability to affect this aggregated value. Additionally, because PreGLAM models emotional responses within a short local time window, any single incorrect prediction or response will only affect the accuracy of the model for a short time. Finally, prospective events can be either confirmed or disconfirmed, and an incorrect prediction will not affect the modeling of an event when it happens. We note that human spectators, when estimating prospective events, are similarly predicting an incoming event that may be wrong. While PreGLAM has additional information and therefore accuracy, it does not need to achieve perfect predictive accuracy to model a human response.

Another advantage to PreGLAMs temporal locality is that it can describe momentum where global models may struggle. For example, a player using a "control" deck in the game *Magic: The Gathering* often loses most of their health during the game, and appear to be

losing given a global model of the player's health and relative powers. Once the control deck player has the cards that they want, the momentum suddenly shifts. In terms of the global game state, the control deck player is simply evening the game, but in the local view of each turn, the control deck player is dominating the game [10].

Overall, while PreGLAM has many theoretical advantages, primarily related to its lightweight and flexible design, we implement and evaluate only a single, specific use-case. While PreGLAM is genre agnostic, we study PreGLAM only within the action-RPG genre, in a single-player game. While EEGEs may be derived from multiple sources, we only evaluate an implementation where EEGEs are derived as part of the game design process. While PreGLAM's output could be used to influence many game features, we only evaluate its use to control a musical score.

We believe that there are two main avenues for future work with PreGLAM. The first involves expanding PreGLAM's capacity for modeling complex game situations by applying ML methods in the prediction and derivation of EEGEs, and comparing PreGLAM's model more directly to other ML techniques for modeling game emotion. The second avenue involves expanding the breadth of PreGLAM's capabilities, by evaluating alternative implementations with differing game mechanics. In short, we believe that the future work on PreGLAM generally involves examining and evaluating the range of its theoretical capabilities.

5.8 Conclusion

PreGLAM presents a novel approach to modeling the perceived emotion of a passive spectator. PreGLAM uses a flexible framework that describes the actions and events of gameplay as they occur in time. We demonstrate the broad applicability of PreGLAM in presenting numerous theoretical applications, and present an actual implementation in a video game. We implement PreGLAM into our game *GalDef*, manually assigning EEGEs and associated values

We evaluate our implementation empirically, comparing the annotations from PreGLAM with ground-truth annotations provided by gameplay spectators. PreGLAM significantly outperforms a random walk time series, even when confounding music accompanies gameplay. Extending previous game emotion models, PreGLAM uses a 3-dimensional VAT model of emotion, and significantly outperforms a random walk time series for dimensions of arousal and tension, but not valence.

As we discuss in Section 5.7, there a numerous possibilities to extend PreGLAM with ML approaches, and to apply PreGLAM in a wide variety of game genres. Overall, PreGLAM presents a new framework for modeling the perceived emotion of a passive gameplay spectator.

Bibliography

[1] Sander C.J. Bakkes, Pieter H.M. Spronck, and Giel van Lankveld. Player behavioural modelling for video games. *Entertainment Computing*, 3(3):71–79, 2012.

[2] Pablo Barros, Nikhil Churamani, Egor Lakomkin, Henrique Siqueira, Alexander Sutherland, and Stefan Wermter. The OMG-Emotion behavior dataset. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2018.

[3] Shakir Belle, Curtis Gittens, and T.C. Nicholas Graham. Programming with affect: How behaviour trees and a lightweight cognitive architecture enable the development of non-player characters with emotions. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–8. IEEE, 2019.

[4] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. DotA 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[5] Paul Bertens, Anna Guitart, and África Periáñez. Games and big data: A scalable multi-dimensional churn prediction model. In *2017 IEEE conference on computational intelligence and games (CIG)*, pages 33–36. IEEE, 2017.

[6] Julia Ayumi Bopp, Elisa D. Mekler, and Klaus Opwis. *Negative Emotion, Positive Experience? Emotionally Moving Moments in Digital Games*, page 2996–3006. Association for Computing Machinery, New York, NY, USA, 2016.

[7] Zoraida Callejas and Ramon Lopez-Cozar. Influence of contextual information in emotion annotation for spoken dialogue systems. *Speech Communication*, 50(5):416–433, 2008.

[8] J. Clement. Number of hours of video games streamed online 2021. URL: `https://www.statista.com/statistics/1125469/video-game-stream-hours-watched/`, May 2021. Last accessed: 04-25-2022.

[9] NWH Coding. (Unity Plugin) Grapher. URL: https://assetstore.unity.com/packages/tools/utilities/grapher-graph-replay-log-84823, Dec 2017. Last accessed: 04-25-2022.

[10] Paulo Vitor Damo da Rosa. The only point of life that matters is your last. URL: https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/the-only-point-of-life-that-matters-is-your-last/, Oct 2017. Last accessed: 04-25-2022.

[11] Panteleimon Ekkekakis. Affect, mood, and emotion. *Measurement in sport and exercise psychology*, 321, 2012.

[12] Luz Fernández-Aguilar, Beatriz Navarro-Bravo, Jorge Ricarte, Laura Ros, and Jose Miguel Latorre. How effective are films in inducing positive and negative emotional states? A meta-analysis. *PloS one*, 14(11):e0225040, 2019.

[13] N.H. Frijda, Batja Mesquita, J. Sonnemans, and S. Van Goozen. The duration of affective phenomena or emotions, sentiments and passions. In *International Review of Studies on Emotion*, pages 187–225. Wiley, 1991.

[14] Nico H. Frijda. *The laws of emotion*. Lawrence Erlbaum Associates, Mahwah, N.J., 2007.

[15] Riot Games. How to play League of Legends. URL: https://na.leagueoflegends.com/en-us/how-to-play/, 2021. Last accessed: 04-25-2022.

[16] Gege Gao, Aehong Min, and Patrick C. Shih. Gendered design bias: Gender differences of in-game character choice and playing style in League of Legends. In *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, pages 307–317, 2017.

[17] Patrick Gebhard. ALMA: A Layered Model of Affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36, 2005.

[18] Patrick Gebhard, Michael Kipp, Martin Klesen, and Thomas Rist. Adding the emotional dimension to scripting character dialogues. In *International Workshop on Intelligent Virtual Agents*, pages 48–56. Springer, 2003.

[19] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software, Articles*, 31(7):1–24, 2009. Last accessed: 04-25-2022.

[20] Christina Gough. League of Legends Championships viewers 2020. URL: https://www.statista.com/statistics/518126/league-of-legends-championship-viewers/, Mar 2021. Last accessed: 04-25-2022.

[21] Richard L. Hazlett. Measuring emotional valence during interactive experiences: boys at video game play. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1023–1026, 2006.

[22] D. Herremans, Ch. Chuan, and E. Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys*, 50(5), 2017.

[23] Suvi K. Holm, Johanna K. Kaakinen, Santtu Forsström, and Veikko Surakka. Self-reported playing preferences resonate with emotion-related physiological reactions during playing and watching of first-person shooter videogames. *International Journal of Human-Computer Studies*, page 102690, 2021.

[24] Patrick Hutchings and Jon McCormack. Adaptive music composition for games. *IEEE Transactions on Games*, 2019.

[25] Kostas Karpouzis and Georgios N. Yannakakis. *Emotion in Games*. Springer, 2016.

[26] Dae Hee Kwak, Yu Kyoum Kim, and Edward R. Hirt. Exploring the role of emotions on sport consumers' behavioral and cognitive responses to marketing stimuli. *European Sport Management Quarterly*, 11(3):225–250, 2011.

[27] Daniel Leite, Volnei Frigeri Jr., and Rodrigo Medeiros. Adaptive gaussian fuzzy classifier for real-time emotion recognition in computer games. *arXiv preprint arXiv:2103.03488*, 2021.

[28] Antonios Liapis, Georgios N. Yannakakis, Mark J. Nelson, Mike Preuss, and Rafael Bidarra. Orchestrating game generation. *IEEE Transactions on Games*, 11(1):48–68, 2018.

[29] Hendi Lie, Darren Lukas, Jonathan Liebig, and Richi Nayak. A novel learning-to-rank method for automated camera movement control in e-sports spectating. In *Australasian Conference on Data Mining*, pages 149–160. Springer, 2018.

[30] Nintendo Life. (Image) Dark Souls: Remastered screenshot. URL: https://www.nintendolife.com/games/nintendo-switch/dark_souls_remastered/screenshots, Jul 2020. Last accessed: 04-25-2022.

[31] Phil Lopes. *Generating multifaceted content in games: a study on levels and sound*. PhD thesis, University of Malta, 2017.

[32] Phil Lopes and Ronan Boulic. Towards designing games for experimental protocols investigating human-based phenomena. In *International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery.

[33] Tobias Mahlmann, Anders Drachen, Julian Togelius, Alessandro Canossa, and Georgios N. Yannakakis. Predicting player behavior in Tomb Raider: Underworld. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 178–185. IEEE, 2010.

[34] Nikos Malandrakis, Alexandros Potamianos, Georgios Evangelopoulos, and Athanasia Zlatintsi. A supervised approach to movie emotion tracking. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2376–2379. IEEE, 2011.

[35] Ivan Marusic. How many people play League of Legends? URL: `https://leaguefeed.net/did-you-know-total-league-of-legends-player-count-updated/`, Jan 2022. Last accessed: 04-25-2022.

[36] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. PAGAN: Video Affect Annotation Made Easy. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 130–136. IEEE, 2019.

[37] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv preprint arXiv:2104.02643*, 2021.

[38] Belkacem Mostefai, Amar Balla, and Philippe Trigano. A generic and efficient emotion-driven approach toward personalized assessment and adaptation in serious games. *Cognitive Systems Research*, 56:82–106, 2019.

[39] Karen Niven. *Affect*, pages 49–52. Springer New York, New York, NY, 2013.

[40] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[41] Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. Deep player behavior models: Evaluating a novel take on dynamic difficulty adjustment. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[42] Daniel Wei-Shen Phang. *Intelligent camera control in game replays*. Lehigh University, 2014.

[43] Winifred Phillips. *A Composer's Guide to Game Music*. The MIT Press, Cambridge, MA, 2014.

[44] 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.

[45] Cale Plut. (Source code) Galactic Defense. URL: `https://github.com/CalePlut/Galactic_Defense`, Jul 2020. Last accessed: 04-25-2022.

[46] Cale Plut and Philippe Pasquier. Music matters: An empirical study on the effects of adaptive music on experienced and perceived player affect, 2019.

[47] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games. *Foundations of Digital GAmes*, 2022.

[48] Alexandru Popescu, Joost Broekens, and Maarten Van Someren. GAMYGDALA: An emotion engine for games. *IEEE Transactions on Affective Computing*, 5(1):32–44, 2013.

[49] Anthony Prechtl. *Adaptive music generation for computer games*. PhD thesis, Open University (United Kingdom), 2016.

[50] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345, 2000.

[51] Marco Scirea. *Affective music generation and its effect on player experience*. PhD thesis, IT University of Copenhagen, Digital Design, 2017.

[52] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines*, 18(4):433–465, 2017.

[53] Valve Software. Left 4 dead. URL: `https://web.archive.org/web/20090327034239/http://www.l4d.com/info.html`, Mar 2009. Last accessed: 04-25-2022.

[54] Jochen Steffens. The influence of film music on moral judgments of movie scenes and felt emotions. *Psychology of Music*, 48(1):3–17, 2020.

[55] Lijun Sun, Chen Feng, and Yufang Yang. Tension experience induced by nested structures in music. *Frontiers in Human Neuroscience*, 14:210, 2020.

[56] Michael Sweet. *Writing interactive music for video games: A composer's guide*. Addison-Wesley, Upper Saddle River, NJ, 2015.

[57] FrostedSloth (Username), Kyle (Username), and Shawn Saris. (Wiki entry) Dark Souls Game mechanics. URL: https://www.ign.com/wikis/dark-souls/Game_Mechanics, Feb 2013. Last accessed: 04-25-2022.

[58] Nixius (Username). (Image) Wildstar attack telegraph. URL: https://wildstar.fandom.com/wiki/Telegraph, 2014. Last accessed: 04-25-2022.

[59] Thedarkcave (Username). (Image) Screenshot from The Sims. URL: https://thefinkedfilms.com/2020/02/04/the-sims-20th-anniversary/, Feb 2020. Last accessed: 04-25-2022.

[60] Philippe Verduyn, Pauline Delaveau, Jean-Yves Rotgé, Philippe Fossati, and Iven Van Mechelen. Determinants of emotion duration and underlying psychological and neural mechanisms. *Emotion Review*, 7(4):330–335, 2015.

[61] Philippe Verduyn and Saskia Lavrijsen. Which emotions last longest and why: The role of event importance and rumination. *Motivation and Emotion*, 39(1):119–127, 2015.

[62] Markus Viljanen, Antti Airola, Anne-Maarit Majanoja, Jukka Heikkonen, and Tapio Pahikkala. Measuring player retention and monetization using the mean cumulative function. *IEEE Transactions on Games*, 12(1):101–114, 2020.

[63] Duncan Williams, Jamie Mears, Alexis Kirke, Eduardo Miranda, Ian Daly, Asad Malik, James Weaver, Faustina Hwang, and Slawomir Nasuto. A perceptual and affective evaluation of an affectively driven engine for video game soundtracking. *ACM Computers in Entertainment*, 14(3), 2017.

[64] G. Yannakakis, P.H.M. Spronck, D. Loiacono, and E. Andre. *Player modeling*, pages 45–59. Number 6 in Dagstuhl Follow-Ups. Dagstuhl Publishing, 2013.

[65] Georgios N. Yannakakis and Julian Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.

Chapter 6

PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games

As submitted to Plut, C., Pasquier, P., Ens, J., & Tchemeube, R. (2022). *PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games* Foundations of Digital Games

Abstract

We present and evaluate an application of affective adaptive generative music in a single-player, action-RPG video game. We create a score that serves as an audience to the gameplay, based on the output of PreGLAM, which models the emotional perception of a game audience. We use the Multi-track Music Machine to expand and extend a composed adaptive musical score, and we use industry-standard production techniques to synthesize and perform all of our musical scores. We evaluate our application of generative music in comparison to two composed scores, one adaptive and one linear. Our generative score is rated as nearly equivalent to a composed linear score in perceptions of emotional congruency, immersion, and preference.

6.1 Introduction

Music is present in some form in almost all video games. Most music in games is composed by one or more humans, and is either performed by human musicians and/or synthesized into audio format. While music is generally linear, and plays without reacting to external

input, video games are interactive, and respond to the inputs of one or more players. To create music that matches gameplay, video game composers may use "adaptive music", sometimes called "interactive music", which is music that can be altered based on a control input. Adaptive music is a powerful tool for creating music that matches gameplay, but using adaptive music requires specific techniques that can significantly increase a composers workload. Adaptive music is primarily used when music is serving as an "audience" to the gameplay, commenting on the successes and failures of the player [16].

Generative music is created with some degree of systemic autonomy from its input. Because video games almost universally have some degree of systemic autonomy from their input, it may be argued that all game music is generative. However, we define generative music in games as having systemic autonomy from the game logic [21]. For example, if a single piece is cued when the game state changes in an identical fashion each time, we do not consider this generative. Depending on the algorithm, generative music systems are capable of producing large amounts of musical content in minutes, seconds, or even in real-time.

There are two main approaches to applying generative music in video games [21], which we will discuss further in Section 6.2.2. Academic research generally focuses on the use of novel algorithms for online real-time generation of symbolic music to entirely replace a composed score [28, 31, 11, 17], while approaches from the games industry primarily use stochastic methods to target real-time sequencing of audio stems.

Academic systems most commonly generate and synthesize music in real time with General MIDI sounds. These systems mostly use some form of player experience model, commonly affect-oriented, to control the adaptivity of the generative music. These systems produce novel music that can theoretically match the events of a game, but lack timbral and performative features when compared to contemporary video games.

Systems from the games industry generally use pre-rendered or recorded audio stems, sequenced together with stochastic methods. These systems generally extend adaptive musical methods of *horizontal resequencing* and *vertical remixing* [29]. This approach produces music that has equal performative fidelity to linear music, but can often reduce the expressive range of the music, as the music must be composed so that the combined arrangements won't clash with each other [16].

We present a hybrid approach to utilizing generative music in video games, discussed further in Section 6.3.2. We use the Multi-track Music Machine (MMM) [5] transformer model to generate multi-track symbolic variations of a composed adaptive score, as we discuss in Section 6.3.3. The composition of our adaptive score is informed by previous research in composing music to express desired affect in a Valence-Arousal-Tension (VAT) model of emotion [23]. We use the musically-focused audio middleware program *Elias* [4] to control our adaptive scores based on the output of PreGLAM. We also compose a linear score that is based on the adaptive score.

To model the gameplay emotion and inform the musical adaptivity, we use the *Predictive Gameplay-based Layered Affect Model* (PreGLAM) [22] as discussed in Section 6.2.3. PreGLAM is an artificial cognitive agent with privileged game information, that models the real-time perceived affect of a biased game spectator. We implement PreGLAM as biased towards the player winning the game, though other biases may be provided.

We use VST instruments to render our scores into audio, to increase the quality of synthesis compared to previous uses of General MIDI. At the time of this writing, real-time synthesis of symbolic music during gameplay is unable to match the quality and fidelity of offline synthesis, such as by VST instruments. We therefore render our symbolic tracks via Ableton Live.

Our approach focuses on providing an application of generative music that builds on previous literature in the area, while increasing the synthesis, production, and performance fidelity of musical scores from previous applications. Our approach also increases the expressive range of the music compared to previous attempts, by utilizing a 3-dimensional VAT model of emotion. We additionally evaluate our generative score in comparison to an adaptive score and a linear score that share identical production methods. Effectively, we target an increase in external validity compared to previous applications, without sacrificing experimental control.

We empirically evaluate our application of generative music in a study with 48 participants, and find that our application of generative music performs consistently with linear music, and outperforms composed adaptive music in participant perception of emotional congruency, player immersion, and preference. Our approach is directly compared to music that is produced using industry-standard techniques.

6.2 Background

6.2.1 Adaptive music in games

Music can serve multiple functions in games, and occasionally serves multiple functions simultaneously. Winifred Phillips describes one function of music in games as acting as an "audience", which is described as creating a feeling that the music is "essentially watching the gameplay and commenting periodically on the successes or failures of the player" [16].

When composing music to act as an audience, there is an inherent mismatch in the relationship that games and music have with time. Games are interactive, and react to the actions and events of one or more player or non-player agents. Music is most often linear, and generally does not react to external changes. Adaptive music allows for music to be altered based on some control input, such as player health, number of enemies, or game progress [29].

