Expressive Performance Rendering for Reduced Music Repetition in Video Games

Kyle Worrall

PhD

University of York
Computer Science
September 2025

Abstract

The study of video game music (or *Ludomusicology*) is a relatively young field, and audio often receives fewer resources than other disciplines within video game development. As games have increased in length and complexity, this imbalance, combined with the slow adoption of algorithmic or procedural music solutions, has led to a continued reliance on the typical practice of fixed, looping musical content. However, excessive repetition is a documented issue, contributing to listener fatigue, reduced enjoyment, and diminished immersion. This thesis argues that contemporary video games have reached a scale for which traditional implementation methods are increasingly insufficient, making computational approaches to supplementing music necessary.

To investigate this, four studies were conducted. The first, a comparative case study of *Final Fantasy VII* (SquareSoft, 1997) and its 2020 *Remake* (Square Enix, 2020), establishes that repetition-related issues persist despite technical advancement. The second study uses reflexive thematic analysis of interviews with professional video game composers to examine barriers to AI adoption. While identifying potential benefits, composers cited concerns regarding creative control and a lack of expressive nuance. The third study addresses these concerns by introducing two novel expressive rendering systems: Cue-Free Express and Cue-Free Express + Pedal. These models enhance symbolic music playback using only pitch and quantised timing data as input to generate expressive variation, significantly improving perceived expressivity over existing baselines. The final study evaluates these systems *in situ* within a role-playing game; results indicate that while expressive variation is less perceptible during gameplay than in isolated listening conditions, it significantly influences player perceptions of repetition, performance, and compositional quality. Overall, this thesis adopts a pragmatic approach to contribute: 1) a framework for analysing repetition; 2) a set of design guidelines for AI music tools; 3) a novel expressive rendering architecture; and 4) the first empirical investigation of expressive rendering in a game context.

Acknowledgements

I must begin where my PhD journey does: with my son. Thank you, Erik, for your joyful company during this adventure; for being my best friend; and for teaching me the true value of work-life balance. Thank you to Salina for encouraging me to return to university and pursue my goals, for her unwavering support, and for the necessary reminders that holidays are meant to be taken. I am also grateful to Nannie Bev and Nannie Deb for caring for Erik every week, providing me with the vital time required to focus on my doctoral research. More generally, I wish to thank my family and friends for their endless support of my "nerdy" research interests.

I am deeply grateful to my supervisors, who have been fantastic to work with. Thank you to Prof. Josh Reiss for the guidance and the experience of working with Nemisindo; to Prof. Jon Hook for stepping in to support my PhD in the final year; and especially to Prof. Tom Collins, for his unwavering support over the last five years. The journey into parenthood, pivoting from music to STEM, and navigating this century's plague was certainly a challenging adventure; however, it was one made significantly easier by excellent mentorship.

My thanks also go to my colleagues in the Music Computing and Psychology Lab, who accompanied me on this journey. I am grateful for the unending patience and friendship of Zongyu Yin, Jemily Rime, Adrián Barahona-Ríos, Mark Hanslip, and Chenyu Gao. Thank you to my fellow IGGIs for their friendship and support, their acceptance, and for providing a community (albeit remotely) during the pandemic. Thank you to my colleagues at Edge Hill University for their support and advice during the write-up year. Last but not least, I thank the Centre for Doctoral Training in Intelligent Games and Game Intelligence (IGGI) for funding this research [EP/S022325/1].

Finally, a special thanks goes yet again to Erik. In truth, *while I am proud of this academic achievement, I will never be as proud of anything as I am of you.*

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References. Publications that contribute to this thesis are listed below:

Chapter 4 is informed by and evidenced in work published in the following peer reviewed journal paper:

> Kyle Worrall. Final Fantasy VII: Remaking Music for Modern Sensibilities. *Journal of Sound and Music for Games*, 5(4):34-57, 2024, doi:10.1525/jsmg.2024.5.4.34.

Chapter 5 is based on research published in the following peer reviewed journal:

> Kyle Worrall and Tom Collins. Considerations and concerns of professional game composers regarding artificially intelligent music technology. *In IEEE Transactions on Games*, 16(3):586-597, 2023, doi: 10.1109/TG.2023.3319085.

Chapter 6 is based on and informed by material published in the following peer reviewed journal:

> Kyle Worrall, Zongyu Yin and Tom Collins. Comparative evaluations in the wild: Systems for the expressive rendering of music. *In IEEE Transactions on Artificial Intelligence*, 5(10):5290-5303, 2024, doi: 10.1109/TAI.2024.3408717.

Other related work undertaken during the course of this thesis is listed below.

Conference paper (co-author):

> Corey Ford, Ashley Noel-Hirst, Sara Cardinale, Jackson Loth, Pedro Sarmento, Elizabeth Wilson, Lewis Wolstanholme, Kyle Worrall, and Nick Bryan-Kinns. Reflection Across AI-based Music Composition. *In ACM Creativity and Cognition.* 2024. Awarded honourable mention.

Journal paper (co-author):

> Tom Collins, Kyle Worrall, Chenyu Gao, Jemily Rime, Mark Hanslip and Ashay Dave. Perspectives of Professional Producers, Songwriters, and Composers on Roles and Rewards in New Cocreative Systems for Music-Making. *Submitted to Springer Nature AI and Ethics* Under review.

Book chapter (co-author):

> Catherine Flick and Kyle Worrall. The Ethics of Creative AI. *In Vear and Poltronieri, eds. The Language of Creative AI: Practices, Aesthetics and Structures.* Springer Series on Cultural Computing, pp. 73-91. 2023, doi: 10.1007/978-3-031-10960-7_5.

Related grants received during the PhD, related to PhD work:

> Kyle Worrall. "Artificial Intelligence Super Connector in connection with Google (Alphabet Inc.), Imperial College London, University of York, University of Leeds and University of Liverpool" Grant Code: G0108703.

Quotes

The desire for knowledge shapes a man.

Kvothe, in Patrick Rothfuss's *The Wise Man's Fear*

Focused hard work is the real key to success. Keep your eyes on the goal and just keep taking the next step towards completing it. If you aren't sure which way to do something, do it both ways and see which works better.

John Carmack

I have found, through painful experience, that the most important step a person can take is always the next.

Dalinar Kholin, in Brandon Sanderson's *Oathbringer*

Acronyms

AI Artificial Intelligence
ANN Artificial Neural Network
ASAP Aligned Scores and Performance Dataset
BFA Bayes Factor Analysis
CE Cross Entropy Loss
CFE Cue-Free Express
CFE+P Cue-Free Express + Pedal
CNN Convolution Neural Network
DAW Digital Audio Workstation
DL Deep Learning
FFNN Feedforward Neural Network
GNN Graph Neural Network
GGNN Graph Gated Neural Network
HRNN Hierarchical Recurrent Neural Network
IOI Inter-Onset Interval
LSTM Long Short-Term Memory
MAE Mean Absolute Error (L_1 Loss)
MIDI Musical Instrument Digital Interface
ML Machine Learning
MPS Music Performance Science.
MP3 MPEG-1/2 Audio Layer III Audio File Format
MSE Mean Square Error (L_2 Loss)
OGG Ogg Vorbis Audio File Format
OPD Ontime, Pitch, Duration.
PSBM Audio Symbolic Beat Map
RENCON Performance Rendering Contest.
RNN Recurrent Neural Network
UE Unreal Engine
VAE Variational Autoencoder
VR Virtual Reality
WAV Waveform Audio File Format

Contents

Acronyms **xi**

List of tables **xvii**

List of figures **xviii**

1 Introduction **1**
1.1 Introduction 1
1.2 Problem Area 4
1.3 Research Questions and Objectives 8
1.4 Clarification of Scope 9
1.5 Contribution to Knowledge 11
1.6 Statement of Ethics 13
1.7 Thesis Structure 14

2 Literature Review **17**
2.1 Music in Video Games 18
2.1.1 Definitions 20
2.1.2 Non-Linearity and Music Implementation 30
2.2 Contextualising the Problem of Listener Fatigue 37
2.2.1 Listener Fatigue 38
2.2.2 Contextualising Listener Fatigue in Video Games . . . 41
2.2.3 Procedural Design as a Potential Solution 47
2.3 Music Generation 51
2.3.1 A History of Music Generation 52
2.3.2 Terminology and Conceptual Scope of AI Music Systems 56
2.3.3 Fundamentals of Deep Learning 59
2.3.4 Deep Symbolic Music Generation 77
2.3.5 Expressive Rendering of Symbolic Music 82

2.3.6 Contextualising Music Generation and Expressive Rendering within Video Games . . . 95
2.4 Chapter Summary . . . 101

3 Methodology 103
3.1 Introduction . . . 103
3.2 Case Study . . . 105
3.3 Semi-Structured Interviews . . . 107
3.4 Reflexive Thematic Analysis . . . 108
3.5 Listening Studies . . . 109
3.6 Gameplay Study . . . 112
3.7 Bayes Factor Analysis . . . 113
3.8 Computational Analysis and Software Development . . . 118
3.8.1 Data Preprocessing and Model Training . . . 118
3.8.2 Predicted Feature Analysis . . . 123
3.8.3 Self-Attention Visualisation . . . 125
3.9 Chapter Summary . . . 126

4 The Evolution of Music Systems in Role-Playing Games 129
4.1 Introduction . . . 129
4.2 Contextualising Evolving Expectations . . . 130
4.3 Methodology . . . 133
4.3.1 Contextualising the Music Design of FFVII . . . 134
4.3.2 Contextualising the Music Design of FFVIIR . . . 136
4.4 Results . . . 138
4.4.1 A Comparison of OSTs . . . 138
4.4.2 A Comparison of Battle Music . . . 140
4.4.3 A Comparison of Bombing Mission . . . 146
4.5 Discussion . . . 151
4.6 Chapter Summary . . . 156

5 Factors Affecting the Adoption of AIMT 157
5.1 Introduction . . . 157
5.2 Background . . . 160
5.2.1 Technology Acceptance . . . 160
5.2.2 Contextualising PCG Acceptance . . . 161
5.2.3 Contextualising PMG Acceptance . . . 163
5.2.4 PMG Acceptance in Video Games . . . 166
5.2.5 AIMT Acceptance . . . 167
5.3 Method . . . 171
5.3.1 Recruitment and Demographics . . . 171

5.3.2 Data Collection 171
5.3.3 Data Analysis 173
5.4 Results 179
5.4.1 Benefits of Music AI 179
5.4.2 Complex Concerns about AIMT 183
5.4.3 The Clash between Ego and AI 184
5.4.4 No Standardisation of Support/Understanding 187
5.4.5 Lack of Understanding and Trust in AIMT 189
5.5 Discussion 191
5.5.1 Limitations 197
5.6 Chapter Summary 198

6 Modelling Musical Expression With Minimal Input Features 201
6.1 Introduction 201
6.2 Dataset and Learning Aims 206
6.2.1 Applicability of this Dataset to Video Games 207
6.3 Method 209
6.3.1 Data Preparation 209
6.3.2 Architecture 210
6.3.3 Training 215
6.4 Experiments 217
6.4.1 Listening Study 1 217
6.4.2 Listening Study 2 224
6.4.3 Listening Study 2 Discussion 230
6.4.4 Quantitative Analysis of Predictive Distributions . . . 231
6.4.5 Attention Mapping 239
6.5 General Discussion 243
6.5.1 Limitations 246
6.5.2 Future work 248
6.6 Chapter Summary 253

7 Evaluating AI Music Performances of Video Game Music 255
7.1 Introduction 255
7.2 Learning Aims and Reasoning 260
7.3 Methods 262
7.3.1 Creating a Video Game as a Testing Environment . . . 262
7.3.2 Implementation of AI Music Technology 267
7.4 Experiment 270
7.4.1 Experimental Design 270
7.4.2 Hypotheses 272
7.4.3 Results 273

7.4.4 Experiment Discussion 281
7.5 General Discussion 286
7.5.1 Limitations and Future Work 290
7.6 Chapter Summary 294

8 Coda: Conclusions and Future Work 297
8.1 Answering the Research Question 297
8.1.1 RO1: Evaluating the Evolution of Music Design 298
8.1.2 RO2: Interviewing Composers to Inform Design 300
8.1.3 RO3: Enhancing Music Playback with AI 303
8.1.4 RO4: Evaluating AI Performers In-Game 305
8.2 Future Work and Limitations 307
8.3 Concluding Remarks 309

References 311

List of Tables

1.1 The mean time (and standard deviation) in hours taken to complete the top ten selling video games between (and inclusive of) 2018 and 2022, separated by genre. 3

2.1 Summary of expressive rendering systems. OPD stands for ontimes, pitches, and durations. ANN stands for score annotations. 85

3.1 Bayes factor interpretation 117

4.1 The total time taken to complete both Final Fantasy VII and Final Fantasy VII Remake, averaged from self-reported data taken from HowLongToBeat.com, taken on 3rd October 2023. 134

5.1 Example of initial coding applied to transcribed interview excerpts. 176
5.2 Grouping of initial codes into broader codes and sub-themes during the thematic analysis process. 177
5.3 Candidate themes and relevant sub-themes 178
5.4 The sub-themes identified in the data, their count across the data set and the thematic groupings that they represent. UT is a shortening of Understanding/Trust. 181

6.1 The specifications of all models within the CFE & CFE+P architectures. 215
6.2 Final loss ratings for each model during training. 216

List of Figures

1.1 On the left, *Ultima: First Age of Darkness* (Garriott, 1981) one of the earliest commercial RPGs; on the right, *Final Fantasy VII Rebirth* (Square Enix, 2024). 2

2.1 The player avoiding a boulder rolling down a slope during a set piece in a level of *Crash Bandicoot*. 26

2.2 The play pathway from (Summers, 2016, p.14), outlining the typical musical architecture of a video game. 27

2.3 An example of two decisions in Until Dawn, which can lead to two very different musical outcomes. 31

2.4 Two scenes from *Final Fantasy IX*, the left showing exploration of the Evil Forest from a set angle, and the right showing changed perspective and mode for an instanced boss battle. 35

2.5 The end of the animation where Link opens a chest in *The Legend of Zelda: Ocarina of Time*, which is always accompanied by the same musical stinger. 35

2.6 An inverted-U graph, visualising the relationship between exposure to audio, and corresponding enjoyment of the same stimuli, after Berlyne (1971). 38

2.7 Conceptual model of hedonic response as a function of stimulus complexity and familiarity, showing optimal pleasure at moderate levels of novelty and complexity (after Berlyne (1970)). 40