Adaptive music can be a powerful tool for using music as an audience, and Phillips describes adaptive music as "constituting the most complex realization of the music-as-

audience approach" [16]. Perhaps the strongest drawback of adaptive music is that it requires a large amount of time investment, and requires early integration into the game design to be effective. Compounding these issues is that music and sound often have fewer resources, lower budgets, and may be added later in the development process than other game features [29, 16].

There are two main techniques for creating adaptive music: Horizontal resequencing, and vertical remixing [29]. Music is often read left-to-right through time, and horizontal resequencing refers to the adaptive alteration of music through time. In horizontal resequencing, the music generally adapt to game state — musical cues will loop until certain conditions are met or the game state changes, a transition is played, and a new musical cue begins looping, matching the new game state.

Instruments in sheet music are vertically aligned, and vertical remixing refers to the adaptive addition or subtraction of audio stems, depending on the input. When using vertical remixing, the music generally responds to some variable such as "intensity", and adds or subtracts tracks based on a provided mapping. *Mass Effect 2* presents a common use of vertical remixing: the music in *Mass Effect 2* adapts based on a measure of combat intensity, adding additional layers as combat becomes more intense [2].

6.2.2 Generative music in games

Generative music, also known as procedurally generated music or algorithmic music, is music that is partially or wholly created by some form of systemic autonomy [15]. Depending on the specifics of a particular system, generative music algorithms are capable of generating music quickly, potentially in real-time, based on a set of input parameters. Because generative music can produce music quickly and produce large amounts of music based on the provided input, it may be used to address the drawbacks to using adaptive music.

Plut and Pasquier survey uses of generative music in video games in both the games industry and academic research, and identify several trends [21]. Primarily, generative music in the games industry is used to extend composed scores in the audio domain, mostly through stochastically re-arranging musical cues and stems based on input from the game. In contrast, generative music in academic research mostly targets the replacement of a composed score with adaptive music generated and synthesized in real time.

Academic applications

Academic approaches primarily focus on applying novel generative algorithms to create a general system capable of real-time, adaptive symbolic music generation. These systems commonly use generative music instead of composed music, with the musical adaptivity most often based on an affective model of player experience. These affective models generally map a set of game variables to one or more affective dimensions. Academic systems are often

empirically evaluated, and the evaluation is often focused on whether the generated music is perceived as expressing similar affect to the game.

Plans and Morelli create a system that generates music for the MarioAI Championship engine, a game used in procedural level generation research [17]. Plans and Morelli describe an "excitement" metric based on aggregate counts of game events and variables, and map several musical features to the excitement metric. A harmonic sequence are generated by a genetic algorithm design, using notes from the C major scale or a subset of notes from the C major scale. Additionally, a melody is created, first by creating set of phrases are generated by applying minor transformations to a smaller set of composed phrases, and then combining a sequence of these phrases into a melody. The music is synthesized by the "SawLPFInstRT2" instrument, from the Jmusic library [3].

Plans and Morelli evaluate their system by comparing results from the generative system to a precomposed linear MIDI track. Plans and Morelli collect the output of their affect model from playthroughs, and ask player-participants to rate a level of enjoyment after playing. While the gameplay-derived frustration value was on average lower when utilizing generative music, other measures, including self-reported enjoyment, are consistent between the two conditions.

Prechtl presents a system that uses weighted Markov models to generate real-time chord progressions, which can be played by a single loaded VST instrument. The chords are played both as a block chord and an arpeggio, and the chord contents are selected base on an input "tension" value. Prechtl also presents a horror-genre game *Escape Point*, created to implement and evaluate the generative system. Prechtl maps a tension value to the distance between the player and the nearest mobile object (mob) while navigating a maze. Mobs follow a pre-determined path, and if the player comes into contact with a mob, they lose the game.

Prechtl evaluates his system, and finds that the adaptive generative score invoked more tension and excitement based on skin conductance. After playing *Escape Point*, participants report perceiving more tension and excitement with the adaptive generative score than with linear generative music or no music [24]. Participants who like the horror genre prefer the generative score to the linear score or no music, and find the game more fun to play with the generative score. However, all three conditions are evaluated as roughly equal in preference and fun ratings among participants who do not like the horror genre.

Scirea presents *Metacompose*, which uses hybrid evolutionary techniques to generate music [27]. Metacompose generates a chord progression, and evolves a melody based on that chord progression. Metacompose then realizes the chord progression into an accompaniment in the form of a block or arpeggiated chord. The music generation responds to input values for the dimensions of valence and arousal.

Scirea implements and evaluates Metacompose in the game of checkers, synthesized via a solo piano. A valence value is determined by evaluating "how good the current board

configuration is for the human-player", and an arousal value is determined by evaluating the range of evaluations for possible moves, described as reflecting the sentiment "How much is at stake for the next move?". Metacompose outperformed random music and non-adaptive music in an empirical user preference study.

Williams et al. present an "affectively-driven algorithmic composition" system that primarily uses Markov generation with post-hoc transformations for affective expression [31]. While this system is capable of real-time generation, generated sequences were rendered into audio files, played on a solo piano, for the evaluation.

To evaluate their generative system, Williams et al. select a specific in-game section of the MMO *World of Warcraft*. Situations that occur within the section are manually tagged with affective targets, and the music system selects generated clips to match the affective target. Williams et al.'s system outperforms both the composed score and silence in user ratings of "emotional congruence", in gameplay, and outperforms silence in user ratings of immersion. However, the generated score shows a "marked decrease" in user ratings of immersion compared to the composed score.

Industry applications

Industry applications of generative music most commonly sequence composed and pre-rendered or recorded audio stems together in new ways. Mick Gordon describes an example of using generative music to extend horizontal resequencing in *DOOM (2016)* [25]. Gordon assigns fully arranged clips into "buckets", that generally follow a structure such as "verse", "chorus", and "bridge". During gameplay, while certain conditions are met, the system continuously randomly selects clips from within a bucket. When conditions change, the system plays a transition as in typical horizontal resequencing, and then being playing randomly selected clips from the new bucket.

Red Dead Redemption makes aggressive use of generative music addressing the arrangement task [26] — in *RDR*, all music is written at 130 beats per minute, in the key of a minor. The music in *RDR* is divided into orchestral function e.g. "melody" or "bass", and associated game states e.g. "riding horse" or "combat". When the game state changes, the generative system in *RDR* selects a set of instruments/functions, and randomly selects a loop for each selected instrument. Additionally, the system adds or removes layers based on game variables within some situations, using elements of both horizontal resequencing and vertical remixing.

6.2.3 PreGLAM

As mentioned in Section 6.2.2, the most common application of generative music in games uses an affect-based model of player experience to influence the adaptivity of the generative score. The adaptivity of our score is influenced by the Predictive Gameplay-based Layered Affect Model, or *PreGLAM* [22]. *PreGLAM* is a cognitive agent that models a spectator

Figure 6.1: A possible flank in XCOM.

with a provided bias. In our implementation, we use PreGLAM to model an audience who is biased in favour of the player to create music that affectively comments on the successes and failures of the player.

PreGLAM is a layered, gameplay-based affect model, based on NPC models of affect [6, 1]. A base mood value is provided to PreGLAM that represents a general, environmental affective feeling. PreGLAM models emotions as the responses to emotionally evocative game events (EEGEs). EEGEs have a provided base emotion value, a set of intensity modifier variables, and a time scalar. PreGLAM calculates an output affect value for each dimension based on the provided mood value, as well as the summed emotional responses to EEGEs, modified by their intensity modifier variables and time scalar.

PreGLAM models EEGEs that occur in the game, and also models emotional responses to prospective EEGEs. Prospective events are events that PreGLAM expects to happen, given the current state of the game. One example of how we model prospective events can be seen in Figure 6.1, which shows a scenario from *XCOM: Enemy Within*. In Figure 6.1, the player's selected unit is able to flank an opposing unit. The opposing unit is otherwise in cover, which gives it a tactical advantage that can be removed when flanked. Because the gameplay in *XCOM* primarily involves manipulating tactical positioning in relation to cover, we can predict that the player will move their unit into a flanking position and attack the opposing unit. Importantly, PreGLAM models that a spectator emotionally perceives the possibility of a flank even if the player does not take the predicted action — the prospect of the flank is not affected by whether or not it is realized.

PreGLAM is integrated into the game *Galactic Defense* (GalDef), which will be further described in Section 6.2.5. PreGLAM's application in GalDef is based on informal experiential playtesting, with all EEGEs, mood values, and conditions for predicting EEGEs created

during the design process. We assign threshold values for 5 levels of each dimension, which influences the adaptivity of our musical score.

6.2.4 IsoVAT Composition guide

Plut et al.'s IsoVAT composition guide presents a set of Western musical features, and the perceived changes in emotional expression that changes in these musical features are associated with [23]. This guide is aggregated from a broad overview of research in music and emotion. The IsoVAT guide represents emotion using a VAT model, and is intended to be used across Western pop, jazz, and classical genres. For example, increases in the melodic range, contour, and direction are strongly associated with increases in arousal, while decreases in harmonic consonance are strongly associated with increases in tension.

The IsoVAT composition guide is empirically evaluated by producing a corpus of clips that express varying levels of valence, arousal, or tension. These clips are organized by the affective dimension that they manipulate, and further divided into sets of 3. Each set shares an instrumentation and genre, and is composed to express a low, medium, and high level of the assigned emotional dimension.

6.2.5 Galactic Defense

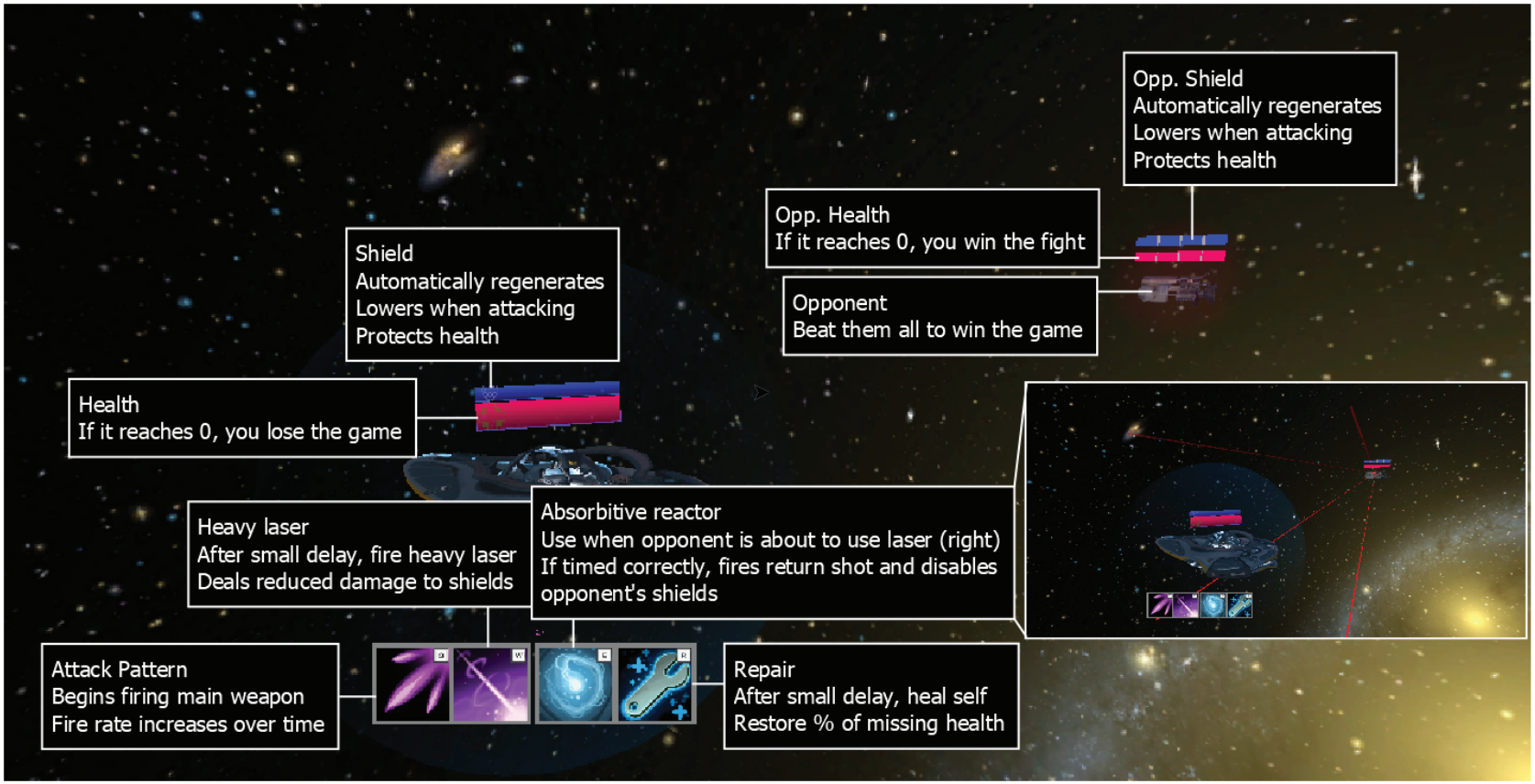


Figure 6.2: Visual tutorial for Galactic Defense.

Following a common approach in research into games and emotion [10, 11, 13, 24], we use a custom-designed game to implement and evaluate our application of generative music in games. *Galactic Defense*, or *GalDef*, is further described alongside *PreGLAM*, as *GalDef* implements PreGLAM [22]. Figure 6.2 provides an annotated screenshot of gameplay. We integrate our adaptive scores into *GalDef*, using PreGLAM's output to control the adaptivity. *GalDef* also serves as an environment for evaluating our application of generative music.

GalDef is an action-RPG game, where the player uses a set of abilities to defeat a series of opposing units in real-time. The abilities that the player has access to have situational strengths and weaknesses, with the intent of encouraging moments of gameplay where the player is appraising the current game state, and using that appraisal to make choices about their next move. The player must manage a small, recharging limited resource pool for both themselves and the opponent, and must take care not to use certain abilities while under threat of attack.

The player controls a spaceship in *Galactic Defense*, and must defeat several opposing AI-controlled spaceships to win the game. The player has four moves, which are shown in Figure 6.2. In terms of resources, the player has a weak shield that constantly recharges, and a pool of health points. When the player uses any ability, the shield is temporarily deactivated, and therefore any incoming attack will directly drain health points. All opponents have the moves of *attack pattern*, *heavy laser*, and *repair*.

Both the heavy laser and repair abilities are interruptible when used by the player. If the player receives any damage while using these abilities, the damage will be multiplied and the ability will be cancelled. Most of the gameplay in *GalDef* is in tactical decisions of when to use each of the four moves. The basic attack pattern does small but consistent damage, the heavy laser deals large damage in some situations, but is vulnerable to counterplay. The "absorbitive reactor" parry ability is extremely powerful, but requires precise timing and is purely situational. Self-repair is often necessary, but as with the heavy laser, the player is vulnerable while using it. This design provides fluid gameplay and highlights the contextual nature of game emotions.

6.2.6 PreGLAM implementation

Figure 6.3 shows how PreGLAM appraises game data to select music, based on a perceived valence, arousal, and tension, acting as an audience.

Mood values are provided to PreGLAM based on the designed difficulty levels of each gameplay segment. Each gameplay segment involves 2-3 combat encounters, which rise in difficulty as the game progresses. We model PreGLAM with a desire of the player winning, and derive a set of EEGEs, shown in Table 6.1. In Table 6.1, we abbreviate "Player" to "P", and "Opponent" to "O". These EEGEs are created through an iterative process of playtesting with a focus on informal evaluation of experienced emotions.

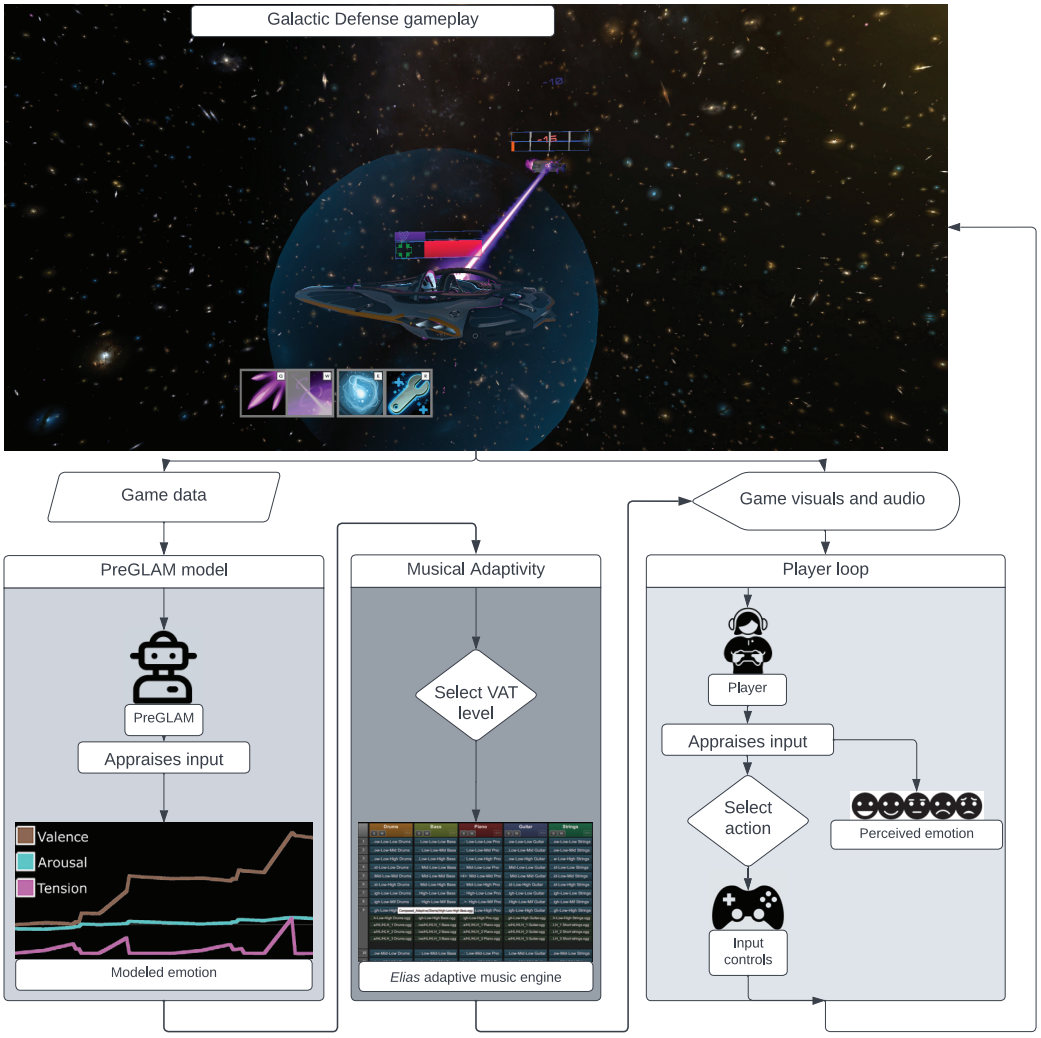


Figure 6.3: Diagram showing how PreGLAM-MMM fits into game loop.

Table 6.1 gives the base assigned value for the associated emotional perception of each event. These values are based on an initial unit of 1, and values represent the intensity of the emotional response to the EEGE. We represent all intensity modifiers as percentages, which scale the emotional values between 100 and 200%. Tension values are only computed for prospective events, as tension arises from the prospect of events [14]. As an example, the "Player shield down" EEGE has a base value of -2 valence, 1 arousal, and 2 tension. These values are modified based on how much health the player has remaining — losing the shield is more of a problem if the player's health is also low. If the player has, e.g. 50% of their maximum health and is expected to lose their shield, output values will scale to 150% of their base value, and the output values at the moment that the shields are expected to go down are 3 valence, 1.5 arousal, and 3 tension. We note that during actual gameplay, these

values are additionally scaled through time. As mentioned, PreGLAM is further described in its own paper [22].

Table 6.1: Emotionally evocative events in *GalDef*.

Event	Valence	Arousal	Tension	Modifiers
P. complete atk combo	1	1	1	Missing O. shield
P. heavy atk	1	1	1	Missing O. health
O. atk combo	-1	1	1	Missing P. shield
O. heavy atk	-2	1	2	Missing P. health, Parry active
P. shields down	-2	1	2	Missing P. health
O. shields down	2	1	2	Missing O. health
P. exploit O. disable	3	1	2	Missing O. health
P. death	-3	1	3	P. shield recharge time
O. death	3	1	3	O. shield recharge time
P. heal	2	1	2	Missing P. health
O. heal	-2	1	2	Missing O. health

6.3 Musical scores

6.3.1 Linear score

We compose a linear score that attempts to create moments of "serendipitous sync" [29], where a linear score that is written with changes in emotion over time occasionally synchronize with the changing emotions of gameplay. This score is musically based on the adaptive score, and mostly consists of manually re-arranged tracks and sections of tracks from the adaptive score. We arrange the musical ideas from the adaptive score into a linear score that has varying rises and falls in valence, arousal, and tension. The approximate levels of each dimension through the linear score's 128 bars is shown in Figure 6.4. As we expect the gameplay of GalDef to also demonstrate moments of rising and falling valence, arousal and tension, we expect that there may be moments where the linear music aligns with the GalDef's perceived emotion. The linear score is available to listen on SoundCloud [18].

6.3.2 Adaptive score

We compose our affectively adaptive score following the IsoVAT composition guide [23], as described in Section 6.2.4. The IsoVAT guide provides an ordinal description of how musical features affect emotional perception, and we use the guide to create clips that express three levels of each dimension: low, medium, and high. While the IsoVAT corpus adjusts music along individual dimensions, we use the guide to compose a 3-dimensional adaptive score, that can express any combination of 3 levels of 3 affective dimensions. Therefore, we compose 3^3, or 27 clips.

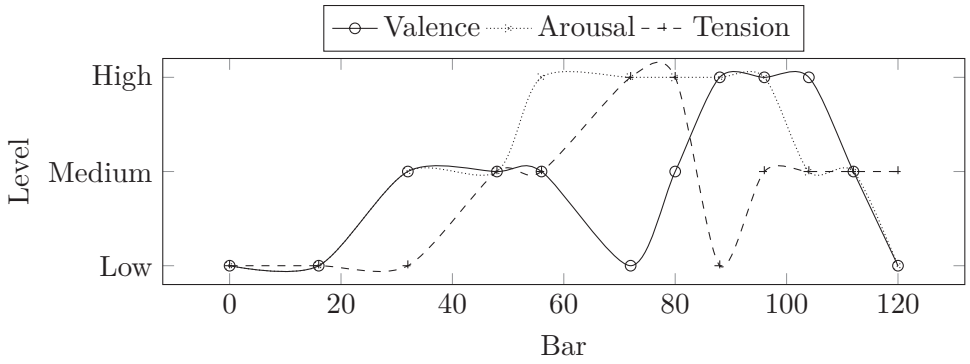


Figure 6.4: Affective levels by bar in the linear score.

Each clip is at a tempo of 130 beats per minute. Each clip has 5 tracks, and is composed for the same instrumentation, divided into "melody" and "rhythm/harmony" sections, as shown in Table 6.2. Table 6.2 also provides the VST instrument used for each instrument. We note that the guitar part alternates between using a distorted electric effect and using an acoustic guitar, and the two guitar parts share a track. We also note that while piano is often considered a rhythm section instrument, we use piano as a melody instrument in our adaptive score. The use of VST instruments will be further discussed in Section 6.3.4.

We expand our 3 levels of adaptivity into 5, adding medium-low and medium-high levels via adaptive re-sequencing. These levels are differentiated by instrument section — only the melody section adapts to medium-low and medium-high levels. The rhythm section, in contrast, only adapts to levels of low, medium, and high. Section levels are independently set, so when transitioning from a high or low level to a medium-high/low level, the rhythm section continues to play the high/low clips until the corresponding dimension reaches a medium level. This further expands our adaptivity from 5 to 7 possible output levels for each dimension: low, low→medium, medium→low, medium, medium→high, high→medium, and high, creating a total of $7^3 = 343$ unique arrangements. Due to the interactive nature of our adaptive and generative scores, we implement a "Music explorer" in *GalDef*, where users can freely navigate the emotion space of the score outside of the gameplay.

Table 6.2: Instrumentation of *Galactic Defense* score.

Instrument	Section	VST bank	VST instrument	VST source
Bass	Rhythm	Analog Essentials	80ties Dance	Applied Acoustic Systems
Drums	Rhythm	LABS	Drums	Spitfire Audio
Strings	Rhythm	BBC Symphony Orchestra	Violas	Spitfire Audio
E. Piano	Melody	Lounge Lizard Session	Bite	Applied Acoustic Systems
Guitar (Electric)*	Melody	Strum Session	Ballistic Squeeze	Applied Acoustic Systems
Guitar (Acoustic)*	Melody	Strum Session	Dreadnought Smooth	Applied Acoustic Systems

In terms of harmonies, the keys/modes used in the clips are:

1. b minor/aeolian, primarily used for low valence

Table 6.3: MMM Generation parameters.

Parameter	Tracks per step	Bars per step	Shuffle	Percentage	Temperature	Model size
Value	1	4	True	90%	1.0	8-bar

2. D Major/Ionian, primarily used for high valence

3. G Lydian, primarily used for high valence with high tension

These keys and modes share a key signature, and therefore the adaptive score can theoretically navigate the harmonic space without jarring transitions.

6.3.3 Generative score

We utilize Ens and Pasquier's *MMM* 8-bar transformer model, which generates symbolic multi-track music [5], using the parameter settings in Table 6.3. While MMM has a host of features, we primarily use *Bar inpainting*. Bar inpainting involves resampling a subset of the bars present in one or more tracks, or altering a subset of musical material conditioned on the remaining unaltered musical material.

As mentioned in Section 6.3.2, our score is composed as a set of 8-bar clips, and the instrumentation is separated into sections. Because the melody and rhythm sections adapt as groups, we condition the generation of new melody bars on existing rhythm bars, and the generation of new rhythm bars on existing melody bars. We create 3 additional variations per section, for each of the 27 clips in our adaptive score. When the music adapts, we randomly select from the 4 possible variations (1 composed and 3 generated) independently for each instrument. This creates a total of $343^4 = 13,841,287,201$ unique arrangements.

The MMM model is currently too heavy to generate music in real-time. However, we believe that the amount of generative musical content is indistinguishable from real-time generated music during gameplay in terms of musical variety. By utilizing offline generation, we are able to increase the audio quality over previous real-time uses of symbolic generative music. As technology improves, we believe that our approach could implement real-time generation.

6.3.4 Synthesis and Arrangement

Video game composers commonly use libraries of virtual instruments to provide some or all of the synthesis of their music [16]. These virtual instruments are generally controlled via MIDI, though may be controlled with tracker or piano roll format, and input can be recorded on MIDI controllers and/or manually adjusted. As our score is in MIDI format, we use VST instruments to synthesize both our composed and generative score.

For our composed scores, we record data from a MIDI keyboard directly into *Ableton Live*, a common digital audio workstation (DAW). We primarily use VST sources from

Figure 6.5: Screenshot of participant annotation interface.

Spitfire audio's LABS libraries [8] and libraries from Applied Acoustic Systems [30]. We record the performance at 1/2 speed, played on a MIDI keyboard - this ensures precision in following the composed score while allowing for human articulation and velocity data.

Each instrument part has 27 unique levels, encompassing the VAT space that the adaptive composition expresses in total. As mentioned in Section 6.2.3, we label thresholds for PreGLAM's output to trigger a corresponding categorical level of each emotional dimension. We use "smart transitions" in *Elias*, which attempts to transition individual parts only during silence based on an analysis of the audio file. This creates transitions that somewhat more resemble transitions using symbolic notation instead of rendered audios, as there is some musical consideration for the timing of the transitions.

6.4 Empirical Evaluation

6.4.1 Empirical Methodology

To evaluate our application of generative music in video games, we collect real-time user annotations from 48 video spectators. Our annotation software is available on GitHub [19], and is similar to *RankTrace* [9] and *PAGAN* [12]. Our annotation interface is shown in Figure 6.5. While watching a video of gameplay, user can press the up or down arrows to indicate an emotional change. As with RankTrace and PAGAN, unbounded input is collected every 250 ms, and the user is provided a visual graph of their annotation so far.

Table 6.4: Empirical study conditions.

Condition	Music Source	Adaptivity	Relevant Section
No music	None	N/A	N/A
Linear score	Composed	Linear	6.3.1
Adaptive score	Composed	Adaptive	6.3.2
Generative score	Generative	Adaptive	6.3.3

We create 20 videos of *Galactic Defense* gameplay. Each video is $\approx 3-4$ minutes in length, and we select clips that have clear changes to their emotional expression, particularly within a single affective dimension, based both on PreGLAMs output during the video and our informal evaluation. Each video has an accompanying output file generated by PreGLAM. We divide these videos into 4 sets of 10, based on the source and adaptivity of the musical accompaniment, as shown in Table 6.4.

Prior to annotating video, participants familiarize themselves with the gameplay of *GalDef*. Figure 6.2 shows an image tutorial, and a video tutorial is available for them to watch [20]. Participants are given 25 minutes to familiarize themselves with *GalDef*. During this 25 minutes, after downloading and completing the tutorials for the game, players freely play *GalDef*. After the 25 minutes, participants begin the annotation tasks.

Each participant completes one annotation curve per condition per video, annotating a single affective dimension, for a total of four annotation curves per participant. After completing their annotation, participants are presented with the four videos that they provided annotations for. They are then asked to select one video for each of the following questions:

1. In which video do you feel the music most closely matches the events and actions of the gameplay? (Gameplay match)

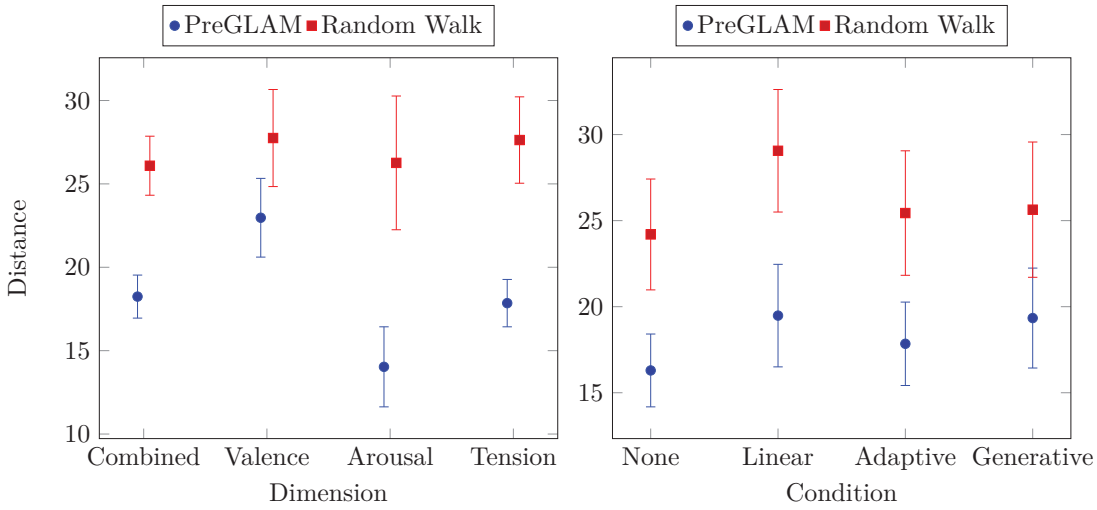
2. In which video do you feel that the music most closely matches the emotion that you perceive from the gameplay? (Emotion match)

3. In which video did you feel most immersed in the gameplay? (Immersion)

4. Which video's music did you enjoy the most? (Preference)

6.4.2 Results

48 participants take part in our study. Of these, 23 use he/him pronouns, and 25 use she/her. 55% of participants report playing between 0-4 hours of games per week, and the average age of participants is 23.60 years old. 39 participants are recruited from undergraduate students at the School of Interactive Arts and Technology at Simon Fraser University, 4 participants are recruited via email and message boards, and 5 participants are recruited using Amazon's Mechanical Turk platform. For all participants, the study is identical.



(a) Average distances and standard error by affective dimension.

(b) Average distances and standard error by musical condition.

Figure 6.6: DTW-Distance between PreGLAM and annotations, compared with Distance between random walk and annotations.

We analyze our results using Dynamic Time Warping (DTW), with the dtw-python library [7], and calculate the Root Mean Squared Error (RMSE) based on z-score scaling. Table 6.5 shows these values, and Figure 6.6 shows the DTW Distance and 95% confidence interval. DTW is a measurement of similarity between two time series that may vary in speed. RMSE is a commonly used measure of the similarity between predicted and actual values. These measures provide both a measure of contour similarity with DTW, and overall similarity with RMSE.

In Table 6.5, the responses for musical condition are aggregated across affective dimension, and the dimension responses are aggregated across conditions. In other words, the DTW Distance between PreGLAM and the ground-truth annotations for the "linear" condition represents the combined average distance of valence, arousal, and tension annotations when the linear score is played.

Each participant's annotation curve is compared directly to PreGLAM's output annotation. Additionally, we provide a more absolute measure by comparing each participant's annotation curve to a random walk time series. These results therefore demonstrate the distance measures between PreGLAM and ground-truth annotations, in comparison to the distance measures between the random walk and the ground-truth annotations.

We test the assumption of normality, and find that the data is normally distributed in all four measures. We perform a t-test to compare results and find significant difference between PreGLAM and random walk compared to user annotations, $p < 0.01$ for both metrics. We perform post-hoc two-way t-tests separated by condition and dimension. Results of these t-tests are shown in Table 6.6

Table 6.5: Results by musical condition and dimension.

Measure	Model	Result	None	Linear	Adaptive	Generative	Valence	Arousal	Tension
DTW	PreGLAM	Distance	16.30	19.48	17.84	19.33	22.52	13.52	17.39
		SEM	1.05	1.48	1.20	1.44	1.19	1.20	0.71
	Random walk	Distance	24.20	25.63	25.44	25.64	27.53	24.61	25.71
		SEM	1.60	1.95	1.80	1.96	1.46	2.00	1.30
RMSE	PreGLAM	RMSE	1.06	1.04	0.99	1.07	1.23	0.73	1.08
		SEM	0.06	0.06	0.05	0.06	0.04	0.05	0.04
	Random walk	RMSE	1.34	1.38	1.35	1.38	1.36	1.28	1.41
		SEM	0.05	0.06	0.06	0.06	0.04	0.06	0.04

Table 6.6: T-test results by musical condition and dimension.

Measure	None	Linear	Adaptive	Generative	Valence	Arousal	Tension
Dtw-Distance	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p = 0.08$	$p < 0.01$	$p < 0.01$
RMSE	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p = 0.09$	$p < 0.01$	$p < 0.01$

PreGLAM significantly outperforms the random walk in both DTW-Distance and RMSE across all conditions. We perform an ANOVA across all conditions, and find no significant effects from changes in musical condition. Separated by dimension, PreGLAM significantly outperforms the random walk for arousal and tension, but does not significantly outperform the random walk for valence measures. We perform an ANOVA across all dimensions, and find that the three dimensions are significantly differentiated from another. Post-hoc Tukey tests show that all pairwise comparisons of dimensions are also significantly different — modeled arousal is significantly more accurate than modeled tension, which is significantly more accurate than modeled valence.

Figure 6.7 shows the distribution of questionnaire responses. In these responses, the composed linear score is rated as the highest in all questions. In terms of emotional congruency, immersion, and preference, the generative score is rated as a close second, with the composed adaptive score in a more distance third. In terms of matching the events and actions of the gameplay, the adaptive score slightly outperforms the generative score.

6.4.3 Discussion

Overall, PreGLAM presents a viable emotion model for controlling adaptive music, outperforming a random walk in matching ground-truth annotations. There is a marginal increase in the distance between PreGLAM's ouput and user annotations when any music is introduced. Within the musical conditions, the distance is lowest with composed adaptive music, and highest with the composed linear score. There are no significant differences between the distances between PreGLAM and ground-truth annotations when separated by the musical condition. In other words, the musical conditions are not significantly differentiated from each other according to real-time perceived ground-truth annotations.