2.8 A selection of Reddit threads showing the consensus of criticisms for Chalice Dungeons, a procedurally generated aspect of *Bloodborne*. 43

2.9 An example of Linear Regression on a Scatter Plot. The red line is the line of best fit, and if new data is shown to a model, it will make predictions to fit this line, depending on the variables of the input data (i.e. x), where output y is a continuous value for the predicted feature (i.e. a number within a range). 66

2.10 An example of Logistic Regression plotted on a Scatter Plot. The red line is the line of best fit, and if new data is shown to a model, it will make predictions to fit this line, depending on the variables of the input data (i.e. x). These predictions y will be a probability representing the classification of the group the data fits (i.e. 0 or 1). 67

2.11 An image taken from Fradet et al. (2021) demonstrating how a basic score is tokenised in both the MIDI-Like and REMI representations. 68

2.12 An illustration of the perceptron after Minsky and Papert (1969). Inputs X are multiplied by their corresponding weights W, summed together, and combined with a bias term. The resulting value is passed through an activation function to produce the output Y. 70

2.13 An image of the transformer architecture (Vaswani et al., 2017), the left hand portion of the diagram shows the encoder, and the right hand portion the decoder. 73

3.1 The Exploratory Sequential Mixed-Methods Process (Creswell, 2019) 104

3.2 The design of the research carried out in this thesis, after Creswell (2019). 106

3.3 Experimental design for Listening Study 1. Participants listened to multiple musical excerpts and rated each excerpt across four conditions (CFE, VirtuosoNet, Human, and Inexpressive). Condition order was counterbalanced across participants. 110

3.4 Experimental design of Listening Study 2. Participants listened to each musical excerpt rendered under seven conditions (VirtuosoNet, Basis Mixer, CFE, CFE+P, Human, Inexpressive, Randomised). Condition order was counterbalanced across participants. 111

3.5 Experimental procedure for the gameplay study. Participants played a game level under one music condition, rated the music across three dimensions, and then repeated this play–rate cycle for the remaining conditions. Data were exported for Bayesian Factor Analysis. 114
3.6 1. A note's `ontime` is extracted from the score; 2. This is combined with the PSBM to estimate the `onset` of the corresponding note in a performance; 3. This estimated `onset` is compared to the actual time at which the note is performed; 4. The difference is combined with the PSBM again to give `ontimeAdj/offtimeAdj`. The values indicated in the figure are exaggerated for sake of readability. 122
3.7 Raincloud plots of the distributions of dynamic predictions for each system across four stimuli in Listening Study 2. 124
3.8 A visualisation of the piano roll highlighting the fugue subject and the countersubject in Bach's *Fugue BWV 862* (Bach, 1722a). 127
3.9 A visualisation of the attention that is being paid in layer 6 of the transformer velocity model to previous notes in Bach's *Fugue BWV 862* (Bach, 1722a), originating from the fugue subject in the left hand. Each line colour denotes a different origin note in the restatement in the left hand. 128

4.1 A graph, where the x axis denotes low to high adaptation of original material to meet modern game standards, and where the y axis denotes low to high loyalty to original material. . . 133
4.2 Final Fantasy VII's engine modules, and arrows demonstrating possible game state transitions. 136
4.3 A timeline of the music in the first mission of both Final Fantasy VII and Final Fantasy VII Remake, and how this music aligns with sections of gameplay. VF denotes Victory Fanfare, and empty slots denote silence. 152

5.1 The demographic information for all participants, including: gender (TL), Experience (TR), Affiliation (BL), and Industry Level (BR). 172
5.2 Semi-structured interview questions used in the interviews with video game composers. 174
5.3 The percentage of coverage over individual codes that each theme represents. 180
5.4 A bar chart showing the participants openness to different AI-driven technologies. 200

6.1 1. A note's `ontime` is extracted from the score; 2. This is combined with the PSBM to estimate the `onset` of the corresponding note in a performance; 3. This estimated `onset` is compared to the actual time at which the note is performed; 4. The difference is combined with the PSBM again to give `ontimeAdj/offtimeAdj`. The values indicated in the figure are exaggerated for the sake of readability. 208
6.2 The inference pipeline of Cue-free Express (CFE) and Cue-free Express + Pedal (CFE+P) models (left); and a diagram after indicating the transformer architecture used (right). . . . 214
6.3 Raincloud plots of my results for Listening Study 1; expressivity, dynamics, timing and tempo ratings.Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side. 223
6.4 Raincloud plots of my results for Listening Study 2; expressivity, dynamics, timing and tempo ratings. The scores are integers between 1 and 7. Jittering of data means that there may be slight discrepencies between scores of the same value, i.e. if they sit around the same number but on either side. . . 229
6.5 Raincloud plots of the distributions of dynamic predictions for each systems across four stimuli. 233
6.6 Raincloud plots of the distributions of ontime adjustments made to each note in a piece for four stimuli. 236
6.7 Raincloud plots of the pruned distributions of offtime adjustments made to each note in a piece for four stimuli. 238
6.8 A smaller visualisation of the piano roll highlighting the fugue subject in Bach's BWV 862 (Bach, 1722a). A larger version appears in Chapter 3. 240
6.9 A visualisation of the attention that is being paid in layer 6 of my transformer velocity model to previous notes in Bach's BWV 862 (Bach, 1722a), originating from the fugue subject in the left hand. Each line colour denotes a different origin note in the restatement in the left hand. A larger version appears in Chapter 3. 240

7.1 A screenshot of gameplay seen in a 2D *Legend of Zelda* (Nintendo, 1986) clone, made in PyGame, produced by Clear Code. Zelda (and this clone) is an Action RPG where players participate in combat in real-time rather than instanced battles. . . 264

7.2 A scene from the inspiration for the video game in this study, namely *The Legend of Zelda: A Link to the Past* (Nintendo, 1992b). 265
7.3 A screenshot of gameplay seen in a 2D *Pokemon* (Game Freak, 1996) clone, made in PyGame, produced by Clear Code. Pokemon (and this clone) is an RPG where players explore the world in real-time (left), but battles take place in instances (right). 266
7.4 Raincloud plots of the results for the Gameplay Study, denoting ratings for Perceived Expressivity of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side. 274
7.5 Raincloud plots of the results for the Gameplay Study denoting ratings for Perceived Repetitiveness of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side. 276
7.6 Raincloud plots of the results for the Gameplay Study denoting ratings for Perceived Musical Quality of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side. 278

1 Introduction

1.1 Introduction

Video games are a constantly evolving medium, supported by technology that advances significantly each year (Prechtl, 2016; Wilde, 2014). This advancement has yielded marked improvements in graphical fidelity, real-time physics, and artificial intelligence (AI) (Cutajar, 2020; Prechtl, 2016). As the scope and duration of digital titles expand, procedural generation methods are increasingly employed to automate the creation of visual and mechanical content, facilitating the development of vast, immersive environments (Wilson, 2019; Summerville et al., 2018). The inclusion of these algorithms allows developers to extend gameplay longevity by algorithmically generating levels, producing a volume of content that would be manually unfeasible.

A genre of video games that demonstrates this growth particularly well is role-playing games (RPGs). The term RPG has varied definitions; its origins derive from the emergence of tabletop role-playing games from earlier tabletop war games, while also incorporating the early development of text-based and graphical video games on university mainframes prior to commercialisation (Kushner, 2003). However, in this thesis, RPG refers to the following definition, taken from a video documenting the history of the genre.

"RPGs are games which focus on player characters through features such as character creation, dialogue choices, narrative choice and consequence, and other types of player character expression, and combine this with statistical rules and gameplay such as experience points and levelling up, character skills and stats, loot progression and combat involving statistical calculations like hit chance or variation in damage numbers..." (NeverKnowsBest, 2024).

RPGs have advanced significantly over the last 40 years, at least visually, as evidenced by Figure 1.1. As a genre, they possess a noticeably longer expected runtime than many others (Gibbons and Reale, 2020; Armstrong, 2021). This is not unexpected, as this extensive duration is supported by features such as: the ability to increase character and weapon statistics in *Elden Ring* (FromSoftware, 2022); randomised "loot" (or items) that drop by enemies in *Diablo Immortal* (Blizzard Entertainment, 2022); the inclusion of additional party members to level up in *Persona 5 Royal* (Atlus, 2019); and alternative character classes to level up in *Final Fantasy XIV* (Square Enix, 2010).

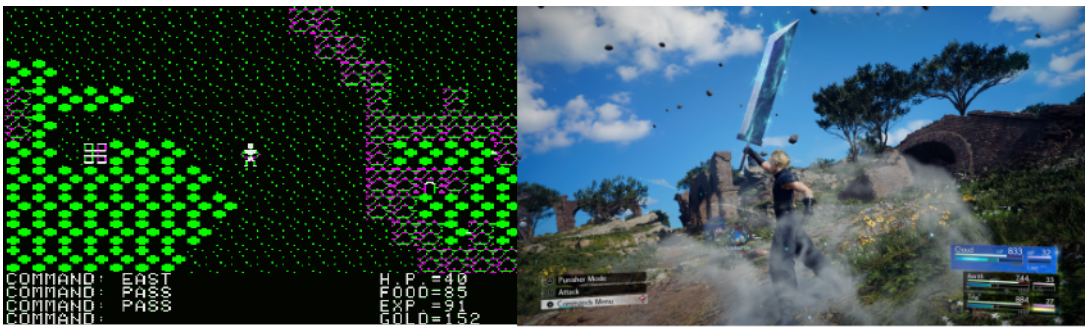

Figure 1.1: On the left, *Ultima: First Age of Darkness* (Garriott, 1981) one of the earliest commercial RPGs; on the right, *Final Fantasy VII Rebirth* (Square Enix, 2024).

Table 1.1 demonstrates that RPGs, as a genre, generally require more time to complete than many other categories. This table denotes the mean time (and standard deviation) required to finish titles across various genres. The values shown here are the average of the time taken to complete all games within a genre category, specifically for those ranked in the top ten games per year between 2018 and 2022.[1] While simulation games may initially seem to offer longer play time, these titles often include RPG features that extend their longevity.

	Main Story	Main Story + Extras	Completion
RPG	**36.55** (12.98)	**64.10** (26.78)	**128.35** (93.84)
Action Adventure	26.04 (13.23)	43.09 (25.81)	75.50 (54.13)
Simulation	42 (26.16)	85 (74.95)	241.5 (176.06)
FPS/Shooter	12.55 (8.77)	22.5 (18.10)	51.55 (41.10)
Racing	14.50 (12.73)	25.50 (9.19)	79.25 (53.39)
Platformer	11.25 (1.77)	21 (8.49)	43.75 (25.81)
Horror	9.50 (0)	12 (0)	37 (0)
Fighting	14.75 (12.38)	28.75 (15.20)	59 (12.73)

Table 1.1: The mean time (and standard deviation) in hours taken to complete the top ten selling video games between (and inclusive of) 2018 and 2022, separated by genre.

[1] All time data was sourced from HowLongToBeat.com, a website where users self-report the time required to complete different titles.

1.2 Problem Area

Before video games were considered a serious field of research, early psychology studies by Daniel Berlyne coined the term *listener fatigue* to describe fatigue driven by the repetition of audio stimuli, resulting in increased annoyance and decreased enjoyment (Berlyne, 1971). With linear media, such as film, the likelihood of suffering listener fatigue is low, as the music is composed to align precisely with the pre-determined run-time. By comparison, video games are a nonlinear experience in which the player's actions and choices dictate the flow of gameplay, allowing them control over how long sections of the experience may last. This, in turn, results in the music looping for durations determined by the player. This is significant because when comparing the average run-time of an RPG (36-128 hours) to its soundtrack (approximately 2-4 hours), it becomes evident that video games as a medium present a much higher risk of listener fatigue.

Although musical repetition is essential for establishing thematic material, audio repetition remains a well-established historical problem (O'Donnell, 2002; Vachon, 2009; Forss, 2016; Wilde, 2014). This issue is exacerbated by the limited computational resources allocated to audio teams regarding audio streaming and storage (Wilde, 2014; Marks, 2008; Plut and Pasquier, 2020). Contemporary video games use dynamic music integration techniques to account for this non-linear nature and to mitigate listener fatigue (Stevens and Raybould, 2015; Phillips, 2014). However, the adoption rate of computational or procedural methods to supplement musical material remains low; consequently, as titles grow in scale, existing techniques become less effective.

This is exemplified by titles such as *Path of Exile* (Grinding Gear Games, 2013) (an Action-RPG) designed to be "played forever" (Wilson, 2019). In this title, a player can invest thousands of hours across multiple characters, navigating extremely complicated skill trees and engaging with new quarterly content. Although seasonal updates occasionally include new music, the base soundtrack is approximately six and a half hours long and is utilised for the majority of play time. Even with a **larger than average soundtrack**, this case demonstrates how susceptible players are to listener fatigue.

Returning to Table 1.1, an examination of the top-selling RPGs between 2018 and 2022 reveals that these titles feature 4.6 hours of music on average. By comparison, the titles themselves have an average completion time ranging between 37 and 135.5 hours. Even in a best-case scenario, where music is distributed evenly across all sections of the game, this necessitates approximately eight repetitions of the entire soundtrack in the shortest case, and thirty repetitions in the longest. In reality, video game music is situation- or state-driven, leading to higher exposure to individual tracks, particularly during frequently encountered scenarios such as combat (Armstrong, 2021; Gibbons and Reale, 2020; Tate, 2021).

This thesis investigates the reasons why procedural and AI music is not widely adopted in the video game industry and explores the application of machine learning approaches[2] to augment video game music in ways that mit-

[2]The term *machine learning* is used here, and throughout the research questions, to frame the problem at a method-agnostic level. Deep learning approaches, including transformer architectures, constitute a subset of machine learning and are explored later in this thesis as specific instantiations of this broader class. Framing the research question in terms of machine learning avoids presupposing a particular model complexity; it also reflects prior findings in music generation research showing that increased architectural complexity does not necessarily result in improved musical outcomes (Yin et al., 2023).

igate listener fatigue while ensuring the generative output remains emotive and listenable. Machine learning approaches are adopted because they provide a principled middle ground between hand-crafted rule-based systems, which often struggle to generalise expressive nuance, and highly complex, data-hungry models that are frequently impractical for real-time interactive contexts. Machine learning enables expressive performance behaviours to be learned directly from data, remaining flexible enough to accommodate constraints on latency and authorial control. However, it is important to first define AI and AI music. A comprehensive definition of AI, machine learning, and deep learning is provided in Chapter 2; however, for the purposes of this research, AI is defined as "an algorithm which carries out a task that, if performed by a human, would be deemed to require intelligence" (Wiggins, 2006). Therefore, in this thesis, the term "music AI" refers to algorithms that carry out musical tasks typically requiring human intelligence. It is also necessary to define the terms "responsible" and "acceptable" with regards to music AI. A key motivator of this research is the pragmatic approach, in which the design of music AI tools is intended to be responsible and acceptable by not replacing the music creator within their own workflow (discussed further in Chapter 5). Although not detailed here in full, the premise of this research is to enhance or augment human creativity rather than replace it.

Throughout this thesis, various terms are used to describe computational systems that generate, transform, or analyse musical material, including AI music systems, procedural music, generative music, and machine learning–based music systems. While these terms may originate from different research traditions and are not strictly synonymous, they are frequently used

interchangeably in both academic and applied contexts. In this work, “AI music systems” and “AI music tools” serve as umbrella terms for computational approaches that autonomously or semi-autonomously support musical activity; this includes non-generative applications such as automated mixing, mastering, and music analysis. Where technical distinctions between these traditions are critical to the argument being made, they are explicitly clarified within the literature review (Chapter 2).

1.3 Research Questions and Objectives

The research question addressed in this thesis is as follows:

How can machine learning be used to responsibly and acceptably augment video game music to reduce the likelihood of listener fatigue in RPGs?

This main research question can be broken down into objectives, each of which will be addressed throughout this dissertation.

RO1: *To evaluate the evolution of music design and implementation in video games and to establish whether there is space for AI music systems in video games.*

RO2: *To evaluate and potentially impact the workflow, programming knowledge, and opinions of video game composers to inform tool design and use cases of AI music systems.*

RO3: *To explore the degree to which the quality of procedural music for video games can be improved by modelling expressive musical performance.*

RO4: *To evaluate how an AI music system impacts the perception of listenability and listener fatigue in role-playing video games.*

1.4 Clarification of Scope

There are well over one hundred different musical tasks that can be tackled with machine learning within the field of Music Information Retrieval (MIR). Examples of such tasks and the research around them include, but are not limited to: audio metadata tagging (Kong et al., 2020; Rime et al., 2023); expressive performance modelling for MIDI (Cancino-Chacón, 2018; Jeong et al., 2019b); mastering (Sterne and Razlogova, 2019); mixing (Moffat, 2021); beat detection (Estolas et al., 2020; Jia et al., 2019); transcription (Liu and Benetos, 2021); audio generation (Nistal et al.; Ghosal et al., 2023); music production (Deruty et al., 2022); sound source identification (Hou et al., 2023); symbolic (MIDI) generation (Huang et al., 2018; Collins and Laney, 2017; Guo et al., 2022); musical in-painting (Hadjeres and Crestel, 2021); sound effect generation (Barahona-Ríos and Pauletto, 2020); generative music evaluation (Yin et al., 2022, 2023); pattern recognition (Gao and Collins, 2023); and instrument sound synthesis (Hurle et al., 2020). Given this breadth, this thesis focuses exclusively on research concerning symbolic (MIDI) music generation, expressive performance modelling, human–AI interaction, and music for video games.

Additionally, while Large Language Models (LLMs) facilitate music generation via text-based conditioning, these approaches lie outside the scope of this research. This decision is informed by the findings in Chapter 5, which indicate that musicians tend to reject tools that diminish their musical agency, especially when such tools are accessible to users who are not musically trained, as these tools can make the professional landscape unfairly competitive in their eyes.

Furthermore, while contemporary research investigates generative music within tabletop and non-digital contexts, this thesis focuses exclusively on digital video games. Although non-digital games increasingly utilise software to enhance the player experience, video games by virtue of their native computational and non-linear architecture represent the primary platform for procedural music algorithms (Collins, 2009). While tabletop games present unique challenges, the research presented hereafter is concerned solely with the digital medium. Henceforth, any use of the term "game" refers to a digital video game.

1.5 Contribution to Knowledge

This thesis contributes:

1. A novel comparison of music systems across video game remakes spanning multiple console generations. This analysis establishes a new framework for ludomusicologists and video game researchers to examine the evolution of music design approaches, and the shifting priorities of players and developers alike.

2. The first empirical study specifically investigating the perspectives of professional video game composers regarding procedural music systems and music AI. Expanding upon foundational work such as Doornbusch (2002) and Knotts and Collins (2020), which focus on composers in other media, this research offers five key findings regarding industry concerns and expectations:

    - Composers identify clear potential benefits for AI music tools;
    - Professional concerns regarding music AI are multi-faceted and extend significantly beyond the fear of replacement;
    - A lack of standardised industry support for novel music AI is exacerbated by terminological ambiguity regarding procedural and AI systems;
    - A deficit in trust and understanding of AI-based systems persists, further complicated by current linguistic inconsistencies;
    - Creative AI inherently conflicts with the "creative ego," as these tools often perform tasks that overlap with the user's vocational identity.

3. The development and application of a novel ensemble-of-transformers architecture for expressive performance rendering. Directly addressing the industry view (established in the interview study) that AI-generated music lacks human-like expressivity, this work demonstrates that the proposed architecture can successfully render expressive nuances with minimal input features.

4. The first implementation and assessment of *Cue-Free Express + Pedal* and related algorithms within a real-time video game environment. This study provides empirical data on how AI-driven musical performance affects player perceptions of quality, repetition, and compositional intent.

1.6 Statement of Ethics

This thesis utilises empirical methods (i.e. listening studies and semi-structured interviews) involving human participants. All participants were fully informed regarding the nature of the studies, which were approved by the Physical Sciences Ethics Committee of the University of York, in accordance with the institution's Code of Practice and Principles for Good Ethical Governance.[3] All participant information sheets and questionnaires are provided as digital supplementary material, alongside the raw data and audio stimuli for experiments. These files are accessible via the links provided in the Methodology (Chapter 3) and subsequent research chapters.

It is essential to acknowledge that the research presented here contributes to the automation of human creative processes, such as music composition. While this could pose negative impacts for those working professionally in the music industry, this thesis adopts a proactive stance to avoid such collateral damage during the research and development of AI solutions. Specific mitigation measures include: integrating professionals and consumers into the design process to inform decision-making; avoiding the development of entirely generative algorithms; and omitting LLM-based tools to prevent lowering the barrier to entry in a manner that could destabilise the competitive landscape of the profession.

[3] https://www.york.ac.uk/staff/research/governance/research-policies/ethics-code/

1.7 Thesis Structure

This thesis is organised as follows. Chapter 2 provides an in-depth literature review encompassing ludomusicology, video game music, and listener fatigue to contextualise the motivation behind this research. This chapter also introduces terminology relevant to artificial intelligence, machine learning, and deep learning; it further reviews the application of computational technologies to music generation and expressive performance modelling, and explores technology acceptance literature. Chapter 3 outlines the methodologies employed, including semi-structured interviews, reflexive thematic analysis, Bayes Factor Analysis, and the development of software solutions for data analysis.

Chapter 4 evaluates the evolution of music design over 23 years through a case study comparing an original title with its remake. It also identifies gaps in music design where generative algorithms may benefit contemporary video games, addressing Research Objective 1 (RO1). Chapter 5 presents a series of semi-structured interviews with professional composers to assess the factors affecting the adoption of AI-powered music tools. This chapter addresses Research Objective 2 (RO2). Chapter 6 describes the development of two expressive rendering algorithms (*Cue-Free Express* & *Cue-Free Express + Pedal*) designed for use within Digital Audio Workstations or game engines. This chapter also details two listening studies and the quantitative analysis of model performances, addressing Research Objective 3 (RO3). Chapter 7 reports on the integration of the preceding work within a video game demo, alongside an experiment measuring how expressive variations affect player perception of repetition *in situ*. This chapter also outlines the technical de-

tails of system implementation and advises on future development pipelines, addressing Research Objective 4 (RO4).

Finally, Chapter 8 summarises the findings of this thesis and outlines potential directions for future research.

2 Literature Review

This chapter discusses previous work relating to multiple research areas, each of which is related to the later chapters of this thesis. As each of these chapters is based on research published or submitted to journals, specific parts of this literature review are taken from the peer-reviewed papers. Section 2.1 discusses why and how music is used in video games and defines key terminology required for a deeper understanding of audio in games[1] (relevant to Chapter 4). Section 2.2 contextualises the problem area of listener fatigue by discussing the phenomenon in more detail (relevant to Chapter 4). Section 2.3 introduces terminology pertinent to the discussion and understanding of machine learning and deep learning, and identifies symbolic music generation and expressive rendering algorithms outside of video games before comparing them with existing methods that have been used within video game settings (relevant to Chapters 6, and 7). Finally, Section 2.4 discusses factors affecting technology and innovation acceptance, and relates these models to contemporary technologies outlined in Section 2.3 (relevant to Chapter 5).

[1] While practitioners generally use the colloquial term "game audio", this chapter employs "video game audio" to maintain clarity regarding the specific medium and its audio practices. References to a "game" hereafter specifically denote the ludic structures, internal game states, or metaphorical frameworks of play within that medium.

2.1 Music in Video Games

The video game industry has become one of the largest industries in the world, with the UK consumer video games market alone reaching a valuation of £7.82bn in 2023 (a 4. 4% increase over the previous year) (UKIE, 2024). The global video game market in 2022 generated £140.3bn[2] and is estimated to have reached £141.2bn in 2023 (a 0.6% increase on the previous year) (Wijman, 2023).

The video game music market has seen similar growth, unsurprisingly, with major record labels and music artists contributing to soundtracks (Tessler, 2008), where entire video game soundtracks are sometimes comprised entirely of popular music (Cutajar, 2020, p.12). Such examples of these types of soundtracks include *Grand Theft Auto V* (Rockstar North, 2013) and *Forza Horizon 5* (Playground Games, 2021). The global video game music market was valued at £1.31 billion[3] in 2023 (Custom Market Insights, 2023), a small fraction of the global market, but shows the increase in interest in video game music.

Similarly to how non-video game music has crossed the boundary into video games, such as the use of popular music in video game soundtracks, so too has video game music escaped the medium, denoting the rising popularity of the genre, as well as its acceptance as an art form (Cutajar, 2020, p.12). Examples of this include: the increasing number of live orchestra tours that play exclusively video game music such as Distant Worlds[4] and Elden Ring Symphonic Adventure[5]; the number of Grammys that have been won by

[2]This figure was originally reported in USD.
[3]This figure was originally reported in USD.
[4]https://ffdistantworlds.com/
[5]https://en.bandainamcoent.eu/elden-ring-concert

video game soundtracks (Rivera, 2024); and the inclusion of its own category at the Grammys in 2023 (Rivera, 2024).

While there is widespread interest in video game music from a consumer standpoint, the study of *Ludomusicology* or video game music as it is commonly known is a relatively new field of academic research compared to related fields such as film, contemporary or computer music (Wilde, 2014; Worrall and Collins, 2023). This is unsurprising, yet parallels the treatment that video game audio receives – and has historically received – where it is often considered an afterthought in game development, with minimal allocation of resources (Marks, 2008, p. 3; Wilde, 2014, p.1; Plut and Pasquier, 2020, p. 11; Worrall and Collins, 2023, p. 2).

Literature (and the industry as whole) generally uses *game audio* or *video game audio* as an umbrella term to describe all audio within a video game, including *music*, *voice over* and *sound effects*. However, throughout this thesis: *music* refers to any in-game music that accompanies a video game experience (i.e. during gameplay, credits, the loading of the game) (Cutajar, 2020); *sound effects* refers to the general definition of generally "non-musical sounds" such as user interface sounds, ambiances (environmental sounds) or foley (player character sounds) (Collins, 2008, p.165); *voice over* refers to any dialogue that is captured, recorded or processed as part of the in-game experience.

I will now define some necessary terminology in the following section that relates to video game audio and video games as a whole, before discussing the role and the importance of video game audio and, more specifically, music in video games.

2.1.1 Definitions

Defining Games

The term *game* is nebulously defined, with many researchers arguing about the nuance of what factors truly define a game (Stenros, 2017). For example Wittgenstein (1986, pp.31–32) attempts to define games regardless of the medium in which it is played (i.e. tabletop, video [computer], cards or sports) and proposes the idea that all games have "family resemblances" but that no set features can be defined to fit all kinds of games. In the same decade, however, Ellington et al. (1982, p.9) state that the definition "proposed by Clark C. Abt in 1968" was accepted by the majority of workers. This definition states that a game is "any contest (play) among adversaries (players) operating under constraints (rules) for an objective (winning, victory, or pay-off" (Ellington et al., 1982, p.9). Stenros (2017, p.1) explores this issue further, comparing over 60 definitions since the 1930s to better understand the concept of a game, and finds that this terminology is constantly evolving. As such, given that there is no widely accepted definition of a "game" I determine that such work is beyond the scope of this thesis, and follow the precedent of earlier music generation work by describing games using the genres and sub-genres in which this work focuses (Cutajar, 2020, pp.13–15).

In this work, as explained in Chapter 1, I define Role-Playing Games (RPGs) as a genre of video games as described in an oral history of the genre: "RPGs are games which focus on player characters through features such as character creation, dialogue choices, narrative choice and consequence, and other types of player character expression, and combine this with statistical

rules and gameplay such as experience points and levelling up, character skills and stats, loot progression and combat involving statistical calculations like hit chance or variation in damage numbers...” (NeverKnowsBest, 2024).

Music’s Role in Video Games

Video game music research, like video game research, suffers from a disproportionately large dictionary of terms, with opposing definitions, such as those for the terms “procedural audio”, “procedural music” and “procedural generation” (Worrall and Collins, 2023). However, before focusing on the more technical language of video game music, it is important to first consider how the music is experienced during play. Cutajar (2020, p.21) notes that language surrounding video game music analysis is often derived from filmic traditions, which is not surprising given the similarity between the mediums, and describes the two main levels of music use in video games as follows.

Intra-diegetic audio: audio which occurs within the game’s world itself, and can be heard by players, but also characters/agents within the video game world. Diegetic music therefore, is music whereby the source of the music is within the game’s world. Examples include radios, jukeboxes and bards/street musicians in taverns.

Extra-diegetic audio: audio which occurs outside of the game’s world, and cannot be heard by the player character or other agents. This might include musical passages that accompany opening chests (as in The Legend of Zelda: Ocarina of Time (Nintendo, 1998)) or victory music after a level finishes.