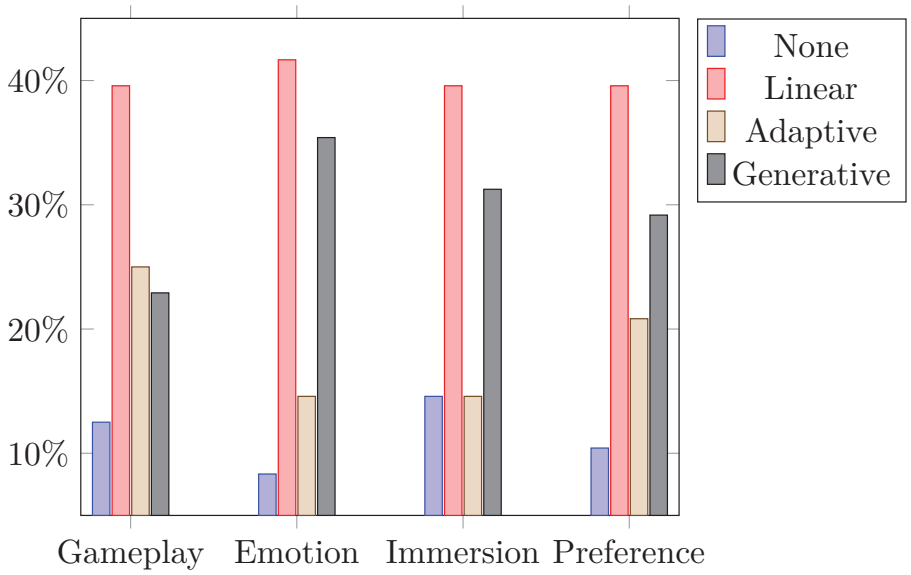


Figure 6.7: Distribution of questionnaire responses.

The post-hoc questionnaire questions indicate support for the generative approach. As mentioned in Section 6.2, Williams et al. find that participants report a decreased immersion when playing with a single-instrument MIDI track, compared to an original orchestral score. We address this by using identical production processes across our three musical conditions, therefore isolating the compositional element of the musical generation. Participants judge the generative score slightly lower than the linear score in all questions, but generally much higher than the composed adaptive score. The composed adaptive score outperforms the generative score in terms of matching the actions and events of gameplay, but the generative adaptive score presents an increase in participant ranking for perceived emotional congruency, immersion, and preference.

The linear score is the only score that has composed transitions. While the linear score does not adapt its emotional expression based on gameplay, it does have rising and falling emotional arcs through time, and may produce serendipitous sync [16]. In Williams' previous research, the generative affective score is compared with a linear score that has a mostly consistent emotional expression. While Williams' generative system outperforms their compared linear score in emotional congruency, the linear score outperformed the generative score in immersion [31].

While our generative score is close in ranking to our linear score, our linear score outperforms our generative score in all questionnaire responses. This may seem to show a step backwards from the work presented by Williams et al. [31]. We draw attention to several differences that may explain some of this discrepancy. Our emotional model adapts in response to the actions and events of gameplay in real-time, rather than associating each emotion with a single game state. Our linear musical score changes in emotion over time,

rather than expressing a mostly static affect. Additionally, our linear, generative, and adaptive scores are synthesized using identical production techniques, bringing musical features such as instrumentation, timbre, tempo, genre, synthesis, and production quality to parity with the generative music. We believe that this provides a more isolated understanding of the compositional aspects of the generative score.

A linear score may be preferred by listeners due to the smoothness of transitions, and the pre-determined intentionality of its emotional expression. Contrastingly, our composed adaptive score has a limited amount of musical content for each adaptive level compared to the generative score, which may lead to the musical transition point between adaptive levels in the composed score being jarring and/or repetitive. While the application of generative music does not bring an adaptive score to full parity with a composed linear score in post-hoc participant responses, the generative score improves upon our adaptive score and upon previous applications of generative music in games.

Overall, these results indicate that while the real-time perceived effects of musical accompaniment to gameplay shown in Table 6.5 and Figure 6.6 are small, our approach to generative music is mostly comparable to linear music in terms of the emotional congruency, immersion, and preference in post-hoc responses from participants, and improves upon these features compared to purely human-composed adaptive music. This demonstrates the strength of MMM in assisting a composer to create and extend a highly adaptive score with generative music.

6.5 Conclusion

We identified several differences between academic approaches to using generative music in games, and approaches taken from the games industry. Academic systems tend to use MIDI synthesis of symbolic generative music, often with a single piano instrument. Academic systems generally use an emotion model that directly relates the absolute values of game variables to emotion values for one or two dimensions. Systems from the game industry generally use audio recordings of instruments and/or VST instruments to synthesize and produce the music offline. Industry systems rarely use an abstracted model of emotion, instead directly relating a set of game variables to musical adaptivity.

We present a hybrid approach to using generative music in video games that uses generative composition to extend and expand a composed adaptive score. This approach attempts to utilize the advantages of using advanced generative music algorithms within a score that is aesthetically similar to scores from commercial video games. We believe that this represents an evaluation of generative music in games that more closely measures how generative music may be used in real-world games than previous approaches.

This approach presents a somewhat idealized version of generative music used in video games, given current technological constraints. While our generative score technically pro-

duces unique music that matches gameplay, it does not compose music in real-time to match gameplay as the MMM algorithm is not currently capable of real-time generation. Our generative score is generated using symbolic notation, but tracks are rendered into audio files, as real-time synthesis cannot currently match the fidelity or computing performance of offline synthesis.

Our results are consistent with previous approaches to using generative music in games. While the differences are marginal, real-time annotations of perceived emotions match our predicted perceived emotion more with generative and adaptive music than with the composed linear score. Participants rank our generative score as on par with our linear score in terms of emotional congruency, immersion, and preference, and far above our composed adaptive score.

6.6 Future work

In focusing on the aesthetic fidelity of our application of generative music in games, we do not necessarily exploit the full strength of generative music. While PreGLAM outputs unbounded floating point values for valence, arousal, and tension, we use 5 categorical levels of emotion — We control the adaptivity of the score separately from the composition in order to use adaptive music techniques from the industry.

Additionally, we manually design the adaptivity of our score, and compose a score that has the same 3-dimensional adaptivity as the generated score. While the use of generative music allows us to easily and quickly expand the composed adaptive score, the original composition, and therefore the generated music that is based on the composition, is still somewhat restricted in expressive range to allow for relatively smooth musical transitions.

Generative music that is composed and synthesized in real-time could exhibit more musical flexibility than our composed score, and could provide more continuous adaptivity. Additionally, generative music that is composed and synthesized in real-time could have smoother transitions, as the transitions could be directly generated.

In addition to future work in the technological implementations of generative music in games, we note that we evaluate generative music acting as an audience for a single-player action-RPG game. There are many ways to use music in games, and this application of generative music may not be suitable for all of them — for example, Phillips describes the use of music as “branding”, which uniquely generated music may be very poorly suited to. While we believe that our application represents a scenario for which generative music is most well suited, there are many other possible applications of generative music in video games.

Bibliography

[1] Shakir Belle, Curtis Gittens, and T.C. Nicholas Graham. Programming with affect: How behaviour trees and a lightweight cognitive architecture enable the development of non-player characters with emotions. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–8. IEEE, 2019.

[2] BioWare. (Game) Mass Effect, 2007.

[3] Andrew R. Brown and Andrew C. Sorensen. Introducing jMusic. In *Australasian computer music conference*, pages 68–76, 2000.

[4] Kristofer Eng and Philip Bennefall. (software) elias adaptive music. URL: `https://www.eliassoftware.com/`, 2015. Last accessed: 04-25-2022.

[5] Jeff Ens and Philippe Pasquier. MMM: Exploring conditional multi-track music generation with the transformer, 2020.

[6] Patrick Gebhard. ALMA: A Layered Model of Affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36, 2005.

[7] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software, Articles*, 31(7):1–24, 2009. Last accessed: 04-25-2022.

[8] Christian Henson and Paul Thomson. (Virtual instrument library) LABS. URL: `https://labs.spitfireaudio.com/`, 2007. Last accessed: 04-25-2022.

[9] Phil Lopes. *Generating multifaceted content in games: a study on levels and sound.* PhD thesis, University of Malta, 2017.

[10] Phil Lopes and Ronan Boulic. Towards designing games for experimental protocols investigating human-based phenomena. In *International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery.

[11] Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis. Sonancia: Sonification of Procedurally Generated Game Levels. *Proceedings of the 1st Computational Creativity and Games Workshop.*, 2015.

[12] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. PAGAN: Video Affect Annotation Made Easy. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 130–136. IEEE, 2019.

[13] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv preprint arXiv:2104.02643*, 2021.

[14] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[15] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to Musical Metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[16] Winifred Phillips. *A Composer's Guide to Game Music*. The MIT Press, Cambridge, MA, 2014.

[17] 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.

[18] Cale Plut. (SoundCloud) Galactic Defense linear score. URL: `https://soundcloud.com/cale-plut/galactic-defense-linear-score`, 2021. Last accessed: 04-25-2022.

[19] Cale Plut. (Source code) GalDef annotation software. URL: `https://github.com/CalePlut/GalDef_Annotation`, 2021. Last accessed: 04-25-2022.

[20] Cale Plut. (Video) How to play Galactic Defense. URL: `https://www.youtube.com/watch?v=YQtF9s5fVyc`, Sep 2021. Last accessed: 04-25-2022.

[21] Cale Plut and Philippe Pasquier. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing*, 33:100337, 2020.

[22] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) PreGLAM: A predictive gameplay-based layered affect model. *Entertainment Computing*, 2022.

[23] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) the IsoVAT corpus: Parameterization of musical features for affective composition. *Transactions of the International Society for Music Information Retrieval*, 2022.

[24] Anthony Prechtl. *Adaptive music generation for computer games*. PhD thesis, Open University (United Kingdom), 2016.

[25] Lucas Reycevick. (Video) The Brilliance of DOOM's Soundtrack. URL: https://www.youtube.com/watch?v=7X3LbZAxRPE, 2016. Last accessed: 04-25-2022.

[26] Rockstar Games. (Game) Red Dead Redemption. Game, May 2010.

[27] Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. Evolving in-game mood-expressive music with metacompose. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, pages 1–8. Association for Computing Machinery, 2018.

[28] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines*, 18(4):433–465, 2017.

[29] Michael Sweet. *Writing interactive music for video games: A composer's guide*. Addison-Wesley, Upper Saddle River, NJ, 2015.

[30] Marc-Pierre Verge. (Virtual instrument library) Applied Acoustic Systems. URL: https://www.applied-acoustics.com/, 1998. Last accessed: 04-25-2022.

[31] Duncan Williams, Jamie Mears, Alexis Kirke, Eduardo Miranda, Ian Daly, Asad Malik, James Weaver, Faustina Hwang, and Slawomir Nasuto. A perceptual and affective evaluation of an affectively driven engine for video game soundtracking. *ACM Computers in Entertainment*, 14(3), 2017.

Chapter 7

Conclusion

7.1 Summary

In applications of generative music for video games, we found several trends. There is a gulf between academic and industry approaches to using generative music in games, each with distinct advantages and disadvantages. Academic approaches mostly focus on generalizable symbolic musical generation systems, while the industry focuses on creating a highly polished entertainment product.

The gulf between these systems is wide enough that they sometimes seem to be solving two entirely different problems. Academic approaches mostly focus on the creation of a new generative algorithm, targeting real-time affective generation of music that will completely replace a composed score. These systems primarily generate for one or two instruments, which are synthesized using general MIDI. Industry systems mostly focus on extending and expanding a composed, adaptive score using simple stochastic methods. These scores are mostly recorded by live musicians or synthesized offline.

Generative music has not yet achieved widespread use in games, and we believe that part of this is due to this gulf between approaches. To address this, we target the use of generative music in games, rather than the creation of generative music for games. Additionally, we evaluate our use of generative music in comparison to scores that use standard industry approaches to production and synthesis. Because we use the same production tools to create our generative score as our linear and adaptive score, our generative score is consistent with our linear and adaptive scores in genre, timbre, instrumentation, synthesis, and production quality.

In short, we produce a prototype real-world implementation of generative music in games, and evaluate it in a consistent, controlled environment that is similar to real-world video games. This prototype is created using industry standard tools, and extends common industry approaches to adaptive game music. This focuses our examination of generative music on how these technologies might be used, rather than on the development of the technologies themselves.

We believe that our approach provides multiple benefits when compared to previous approaches. Game music is a popular musical genre, with sales of game soundtracks [39], and live concerts of game music [152] demonstrating the appeal outside of gameplay. Most video games have music, and most of that music is composed by humans. Speaking personally as both a composer and game designer, I am much more open to the idea that generative music can be an assistive tool than I am to the idea that a set of equations and algorithms will completely replace an entire mode and genre of musical composition. As mentioned in Chapter 1, this is consistent with contemporary recommendations for the use of generative music [104, 29]. Also, using generative music as an assistive tool allows us to leverage the strengths of human composition, such as by allowing for more complex musical adaptivity, bolstered by the generative system, as we demonstrate.

In addition to producing an application of generative music that mimics the theoretical potential of generative music, we iteratively improve on the quality or fidelity of each of our system's components compared to previous research. In terms of game design, we include a scaled-down version of combat mechanics inspired by existing Action-RPG games, while previous approaches mostly focus on navigating a simple game space. In terms of our musical production, we use offline synthesis with VST instruments in a 5-piece musical ensemble, while previous approaches generally use General MIDI to synthesize one or two instruments, such as solo piano.

In terms of manipulating music for perceived affect, previous academic systems primarily manipulate one or a set of musical features to manipulate affect, based on general Western music theory. We instead collate empirical results from a wide range of studies on music and emotion, to create a central guide that ordinally describes the direction of change in perceived affect, based on a change in a set of composition and performance features. Rather than attempting to directly control the music generation via data variables, we use this guide to direct the manual composition of a musical score that expresses given emotions in a VAT space.

In terms of our generative algorithm, we use the MMM transformer model to expand our manually composed score into a generative score. This allows us to produce multi-track music with flexible instrumentation. This additionally allows us to condition the output music not on a set of parameters, but on a provided piece of music. We condition MMMs generation on our composed adaptive score. Because we compose our score following the IsoVAT composition guide, we know that the composed score manipulates musical features that are associated with affective perception. Because of this, we assume that the music generated by the transformer, conditioned on the composed score, will have many of the same feature manipulations. Therefore, we assume that the adaptive score generated by MMM will have a similar affective perception to the composed adaptive score.

In terms of modeling gameplay for adaptive music, previous systems generally create affect models inspired from EDPCG, which mostly follow a design of assigning affective

values based on a formula consisting of one or more game variables. We extend Phillips' metaphor of music acting as an audience, and create an affect model that is based on previous NPC design, which acts as an audience to the game. Rather than calculating an absolute affect value based on game variables, PreGLAM uses an appraisal model that is based on emotionally evocative game events, as modified by game variables. Essentially, we model our affective response to gameplay based on the context of what is happening in gameplay through time, rather than modeling based on a snapshot of the current state of the game. Returning to Phillip's metaphor of music acting as an audience, our emotion model responds to the successes and failures of the player, rather than an assessment of how the player is doing.

Overall, we created an application of generative music that mimics the appearance of a theoretical implementation of real-time affective adaptive generative music, for the purpose of exploring and evaluating the use of generative music, rather than the generation of music. Additionally, we present improvements in multiple individual features of an affective adaptive generative music system for games.

7.2 Reflection and Future work

I recall a joke that goes something like this:

1. Application: "I'm going to pet all the dogs in the world!"
2. Proposal: "I'm going to pet all the dogs in Vancouver!"
3. Dissertation: "I waved at a cat down the street from here."

I successfully waved at a cat down the street from here, but there are many animals yet to pet. We created an application of a 3-dimensional generative score, controlled by a new emotion model, within an action-RPG. There are clear extensions to each individual feature of this work.

While we bypassed the obstacles normally present in using generative music in games, we generally did so with offline, time and labour-intensive processes. While we created a facsimile of generative music extending adaptive music, we mostly accomplished this illusion by manually performing the labour that was outside of current technological limitations. Given that one of the primary motivations for the applicability of generative music in games is that it may reduce the overall labour of composing adaptive music, this may seem to present a significant roadblock.

Some of the issues that we faced may be dealt with, as assumed in much research in this area, by technological advancement. There is active research into the usability of musical AIs and the real-time synthesis of symbolic music, which will reduce the amount of additional labour required to use generative music. As these technological aspects improve, less labour

will be required to use generative music, and theoretically more attention can be paid to the application of the technology, rather than just the development.

Some of the roadblocks that we encountered may only be a result of the requirements of academic research, and may not be an issue when applying generative music to real-world game scenarios. Because the implementation of our design required experimental control for empirical evaluation, and because we based many of our decisions on previous literature, many of the specific requirements of this work are simply not present when designing a commercial and/or artistic product.

Industry uses of generative music in games generally have different requirements, and therefore one may question the applicability of this research in industry applications at all. We note that while the *implementation* of our approach is primarily influenced by its nature as research, the *design* of our approach is flexible and scalable. The design of our approach generally cast music as supportive to the game mechanics — EEGEs for PreGLAM were determined while playtesting the game, and while the adaptive score's musical adaptivity was primarily based on the IsoVAT guide, elements of musical genre, instrumentation, tempo, and keys were musically chosen to support the game, constrained only in their consistency across conditions. We believe that this research presents a step towards increased interoperability and communication between academic and industry approaches to generative music.

In addition to future work involving technological and design elements, we believe that there are still many questions about implementing generative music in games with our current research-oriented design and technology. We targeted one particular function of games music — music acting as an audience. Phillips provides 5 other functions of music in games, several of which could theoretically benefit from generative music [177].

We target the real-time adaptation of music to moment-to-moment gameplay. As discussed in Chapter 1, game lengths can extend to 100s of hours, and composing music, particularly adaptive music, for such lengths is beyond the budget of almost every game. As with generative music providing assistance in composing highly adaptive music, generative music could also provide assistance for composers in longitudinal aspects of the music.

In terms of game genre, we applied generative music only within an action-RPG genre game, and only within active gameplay in the form of combat. Combat presents one of the simpler game paradigms to implement, with many examples of combat-based game design. In focusing on the application of music and the emotion model, we targeted a straightforwards game genre to implement. However, much of gameplay in real-world games is outside of combat, and there are many types of nonviolent gameplay mechanics that may have perceived emotional arcs; As Murray notes, even Tetris has a story [155]. Exploring alternate game genres may provide insights for PreGLAM's architecture, and for game music in general.

We model gameplay emotion using a set of manually derived EEGEs for PreGLAM. PreGLAM's framework can implement machine learning techniques for the determination and prediction of EEGEs, and we believe that any implementation in PreGLAM into more complex and complete game structures will require the use of ML methods. Additionally, there are several different applications of ML techniques within the PreGLAM architecture, from automatically deriving EEGEs from ground-truth annotations, to predicting manually derived EEGEs, or a hybrid approach. Further empirical evaluation may shed light on where human design is best applied alongside ML methods for PreGLAM.