While there are similarities between the ludic descriptions of intra-diegetic and extra-diegetic audio in video games vs the filmic terminology of diegetic and non-diegetic, there are important factors to consider in why they are different. Firstly, while film theory can describe audio as within the narrative diegesis of the film (i.e. sound effects, dialogue), or not within the diegesis (i.e. narration or score), for video games this is more complicated. In films, the audience is a passive listener or observer, where the story is fixed, and there is a single frame of reference for the viewer. However, this is not true for video games. Within the game world, there are two separate listening positions: the first is that of the character (within the diegesis), and the second is that of the player (outside of the diegesis). The distinction of intra- vs extra-diegesis builds on diegetic vs non-diegetic by describing where the sound is operating (and not just whether or not it is sourced from within the diegesis/narrative). Furthermore, while intra-diegesis aligns closely with diegetic audio in films, this audio is intended to be heard by the character, and it is spatialised and behaves accordingly. Extra-diegetic audio, by comparison, does not merely exist outside of the video game world or diegesis, but typically addresses the player and not the character, communicating important ludic information to the player (i.e. UI sounds and health warnings).

To clarify, while there are similarities between how film and video game music can be described and understood, there are additional roles that music can play in a video game such as informing player behaviour which mean that diegetic and non-diegetic terminology are not necessarily specific enough to fully capture the relationship between different types of audio in video games (Jørgensen, 2007).

Jørgensen (2007, p.105) defines a third type of diegesis within video game music, namely **Trans-diegesis**. Trans-diegesis describes sounds which cannot be cleanly defined as either intra-diegetic or extra-diegetic, as they blur the lines between diegetic boundaries, operating simultaneously across both. This language is increasingly necessary as audio systems in video games evolve to be more responsive to real-time parameter changes and communicate ludic information to the player from within the diegesis of the narrative. An example of trans-diegetic sound is character heartbeats becoming more audible as their health drops. This change in volume and processing is occurring within the diegesis of the video game world as the player character is hearing their own heartbeat increase with the tension in the game; however, this is also performing a ludic function by alerting the player to increased risk. A musical example of trans-diegesis is demonstrated by the adaptive music system featured in the boss battle with "Becki the Bride" in *Dead Island 2* (Dambuster Studios, 2023). In this example, a spatialised version of "Sad Wedding" (Gaye and Terrell, 2023) is playing within the diegesis of the game world during narrative exposition, but the music system transitions this piece of music to outside of the diegesis as a non-spatialised sound file when the boss monster enters the area (Lightbound and Williams, 2025). With both examples, merely describing them as intra-diegetic or extra-diegetic misses their designed ambiguity.

Beyond academic research, colorrev some industry professionals use filmic language to describe the role of the music they write, describing music as diegetic or non-diegetic, avoiding the complexities of extra vs. trans-diegesis (Thomas, 2016, pp.18-19). However, this is not true of all video game audio

professionals, and as video game technology and audio middleware evolve, terms such as trans-diegetic are increasingly discussed at video game industry conferences (Lightbound and Williams, 2025). While this may seem inconsistent, terminology between industry and academia is not always exclusively aligned (which will be discussed in more depth in Chapter 5). Now that I have covered the roles of music within video games on this broader level, we will step a level deeper in video game audio terminology and define the functions that music plays.

Music's Function in Video Games

Beyond the three levels of musical diegesis in video games, the role of music in video games can also be described by the music's function. Unlike linear media such as film, where the above terminology originates and music is generally written to synchronise with visuals, music in video games is non-linear. This means that, like the medium itself, the flow of action and story is dictated or, at the very least, influenced by the actions of players. With this in mind, different versions of musical pieces, or even entirely different pieces, can be experienced as the narrative changes based on player actions.

To understand this non-linearity further, and before discussing implementation, it is necessary to consider the player experience. (Summers, 2016, p.14) visualises a generalised model of the sonic experience of the player in the *play pathway*, which denotes the player's experience from the booting up of a video game to the finishing of the gameplay experience. Figure 2.1.1 demonstrates that players go through a series of loading screens and menus with accompanying music before being introduced to a level using a cutscene

(a typically short linear film that introduces the narrative). Within a level, this cutscene is considered the beginning of any **Gameplay Music**. Gameplay music is the music that is experienced during play. After the starting cutscene, players hear *loops* of music (a term used to define a piece of music that can seamlessly repeat until the game determines it is time for the music to change). Over the top of these loops, a player might hear stingers, which are short musical passages or segments that accompany behaviours within the game.

An example of this is the opening of chests in the *Legend of Zelda* (Nintendo, 1986) series. With the non-linear nature of gameplay, after set loops, there are multiple layers or branches that could play to correspond to a player going into a boss room, into a safe zone, or hiding from enemies in stealth. The implementation of these is discussed later in this chapter, but for now, knowing that these are possible and that they affect music playback is requisite for further discussion. Within the play pathway, music will transition between set-pieces and cues (action segments that are timed, such as when a boulder falls from high and chases Crash Bandicoot down a hill in *Crash Bandicoot* (Naughty Dog, 1996), or Nathan Drake enters an area and is attacked by enemies in *Uncharted: A Thief's End* (Naughty Dog, 2016)). These set-pieces are usually determined in length, but the music can loop endlessly if the player fails to end the sequence. The end of the level can have multiple different pieces, such as for success, failure, or additional end states, such as secret endings.

Figure 2.1: The player avoiding a boulder rolling down a slope during a set piece in a level of *Crash Bandicoot*.

This generalised model is useful for describing the general shape of music during gameplay as experienced by the player and introduces some basic terminology, such as loops. However, it does not explore the function of the music being used in video games. Function and implementation of video game audio is an area in which academic and professional texts discuss similar ideas with varying terminology, making for overly complicated discussions (Worrall and Collins, 2023). For example, Winifred Phillips (2014, pp.97-116) award winning composer for video games and author, writes about the functions of music in video games and describes them as follows: music as a state of mind (where music supports being in the correct state of mind for playing a specific genre such as horror); music as a world builder (where music can be used to support world building with in-world musicians or songs); music as a pace setter (where music can be used to trigger rhythmic

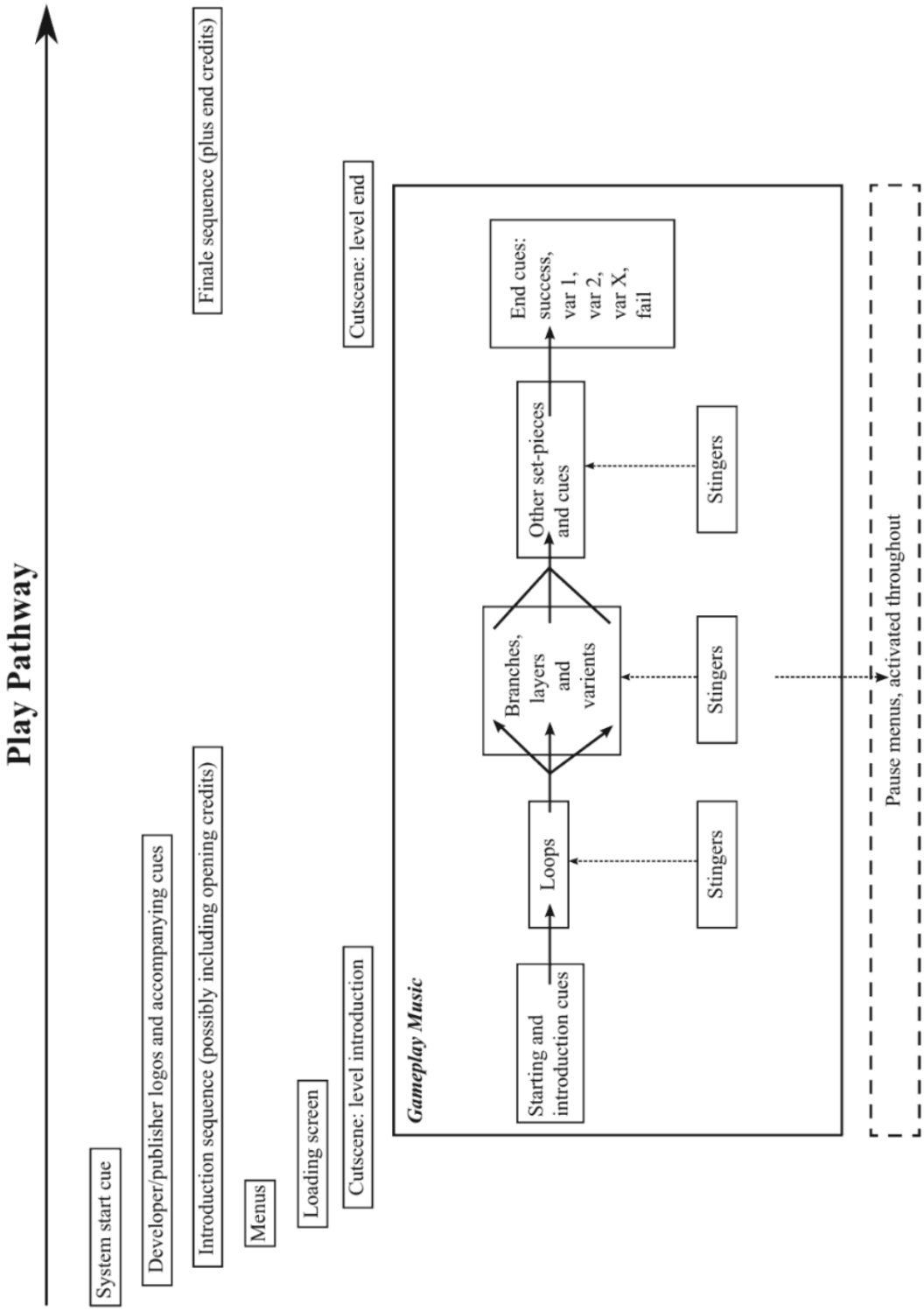

Figure 2.2: The play pathway from (Summers, 2016, p.14), outlining the typical musical architecture of a video game.

entrainment [6] and increase the psychological pace of gameplay; music as an audience (where music is the audience to player action, playing a reactive role – widely applying to most video game music); music as branding (where music becomes part of a franchise brand such as the Halo theme). Furthermore, Phillips (2014) differentiates between scoring for linear media (such as cut-scenes) and scoring for non-linear scenes (gameplay), but states that the functions of the music are typically the same regardless of linearity.

Another award-winning composer and author, Chance Thomas (2016, pp.20-26), similarly describes the functions of music in video games as: setting the mood; heightening emotion; propelling action; providing contextual clues; enhancing aesthetic; and contributing to structural unity. Audio director and author Aaron Marks (2008, p.378) supports these ideas stating that "music is primarily designed to create a certain atmosphere or feeling for the player whole in the game world," and that "music [...] can also add a sense of realism" and even "place us in a different world." While many of the ideas here are similar in nature, there are some differences, and implementation terminology in particular will make the differences more apparent.

Academic research supports all these claims with papers across many genres of video games giving evidence that both music and sound play a vital role in: establishing and maintaining fear or terror in horror video games (Roberts, 2014); supporting immersion (Sanders and Cairns, 2010); providing ludic information to improve player performance in stealth and first-person shooters (Stevens et al., 2015; Grimshaw et al., 2011, 2008); directly improving player dexterity (Shah and Oberoi, 2021). Additionally, audio has

[6] Rhythmic entrainment is a music psychology term describing how biological rhythms such as heart rate attempt to match and keep pace with external musical rhythms (Trost and Vuilleumier, 2013).

been shown to play an important role in supporting immersion within the verisimilitude of the video game world by creating an "immersive fallacy" (Collins, 2008, p.133). Furthermore, even music choice is shown an impact on increasing immersion and reducing awareness of time passing during play (Sanders and Cairns, 2010) and supporting flow states (Grimshaw et al., 2013).

Beyond the more philosophical functions of music within video games, it is also widely agreed that video game music can be described by the type of set-piece (gameplay section) that the music accompanies (i.e. by the requirements of the game). For example, (Phillips, 2014, pp.145–155) describes the requirements of music within a video game as: trailer music; main theme; tutorial music; action and ambient tracks; puzzle tracks; scripted cut-scenes and sequences; experimental tracks; and attract mode (which dates back to when arcade machines had to loudly vie for the attention (and coins) of players). Furthermore, she relates these requirements back to the function and roles described earlier. For example, main themes, atmosphere tracks, tutorial music and trailer music establish the atmosphere and "scene". By comparison, action music and puzzle music support the player's immersion in specific tasks, and the music of scripted scenes can play a variety of roles. (Marks, 2008, pp.231–233) uses similar but less granular language to describe these musical requirements: intro, closing and credits; menu screens, gameplay music, loops and ambient tracks (which broadly encapsulates most of the topics Phillips (2014) covers; cinematic sequences and plot advancement, cut-scenes and tie-ins (similar to scripted/cut-scenes in Phillips (2014)) and win/lose finale cues. However, being an older text, focused on earlier con-

soles, (Marks, 2008, p.234) uses a broader category to describe "Interactive Music" to describe the differences between music for linear and non-linear mediums.

2.1.2 Non-Linearity and Music Implementation

The non-linearity of video games contextualises the functions and roles of music and complicates the music composition process more than with linear media. Ludomusicologist and author Karen Collins (2008, p.142) goes as far as to describe non-linearity as the "most significant problem facing game composers." Andrew Glassner (2004, p.244) supports this, outlining in his text on interactive storytelling that if a story contains ten branches to its narrative, and each branch offers three choices, then the author must "write a few more than 59000 stories" to account for all the possible directions the story could go in.

For example, in horror films, music and sound play a role in establishing and maintaining fear and terror. However, when playing a video game, the player is an active participant in the experience, rather than a viewer, and if they choose to move away from danger, then the music needs to appropriately respond to this. A very successful blockbuster video game that exemplifies this is *Until Dawn* (Supermassive Games, 2015). *Until Dawn* is an interactive horror drama video game, where players decide how teenagers in a horror film-esque scenario respond to dangerous situations, simulating an interactive horror film experience. With this type of branching narrative, it is necessary to compose more music than would be needed for a linear experience (to account for and support the diversifying narratives that can

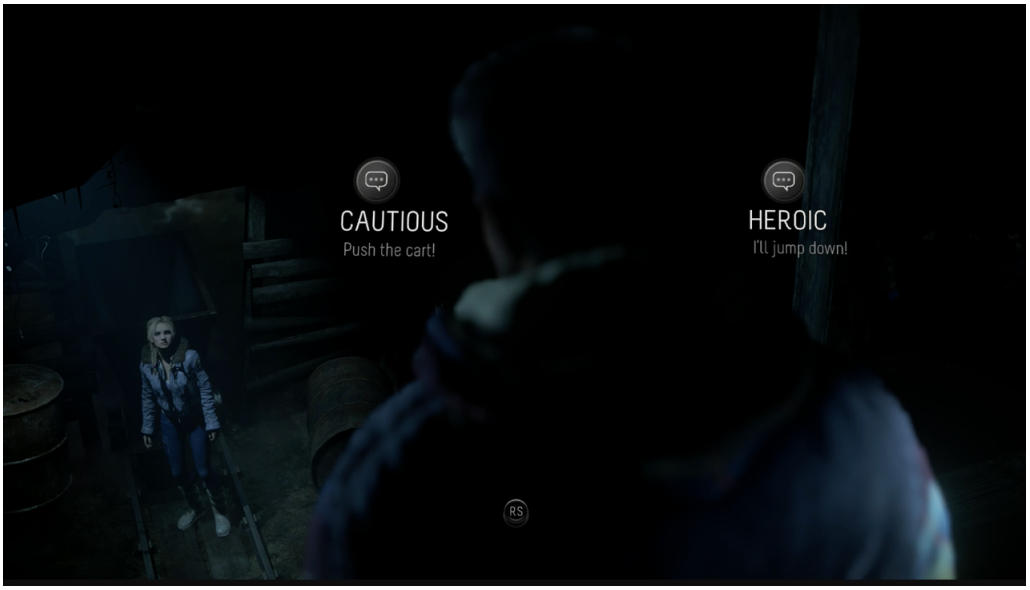

Figure 2.3: An example of two decisions in Until Dawn, which can lead to two very different musical outcomes.

occur). However, branching narratives do not impact the composition process solely, but require music to be implemented in ways that can account for sudden in-game changes driven by player behaviour.

Before discussing the technical implementation of music in video games, and the effects that different approaches can have on the player experience, it is important to also consider how music in video games is described beyond its role, function and level within the diegeses. The lexicon of video game audio is rife with contrasting and competing definitions, so this is not always a simple task (Worrall and Collins, 2023).

Collins (2008, pp.184–185) being one of the earliest academic texts on video game audio describes audio as being:

1. *Adaptive*, which describes any sound that occurs in response to the game environment or gameplay, as opposed to the user;

2. *Interactive*, which describes any sound or music that occurs in response to gameplay related to the player, such as to controller input;

3. *Dynamic*, which describes any audio that is designed to be changeable and reacts to changes in the gameplay – thus "encompassing both interactive and adaptive audio".

However, Stevens (2021, p.76) argues that the distinction between adaptive and interactive audio is up for much debate, citing differences between distinctions made by (Collins, 2009), (Sweet, 2015), and (Selfon, 2003). Furthermore, Stevens (2021, p.91) explains that the term "interactive" is generally used to describe music during active episodes of gameplay, but describes most music in video games as "not truly interactive, in that it lacks a reciprocal relationship with the game's systems," determining that *audio that was previously considered interactive should now be considered reactive.* This is important to consider, as technology is evolving in a way that allows for players to react to the game, and the game to react to the player, creating "true interactivity," where a truly interactive music system should be able to communicate with the game and vice versa to move beyond reactionary behaviours.

Music Implementation

Now that the functions, roles and descriptors of video game audio (adaptive, interactive and dynamic) have been outlined, and the problem that accompanies the non-linear nature of gameplay has been understood, it is time to discuss music implementation. Historically, during video game development, it has been common practice for composers to create individual loops of music that can be handed off to development teams and implemented in a semi-static fashion, where these tracks were transitioned between to support the situation. This approach has two competing definitions, the transitional approach (Stevens and Raybould, 2015) and horizontal resequencing (Phillips, 2014), and henceforth I will use the transitional approach to refer to this style of implementation. The transitional approach utilises the transition of different pieces of music for different sections of the game, with a focus on retaining some type of musical coherence and quality, while attempting to enact a timely change. A second method is the parallel approach (Stevens and Raybould, 2015) or vertical layering (Phillips, 2014), which utilises changes in layers of music based on real-time changes in-game parameters. With these terms, they are interchangeable, so henceforth I will use the term parallel to describe this type of implementation, in keeping with transitional as both originate in Stevens and Raybould (2015). Compared to the parallel approach, the transitional approach can be slower to react to in-game changes, as it is necessary to wait for the right moment to transition to ensure that transitions are musically satisfying. To exemplify these approaches, in *Final Fantasy IX* (SquareSoft, 2000) they player will encounter "Dangers in the Forest" (Uematsu, 2000c) when exploring the Evil Forest area. However,

when they enter an instance of randomised combat (a separate section of gameplay which the game transitions to), this piece of situation music ends, and the piece "Battle 1" (Uematsu, 2000a) plays. At the end of combat, the battle music fades out and "Victory Fanfare" (Uematsu, 2000b) plays briefly as the player sees the experience and items their characters have gained, before "Dangers in the Forest" (Uematsu, 2000c) resumes during exploration. By comparison, in stealth and stealth action video games, the parallel approach is used to manage the volume of layers of music to match intensity. For example, in *Rise of the Tomb Raider* (Crystal Dynamics, 2015), percussion will be muted during exploration, and some slower layers of percussion join the mix when enemies are near but the player is being stealthy. However, if the player is spotted during a stealth sequence or fails to take out an enemy stealthily, then all the additional layers of percussion will fade in to increase the player's excitement via emotional induction (Trost and Vuilleumier, 2013). With the parallel approach, this allows for musical coherence as the underlying piece remains the same, but the shape of the arrangement evolves in real-time, vis-a-vis the transitional approach, where different themes and pieces swap in and out to match the situation.

Transitional and parallel music design are the most common approaches used in the video games industry, but Stevens and Raybould (2015, pp.227–229) define two more approaches. These approaches are: *ornamental*, where musical stingers (or fragments) play over the top of a soundscape/background piece, such as when Link opens an item chest in *Legend of Zelda: Ocarina of Time* and the opening of the chest is accompanied by a short ostinato; and *algorithmic* (sometimes described as procedural), which is less common, and

Figure 2.4: Two scenes from *Final Fantasy IX*, the left showing exploration of the Evil Forest from a set angle, and the right showing changed perspective and mode for an instanced boss battle.

Figure 2.5: The end of the animation where Link opens a chest in *The Legend of Zelda: Ocarina of Time*, which is always accompanied by the same musical stinger.

typically constitutes the timing, playback, and generation of note-level music events, such as the stochastic generation of endless melodies in Ballblazer (Lucasfilm Games, 1985).

The definition of procedural music is not necessarily a concrete one, and misunderstandings persist between academic and industry terminology, as will be expanded upon in Chapter 5. However, generally in video game research, procedural and algorithmic are used interchangeably (Cutajar, 2020;

Prechtl, 2016; Worrall and Collins, 2023). Procedural music was first defined in Wooller et al. (2005), which describes two distinct procedural processes in algorithmic music systems. Wooller et al. (2005) defines: transformational systems, where music data is computationally re-arranged to create what appears to be new music; and generative systems, where new music data is created at run-time. The main distinction in this initial definition is whether or not the system creates new data. These definitions were later expanded upon by (Plut and Pasquier, 2020, p.5), who define procedural systems by the tasks that they perform:

1. Composition, where "the algorithm addresses the creation of new music and structures";

2. Arrangement, where "the algorithm addresses the recombination of extant musical layers or elements";

3. Performance, where "the algorithm addresses the interpretation of extant music within the game".

Procedural music systems are the least common approach within the video games, and it is not fully understood why, although it is suggested that the historic lack of resources for audio plays a role (Plut and Pasquier, 2020), and that the need for robust timing systems that are not reliant on frames, and the lack of audio programming expertise could be contributing factors (Weir et al., 2021). This is a motivation of this thesis, and will be discussed in more detail in chapter 5. Regardless, it is accepted that procedural approaches are a valid system for offering less repetition to players (Dutta, 2020, p.174).

2.2 Contextualising the Problem of Listener Fatigue

Now that we have covered the positive impacts of video game music, how music is implemented and the effects that it can have, and defined the requisite lexicon, it is time to turn to the elephant in the room, and what I consider to be the second biggest problem for composers after non-linearity: repetition.

Repetition of assets in video game audio (and games as a whole) is largely a problem due to the non-linear nature of video games. Compared to a film, where the playback of music and sounds is linear and cut directly to the visuals, video games are driven by player actions. Considering the previously explained implementation methods, if a player leaves a specific area with a backing track, the music will transition to music appropriate for the new area; however, when and if they make this transition is dictated by the player (beyond finite examples where levels are timed). This means that in most cases, the player is in control of the amount of time they experience audio (and visual stimuli). As this is not a linear experience and the data allowances for the audio files are not infinite, this means that music and audio will be repeated indefinitely unless it is faded out. As the literature to follow describes, this repetition has an effect on player experiences by impacting immersion, engagement and performance.

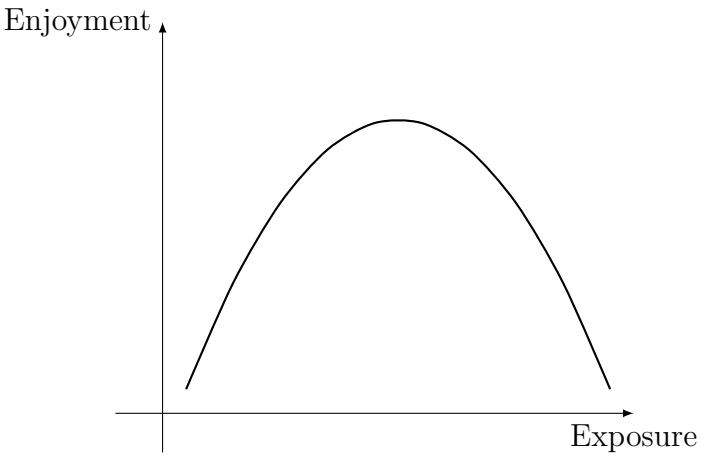

Figure 2.6: An inverted-U graph, visualising the relationship between exposure to audio, and corresponding enjoyment of the same stimuli, after Berlyne (1971).

2.2.1 Listener Fatigue

Long before research into video game audio began, psychological research by Daniel Berlyne defined listener fatigue as **desensitisation to and lack of enjoyment stemming from over exposure to audio** (Berlyne, 1971). Berlyne further hypothesised that the relationship between repeated exposure and enjoyment would be that of an inverted-U shape curve (Berlyne, 1971). The general premise of this theory is that prolonged exposure to repeated audio stimuli will, for a while, increase enjoyment, but will eventually lead to disdain or apathy.

Support for the proposed generality of the inverted-U curve has been inconclusive (Orr and Ohlsson, 2001). For example, supporting evidence for this hypothesis is demonstrated in Crozier (1974), Hargreaves (1984), and Vitz (1966). By contrast, this hypothesis is contradicted by Smith and Melara (1990), Russell (1982) and Orr and Ohlsson (2001), all of which found that the complexity of the music can affect the relationship between exposure

and enjoyment. Furthermore, other factors can affect the generalisability of the inverted-U. Such factors include but are not limited to: genre (Orr and Ohlsson, 2001); major versus minor tonality in focused listening situations (Schellenberg et al., 2008); and hearing loss in participants (Schneider et al., 2019).

In particular, a factor to consider in this thesis is the effects of musical complexity on enjoyment and the arousal potential of stimuli. Berlyne (1970) found that homogeneous sequences declined in pleasantness ratings, more than sequences which involved several stimuli interspersed. Furthermore, Berlyne (1970) found that simpler stimuli became less enjoyable as they became more familiar, whereas more complex stimuli decreased in enjoyment less than simpler stimuli, or in some cases increased in perceived enjoyment. Berlyne (1970) further describes this as the Wundt curve, which denotes the relationship between exposure, hedonic value and arousal potential depending on the simplicity/complexity of stimuli, and familiarity/novelty of the stimuli. The importance of this will become more apparent in Section 2.2.2.

However, before moving on to the impact of such listener fatigue on video game audio, it is necessary to further explain Wundt's curve. The Wundt Curve was originally proposed by the general works of Wilhelm Wundt, but popularised and applied by Berlyne. This curve is similar to the original inverted U, but builds upon it, and Berlyne (1970) states that too little or too much stimulation can lead to boredom or anxiety, while a moderate amount leads to the most pleasure/engagement. Within the context of music and repetition, stimulation can be considered as changes in the music, so too few changes can lead to boredom, or too frequent changes can lead to anxiety

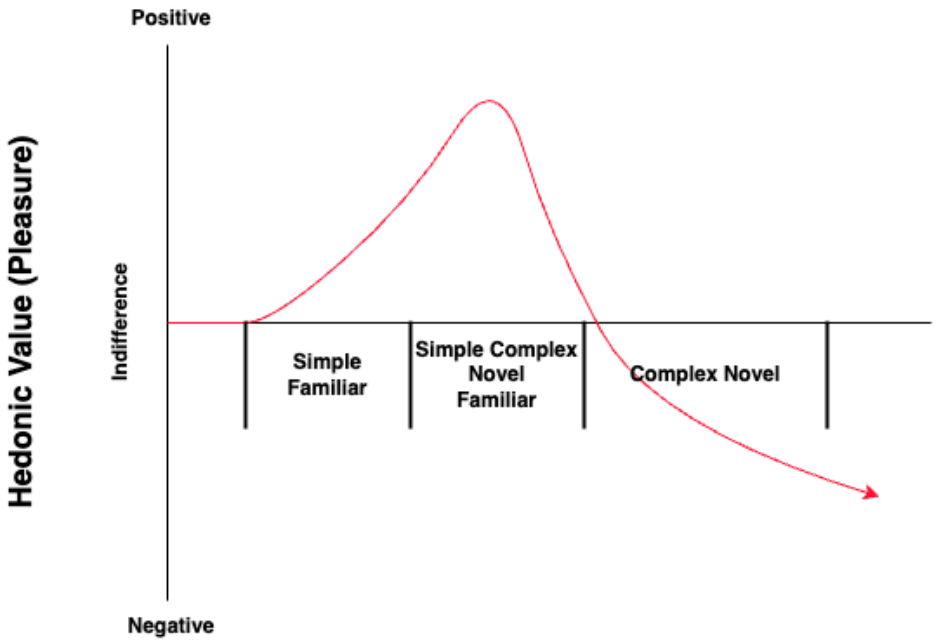

Figure 2.7: Conceptual model of hedonic response as a function of stimulus complexity and familiarity, showing optimal pleasure at moderate levels of novelty and complexity (after Berlyne (1970)).

as the listener is never allowed to settle into the music and become familiar. Figure 2.2.1 depicts the Wundt Curve in Berlyne (1970).