With the benefit of hindsight, there are several aspects that I believe could be polished and expanded in our implementation. Mostly, these changes involve expanding the empirical evaluation of our work by implementing additional conditions and variations on our application. For example, while we follow the most common approach in academia of using an emotional model to adapt our musical score, such an approach is almost unheard of in the games industry. Without an actual comparison to industry-standard techniques, it is difficult to gauge the utility of the PreGLAM framework.

Additionally, while we believe that we improve on each feature of generative music in games compared to previous implementations, we do not have direct comparisons to the designs of previous implementations. While PreGLAM performs generally well, we do not know how it performs in relation to the game-state models such as used in the AGAIN database [148], nor how it directly compares to real-time symbolic generative systems.

As mentioned, the design of our approach is theoretically genre agnostic, as demonstrated by multiple theoretical implementations. However, we do not implement or evaluate PreGLAM in games with different genres. Given the importance of experiential playtesting in the construction of PreGLAM within GalDef, it follows that implementing PreGLAM into different genres, informally evaluated during the implementation, would shed further light on PreGLAM's cross-genre capabilities.

Overall, we present one particular application of generative music. Our approach provides a new framework for modeling and approaching generative music in games by extending the metaphor of the music acting as an audience. I believe that we were successful in applying this approach, but there is much more room to explore.

Bibliography

[1] Airtight Games, Jose Perez III, and Jason Lamparty. (Game) Dark Void, 2010.

[2] Anna Aljanaki, Frans Wiering, and Remco Veltkamp. Collecting annotations for induced musical emotion via online game with a purpose emotify, 2014.

[3] Fernando Arroyo Garcia Lascurain. *Affect and Feelings: The Persuasive Power of Film Music*. PhD thesis, University of California, University of California, 2016.

[4] Jack Atherton and Ge Wang. Chunity: Integrated audiovisual programming in unity. *New Interfaces for Musical Expression*, 2018.

[5] Audiokinetic. (Software) Wwise, 2017.

[6] Sander C.J. Bakkes, Pieter H.M. Spronck, and Giel van Lankveld. Player behavioural modelling for video games. *Entertainment Computing*, 3(3):71–79, 2012.

[7] Laura-Lee Balkwill and William Forde Thompson. A cross-cultural investigation of the perception of emotion in music: Psychophysical and cultural cues. *Music perception*, 17(1):43–64, 1999.

[8] Tonio Ball, Benjamin Rahm, Simon B. Eickhoff, Andreas Schulze-Bonhage, Oliver Speck, and Isabella Mutschler. Response properties of human amygdala subregions. *PLoS ONE*, 2(3):e307, 2007.

[9] Pablo Barros, Nikhil Churamani, Egor Lakomkin, Henrique Siqueira, Alexander Sutherland, and Stefan Wermter. The OMG-Emotion behavior dataset. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2018.

[10] Richard Bartle and Chris Bateman, editors. *Beyond game design: Nine steps towards creating better videogames*. Charles River Media/Course Technology, Boston, Mass., 2009.

[11] David Bashwiner. Tension. In *Music in the Social and Behavioral Sciences: An Encyclopedia*, volume 2, pages 1113–1115, Thousand Oaks, 2014. SAGE Publications, Inc.

[12] Russell Belk. Out of sight and out of our minds: What of those left behind by globalism? In *Does Marketing Need Reform?: Fresh Perspectives on the Future*, pages 217–224. Routledge, 2015.

[13] Shakir Belle, Curtis Gittens, and T.C. Nicholas Graham. Programming with affect: How behaviour trees and a lightweight cognitive architecture enable the development of non-player characters with emotions. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–8. IEEE, 2019.

[14] Luciano Berio. *Two interviews*. New York : M. Boyars, 1985.

[15] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. DotA 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[16] Paul Bertens, Anna Guitart, and África Periáñez. Games and big data: A scalable multi-dimensional churn prediction model. In *2017 IEEE conference on computational intelligence and games (CIG)*, pages 33–36. IEEE, 2017.

[17] John A. Biles. GenJam: A genetic algorithm for generating jazz solos. *Proceedings of the International Computer Music Conference*, pages 131–137, 1994.

[18] BioWare. (Game) Mass Effect, 2007.

[19] BioWare, Black Isle Studios, and James Ohlen. (Game) Baldur's Gate, 1998.

[20] Julia Ayumi Bopp, Elisa D. Mekler, and Klaus Opwis. *Negative Emotion, Positive Experience? Emotionally Moving Moments in Digital Games*, page 2996–3006. Association for Computing Machinery, New York, NY, USA, 2016.

[21] Matthew Bribitzer-Stull. *Understanding the leitmotif*. Cambridge University Press, 2015.

[22] Andrew R. Brown and Andrew C. Sorensen. Introducing jMusic. In *Australasian computer music conference*, pages 68–76, 2000.

[23] JaeHwan Byun and Christian Loh. Audial engagement: Effects of game sound on learner engagement in digital game-based learning environments. *Computers in Human Behavior*, 05 2015.

[24] Zoraida Callejas and Ramon Lopez-Cozar. Influence of contextual information in emotion annotation for spoken dialogue systems. *Speech Communication*, 50(5):416–433, 2008.

[25] Clifton Callender, Ian Quinn, and Dmitri Tymoczko. Generalized voice-leading spaces. *Science*, 2008.

[26] Emilios Cambouropoulos, Maximos A. Kaliakatsos-Papakostas, and Costas Tsougras. An idiom-independent representation of chords for computational music analysis and generation. In *ICMC*, 2014.

[27] Canadian League of Composers. Commissioning Rates. URL: `https://www.composition.org/commissioning-rates/`, 2015. Last accessed: 04-25-2022.

[28] Capcom and Akira Kitamura. (Game) Mega Man, 1987.

[29] Luca Casini, Gustavo Marfia, and Marco Roccetti. Some reflections on the potential and limitations of deep learning for automated music generation. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 27–31, 2018.

[30] G.G. Cassidy and R.A.R. Macdonald. The effects of music on time perception and performance of a driving game. *Scandinavian Journal of Psychology*, 51(6):455–464, 2010.

[31] Carmine Cataldo. Towards a music algebra: fundamental harmonic substitutions in jazz. *International Journal of Advanced Engineering Research and Science*, 5(1):237359, 2018.

[32] Julian Cespedes-Guevara and Tuomas Eerola. Music communicates affects, not basic emotions–a constructionist account of attribution of emotional meanings to music. *Frontiers in psychology*, 9:215, 2018.

[33] Tim Challies. "No Man's Sky" and 10,000 Bowls of Plain Oatmeal. URL: `https://www.challies.com/articles/no-mans-sky-and-10000-bowls-of-plain-oatmeal/`, Oct 2016. Last accessed: 04-25-2022.

[34] Pei-Chun Chen, Keng-Sheng Lin, and Homer H. Chen. Emotional accompaniment generation system based on harmonic progression. *IEEE transactions on multimedia*, 15(7):1469–1479, 2013.

[35] Christiaan Aaron Clark. *Gameplay as Discrete Form: Leveraging Procedural Audio for Greater Adaptability in Video Game Music*. University of California, Riverside, 2021.

[36] J. Clement. Number of hours of video games streamed online 2021. URL: `https://www.statista.com/statistics/1125469/video-game-stream-hours-watched/`, May 2021. Last accessed: 04-25-2022.

[37] J. Clement. Video game market value worldwide 2015. URL: `https://www.statista.com/statistics/292056/video-game-market-value-worldwide/`, Nov 2021. Last accessed: 04-25-2022.

[38] NWH Coding. (Unity Plugin) Grapher. URL: `https://assetstore.unity.com/packages/tools/utilities/grapher-graph-replay-log-84823`, Dec 2017. Last accessed: 04-25-2022.

[39] Karen Collins. *Game sound: an introduction to the history, theory, and practice of video game music and sound design*. Mit Press, 2008.

[40] Karen Collins. An introduction to procedural music in video games. *Contemporary Music Review*, 28(1):5–15, 2009.

[41] Karen Collins. *From Pac-Man to Pop Music Interactive Audio in Games and New Media*. Farnham : Ashgate Publishing Ltd, Farnham, 2011.

[42] Karen Collins. *Playing with sound: a theory of interacting with sound and music in video games*. The MIT Press, Cambridge, MA, 2013.

[43] Darrell Conklin and Ian H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.

[44] Joel Cooper and Kevin M. Carlsmith. Cognitive dissonance. In *International Encyclopedia of the Social and Behavioral Sciences*, pages 76–78. Elsevier ltd, Amsterdam, 2015.

[45] Greg Costikyan. *Uncertainty in games*. Mit Press, 2013.

[46] Chris Crawford. *Chris Crawford on game design*. New Riders, 2003.

[47] Jason Cullimore, Howard Hamilton, and David Gerhard. Directed transitional composition for gaming and adaptive music using q-learning. In *ICMC*, pages 332–338, 2014.

[48] Gabe Cuzzillo. (Game) Ape Out, 2019.

[49] Paulo Vitor Damo da Rosa. The only point of life that matters is your last. URL: `https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/the-only-point-of-life-that-matters-is-your-last/`, Oct 2017. Last accessed: 04-25-2022.

[50] Die Gute Fabrik. (Game) Sportsfriends, 2014.

[51] Monica Dinculescu, Jesse Engel, and Adam Roberts, editors. *MidiMe: Personalizing a MusicVAE model with user data*, 2019.

[52] Christopher Doll. Definitions of 'chord' in the teaching of tonal harmony. *Dutch Journal of Music Theory*, 2013.

[53] Tuomas Eerola and Jonna Vuoskoski. A review of music and emotion studies: Approaches, emotion models, and stimuli. *Music Perception*, 02 2013.

[54] Tuomas Eerola and Jonna K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, jan 2011.

[55] Tuomas Eerola and Jonna K. Vuoskoski. A review of music and emotion studies: Approaches, emotion models, and stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, 2012.

[56] Arne Eigenfeldt and Philippe Pasquier. Realtime generation of harmonic progressions using controlled markov selection. In *Proceedings of ICCC-X-Computational Creativity Conference*, pages 16–25, 2010.

[57] Panteleimon Ekkekakis. Affect, mood, and emotion. *Measurement in sport and exercise psychology*, 321, 2012.

[58] Eran Eldar, Ori Ganor, Roee Admon, Avraham Bleich, and Talma Hendler. Feeling the real world: Limbic response to music depends on related content. *Cerebral Cortex*, 17(12):2828–2840, 2007.

[59] Andrew Elmsley, Ryan Groves, and Valerio Velardo. Deep Adaptation: How Generative Music Affects Engagement and Immersion in Interactive Experiences. *Digital Music Research Network One-day workshop*, 12:7, 2017.

[60] Kristofer Eng and Philip Bennefall. (software) elias adaptive music. URL: `https://www.eliassoftware.com/`, 2015. Last accessed: 04-25-2022.

[61] Steve Engels, Fabian Chan, and Tiffany Tong. Automatic real-time music generation for games. In *11th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 220–222, Palo Alto, CA, 2015. AAAI Press.

[62] Jeff Ens and Philippe Pasquier. MMM: Exploring conditional multi-track music generation with the transformer, 2020.

[63] Jeff Ens and Philippe Pasquier. MMM: exploring conditional multi-track music generation with the transformer. *arXiv preprint arXiv:2008.06048*, 2020.

[64] Blizzard Entertainment. (game) world of warcraft, 2004.

[65] Q Entertainment and Tetsuya Mizuguchi. (Game) Child of Eden, 2011.

[66] Entertainment Software Association. 2019 essential facts about the computer and video game industry. URL: `https://www.theesa.com/resource/essential-facts-about-the-computer-and-video-game-industry-2019/`, 2019. Last accessed: 04-25-2022.

[67] Epic Games and Tim Sweeney. (Game Engine) Unreal Engine. URL: `https://www.unrealengine.com/`, 1998. Last accessed: 04-25-2022.

[68] Xiaowen Fang, Susy Chan, Jacek Brzezinski, and Chitra Nair. Development of an instrument to measure enjoyment of computer game play. *Intl. Journal of Human–Computer Interaction*, 26(9):868–886, 2010.

[69] Luz Fernández-Aguilar, Beatriz Navarro-Bravo, Jorge Ricarte, Laura Ros, and Jose Miguel Latorre. How effective are films in inducing positive and negative emotional states? A meta-analysis. *PloS one*, 14(11):e0225040, 2019.

[70] L. Festinger and J. M. Carlsmith. Cognitive consequences of forced compliance. *Journal of abnormal psychology*, 58(2):203–10, 1959.

[71] Firaxis Games and Jake Solomon. (Game) XCOM 2, 2016.

[72] FMOD. (Software) FMOD Studio, 2016.

[73] N.H. Frijda, Batja Mesquita, J. Sonnemans, and S. Van Goozen. The duration of affective phenomena or emotions, sentiments and passions. In *International Review of Studies on Emotion*, pages 187–225. Wiley, 1991.

[74] Nico H. Frijda. *The laws of emotion*. Lawrence Erlbaum Associates, Mahwah, N.J., 2007.

[75] Thomas Fritz, Sebastian Jentschke, Nathalie Gosselin, Daniela Sammler, Isabelle Peretz, Robert Turner, Angela D. Friederici, and Stefan Koelsch. Universal recognition of three basic emotions in music. *Current biology*, 19(7):573–576, 2009.

[76] Funcom. (Game) Anarchy Online. URL: https://www.anarchy-online.com/, 2001. Last accessed: 04-25-2022.

[77] Gaijin Games and Alex Neuse. (Game) Bit.Trip Runner, 2011.

[78] Philip Galanter. What is Generative Art? Complexity theory as a context for art theory. *In GA2003–6th Generative Art Conference*, 2003.

[79] Gamelab and Nicholas Fortugno. (Game) Diner Dash, 2004.

[80] Riot Games. How to play League of Legends. URL: https://na.leagueoflegends.com/en-us/how-to-play/, 2021. Last accessed: 04-25-2022.

[81] Gege Gao, Aehong Min, and Patrick C. Shih. Gendered design bias: Gender differences of in-game character choice and playing style in League of Legends. In *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, pages 307–317, 2017.

[82] Brad Garton. (Software) RTcmix. URL: http://rtcmix.org/, 2019. Last accessed: 04-25-2022.

[83] Hans-Peter Gasselseder. Dynamic music and immersion in the action-adventure an empirical investigation. *ACM International Conference Proceeding Series*, 2014, 10 2014.

[84] Patrick Gebhard. ALMA: A Layered Model of Affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 29–36, 2005.

[85] Patrick Gebhard, Michael Kipp, Martin Klesen, and Thomas Rist. Adding the emotional dimension to scripting character dialogues. In *International Workshop on Intelligent Virtual Agents*, pages 48–56. Springer, 2003.

[86] Deyan Georgiev. 45 Video Games Industry Statistics, Facts, and Trends for 2022. URL: https://techjury.net/blog/video-games-industry-statistics/, Dec 2021.

[87] David Gerhard and Daryl H. Hepting. Cross-modal parametric composition. In *ICMC*. Citeseer, 2004.

[88] Sang gil Lee, Uiwon Hwang, Seonwoo Min, and Sungroh Yoon. A seqgan for polyphonic music generation. *CoRR*, 2017.

[89] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. *Journal of Statistical Software, Articles*, 31(7):1–24, 2009. Last accessed: 04-25-2022.

[90] Yosef Goldenberg. Harmony without voice leading? the challenge of interpreting exact leaping transpositions. *Music Analysis*, 35(3):314–340, 2016.

[91] Christina Gough. League of Legends Championships viewers 2020. URL: https://www.statista.com/statistics/518126/league-of-legends-championship-viewers/, Mar 2021. Last accessed: 04-25-2022.

[92] Gaëtan Hadjeres and Léopold Crestel. The piano inpainting application. *arXiv preprint arXiv:2107.05944*, 2021.

[93] Gaëtan Hadjeres, Jason Sakellariou, and François Pachet. Style imitation and chord invention in polyphonic music with exponential families. *arXiv preprint arXiv:1609.05152*, 2016.

[94] Waldie E. Hanser and Ruth E. Mark. Music influences ratings of the affect of visual stimuli. *Psychological Topics*, 22(2):305–324, 2013.

[95] John Harper. *(Game Rulebook) Blades in the Dark*. Evil Hat Productions, Maryland, USA, 2017.

[96] Robert Hasegawa. Creating with constraints. In *The Oxford Handbook of the Creative Process in Music*. Oxford University Press, 2020.

[97] Richard L. Hazlett. Measuring emotional valence during interactive experiences: boys at video game play. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1023–1026, 2006.

[98] Paul Head. ACDA choral workshop, 2005.

[99] Jim Hedges, Kurt Larson, and Christopher Mayer. An Adaptive, Generative Music System for Games. URL: `https://www.gdcvault.com/play/1012710/An-Adaptive-Generative-Music-System`, 2010. Last accessed: 04-25-2022.

[100] Hello Games. (Game) No Man's Sky, 2016.

[101] Christian Henson and Paul Thomson. (Virtual instrument library) LABS. URL: `https://labs.spitfireaudio.com/`, 2007. Last accessed: 04-25-2022.

[102] Carlos Hernandez-Olivan and Jose R. Beltran. Music composition with deep learning: A review. *arXiv preprint arXiv:2108.12290*, 2021.

[103] D. Herremans, Ch. Chuan, and E. Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys*, 50(5), 2017.

[104] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys (CSUR)*, 50(5):1–30, 2017.

[105] Pierre Hoegi. *A Tabular System: Whereby the Art of Composing Minuets is Made So Easy that Any Person, Without the Least Knowledge of Musick, May Compose Ten Thousand, All Different, and in the Most Pleasing and Correct Manner. Invented by Sigr. Piere Hoegi*. Printed at Welcjer's musick shop, 1763.

[106] Holger Hoffmann, Andreas Scheck, Timo Schuster, Steffen Walter, Kerstin Limbrecht, Harald C. Traue, and Henrik Kessler. Mapping discrete emotions into the dimensional space: An empirical approach. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3316–3320. IEEE, 2012.

[107] Suvi K. Holm, Johanna K. Kaakinen, Santtu Forsström, and Veikko Surakka. Self-reported playing preferences resonate with emotion-related physiological reactions during playing and watching of first-person shooter videogames. *International Journal of Human-Computer Studies*, page 102690, 2021.

[108] Ben Houge. Cell-based music organization in Tom Clancy's EndWar, 2012.

[109] HowLongToBeat. How long is Final Fantasy XIII? URL: https://howlongtobeat.com/game?id=3532. Last accessed: 04-25-2022.