These conceptual models lay the foundation for how repetition is understood to lead to damaging the listener's experience. Now to relate this phenomenon to the non-linearity (or complexity and novelty) of gameplay and how non-linearity compounds with listener fatigue to create the larger problem within the field of video game audio.

2.2.2 Contextualising Listener Fatigue in Video Games

Historically, video game audio has long been considered an afterthought during development, demonstrated by the limited resources ascribed to the storage of audio assets (Marks, 2008, p. 3; Wilde, 2014, p.1) and the streaming/processing of audio in real-time (Plut and Pasquier, 2020, p. 11; Worrall and Collins, 2023, p. 2). With these limited resources, an amount of repetition was generally *expected and accepted* by players, especially in early video games (Worrall, 2024). Over time, as these technologies became more familiar, player expectations have evolved, generally becoming more critical of the video games being played. Subsequently, players became more critical of the music they heard, also.

Repetition of music can sometimes be seen positively for evoking feelings of nostalgia in players (Dutta, 2020, p.93), supporting or establishing themes (Armstrong, 2021), and making new segments even more striking (Summers, 2019, p.47). Repetition can be specifically important when considering melody and themes, as the importance of leitmotif and themes in video games cannot be overstated, with whole chapters dedicated to this topic (Phillips, 2014, pp.55-74). With themes in particular, it is accepted that they "become memorable through meaningfully targeted repetition (Thomas, 2016, p.36). Beyond composers, even video game developers have stated the importance of this repetition at leading industry events (Iwamoto, 2017).

However, over-repetition of audio assets is an aspect of video game development that receives well-documented criticism for its negative impact (Vachon, 2009; O'Donnell, 2002; Forss, 2016; Plut and Pasquier, 2020, p.13; Wilde, 2014, p.14; Worrall and Collins, 2023, p.1). In particular, audio di-

rector and composer Martin O'Donnell states in (Thomas, 2016, p.96) that the video game composer equivalent of the Hippocratic Oath is to "do no annoying," because the overstating of themes or leitmotifs will "always, inevitably and inescapably, lead to annoyance. And that just makes people turn off your music."

This is unsurprising, aligning with the ideas proposed in listener fatigue literature (Berlyne, 1970, 1971). Furthermore, given the non-linear nature of video games – where player action dictates the state of the game, and as a consequence the music – video games are capable of being played endlessly, or at least the music in many situations may loop endlessly. This means that there is a higher than normal risk of over-exposure and listener fatigue than in other media such as film (Worrall, 2024).

This idea of listener fatigue is not only applicable to sound within a video game, but visuals as well. Repetitive visuals – or the overuse of visual assets – in video games receive similar complaints to repeated audio (Greuter and Nash, 2014). This is especially true when limited visual resources are combined with procedural content generation algorithms to output "bland" level designs (Compton, 2016).

Procedural content generation is a term used to describe a variety of computational techniques that can be used to generate visual and mechanical content in games, such as Perlin Noise or stochastic randomness. Procedural content generation is often employed in video games in order to generate non-main story content and extend playtime artificially, without requiring the extra work that bespoke levels require from the developers. This is notable in role-playing games, such as *The Elder Scrolls: Daggerfall* (Bethesda

Softworks, 1996), which was heavily criticised for its overuse of procedural content generation, as were the many extra and empty worlds in *Mass Effect* (BioWare, 2007) and the chalice dungeons of *Bloodborne* (FromSoftware, 2015). These notable criticisms support academic literature surrounding repetition fatigue, in a way that is parallel to how the video game industry views audio repetition. While procedural content generation is a widely adopted solution within video games, procedural audio and procedural music are not widely adopted, and this idea will be explored further later in this chapter.

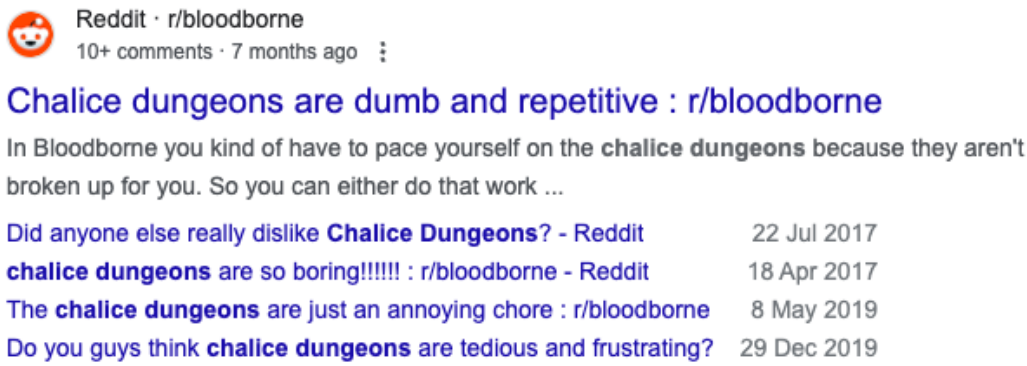

Figure 2.8: A selection of Reddit threads showing the consensus of criticisms for Chalice Dungeons, a procedurally generated aspect of *Bloodborne*.

As video games as a medium have continued to evolve, newer methods have been employed to reduce repetition of sound and music. These approaches include, but are not limited to: the use of smaller MIDI data files instead of audio files to offer larger adaptive soundtracks (Collins, 2008, p.11) as in *Final Fantasy VII* (SquareSoft, 1997); the rearrangement of existing audio files for more variation with a reduced data footprint (Hughes, 2020; Weir, 2017) as seen in *Hades* (Supergiant Games, 2020), *No Man's Sky* (Hello Games, 2016) and *Red Dead Redemption* (Rockstar San Diego, 2010); the

employment of Nyquist's Theorem[7] to limit the sample rate of audio files to double their maximum frequency, ensuring files are smaller in size, but accurately reconstruct the original sound data (Stevens and Raybould, 2015); the generation of new musical content in real-time (Lamperski, 2015; Abel, 2021; Cutajar, 2020; Prechtl, 2016) as seen in *Rise of the Tomb Raider* and *Marvel's Avengers* (Crystal Dynamics, 2020), *Genesis Noir* (Feral Cat Den, 2021) and the procedural synthesis of sounds with minimal or no input data for infinite variations (Farnell, 2009; Su and Joslin, 2019; Barahona-Ríos and Pauletto, 2020; Menexopoulos et al., 2023) as seen in *Spore* (Maxis, 2008) and *No Man's Sky*. While these methods alleviate the problem of repetition-based fatigue to a degree, they further highlight how much more critical audiences are becoming, as the sub-par fidelity of some of these methods (i.e. procedural audio synthesis) is critiqued for not meeting users' expectations (Böttcher, 2013).

The requirement for variation in video game music is seen as so important that many authors discuss methods for increasing the "shelf life" of their music (Collins, 2008; Sweet, 2015; Thomas, 2016; Marks, 2008). Often the ideas discussed encompass similar methods such as variation in: tempo; pitch; rhythm/meter; volume/dynamics; timbres/digital signal processing; melodies, harmony, mixing and musical structure (Collins, 2008, pp.147–165; Sweet, 2015, pp.44–49).

Some of the techniques outlined above are no longer used due to being

[7]The Nyquist-Shannon sampling theorem states that a band-limited signal can be perfectly reconstructed if it is sampled at a rate greater than twice its highest frequency component. Higher sample rates create more data points within the data structure containing the audio data, making the files larger. However, for low frequency sounds, sample rates can be lowered as there is no frequency data contained in the higher samples that will be lost.

outdated (such as raw MIDI scores), nor have some ever received notable widespread adoption (such as procedural sound design). However, music design and implementation methods have standardised and received widespread use across most contemporary video game titles as a method of alleviating repetition as video games get longer and longer (Worrall, 2024). As outlined earlier in the chapter, there are different naming conventions for music design techniques revolving around the term procedural. However, in this thesis, I use a combination of terms defined in Stevens & Raybould's *Game Audio Implementation* (2015) and Plut and Pasquier (2020). To recapitulate these ideas, these terms are typically defined as: *parallel*, where different stems are mixed in real-time to better match changes in the game; *transitional*, where different music tracks are transitioned between to better match game states, for example battle to exploration; and *ornamental*, where musical stingers (or fragments) play over the top of a soundscape/background piece. However, rather than adopt the definition present in Stevens and Raybould (2015) for the *algorithmic* (or procedural) approach, I will rely on the three sub-categories provided in Plut and Pasquier (2020). This definition, however, as will be discussed further in chapter 5, is not necessarily a concrete definition, and within the video game industry it is widely misunderstood (Worrall and Collins, 2023).

Regardless of terminology, the above approaches are often combined to add variation by most music teams and composers. Additionally, these techniques have managed to alleviate repetition-based listener fatigue to an extent. Unfortunately, there is a limit to what can be achieved with finite pre-composed assets, even when combined with these approaches, especially

in role-playing video games that are supported by live service models. A live service video game continues to be delivered on a continuous revenue model, where new content is added at regular or semi-regular intervals. Examples of role-playing video games that feature live service models include: *Path of Exile* (Grinding Gear Games, 2013); *The Elder Scrolls Online* (ZeniMax Online Studios, 2014) and *World of Warcraft* (Blizzard Entertainment, 2004). Many other genres fall into the live service category; many live service video games involve role-playing game mechanics, as these are highly conducive to prolonged play. With this in mind, listener fatigue is a growing problem in role-playing and live service video game development, as between 65% to 95% of studios have stated an interest in moving towards live service models, or are already working within this model (Rousseau, 2024; Muhammad, 2024).

While all genres of video games are in some way non-linear, and the move towards live service models and the repetition of audio are problems across most larger video games, there are certain genres in which this is a greater problem. For example, battle music in Japanese role-playing games (JRPGs). The grinding (purposefully over repetitive) nature of JRPG gameplay and the limited range of musical pieces to accompany battle scenarios lead to higher-than-average exposure to these tracks (Armstrong, 2021). Furthermore, while modern JRPGs/RPGs utilise contemporary music design principles, this genre, when compared to others, takes longer to complete, which exacerbates the issue of over exposure – see Table 1.1 (Gibbons and Reale, 2020, p.1). This higher-than-average exposure can lead to listener fatigue. This is especially true as JRPGs and RPGs have a focus on melodic and leitmotif-laden composition, which is often considered closer to

art-music (Gibbons and Reale, 2020, p.1; Summers, 2019, p.46). In thematically heavy musical material, repetition can be much more noticeable than in horror video games, for example, where music might follow the traditions of musique concréte being constructed of timbres and textures rather than leitmotif (Simard-Poirier, 2017). This can be a problem, as while some timbres are easy to detect, a melody is easier to identify, even when being recontextualised across instruments.

2.2.3 Procedural Design as a Potential Solution

Procedural Content Generation

Procedural Content Generation (PCG) is a well-documented area of research, where a variety of computational approaches are applied to the generation of visual and mechanical content in video games. Examples of these approaches include: constructive PCG (which uses rules-based systems); PCG via machine learning (Summerville et al., 2018); and search-based PCG (those using stochastic search-based algorithms) (Togelius et al., 2010). Within the industry, PCG has been leveraged to generate worlds (McKendrick, 2017; Carrier, 2018), geometry (Sander van Rossen, 2020), world history (Grinblat, 2018), and determine resource/enemy placement throughout generative levels (Bucklew and Grinblat, 2019). In particular, PCG is a very popular technique for independent video game studios, where PCG is a "technical strategy for generating content despite limited production resources" (D'Errico, 2015, p.197). PCG has also seen a wide range of applications at triple-A (AAA) companies, such as the generation of dungeons in *Bloodborne*, infinite new weapons in *Borderlands 3* (Gearbox Software, 2019) and

the world of *Far Cry 5* (Ubisoft Montreal, 2018).

In these examples, PCG has been used at AAA to develop video games faster and with reduced costs (Amato, 2017; Barriga, 2019; Carrier, 2018) while preserving quality (Shaker et al., 2016; Watkins, 2016; Amato, 2017). PCG's application has also played a pivotal role in creating unending experiences for players and supporting live-service games (LSG) (Wilson, 2019). While PCG is often critiqued for the lack of diversity and polish in generated output, where generative content is compared to 10,000 bowls of oatmeal (Compton, 2016), PCG has advantages in a creative aspect, and the use of such techniques in supporting live service games and role-playing games is obvious. Within the industry and academia, PCG is a clear benefit that can help to reduce repetition by adding variety to levels, items, and worlds to assist in alleviating repetition, so long as it is used carefully. Now that the benefits of PCG have been outlined, I can outline how procedural music could play a similar role.

Procedural Music Generation in Video Games

The impact and benefits of Procedural Music Generation (PMG)[8] are similar to those of visual/mechanical PCG, allowing video game developers to computationally generate, perform or transform music in real-time in video games (Wooller et al., 2005; Plut and Pasquier, 2020). With a notably small data allowance for audio in video games (Marks, 2008; Wilde, 2014; Stevens and Raybould, 2015), and the increasing need for variation (Plut and Pasquier, 2020; Worrall, 2021) and interactivity (Phillips, 2014; Stevens and Raybould, 2015), there is justification for the adoption of PMG in the video game industry.

Some successful instances of the use of PMG in video games utilise approaches that do not require composers to change their workflow, or participate in technical work beyond what is expected of them already (Brown, 2019; Lamperski, 2015; Weir, 2017). As an example of successful symbolic music generation in industry, the Dynamic Percussion System is used in *Rise of the Tomb Raider* and utilises machine learning to generate drum patterns in real-time during gameplay, where the composer only needs to give the developers their percussion in symbolic form (MIDI files). Weir (2017) on the other hand creates an audio domain music generation tool (PULSE) for *No Man's Sky*, where they specifically told the band providing the music (65daysofstatic) to compose an album as they typically would, but to de-

[8]It may seem peculiar that music and audio both have their own procedural terminology, compared to broader areas of video game development such as visuals and mechanics, which fall under the umbrella term of PCG. However, literature around procedural music and procedural audio use distinct, non-overlapping terminology specific to music and audio, which encapsulates very different forms of behaviour and interaction (Wooller et al., 2005) and (Farnell, 2009) respectively. By comparison, PCG generally covers the generation of content on instantiation of game objects such as levels, enemies or loot, and is a broader research area where interactions more closely resemble each other.

liver stems that could be manipulated to generate music in-game, without impacting the band's creative process. In another example where the video game composer was not involved, developers leveraged an older open-source tool for jazz solo generation known as Impro-visor (Keller et al., 2007) to overcome the lack of access to machine learning specialists for music in indie video game studios (Abel, 2021). While Plut & Pasquier outline a more comprehensive list of PMG systems in video games, it is clear that when compared to PCG, PMG has seen much less application in the video game industry, giving the impression of less acceptance (Plut and Pasquier, 2020).