[110] Craig Hubbard, Kevin Stephens, Wes Saulsberry, Guy Whitemore, Chris Miller, Samantha Ryan, and Monolith Interactive. (Game) The Operative: No One Lives Forever, 2000.

[111] Johan Huizinga. *Homo Ludens: A Study of the Play-Element in Culture*. Beacon Press, 1971.

[112] Hsiao-Tzu Hung, Joann Ching, Seungheon Doh, Nabin Kim, Juhan Nam, and Yi-Hsuan Yang. Emopia: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. *arXiv preprint*, 2021.

[113] Patrick Hutchings and Jon McCormack. Adaptive music composition for games. *IEEE Transactions on Games*, 2019.

[114] Przemysław Jarząbek. Are new movies longer than they were 10, 20, 50 year ago? URL: https://towardsdatascience.com/are-new-movies-longer-than-they-were-10hh20-50-year-ago-a35356b2ca5b, Dec 2018. Last accessed: 04-25-2022.

[115] Kent Jolly and Aaron McLeran. Procedural Music in Spore. URL: https://www.gdcvault.com/play/323/Procedural-Music-in, 2008. Last accessed: 04-25-2022.

[116] Patrik N. Juslin and John Sloboda. *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[117] Patrik N. Juslin, John Sloboda, Alf Gabrielsson, and Erik Lindström. *The role of structure in the musical expression of emotions*, page 367–400. Oxford University Press, 2012.

[118] Kostas Karpouzis and Georgios N. Yannakakis. *Emotion in Games*. Springer, 2016.

[119] Michael Kennedy and Joyce Bourne. *The Oxford Dictionary of Music*. Oxford University Press, Oxford, 6 edition, 2012.

[120] Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey Scott, Jacquelin A. Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *International Society for Music Information Retrieval (ISMIR)*, volume 86, pages 937–952, 2010.

[121] Peter Kivy. *The corded shell*. Princeton essays on the arts. Princeton University Press, Princeton, 1980.

[122] Konami and Hitoshi Akamatsu. (Game) Castlevania, 1986.

[123] Grace Kramer and Derek Alexander. (Video) Why the Music in Dragon Quest XI is so Terrible. URL: https://www.youtube.com/watch?time{_}continue=200{&}v=xfdFU3O3nf8, 2018. Last accessed: 04-25-2022.

[124] Dae Hee Kwak, Yu Kyoum Kim, and Edward R. Hirt. Exploring the role of emotions on sport consumers' behavioral and cognitive responses to marketing stimuli. *European Sport Management Quarterly*, 11(3):225–250, 2011.

[125] Bjorn Arve Lagim. The Music of Anarchy Online: Creating Music for MMOGs. URL: `https://www.gamasutra.com/view/feature/131361/the_music_of_anarchy_online_.php`, Sep 2002. Last accessed: 04-25-2022.

[126] Peter S. Langston. Six Techniques for Algorithmic Music Composition. *International Computer Music Conference. San Francisco: Computer Music Conference Association*, pages 164–167., 2005.

[127] Daniel Leite, Volnei Frigeri Jr., and Rodrigo Medeiros. Adaptive gaussian fuzzy classifier for real-time emotion recognition in computer games. *arXiv preprint arXiv:2103.03488*, 2021.

[128] David Levine, Peter Langston, David Riordan, and Garry Hare. (Game) Ballblazer, 1984.

[129] Antonios Liapis, Georgios N. Yannakakis, Mark J. Nelson, Mike Preuss, and Rafael Bidarra. Orchestrating game generation. *IEEE Transactions on Games*, 11(1):48–68, 2018.

[130] Hendi Lie, Darren Lukas, Jonathan Liebig, and Richi Nayak. A novel learning-to-rank method for automated camera movement control in e-sports spectating. In *Australasian Conference on Data Mining*, pages 149–160. Springer, 2018.

[131] Nintendo Life. (Image) Dark Souls: Remastered screenshot. URL: `https://www.nintendolife.com/games/nintendo-switch/dark_souls_remastered/screenshots`, Jul 2020. Last accessed: 04-25-2022.

[132] Ben Lindbergh. Length matters. URL: `https://www.theringer.com/2016/8/25/16038806/video-game-length-playtimes-f7b8e38f949f`, Aug 2016. Last accessed: 04-25-2022.

[133] Steven R. Livingstone, Ralf Muhlberger, Andrew R. Brown, and William F. Thompson. Changing musical emotion: A computational rule system for modifying score and performance. *Computer Music Journal*, 34(1):41–64, 2010.

[134] Phil Lopes. *Generating multifaceted content in games: a study on levels and sound.* PhD thesis, University of Malta, 2017.

[135] Phil Lopes and Ronan Boulic. Towards designing games for experimental protocols investigating human-based phenomena. In *International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery.

[136] Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis. Sonancia: Sonification of Procedurally Generated Game Levels. *Proceedings of the 1st Computational Creativity and Games Workshop.*, 2015.

[137] Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis. Sonancia: Sonification of Procedurally Generated Game Levels. *Proceedings of the 1st Computational Creativity and Games Workshop.*, 2015.

[138] Lars-Olov Lundqvist, Fredrik Carlsson, Per Hilmersson, and Patrik N. Juslin. Emotional responses to music: Experience, expression, and physiology. *Psychology of Music*, 37(1):61–90, 2009.

[139] Phillip Magnuson. Basic rules for species counterpoint. URL: `http://academic.udayton.edu/PhillipMagnuson/soundpatterns/speciesctpt/`, 2008. Last accessed: 04-25-2022.

[140] Tobias Mahlmann, Anders Drachen, Julian Togelius, Alessandro Canossa, and Georgios N. Yannakakis. Predicting player behavior in Tomb Raider: Underworld. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 178–185. IEEE, 2010.

[141] Nikos Malandrakis, Alexandros Potamianos, Georgios Evangelopoulos, and Athanasia Zlatintsi. A supervised approach to movie emotion tracking. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2376–2379. IEEE, 2011.

[142] Henry Mancini. *Did they mention the Music? The autobiography of Henry Mancini.* Cooper Square Press, New York City, New York, 2001.

[143] Marauder Interactive. (Game) House of the Dying Sun, November 2016.

[144] Ivan Marusic. How many people play League of Legends? URL: `https://leaguefeed.net/did-you-know-total-league-of-legends-player-count-updated/`, Jan 2022. Last accessed: 04-25-2022.

[145] Kelsey McKinney. A hit song is usually 3 to 5 minutes long. here's why. URL: `https://www.vox.com/2014/8/18/6003271/why-are-songs-3-minutes-long`, Aug 2014. Last accessed: 04-25-2022.

[146] Albert Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, 1996.

[147] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. PAGAN: Video Affect Annotation Made Easy. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 130–136. IEEE, 2019.

[148] David Melhart, Antonios Liapis, and Georgios N. Yannakakis. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv preprint arXiv:2104.02643*, 2021.

[149] Microsoft. (Game) Halo 2. Game, November 2004.

[150] Microsoft. (Software) DirectMusic, Apr 2009.

[151] Eduardo R. Miranda and Duncan Williams. Artificial intelligence in organised sound. *Organised Sound*, 20(1):76–81, 2015.

[152] Peter Moormann. *Music and Game: Perspectives on a Popular Alliance*. Springer, 2012.

[153] Belkacem Mostefai, Amar Balla, and Philippe Trigano. A generic and efficient emotion-driven approach toward personalized assessment and adaptation in serious games. *Cognitive Systems Research*, 56:82–106, 2019.

[154] Wolfgang Amadeus Mozart. Musikalisches Würfelspiel, 1787.

[155] Janet H. Murray. *Hamlet on the holodeck: The future of narrative in cyberspace*. MIT press, 2017.

[156] Naughty Dog, Bruce Straley, and Hennig Amy. (Game) Uncharted 2: Among Thieves, 2009.

[157] Ninja Theory and Tameem Antoniades. (Game) Hellblade: Senua's Sacrifice, 2017.

[158] Nintendo, Yoichi Yamada, Eiji Aonuma, and Yoshiaki Koizumi. (Game) The Legend of Zelda: Ocarina of Time, November 1998.

[159] Nintendo Creative Department and Shigeru Miyamoto. (Game) Super Mario Bros., 1985.

[160] Karen Niven. *Affect*, pages 49–52. Springer New York, New York, NY, 2013.

[161] Dom O'Hanlon. Is length important? URL: `https://www.londontheatre.co.uk/theatre-news/west-end-features/is-length-important`, Aug 2016. Last accessed: 04-25-2022.

[162] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 1990.

[163] François Pachet. A joyful ode to automatic orchestration. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–13, 2016.

[164] Alexey Pajitnov and Vladimir Pokhilko. (Game) Tetris, June 1984.

[165] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4):614–626, 2018.

[166] Renato Eduardo Silva Panda, Ricardo Malheiro, Bruno Rocha, António Pedro Oliveira, and Rui Pedro Paiva. Multi-modal music emotion recognition: A new dataset, methodology and comparative analysis. In *10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2013)*, pages 570–582, 2013.

[167] Rob Parke, Elaine Chew, and Chris Kyriakakis. Quantitative and visual analysis of the impact of music on perceived emotion of film. *Computers in Entertainment (CIE)*, 5(3):5, 2007.

[168] Teri Parker. A modal approach to chord voicing. *Canadian Musician*, 40(1):26, Jan 2018.

[169] Philippe Pasquier. (Kadenze Class) Generative Art and Computational Creativity. URL: https://www.kadenze.com/courses/generative-art-and-computational-creativity-i. Last accessed: 04-25-2022.

[170] Philippe Pasquier and Brahim Chaib-Draa. Agent communication pragmatics: the cognitive coherence approach. *Cognitive Systems Research*, 6(4):364–395, 2005.

[171] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to Musical Metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[172] Ashis Pati, Alexander Lerch, and Gaëtan Hadjeres. Learning to traverse latent spaces for musical score inpainting. *arXiv preprint arXiv:1907.01164*, 2019.

[173] Iván Paz, Àngela Nebot, Francisco Mugica, and Enrique Romero. Modeling perceptual categories of parametric musical systems. *Pattern Recognition Letters*, 105:217–225, 2018.

[174] Bernard Perron, Mark J.P. Wolf, and Thomas H. Apperley. *The video game theory reader 2*, volume 2. Routledge New York, 2009.

[175] Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. Deep player behavior models: Evaluating a novel take on dynamic difficulty adjustment. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.

[176] Daniel Wei-Shen Phang. *Intelligent camera control in game replays*. Lehigh University, 2014.

[177] Winifred Phillips. *A Composer's Guide to Game Music*. The MIT Press, Cambridge, MA, 2014.

[178] Phosfiend Systems. (Game) Fract OSC, April 2014.

[179] Jaromir Plachy and Amanita Design. (Game) Chuchel, 2018.

[180] 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.

[181] Cale Plut. (Game) The Audience of the Singular, 2017.

[182] Cale Plut. (Soundcloud) Examples of Harmonic and Rhythmic Tension. URL: https://bit.ly/2QBwV1a, 2018. Last accessed: 04-25-2022.

[183] Cale Plut. (Soundcloud) Music Matters Musical Examples. URL: https://soundcloud.com/khavall/sets/music-matters-musical-examples/s-5fBfr, March 2019. Last accessed: 04-25-2022.

[184] Cale Plut. (Video) Galactic Escape description. URL: https://youtu.be/3vxXbMeJGkw, 2019. Last accessed: 04-25-2022.

[185] Cale Plut. (Source code) Galactic Defense. URL: https://github.com/CalePlut/Galactic_Defense, Jul 2020. Last accessed: 04-25-2022.

[186] Cale Plut. (SoundCloud) Galactic Defense linear score. URL: https://soundcloud.com/cale-plut/galactic-defense-linear-score, 2021. Last accessed: 04-25-2022.

[187] Cale Plut. (Source code) GalDef annotation software. URL: https://github.com/CalePlut/GalDef_Annotation, 2021. Last accessed: 04-25-2022.

[188] Cale Plut. (Video) How to play Galactic Defense. URL: https://www.youtube.com/watch?v=YQtF9s5fVyc, Sep 2021. Last accessed: 04-25-2022.

[189] Cale Plut and Philippe Pasquier. Music matters: An empirical study on the effects of adaptive music on experienced and perceived player affect, 2019.

[190] Cale Plut and Philippe Pasquier. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing*, 33:100337, 2020.

[191] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) PreGLAM: A predictive gameplay-based layered affect model. *Entertainment Computing*, 2022.

[192] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games. *Foundations of Digital GAmes*, 2022.

[193] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. (Preprint) the IsoVAT corpus: Parameterization of musical features for affective composition. *Transactions of the International Society for Music Information Retrieval*, 2022.

[194] Alexandru Popescu, Joost Broekens, and Maarten Van Someren. GAMYGDALA: An emotion engine for games. *IEEE Transactions on Affective Computing*, 5(1):32–44, 2013.

[195] Anthony Prechtl. *Adaptive music generation for computer games*. PhD thesis, Open University (United Kingdom), 2016.

[196] Rainer Reisenzein. Wundt's three-dimensional theory of emotion. *Poznan Studies in the Philosophy of the Sciences and the Humanities*, 75:219–250, 2000.

[197] Lucas Reycevick. (Video) The Brilliance of DOOM's Soundtrack. URL: https://www.youtube.com/watch?v=7X3LbZAxRPE, 2016. Last accessed: 04-25-2022.

[198] Alan Rich. Harmony before the common practice period. URL: https://www.britannica.com/art/harmony-music/Harmony-before-the-common-practice-period, Sep 1998. Last accessed: 04-25-2022.

[199] Steve Ritchie, Ed Boon, Doug Watson, and Joe Joos Jr. (Game) Black Knight 2000, April 1989.

[200] Judy Robertson, Andrew de Quincey, Tom Stapleford, and Geraint Wiggins. Real-time music generation for a virtual environment. In *Proceedings of ECAI-98 Workshop on AI/Alife and Entertainment*. Citeseer, 1998.

[201] Rockstar Games. (Game) Red Dead Redemption. Game, May 2010.

[202] Rocksteady Sudios and Sefton Hill. (Game) Batman: Arkham Asylum, 2009.

[203] Richard Rouse. *Game Design: Theory and Practice*, volume 2nd ed of *Wordware Game Developer's Library*. Jones & Bartlett Learning, Plano, Tex, 2005.

[204] James A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980.

[205] Stuart J. Russell. *Artificial Intelligence: A modern approach*. Pearson series in artificial intelligence. Pearson, fourth edition. edition, 2021.

[206] Katie Salen and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. MIT Press, Cambridge, MA, 2004.

[207] Josh Sawyer, Bobby Null, and Eric Fenstermaker. (Game) Pillars of Eternity, 2015.

[208] Ulrich Schimmack and Alexander Grob. Dimensional models of core affect: a quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345, 2000.

[209] Ulrich Schimmack and Rainer Reisenzein. Experiencing activation: Energetic arousal and tense arousal are not mixtures of valence and activation. *Emotion*, 2(4):412–417, 2002.

[210] Emery Schubert. Continuous response to music using a two dimensional emotion space. In *Proceedings of the 4th International Conference of Music Perception and Cognition*, pages 263–268. McGill University. Montreal, 1996.

[211] Emery Schubert. Measuring Emotion Continuously: Validity and Reliability of the Two-Dimensional Emotion-Space. *Australian Journal of Psychology*, 51(3):154–165, dec 1999.

[212] Marco Scirea. *Affective Music Generation and its effect on player experience*. PhD thesis, IT University of Copenhagen, 2017.

[213] Marco Scirea. *Affective music generation and its effect on player experience*. PhD thesis, IT University of Copenhagen, Digital Design, 2017.

[214] Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. Evolving in-game mood-expressive music with metacompose. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, pages 1–8. Association for Computing Machinery, 2018.

[215] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines*, 18(4):433–465, 2017.

[216] William A. Sethares. *Tuning, timbre, spectrum, scale*. Springer, London, 2nd ed. edition, 2005.

[217] Peter Silk. (Video) iMUSE Demonstration 2 - Seamless Transitions. URL: https://bit.ly/1R39FPY, May 2010. Last accessed: 04-25-2022.

[218] Valve Software. Left 4 dead. URL: https://web.archive.org/web/20090327034239/http://www.l4d.com/info.html, Mar 2009. Last accessed: 04-25-2022.

[219] Bethesda Softworks. (Game) Fallout 3, October 2008.

[220] Square Enix and Hajima Tabata. (Game) Final Fantasy XV, 2016.

[221] Square Enix and Motomu Toriyama. (Game) Final Fantasy XIII, 2009.

[222] Jochen Steffens. The influence of film music on moral judgments of movie scenes and felt emotions. *Psychology of Music*, 48(1):3–17, 2020.

[223] Marty Stratton, Hugo Martin, Timothy Bell, Jason O'Connell, Billy Ethan Khan, Hugo Martin, Adam Gascoine, Mick Gordon, and id Software. (Game) DOOM (2016), 2016.

[224] Igor Stravinsky. *Poetics of music in the form of six lessons*. Harvard University Press, 1970.

[225] Lijun Sun, Chen Feng, and Yufang Yang. Tension experience induced by nested structures in music. *Frontiers in Human Neuroscience*, 14:210, 2020.

[226] Supergiant Games. (Game) Pyre, 2017.

[227] Michael Sweet. *Writing interactive music for video games: A composer's guide*. Addison-Wesley, Upper Saddle River, NJ, 2015.

[228] Kıvanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, 48(1):56–105, 2019.

[229] Music Theory Team. Learning the harmonic minor scale. URL: http://www.simplifyingtheory.com/harmonic-minor-scale/, 2019. Last accessed: 04-25-2022.

[230] Richard Terell. Tension: Threats and Timers. URL: https://bit.ly/2Ff6cPm, 2009. Last accessed: 01-02-2019.

[231] Julian F. Thayer and Robert W. Levenson. Effects of music on psychophysiological responses to a stressful film. *Psychomusicology: A Journal of Research in Music Cognition*, 3(1):44–52, 1983.

[232] William Forde Thompson. *Music, thought, and feeling*. Oxford University Press, New York, second edition. edition, 2015.

[233] William Forde Thompson and Brent Robitaille. Can composers express emotions through music? *Empirical Studies of the Arts*, 10(1):79–89, 1992.

[234] Richard van Tol and Sander Huiberts. IEZA: A Framework For Game Audio. URL: https://www.gamasutra.com/view/feature/3509/ieza_a_framework_for_game_audio, Jan 2008. Last accessed: 04-25-2022.

[235] Iwai Toshio. (Game) Otocky, 1987.

[236] Than van Nispen tot Pannerden, Sander Huiberts, Sebastiaan Donders, and Stan Koch. The NLN-Player: A system for nonlinear music in games. In *ICMC*, 2011.

[237] Ubisoft Shanghai and Michael de Plater. (Game) Tom Clancy's EndWar, 2008.

[238] United Game Artists, Jun Kobayashi, Tetsuya Mizuguchi, Hiroyuki Abe, Katsuhiko Yamada, and Katsumi Yokata. (Game) Rez, 2001.

[239] Unity3d. (Game Engine) Unity. URL: `https://unity.com/`, 2019. Last accessed: 04-25-2022.

[240] FrostedSloth (Username), Kyle (Username), and Shawn Saris. (Wiki entry) Dark Souls Game mechanics. URL: `https://www.ign.com/wikis/dark-souls/Game_Mechanics`, Feb 2013. Last accessed: 04-25-2022.

[241] Nixius (Username). (Image) Wildstar attack telegraph. URL: `https://wildstar.fandom.com/wiki/Telegraph`, 2014. Last accessed: 04-25-2022.

[242] Thedarkcave (Username). (Image) Screenshot from The Sims. URL: `https://thefinkedfilms.com/2020/02/04/the-sims-20th-anniversary/`, Feb 2020. Last accessed: 04-25-2022.

[243] Valerio Velardo. Melodrive: Adaptive music generation. URL: `https://melodrive.com/index.php`, 2018. Last accessed: 01-02-2019.

[244] Sean Velasco and Yacht Club Games. (Game) Shovel Knight, 2014.

[245] Philippe Verduyn, Pauline Delaveau, Jean-Yves Rotgé, Philippe Fossati, and Iven Van Mechelen. Determinants of emotion duration and underlying psychological and neural mechanisms. *Emotion Review*, 7(4):330–335, 2015.

[246] Philippe Verduyn and Saskia Lavrijsen. Which emotions last longest and why: The role of event importance and rumination. *Motivation and Emotion*, 39(1):119–127, 2015.

[247] Marc-Pierre Verge. (Virtual instrument library) Applied Acoustic Systems. URL: `https://www.applied-acoustics.com/`, 1998. Last accessed: 04-25-2022.

[248] Sandrine Vieillard, Isabelle Peretz, Nathalie Gosselin, Stephanie Khalfa, Lise Gagnon, and Bernard Bouchard. Happy, sad, scary and peaceful musical excerpts for research on emotions. *Cognition and Emotion*, 22(4):720–752, 2008.

[249] Markus Viljanen, Antti Airola, Anne-Maarit Majanoja, Jukka Heikkonen, and Tapio Pahikkala. Measuring player retention and monetization using the mean cumulative function. *IEEE Transactions on Games*, 12(1):101–114, 2020.

[250] Vlambeer. (Game) Luftrausers, 2014.

[251] Lindsay A. Warrenburg. Choosing the right tune: A review of music stimuli used in emotion research. *Music Perception*, 37(3):240–258, 2020.

[252] Paul Weir. (Video) The Sound of "No Man's Sky". URL: `https://bit.ly/2UKShXS`, 2017. Last accessed: 04-25-2022.

[253] Richard Wetzel. *The globalization of music in history*. Routledge, 2013.

[254] Duncan Williams, Jamie Mears, Alexis Kirke, Eduardo Miranda, Ian Daly, Asad Malik, James Weaver, Faustina Hwang, and Slawomir Nasuto. A perceptual and affective evaluation of an affectively driven engine for video game soundtracking. *ACM Computers in Entertainment*, 14(3), 2017.

[255] Kristina Winbladh, Hadar Ziv, and Debra J. Richardson. (Software) iMuse. *ACM SIGSOFT*, page 383, 2010.

[256] Will Wright. (Game) Spore, 2008.

[257] Huiping Wu and Shing-On Leung. Can Likert scales be treated as interval scales? A Simulation study. *Journal of Social Service Research*, 43(4):527–532, 2017.

[258] Wilhelm Max Wundt and Charles Hubbard Judd. *Outlines of psychology*. W. Engelmann, 1902.

[259] Yi-Hsuan Yang and Homer H. Chen. *Music emotion recognition*. CRC Press, 2011.

[260] G. Yannakakis, P.H.M. Spronck, D. Loiacono, and E. Andre. *Player modeling*, pages 45–59. Number 6 in Dagstuhl Follow-Ups. Dagstuhl Publishing, 2013.

[261] Georgios N. Yannakakis and Héctor P. Martínez. Ratings are overrated. *Frontiers in ICT*, 2:13, 2015.

[262] Georgios N. Yannakakis and Julian Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.

[263] Georgios N. Yannakakis and Julian Togelius. *Artificial intelligence and games*, volume 2. Springer, 2018.

[264] Kejun Zhang, Hui Zhang, Simeng Li, Changyuan Yang, and Lingyun Sun. The pmemo dataset for music emotion recognition. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, page 135–142, New York, NY, USA, 2018. Association for Computing Machinery.

[265] Jiulin Zhang Xiaoqing Fu. The influence of background music of video games on immersion. *Journal of Psychology & Psychotherapy*, 05, January 2015.

Appendix A

Cumulative Dissertation information

SIAT Guidelines

WRITING A CUMULATIVE THESIS

Produced by the SIAT Graduate Program Committee, 6 November 2013, amended May 2021

This document is adjunct to SIAT's calendar entry and to SFU's Graduate General Regulations. It describes SIAT's normal practice with respect to the topic it addresses.

It has been approved by SIAT's Graduate Caucus.

The PhD and Masters thesis may have the form of a monograph (i.e., the classic thesis format of one single document), or of a compilation with a number of scholarly peer-reviewed articles ("cumulative thesis"). Students should always consult early on with their senior supervisor and committee, who will discuss with them and decide what form of thesis is the most suitable for a given case.

Please note that, the guidelines suggested here are to outline possible criteria and courses of actions when preparing a cumulative thesis. The students' supervisory committee might have different criteria or requirements and will decide on what contributions constitute a thesis.

1. In the case of a cumulative thesis, the selected scholarly articles are to be connected by an initial introduction chapter (explaining the subject and scope of the work and how the different articles contribute) and a summative final discussion chapter (that includes the overall contributions, main conclusions, and an outlook). This serves to interrelate the contributions as well as discuss and draw conclusions from the entire work. The publications need to be integrated as chapters into the theme of the thesis and must deal with the overall topic of the thesis. The thesis should have continuous pagination and an aggregated bibliography. The individual papers may still have their own bibliography in addition.

2 a) Phd Thesis: The PhD thesis should have a content and depth corresponding to a classic (monograph-style) thesis. E.g., this might be achieved by about 3-6 peer-reviewed conference papers, journal articles, or other written scholarly contributions of high value where the supervisory committee assesses the quality as being appropriate. The articles should maintain such a level that they could be accepted for publication in an international scholarly journal with a rigorous referee procedure. At least two of these articles should already have been accepted for publication or be published. All may already be published. For published articles, the comprehensive bibliographic reference should be stated. For accepted or submitted manuscripts, the venue and date of acceptance/submission should be included.

2 b) Masters thesis: The Masters thesis should have a content and depth corresponding to a classic (monograph-style) thesis. E.g., this might be achieved by at least 2 peer-reviewed conference papers, journal articles, or other scholarly written contributions of high value where the supervisory committee assesses the quality as being appropriate. The articles should maintain such a level that they could be accepted for publication in an international scholarly journal with a rigorous referee procedure. All of these articles need to be submissible (as judged by the supervisory committee), submitted, accepted, or published. For published articles, the comprehensive bibliographic reference should be stated. For accepted or submitted manuscripts, the venue and date of acceptance/submission should be included.

Appendix B

LazyVoice: A multi-agent approach to fluid voice leading

As published in Plut, C and Pasquier, P. (2022) *LazyVoice : A multi-agent approach to fluid voice leading* International Computer Music Conference

Abstract

We outline and describe the interactive *LazyVoice* system for realizing chord progressions into individual voices with fluid voice leading, inspired by choral voice leading techniques. Polyphonic music consists of multiple musical lines that, when taken together, form an implicit or explicit harmonic progression. While generative music systems exist that create harmonic progressions, these systems lack a means to translate the harmonic progression into individual polyphonic musical lines. We apply a technique used to improvise multiple-part harmony in choral settings to generate fluid musical lines from a harmonic progression. *LazyVoice* is a flexible voice leading system that translates abstracted harmonic progressions into multiple fluid musical lines.

B.1 Introduction and motivation

Music is most often performed by groups of musicians or musical instruments. These groups range from the large symphony orchestras of 100+ musicians to a single musician performing on two instruments, such as piano and voice. While music is often described as containing a melody and accompaniment, the composition of music is more complex — each instrument in an ensemble plays its own musical line, and the combinations of these musical lines form the accompaniment to the primary melody. This linear progression of individual musical

lines that make up harmonies is called *voice leading*, and it is an important aspect of composing polyphonic music.

Voice leading is of particular importance to writing for human voices, such as in choral music, due to the inexact nature of the voice as an instrument. On an unprepared piano, each key corresponds to a single note, and pressing that key will always result in the same note, with the same pitch and intonation. There is no such mechanical assistance for vocalists, and they must know the exact pitch of the note that they intend to sing. Because of this, special care must be taken when writing for human voices to create simpler musical lines, to assist the vocalist in knowing where the next note is.

A musical chord is any combination of musical notes that sound together. Chords may have all notes play at the same time, "arpeggiate" and play the notes one after another, or use some combination of the two. Chords are often represented with notation of the root note and any alterations from a major triad[1], e.g. `C7` indicates a C Major triad with an added minor third on top (The 7th scale degree lowered by one semitone). These representations of chords consider an individual chord to be a single musical object [1]. Importantly, these representations provide only information on the notes that constitute the chord, and when compared to a musical score, remove contextual information about the arrangement of the notes compared to each other. While letter and roman-numeral notation of chords can describe the notes in a chord, they cannot describe the voice leading.

This single-chord representation is common in musical notation such as lead sheets, and is often used to provide analysis of harmonies in music theory [3]. In the case of lead sheets, a musician interprets the chord symbol from abstracted object into individual notes, either in advance or in real-time during performance. In the case of harmonic analysis, musical lines are abstracted into a collection of objects, which is helpful to describe the relations of the harmonies to each other. In both cases, the representation of the chord does not include full information about the chord's constituent pitches [1, 5].

Notations for chord extensions depend on the way that the chord is represented — in roman-numeral notation, additional annotations can include information about the order of notes. As an example, a $\texttt{IV}^6_4$ notation indicates a chord built on the 4th scale degree of the key, in the second inversion, as seen in Figure B.1. In Figure B.1, the key is B♭ Major, and therefore the `IV` chord is a major triad based on E♭. The $^6_4$ designation indicates that the chord is in its second inversion — the 6 and 4 correspond to the intervals between the other notes and the root.

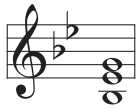


Figure B.1: A voiced $\texttt{IV}^6_4$ chord

Letter-based representation may also use annotations that can provide more information about a chord. The most common extension is a single designation of the chord extensions in

[1]A major triad is a 3-note chord with a minor third stacked on top of a major third

use, though these annotation may also include alternate bass notes. Figure B.2 demonstrates a possible voicing of a D^{9}/F♯ chord — the D indicates that the chord is based on a D Major triad, the ^{9} annotation indicates the extensions of both the dominant 7th and major 9th of the chord, and the annotation of /F♯ indicates that the lowest note of the chord is an F♯.

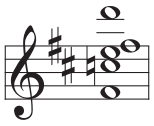


Figure B.2: A voiced D^{9}/F♯ chord

Generative music systems create music via systemic automation, using algorithmic means [14]. Polyphonic harmony generation is a popular feature for generative music, and the most common algorithmic representation of harmonies uses a single-chord representation [11]. This representation is the most obvious representation to use for generating harmonies — single-chord representation heavily restricts the possibility space for generation, there are large corpora of harmonically analyzed musical pieces for training. As discussed, however, this representation requires additional translation into individual musical lines.

B.2 Related work

Several approaches are used to translate between a single-chord representation and a collection of polyphonic voices. Hadjeres, Sakellariou, and Pachet identify three requirements for polyphonic generation; *accuracy* compared to the input corpus, *flexibility* in coping with variety of user input, and *generalization* into new musical forms while maintaining an input style [9]. We note that two of these requirements are particularly relevant to models that attempt to replicate the style of an input corpus. Because *LazyVoice* targets style-agnostic voice leading without an input corpus, we do not consider accuracy to input corpus as a requirement. Instead, we add a requirement of *fluidity* — the individual musical lines that make up a polyphonic generation should follow general melodic guidelines such as avoiding large leaps or multiple leaps in a row.

Chen, Lin, and Chen present one approach to polyphonic generation, in which chords are exclusively represented as a root position triad [4]. This representation is human-readable, particularly in terms of understanding the harmonic relationships between the chords roots and sonorities. Unfortunately, this representation removes almost all of the other contextual data that is important for translating between an abstracted chord representation and polyphonic voice leading. Figure B.3 demonstrates the problems inherent in this chord representation, using a short excerpt from Morten Lauridsen's *O nata lux*. The top system in Figure B.3 presents the harmonies as they originally occur in the piece, and the bottom staff presents the harmonies as they occur using Chen, Lin, and Chen's representation. These root position voicings do not satisfy the requirements of flexibility, generalization, or fluidity. An audio version of this example, as well as any musical figures found in this paper, can be found at `https://bit.ly/2U6pb7L`

Chords may be represented as a set of pitch classes. Systems that implement this approach generally do not differentiate between two different chords, and two different voicings of

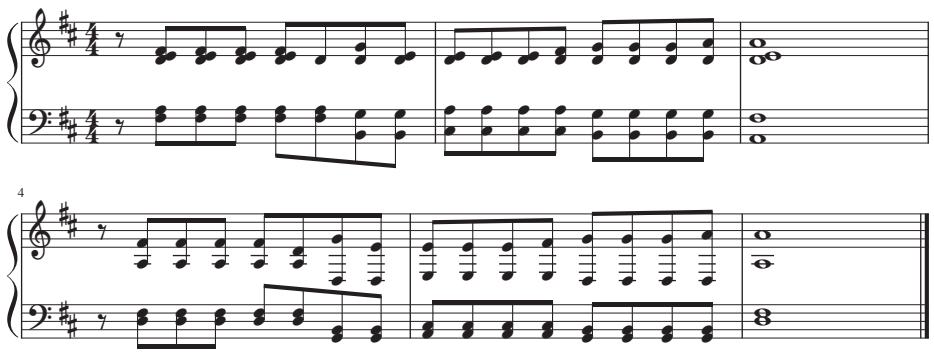


Figure B.3: *O Nata Lux* by Morten Lauridsen, arranged by Cale Plut. ©1997 by Peermusic. Top: Original, Bottom: Root position

a single chord. One such system uses a Markov model to generate progressions of chords, where each chord is a unique set of pitch classes [6]. Essentially, this form of representation assumes that voice leading will be consistent in the input corpus. Another system with similar representation demonstrates that this approach can still produce large leaps and awkward voice leading [7]. While this representation satisfies the requirements of *flexibility* and *generality*, the *fluidity* requirement is left unsatisfied — the individual musical lines contain large and consecutive leaps.

Cambouropoulos, Kaliakatsos-Papakostas, and Tsougras [2] refine Eigenfeldt and Pasquier's representation of chords by including a second data point in their *General Chord Type (GCT)*: a separate integer that corresponds to the root of the chord. This representation allows for different inversions of chords to be stored as distinct but similar, as the vector representing the chord tones itself is unchanged between inversions.

Another approach to modeling chords to voices is through the use of constrained melodic generation, most commonly with a Hidden Markov Model (HMM) providing the harmonic structure, with some other model providing individual melodic lines as constrained by the HMM. This can be seen in Pachet's system of automatic orchestration [12], which uses a maximum entropy model to create melodic lines within a harmonic progression. This approach creates very fluid voice leading between parts as each musical line is created melodically. However, the HMM states in Pachet's system utilize a similar representation of chords to the Eigenfeldt and Pasquier representation, in which chords with varying extensions and inversions are represented as distinct from one another. This reduces the *flexibility* and *generality* of Pachet's model.

Figure B.4 demonstrates the problems with the representation of chord voicings as unrelated entities. In Figure B.4, a `C` Major chord is voiced in four different ways, with three different chord extensions. While the third measure is analyzed as a $\texttt{C}^{add9}$ chord and the fourth measure is analyzed as a $\texttt{C}^{M7}$ chord, any of these voicings may be used to satisfy the function of a `C` Major chord in musical context.

Because choral voice leading must be smooth and singable, as mentioned in Section B.1, techniques from choral composition provide heuristics that can be used to quickly and easily convert a set of chords into a set of interdependent musical lines. We present *LazyVoice*, a heuristic system that takes chords as input and provides customizable smooth voice leading

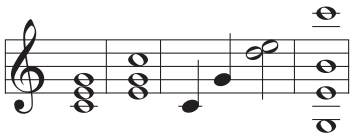


Figure B.4: Various voicings of a C major chord

with trivial complexity. *LazyVoice* provides an intermediary step between chord progression-based generative music systems and full polyphonic music generation. *LazyVoice* can be incorporated into a chord-based generative system, and provides flexible and smooth voice leading without needing additional training or complex rules. This approach also mirrors approaches seen in real-world composition [8].

B.3 LazyVoice

LazyVoice is inspired by a technique from University of Delaware's Director of choral studies Paul Head for improvising choral harmonies [10], that provides flexible, generalizable, and fluid voice leading. This technique can informally be described as being maximally lazy — a chord is initially built in standard open root position voicing. For each subsequent chord, vocalists are instructed to attempt to stay on the same note as the previous chord. If the note that they were holding no longer fits into the harmony, the vocalist moves to the nearest note that is part of the chord. As discussed in Section B.1, voice leading is of particular importance in choral writing , and therefore choral writing requires the most restrictions on the fluidity of the voice leading. We are unaware of any musical styles that require non-fluid voice leading. This means that any system that satisfies the requirements of voice leading for choral writing will also satisfy the requirements of voice leading for non-choral writing.

A drawback to this technique is that it requires vocalists with training in musical theory and harmony, as well as ear training to identify when a note is no longer a harmonic tone, and to determine the nearest note. However, these drawbacks are trivially addressed in a computational reinterpretation. *LazyVoice* requires as input a set of at least 2 chords (as a single chord will not have voice leading), and a desired harmonic complexity. These two inputs provide the information that is critical to follow this technique, without requiring additional external knowledge. A video demonstrating *LazyVoice* is available at `https://youtu.be/9l_P46JWMrE`.