Within academia, music generation for video games as a research topic has seen more interest in recent years than it has in industry, often focusing on rules-based approaches or less computationally taxing approaches such as Markov chains to decrease latency and computational costs in real-time (Engels et al., 2015; Shapiro and Huber, 2021). However, there is a disconnect between industry and academia, so while research has been tackling game-specific use cases such as transition generation (Cutajar, 2020), generating music that adapts to gameplay (Hutchings and McCormack, 2020), using gaussian mixtures to control melodic shape in generative music (Xiang and Guo, 2021) or generating music to match nonplayer character relationships (Washburn and Khosmood, 2020), procedural music systems that generate new material go largely ignored in industry. By comparison, the more popular use of procedural systems in video games is that of the transformational approach (Plut and Pasquier, 2020; Wooller et al., 2005), that uses simple rules-based approaches or Finite State Machines (FSM) to transition between human-authored tracks (Hughes, 2020; Guy Somberg, 2018; Rogier, 2022).

Now, I have introduced the context space (video games) and the problem (music repetition) and the potential impact of procedural music solutions on the problem; before I can fully discuss the existing procedural music approaches, I must introduce music generation beyond video games, how these solutions work, and why they may or may not be applicable to video game music.

2.3 Music Generation

Music Information Retrieval (MIR) is the field of research where computations/algorithms are used to analyse and retrieve information about music (Kaminskas and Ricci, 2012). This area of research has many sub-areas that encompass much more than just the analysis of music, including but not limited to: expressive rendering (or music performance modeling) using traditional machine learning (Widmer, 2002), graph neural networks (Jeong et al., 2019b), feedforward neural networks (Cancino-Chacón, 2018) and multi-variate Markov chain models (Stables et al., 2014); music generation using rules-based, machine learning or deep learning approaches (Collins and Laney, 2017; Huang et al., 2018; Huang and Yang, 2020); and style transfer (Malik and Ek, 2017). However, before discussing how computers and artificial intelligence have been leveraged to generate music in more detail, it is worth exploring their origins.

2.3.1 A History of Music Generation

Ideas that later informed generative and algorithmic approaches to music, such as formal rules, systems for organising pitch and rhythm, and mathematical relationships, can be traced to ancient Greek music theory, including the work attributed to Pythagoras around 500 BC (Nierhaus, 2009; Burkholder et al., 2019; Collins, 2018). While these practices would not be described as generative music in a modern sense, they established early traditions of rules-based musical thinking. Related ideas continued to appear in later Western European classical traditions, including systems such as *Musikalisches Wurfelspiel* (music dice games) in the eighteenth and nineteenth centuries, associated with composers such as Mozart (Alpern, 1995; Hedges, 1978). These systems relied on chance operations constrained by stylistic rules, which produced musical outcomes through the recombination of pre-composed material.

Although random or chance-based compositional techniques predate digital computation, the emergence of computers enabled such processes to be automated, scaled, and formalised in new and interesting ways. In the twentieth century, composers such as John Cage and Marcel Duchamp played a role in popularising indeterminacy and chance as compositional strategies (Schwartz and Godfrey, 1993). Notably, Duchamp's *Erratum Musical* (1913) represents an early exploration of avant-garde practice in using rules-based randomness to draw notes from a hat (Collins, 2018), while Cage's later work formalised indeterminacy as an aesthetic and philosophical position rather than a purely technical one. Importantly, many of these approaches retained a significant degree of human agency, with composers actively shaping sys-

tems, constraints, and interpretations rather than delegating creative control entirely to autonomous processes. This distinction parallels later discussions in computational creativity regarding the difference between systems that merely execute predefined procedures and those capable of more exploratory or unpredictable behaviour (Boden and Edmonds, 2009). While subtle, this distinction becomes increasingly relevant in later work on music generation within interactive and computational contexts, including video games.

Discussions of computer-generated music are often traced back to the nineteenth century, particularly to Charles Babbage's design of the Analytical Engine in 1837 (Nierhaus, 2009). Although the machine itself was never realised and was not intended for musical purposes, Ada Lovelace famously speculated that such a device might one day compose "elaborate and scientific pieces of music of any degree of complexity" (Boden, 2018, p.6). Rather than constituting a concrete proposal for music generation, Lovelace's comments are best understood as articulating an early philosophical vision of computers as general-purpose symbol-manipulating machines. Lovelace's vision was later interpreted as anticipating the possibility of computer-generated music by commentators (Alpern, 1995; Nierhaus, 2009).

Practical examples of computer-assisted music composition began to emerge in the mid-twentieth century. One of the earliest and most frequently cited examples is the *Illiac Suite* (1957), composed by Lejaren Hiller and Leonard Isaacson using rule-based and probabilistic programming, with the resulting output realised as a string quartet score (Maurer, 1999; Collins, 2018). Around the same period, Iannis Xenakis employed stochastic processes and probability theory as formal compositional tools, using computers to support

the generation of large-scale musical structures. These developments formed part of a broader mid-century movement toward algorithmic composition, which also included the work of Gottfried Michael Koenig, whose *Project 1* and *Project 2* systems explicitly formalised composition as an automated, rules-based process, as well as research by figures such as Harry F. Olson, who explored probabilistic and information-theoretic approaches to musical structure in the 1950s (Nierhaus, 2009, p.71). During this period, Markov chains were often employed for style imitation; however, composers such as Hiller and Isaacson, and Xenakis, also used probabilistic models as primary compositional mechanisms in works such as *Experiment 4* and *Analogique A*, respectively (Nierhaus, 2009). Related experiments, including John R. Pierce's use of human "computers" to realise Markov-based music generation, further demonstrate that statistical and probabilistic models were already being applied to music during this period. These approaches are now commonly recognised as foundational techniques within early artificial intelligence and symbolic computational creativity research (Collins, 2018; Boden, 2018). Collectively, these works represent some of the earliest sustained explorations of algorithmic and computational music generation, establishing conceptual and technical foundations that later AI-based approaches would build upon.

Computational research into music generation outside the context of video games is commonly discussed under the broad and sometimes overlapping labels of algorithmic composition and generative music. These terms are not used consistently across the literature and may refer to different degrees of autonomy, formalisation, and creative intent. In recent years, however, ad-

vances in machine learning and deep learning have had a substantial impact on music generation research. Recent systematic reviews indicate a significant increase in research activity in this area, with deep learning for audio emerging as a particularly fast-growing domain (Civit et al., 2022; Briot and Pachet, 2020). This trend is reflected both in academic research and in industry, with large technology companies such as Alphabet (Google), Spotify, TikTok, Meta, and Apple maintaining dedicated music generation research teams (Briot and Pachet, 2020), alongside smaller AI-driven music start-ups such as Infinite Album[9], and AIVA.[10]

These organisations have produced a range of music information retrieval (MIR) and music generation tools, including systems that generate rhythmic patterns, extend melodic passages conditioned on existing material, and assist with tasks such as mixing and mastering within digital audio workstations (Roberts et al., 2019; Tchemeube et al., 2023). More recent examples include piano in-painting models capable of filling gaps between musical phrases based on surrounding context to create coherent transitions (Hadjeres and Crestel, 2021), some of which have been integrated into commercial music production software.

[9] As of the time of finishing this thesis, Infinite Album has now been acquired by Xsolla, a video game development company.

[10] https://www.aiva.ai/

2.3.2 Terminology and Conceptual Scope of AI Music Systems

Having introduced several overlapping approaches to music generation and procedural music thus far, the following section clarifies the terminology used throughout this thesis and situates these terms within their respective research traditions. This clarification is necessary due to the inconsistent and often interchangeable use of terms such as algorithmic composition, generative music, procedural music, computational music, and AI music systems across both academic literature and applied domains, particularly within game audio and music technology research. The following categories are not mutually exclusive and are used here as analytical lenses rather than strict ontological divisions.

Historically, the term *algorithmic composition* has been used to describe systems in which musical material is produced through explicitly defined rules, procedures, or formalised processes, as reflected in many of the early examples discussed in the previous section. In this context, algorithms are typically authored by human composers, and musical outcomes are constrained by symbolic rules, probabilistic models, or deterministic procedures (Collins, 2018). These works are often retrospectively classified under the broader field of *computer music* which encompasses areas such as synthesis, live electronics, composition systems and more (Roads, 1996). Closely related to these is the term *generative music* coined by Brian Eno, which is often used more broadly to describe systems capable of producing musical output with varying degrees of autonomy (Eno, 1996). While generative music is sometimes treated as synonymous with algorithmic composition, it is

also frequently used to emphasise ongoing or open-ended musical processes rather than the generation of discrete compositions. As illustrated by historical and contemporary examples, these distinctions are not consistently maintained across the literature, and the rise of generative AI research gives the term *generative music* new interpretive scope through association.

The term *procedural music* is most commonly associated with video game music, where music is generated, arranged, performed, or adapted in real-time in response to player actions or game state changes (Wooller et al., 2005; Plut and Pasquier, 2020). Procedural music systems may employ pre-authored musical fragments or generate new musical content, and historically (as discussed in a previous section) utilise rule-based recombination (Hughes, 2020), stochastic processes (Langston, 1986), or machine learning models in newer contemporary practice (Brown, 2019). Procedural music systems are often characterised by their functional role within interactive systems rather than by a specific underlying procedural technique. As such, procedural music can be understood as an application context rather than a distinct compositional paradigm, and may encompass algorithmic, generative, transformational, or data-driven approaches depending on implementation.

Computational musicology developed as a parallel research area focusing on the computational analysis of music (Meredith, 2016). This term is used more broadly to refer to the application of computational methods to musical problems, including but not limited to composition, analysis, performance, synthesis, and music information retrieval (Meredith, 2016). This term does not necessarily imply generative intent and may encompass systems designed to analyse existing musical material, assist human composers,

or support technical tasks such as transcription, mixing, or mastering. In this sense, computational music provides a conceptual umbrella under which both generative and non-generative systems may be situated, similar to the use of *Music Information Retrieval* (MIR), which emerged as a more standardised term for this research area.

More recently, advances in machine learning, and particularly deep learning, have introduced new approaches to music generation and music-related tasks that differ from earlier systems, including rules-based and evolutionary approaches such as genetic algorithms. *Machine learning–based music systems* typically operate by learning statistical regularities from data rather than relying on explicitly encoded musical rules. These systems may be applied to a wide range of tasks, including symbolic music generation, audio synthesis, expressive performance modelling, style imitation, music analysis, and production assistance. As discussed later in this thesis, such systems raise distinct conceptual and practical questions regarding authorship, control, interpretability, and creative agency.

Throughout this thesis, the terms *AI music systems* and *AI music tools* are therefore used as umbrella terms to refer to systems that autonomously or semi-autonomously generate, transform, analyse, or support musical material using computational techniques. This includes both generative systems, such as those that produce new musical content, and non-generative systems, such as those designed for analysis, production support, or expressive rendering. Where finer distinctions between algorithmic, procedural, generative, and machine learning–based approaches are analytically relevant, they are explicitly stated. Otherwise, terminology is used pragmatically to reflect

common usage within the relevant research or applied context, particularly within video game music and interactive audio systems.

Given the multi-disciplinary work being presented in this thesis, before examining the specific approaches to music generation and expressive rendering, it is necessary to clarify key terminology, including machine learning, deep learning, neural networks, and conditional generation. The following subsection introduces these concepts to establish a shared foundation for the discussion that follows for non-computer scientist readers.

2.3.3 Fundamentals of Deep Learning

Artificial Intelligence

While the term artificial intelligence was coined by John McCarthy in 1956, the conceptual origins of artificial intelligence draw on earlier work in logic, automata, and cybernetics. In this thesis, when discussing AI, I am referring to the definition from Geraint Wiggins, which is an algorithm which tackles the 'performance of tasks, which if performed by a human, would be deemed to require intelligence' (2006, p.450). However, Wiggins suggests that the origin of the term may be Alan Turing, which is unsurprising given how Turing's (1950) work is generally considered to have given rise to the field of AI, and alludes to the performance of tasks that would seem to require intelligence, by computers, as per Wiggin's definition. While this definition is seemingly broad in its interpretation, this is a strength, as it can be applied to many examples. In video games, for instance, AI is used to control path-finding for non-player agents within a game (often relying on basic algorithms such as the A* path-finding algorithm). In this example, a human player

uses visuals and sounds emitted from the video game to determine where a location of an opponent might be and then uses controls to move towards it. For many players, especially those experienced in playing a variety of video games, that task seems almost intuitive, but it does in fact require contextual knowledge and intelligence. Alternatively, an AI agent could take the following steps to recreate this intelligent behaviour:

1. Determine if noise or visuals occur from the target to let the agent know that they are within a predefined range.

2. Determine the vector of the player in the game space.

3. Determine the vector of the target that was heard.

4. Calculate the distance and direction between the two vectors.

5. Increment their own vector to move toward the target.

6. Repeat as necessary, each frame of the game, to ensure direction is accurate if the target is moving.

7. If the distance between the target and the agent is less than their weapon's range, they could stop their pursuit and fire at the enemy.

Often these AI agents, and many more basic AI for video games and music, are rules-based, search-based, or rely on state-driven machines following a set of pre-determined rules, with elements of stochastic randomness. Within MIR, tasks such as beat tracking and pitch-related processing have been the subject of extensive research (Klapuri, 1999; Müller, 2015). While early systems explored rule-based and heuristic approaches, these methods were

limited in their ability to cope with expressive timing, polyphony, and signal variability. These challenges motivated the development of more sophisticated probabilistic, statistical, and learning-based techniques (Böck et al., 2016).

Machine Learning

Machine learning is a subset of artificial intelligence and is generally defined as the study of giving computers the ability to learn – from experience – without explicitly being programmed (Samuel, 2000). Stuart Russel & Peter Norvig (2020) expand on this definition in their book *Artificial Intelligence: A Modern Approach*, stating that machine learning algorithms are built to adapt to new circumstances and to detect and extrapolate patterns within the data. While hard-coded AI is very reliable, the introduction of machine learning technology reduces production bottlenecks, where agents can be coded once, but then exposed to different datasets to learn rules and be reused across different situations. A game-related example for this is that a single agent could learn the rules for Checkers, Chess, Shogi and Go.