B.3.1 High-level architecture

LazyVoice uses a multi-agent architecture, which is useful for generative music due to its similarity to the ensemble nature of real-world musical performance [15]. This approach is especially useful for our purposes, as it simulates the real-world musical technique that we are duplicating.

A key differentiation between *LazyVoice* and other discussed chord progression systems is in *LazyVoice*'s representation of chords. *LazyVoice* uses a 2-dimensional vector to represent any chord, similar to the *GCT*. Unlike the *GCT*, *LazyVoice* stores the root position of the

most harmonically extended version of each chord — that is to say that *LazyVoice* stores a `C` Major triad and a $\texttt{C}^{13}$ as the same chord. To differentiate between a fully extended chord and a triad, each voice in *LazyVoice* has a user-selectable "chord depth" value, which dictates how far into the extensions the agent may look for its next pitch. This pared-down representation of chords allows for a controllable amount of harmonic complexity, while maintaining the functional harmonies of a chord progression. This abstracted representation is in line with Christopher Doll's separation of the function and content of a chord [5].

Each chord in *LazyVoice* contains all acceptable pitches within the chord, and therefore *LazyVoice* does not implement aspects of tonality such as keys and therefore modulation. This representation allows for a high amount of flexibility in possible chord progressions — chord progressions within *LazyVoice* may be consistent in key, or each chord may outline a new key, without requiring any additional data.

LazyVoice may generate music with between 2 and 8 total voice agents, including the bass agent. Each voice agent produces an individual monophonic line, and therefore the number of voice agents is equal to the number of independent voices. The number of voices is set by the user prior to playback, and may also be altered while the playback is paused. The maximum number of voices was determined during development, as our informal analysis found that when more than 8 agents are set to avoid doubling pitches, the resulting music is often overly crowded.

B.3.2 Progression agent

Because *LazyVoice* targets the realization of chords into voice leading, the progression agent is a translator between a provided chord progression and the voice agents themselves, as well as a metronome that directs voice agents when to change the playing chord. The interface for *LazyVoice* can be seen in Figure B.5.

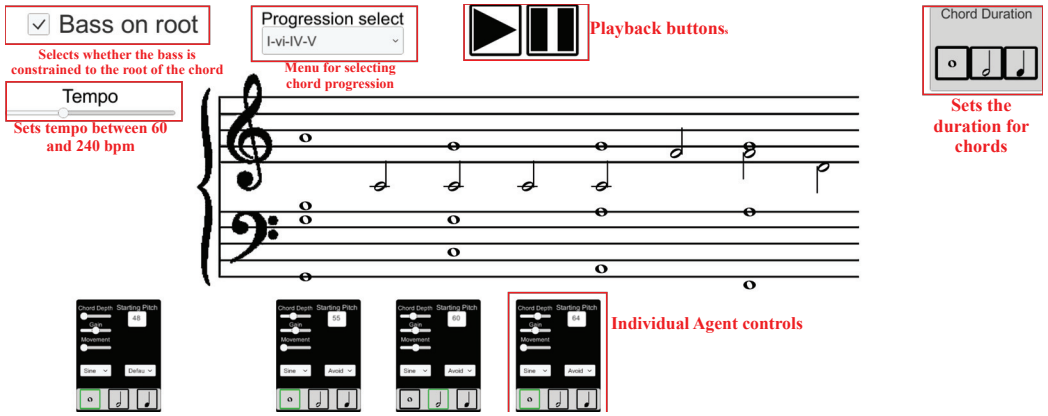


Figure B.5: A screenshot of *LazyVoice* with highlighted and labeled controls

We will first discuss the chord progressions themselves. In its current implementation, *LazyVoice* can select from common chord progressions or randomly shuffle the progressions. Chord progressions may be added to *LazyVoice*'s database directly in the code, as *LazyVoice* stores each chord as a simple vector consisting of the root pitch and the mode that the chord's extensions should follow.

LazyVoice's current scale database includes all standard modes in Western music theory, which are capable of expressing all standard chord extensions as well as several chromatic alterations. These modes can be understood as the scales that result from playing only the white notes of a piano, with each mode starting on a different white note. This means that there are 7 total modes, whose pitches are given in Table B.1. Each mode is stored in a look-up table, and therefore adding a new mode to the database is simply a matter of giving a name and a set of pitches above the root for the scale.

Table B.1: Diatonic white-note modes

Mode name	Pitches						
Ionian	0	4	7	11	2	5	9
Dorian	0	3	7	10	2	5	9
Phrygian	0	3	7	10	1	5	8
Lydian	0	4	7	11	2	6	9
Mixolydian	0	4	7	10	2	5	9
Aeolian	0	3	7	10	2	5	8
Locrian	0	3	6	10	1	5	8

We note that the modes in Table B.1 are not in ascending order of pitch, but instead in order of chord extensions. This order is used to allow for easy truncation of lists based on a user-selectable harmonic complexity, as will be discussed in Section B.3.3. Each chord is therefore stored by the distance in pitches from the key centre and a string corresponding to the mode of the chord. For example, an `e` minor chord in C major (or a `iii` chord using roman numeral notation), is stored as {4, Phrygian}, while an $\texttt{E}^7$ chord in C Major (a modally borrowed $\texttt{III}^7$ chord) is stored as 4, Mixolydian. As mentioned, the progression agent primarily serves to translate the user-selected chord progression to the voice agents — If the user selects a chord progression of `I-vi-IV-V`, the Progression agent translates directly to {{0, Ionian}, {-3, Aeolian}, {-7, Lydian}, {-5, Mixolydian}}. The distance from the key centre may be expressed as either a positive or negative integer — the only difference between the representations is how the user chooses to represent the chords. Because the `I-vi-IV-V` progression most commonly involves a bass movement downwards, we represent the root difference as negative.

The other function of the Progression agent is in acting as a time-keeper. The user may select a unified tempo between 60 and 240 beats per minute (bpm), using a slider, and during playback, the progression agent sends a message to all voice agents each beat.

B.3.3 Voice Agents

Voice agents are the primary agent responsible for implementing the *LazyVoice* algorithm. As previously mentioned, the core goal of these agents is to achieve fluid voice leading via maximum laziness, and can be seen in Algorithm 1. In cases where two potential notes are equidistant from the current note, the choice depends on the user-selected behaviour, with the default behaviour selecting between the pitches randomly with a 50% chance of either note. With the expand or converge behaviours, the agent selects the note that moves it away or towards the other voices, respectively. If an agent is following the avoid behaviour, it will

default to random unless two voices have the same pitch as one of their two equidistant pitches, in which case the agent will randomly select one voice to move to the contested pitch, and the other will select its alternative.

Algorithm 1: LazyVoice note selection algorithm

```
Input: Current note n ;                          // Pitch (36-96)
       Mode M ;                                  // From Table B.1
       Chord depth c ;                           // Triad-13th
Possible notes N ← Truncate M to length c;
if N contains n then
    return n;
else
    find closest note n_x in N;
    return n_x;
```

Each voice agent has a user-selectable harmonic complexity value, corresponding with how deeply into the mode the agent may look for acceptable next notes. While the use of only the 7 diatonic white-note modes may seem to limit flexibility, any heptatonic scale may be used to define the limits of a chord.

In addition to the core behaviour, voice agents have additional optional inputs that control their behaviour. Each voice has a user-provided movement parameter between 0 and 100, which corresponds to a percentage chance for spontaneous movement to prevent overly static harmonies. Each voice also has a duration value between a quarter note and a whole note. While the duration value has no effect on voices that do not spontaneously move, when combined with spontaneous movement, the duration value allows for the emergence of more complex rhythmic figures and quasi-ornamental figures, as the chance for spontaneous movement occurs on each note.

B.3.4 Inter-agent communication

A final behavioural modification for voice agents is an optional inter-agent communication. With their default values and behaviour, any single voice agent is unaware of other voice agents. With highly-spontaneous voice agents at high durations, the harmonies are capable of eventually converging into unison, or expanding beyond desired ranges. To combat this, agents may be set to "Avoid", "Converge" or "Expand". These options only affect the agent's behaviour when multiple possible notes are equidistant from the current note, or when spontaneous movement is triggered.

Expand and Converge behaviours share identical logic, though move in opposite directions. When an agent following these behaviours is spontaneously moving or there are two equidistant notes to choose from, the agent finds the average pitch of the current chord. The agent selects its next pitch by moving towards or away from the centre of the chord.

The Avoid behaviour triggers an additional step in the *LazyVoice* algorithm. When following this behaviour, an agent selects both an upper and lower potential next pitch, and store both as a vector of possible pitches. The agent then compares the neighbours to the possible

pitches that other agents have selected. In the event that there are no possible conflicts, the algorithm proceeds as before. If two agents share a potential pitch, the agents first determine whether one agent has less movement to its secondary pitch than the other agent, in which case the agent with the closer alternative pitch moves to its alternative pitch while the other agent moves to the previously contested pitch. In the event that both agents have the same distance to their alternative pitch, one agent is randomly selected to move to its alternative. *LazyVoice* does not implement any additional voice leading rules, such as avoiding parallel fifths or octaves.

Figure B.6 displays the controls available for each agent, and Table B.2 describes the highlighted controls.

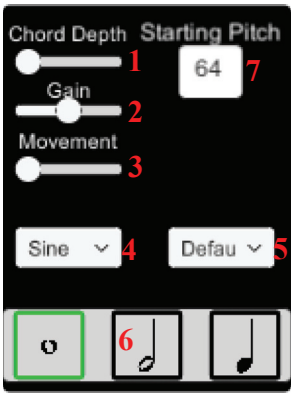


Figure B.6: An individual *LazyVoice* agent with numbered controls

Table B.2: Description of controls in Figure B.6.

#	Name	Description
1	Chord Depth	Acceptable chord extensions
2	Gain	Individual volume of the agent
3	Movement	Spontaneous movement chance %
4	Waveform	Waveform during playback
5	Behaviour	Behaviour i.r.t. other agents
6	Duration	Note duration
7	Starting pitch	Agent's first pitch

B.3.5 Bass agent

A special sub-type of voice agent is used in *LazyVoice* for creating bass lines. Because the function of a chord in music is so commonly related to the lowest note in the chord, the bass agent has an additional behavioural setting that locks it to the root note of each chord, bypassing all other behaviour.

If the bass agent is not constrained to the root of the chord, it functions identically to other voice agents, including the optional behaviours and user-input chord depth. Essentially, this behaviour allows for the user to select whether inversions are acceptable.

B.4 Output and Evaluation

LazyVoice is written in C# in the Unity game engine. The source code for *LazyVoice* is available at `https://github.com/CalePlut/LazyVoice.git`. *LazyVoice* outputs audio, as well as a visual grand staff representation. As before, this can be seen in Figure B.5, and in our demonstration video at `https://youtu.be/9l_P46JWMrE`.

We perform an informal musical evaluation of *LazyVoice*'s output. Figure B.7 demonstrates the output of *LazyVoice* with each voice set to whole notes with no spontaneous motion with the harmonic complexity setting allowing for the use of the 1st, 3rd, 5th, 7th, and 9th scale degree. Figure B.8 illustrates the output of *LazyVoice* with the same settings, but at a complexity that also allows for the use of the 11th and 13th scale degree. Both Figures also have the bass line locked to the root of the chord, for ease of analysis. Finally, for comparison, the top system of each figure is the LazyVoice output, and the bottom system is the chords in static open choral voicing[2]

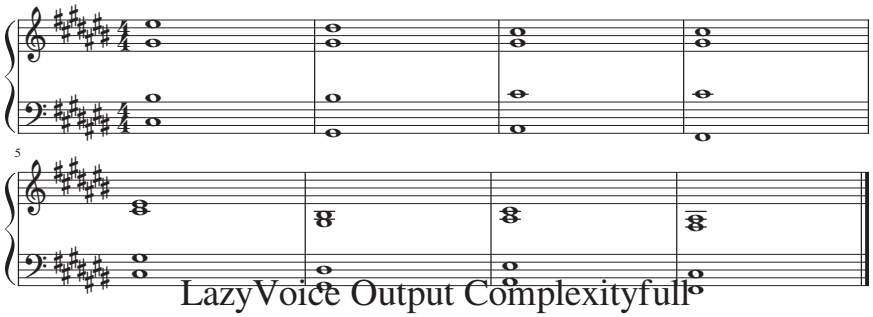


Figure B.7: Sheet music of LazyVoice at complexity 5 for each voice

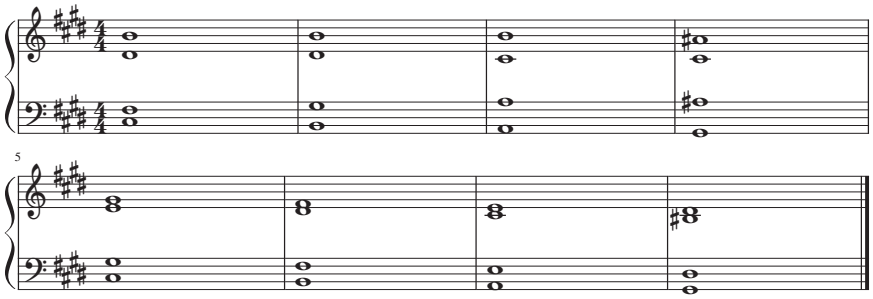


Figure B.8: Sheet music of LazyVoice at complexity 7 for each voice

We draw attention to the limited amount of motion between each voice, especially when compared to the motion that is present in the static voicing. We note that as in Figure B.3, the alto line in Figure B.7's *LazyVoice* output does not move at all, and the tenor line has only a single small motion. We also note that the chords in *LazyVoice*'s output contain chord extensions that do not sound out of place, as the makeup of the individual lines that results in those extensions flows melodically.

[2] A common voicing of chords in 4-part harmony where the 5th is doubled in the Tenor and Soprano line, while the Alto line follows the 3rd, and the bass line follows the root

In Figure B.8, we draw attention to the 4th measure, which contains a chromatic alteration — in an unaltered natural minor key as in the example, the v chord uses notes from the Phrygian mode. The alteration of the v chord from minor to a $\mathtt{V}^7$ chord, or a chord built upon the same root with a Mixolydian scale, is the most common chromatic alteration in minor-key western music, aligning with the harmonic minor scale [16]. Commonly, the 3rd scale degree is considered the primary carrier of scale information in a chord, and we note that while *LazyVoice*'s output does not contain a 3rd scale degree in the voicing of the chord, the modal change remains evident.

We note that *LazyVoice* does not guarantee that all notes of a chord are present. As before, this mirrors real-world voicings of chords as found in actual music. For example, members of jazz ensembles regularly play only notes that signify the chord's sonority such as the 3rd, 7th, and other extensions [13].

B.5 Conclusion

LazyVoice is a rudimentary system for realizing chord progressions into polyphonic voices. We believe that the design of *LazyVoice* provides a bridge between generative music systems that create harmonic progressions and full polyphonic generation. We note that the systems by Chen, Lin and Chen, as well as the system by Eigenfeldt and Pasquier output only a harmonic progression [6, 4], and in the evaluation of the *GCT*, voice-leading was performed manually [2]. *LazyVoice* allows for the translation of these harmonic progressions into polyphonic music with little additional labour.

LazyVoice may also be used as a tool for assisting human composers by suggesting fluid voice leading through a progression from any given start point. While inspiration for *LazyVoice* comes from choral techniques, we see no musical or technological reason that it may not be suited to any instrumentation. As mentioned in Section B.2, because choral voice leading presents the most restriction on voice leading, satisfying requirements for choral voice leading will necessarily satisfy the requirements for less-restrictive voice leading as well.

LazyVoice is currently a rudimentary yet effective system for generating fluid voice leading. Future work involves extending the user input and customization, allowing the user to have more control over the input and output of the system. Additionally, while *LazyVoice* outperforms block and root position chords, further formal evaluation is required to compare *LazyVoice* to other polyphonic generative systems. Overall, while the interaction with the system can be refined, believe that *LazyVoice* is successful at providing fluid voice leading from a harmonic progression.

Bibliography

[1] Clifton Callender, Ian Quinn, and Dmitri Tymoczko. Generalized voice-leading spaces. *Science*, 2008.

[2] Emilios Cambouropoulos, Maximos A. Kaliakatsos-Papakostas, and Costas Tsougras. An idiom-independent representation of chords for computational music analysis and generation. In *ICMC*, 2014.

[3] Carmine Cataldo. Towards a music algebra: fundamental harmonic substitutions in jazz. *International Journal of Advanced Engineering Research and Science*, 5(1):237359, 2018.

[4] Pei-Chun Chen, Keng-Sheng Lin, and Homer H. Chen. Emotional accompaniment generation system based on harmonic progression. *IEEE transactions on multimedia*, 15(7):1469–1479, 2013.

[5] Christopher Doll. Definitions of 'chord' in the teaching of tonal harmony. *Dutch Journal of Music Theory*, 2013.

[6] Arne Eigenfeldt and Philippe Pasquier. Realtime generation of harmonic progressions using controlled markov selection. In *Proceedings of ICCC-X-Computational Creativity Conference*, pages 16–25, 2010.

[7] Sang gil Lee, Uiwon Hwang, Seonwoo Min, and Sungroh Yoon. A seqgan for polyphonic music generation. *CoRR*, 2017.

[8] Yosef Goldenberg. Harmony without voice leading? the challenge of interpreting exact leaping transpositions. *Music Analysis*, 35(3):314–340, 2016.

[9] Gaëtan Hadjeres, Jason Sakellariou, and François Pachet. Style imitation and chord invention in polyphonic music with exponential families. *arXiv preprint arXiv:1609.05152*, 2016.

[10] Paul Head. ACDA choral workshop, 2005.

[11] D. Herremans, Ch. Chuan, and E. Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys*, 50(5), 2017.

[12] François Pachet. A joyful ode to automatic orchestration. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–13, 2016.

[13] Teri Parker. A modal approach to chord voicing. *Canadian Musician*, 40(1):26, Jan 2018.

[14] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An introduction to Musical Metacreation. *Computers in Entertainment (CIE)*, 14(2):1–14, 2017.

[15] Kıvanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, 48(1):56–105, 2019.

[16] Music Theory Team. Learning the harmonic minor scale. URL: `http://www.simplifyingtheory.com/harmonic-minor-scale/`, 2019. Last accessed: 04-25-2022.

fin