Machine learning algorithms are generally described as being of three main types, and also performing three main jobs. The main jobs of machine learning algorithms are:

- Descriptive, where the system or algorithm uses data to explain what is happening;

- Predictive, where the system or algorithm uses data to predict what will happen next;

- Prescriptive, where the system or algorithm will use data to make suggestions about what action to take.

To relate this back to video games, a descriptive machine learning algorithm could be used to watch an e-sports (electronic sports) competition such as *League of Legends* (Riot Games, 2009) and to relay information to the broadcasters or audiences such as which players are alive, what the likelihood of victory for each team is (Kokkinakis et al., 2020). In this sense, it is describing the in-game situation. A predictive machine learning algorithm may use the same information described above to predict what happens next (Ringer et al., 2023). This is not locked in temporally until the next minute, but you could use the results of previous matches to predict who wins the next match. A prescriptive algorithm might take this same data, and then in a coaching situation, it could suggest during training the best course of action to a professional player. To relate these back to music, a descriptive algorithm could use music data to describe tempo or key changes present in a recording. A predictive machine learning algorithm could predict in what chords would play underneath a melody, or vice versa. A prescriptive machine learning algorithm could be used to analyse a musical mix and to recommend to the engineer how to better process the audio files to suit a particular style of mixing.

With regards to the three types of machine learning that occur most popularly, these are:

- Supervised learning, where algorithms are trained with a labelled dataset, which allows them to learn and grow more accurate over time;
- Unsupervised learning, where algorithms are trained with an unlabelled dataset, and it looks for patterns without guidance as to what it is looking for;
- Reinforcement learning, where a model learns through trial and error and takes the best actions with each iteration to reach a goal, where the best action is determined by a reward system that penalises failure and rewards incremental success.

To relate this to video games, a supervised machine learning algorithm may be presented with a variety of animations that are tagged for a specific emotion, and learn to suggest these animations based on the emotional tag for corresponding dialogue (Bosque, 2022). By comparison, an unsupervised machine learning algorithm could be shown thousands of sequences of moves, and then learn the learn reoccurring patterns without requiring tags. Reinforcement machine learning algorithms are quite popular in video games, as they can be used to train agents to play a video game, which can be invaluable for testing and creating realistic non-human agents for players to play against.

For musical examples, a supervised machine learning algorithm could examine audio files labelled by genre and use this information to predict the genre of a new piece. An unsupervised algorithm could analyse patterns in

sequences of musical notes to learn underlying structure or stylistic characteristics, without being given the next note explicitly. A reinforcement learning algorithm could be trained to "jam" alongside a human musician in real time, by practising along with recordings and receiving rewards or penalties based on its note choices

Statistical Models

Before continuing on to deep learning and neural networks, I will briefly introduce statistical models, as they also represent the transition from rules-based AI approaches to more data-driven approaches. Statistical models are mathematical frameworks that can be used to represent and analyse data by capturing the relationships between variables. These models rely on mathematical foundations and make probabilistic assumptions based on what they perceive in the data they are given. Similar to machine learning, these algorithms can be used for descriptive, predictive and prescriptive tasks. One statistical model that is employed widely in procedural music generation is Markov Chains, which are models that describe the likelihood of transitions between states within a finite state space. Markov chains have seen application within: music generation for video games due to their lightweight computation (Prechtl, 2016; Cutajar, 2020; de Oliveira and Tavares, 2018; Ramanto and Maulidevi, 2017); and within symbolic music generation (i.e. the generation of musical scores) (Shapiro and Huber, 2021; Highams and Olszewska, 2023) in particular when supported by pattern recognition algorithms (Collins and Laney, 2017). These pattern recognition supported variants of Markov models have been shown to compete well with much

more complex, deep learning models such as transformers (Yin et al., 2022). Markov-based models are used and will be discussed in later chapters.

Regression, Classification and Natural Language Processing

Before proceeding further, there are three further terms to define, which relate to machine learning/statistical models. The first two correspond to the type of predictive analysis being made by the model. These terms are:

- Regression, where a model is focused on predicting a numerical outcome, this could be the chance of something happening, or a new feature for the model.
- Classification, where a model is focused on assigning an input to a pre-existing group or class.

In general, many models will be performing one of these two tasks, especially when dealing with numeric data. For a music example, a statistical linear regression model may predict the dynamic volume of each note in a MIDI sequence (where each velocity value uses 7-bits, and ranges from 0-127 in value). In this example, the features of each note would be used as input, and then a value from the range would be predicted to match a line of best fit on a graph.

For a classification task, a statistical logistic regression algorithm could be used, where instead of a line of best fit giving a numeric value within the range, the line of best fit determines whether the predicted result is a 0, 1 or any number of groups depending on the feature of the targets. For a musical example, you could analyse an audio file for features such as dance-

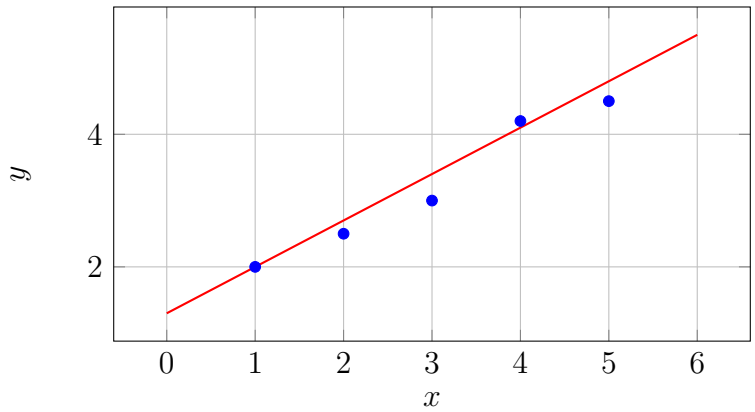

Figure 2.9: An example of Linear Regression on a Scatter Plot. The red line is the line of best fit, and if new data is shown to a model, it will make predictions to fit this line, depending on the variables of the input data (i.e. x), where output y is a continuous value for the predicted feature (i.e. a number within a range).

ability[11], tempo, key, and predict the likelihood of a specific genre. These are explained here, as in later chapters, transformer encoders will be discussed that have classification and regression heads, in order to perform the types of tasks described.

The third type of data and prediction that can be made is Natural Language Processing (NLP). NLP is a field of machine learning in which machines learn to understand natural language as spoken and written by humans (as opposed to numbers or classes – regression and classification).

As I will explain later, some architectures are an advancement of NLP, such as transformers (Yıldırım and Asgari-Chenaghlu, 2021). Transformers will be discussed in more detail later in this literature review, as they underpin many of the popular architectures in contemporary research, and in particular some of the work carried out in this thesis, so introducing NLP is a necessary step. For example, music can be treated as a sequence of

[11]This is a metric used by Spotify in their Music AI API that corresponds to the amount of consistent rhythm within a piece.

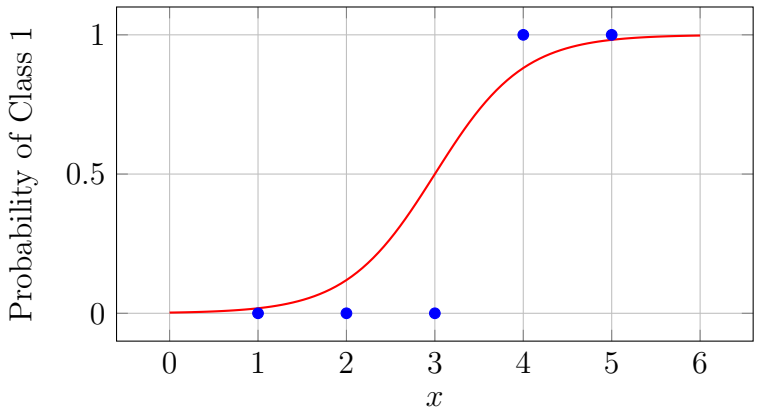

Figure 2.10: An example of Logistic Regression plotted on a Scatter Plot. The red line is the line of best fit, and if new data is shown to a model, it will make predictions to fit this line, depending on the variables of the input data (i.e. x). These predictions y will be a probability representing the classification of the group the data fits (i.e. 0 or 1).

language tokens rather than as numeric points of data. This approach was popularised by the music transformer model (Huang et al., 2018), which is an adaptation of the original transformer architecture (Vaswani et al., 2017). The music transformer uses tokenising, which is the process of converting music data into language tokens that can be understood by the model (Huang et al., 2018). This encoding representation and the newer Revamped MIDI (REMI) representation from the same team (Huang and Yang, 2020) are visualised in Figure 2.11.

As Figure 2.11 depicts, with MIDI-Like encoding, a sequence of tokens begins with a tempo token, and then a note-on token containing the pitch of a note, then a velocity value for the midi note, a timeshift token, then moves linearly along the musical timeline before a note-off token tells the algorithm that a note has ended. This same selection of note-on, velocity, time shift and note-off is used for each note in a piece (or sequence) (Huang

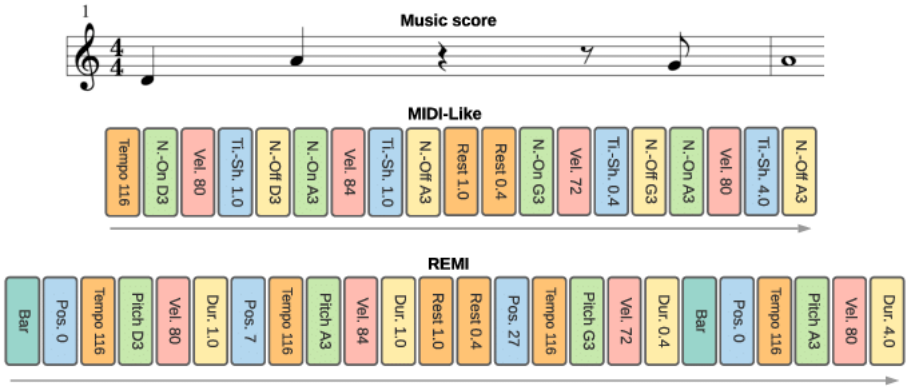

Figure 2.11: An image taken from Fradet et al. (2021) demonstrating how a basic score is tokenised in both the MIDI-Like and REMI representations.

et al., 2018; Fradet et al., 2021). By comparison, REMI encoding provides more information to the model than the MIDI-like representation (Huang and Yang, 2020; Fradet et al., 2021). Figure 2.11 depicts that for a given sequence: a token that denotes the beginning of a bar; a token that represents the current position of a note within the bar (measured in increments of 32nd notes initialised at zero – i.e. beat 2 is position 7); a token representing a notes pitch; a token representing a notes velocity; a token representing the duration of a note (Huang and Yang, 2020; Fradet et al., 2021).

This discussion of tokenising (the encoding of music into language tokens) is premature, but for now it is worth knowing the definition of tokenisation, as well as understanding that it is argued that the more information an encoding can offer, the more likely it is to be successful (Velardo, 2024b; Worrall et al., 2024). This is because the information being provided conditions the output. Conditioning is when an input shapes or guides the generation of content. For example, when looking at music, lyrics can be understood using an NLP approach to get the semantic meaning from the words. These meanings

or emotions can then be used to condition music generation, for example, by suggesting keys or tonalities that are commonly associated with lyrical content, although such mappings are subjective and culturally influenced rather than fixed.

Origins of the Perceptron

Moving to the next layer of the AI hierarchy is deep learning, which is a subset of machine learning. Deep learning generally consists of neural networks that have many layers (which is what makes them deep). However, before deep learning can be properly explained and understood, it is necessary to introduce the perceptron.

The perceptron is an early precursor to the neural network, which was introduced by Frank Rossenblatt in 1958 (Rosenblatt, 1958). A perceptron computes the sum of weighted inputs, adds a bias term, and passes this net input through an activation function (a function that transforms or standardises the value). In early perceptron models, the activation function produced a binary output (0 or 1) (Rosenblatt, 1958). While early perceptrons used binary activation functions, later perceptrons and modern neural networks employ continuous, nonlinear activations (e.g., sigmoid, tanh, ReLU) to enable richer representations and learning (Bishop and Bishop, 2023).

When considering a neural network or deep learning, every node in a neural network resembles a perceptron, where each node receives inputs either from the initial input (if the node is in the first layer) or from the output of the previous layer (in any other layer) (Minsky and Papert, 1969).

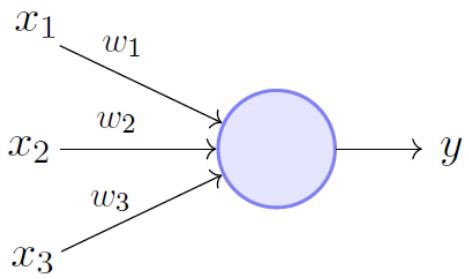

Figure 2.12: An illustration of the perceptron after Minsky and Papert (1969). Inputs X are multiplied by their corresponding weights W, summed together, and combined with a bias term. The resulting value is passed through an activation function to produce the output Y.

Deep Learning and Neural Networks

Neural networks, including deep networks with many layers, are approaches within machine learning in which models learn from data without explicit programming. They are inspired by the structure of the human brain, which contains approximately 86 billion highly interconnected neurons, while artificial networks consist of far fewer nodes organised in layers, with each node capable of processing inputs and producing outputs (Bishop and Bishop, 2023).

Compared with conventional machine learning models, neural networks can capture complex, nonlinear patterns in data, but their performance depends strongly on factors such as the amount and quality of training data, the network architecture, and the number of training epochs. Deep neural networks, with multiple layers, offer additional flexibility and the potential to capture medium- and long-term dependencies. However, they are often considered "black boxes," as interpreting the precise computations leading to their outputs can be challenging. Their versatility allows them to take many forms, each suited to specific tasks, making them a powerful tool in

modern machine learning.

Building on the perceptron examples of the previous section is the multilayer perception (Rosenblatt, 1958). Multilayer perceptrons are a foundational building block of deep learning and are a type of artificial neural network that builds on the individual perceptron by combining at least three layers: one input layer, one hidden layer, and one output layer, where neurons in a layer are connected to all the neurons in the next layer. This type of artificial neural network is a simpler type of architecture, namely feedforward neural networks (Jaiswal, 2025).

Contemporary MIR research leverages many types of deep learning models for a variety of tasks, and I will now outline and define those architectures that have been used popularly across generative music (and expressive rendering – another subset of MIR that I will discuss in Chapter 6) research. The first term I define is **Recurrent Neural Network (RNN)**, which is an adaptation of the more general neural network and is trained to process and work with sequential data (i.e. words, sentences, notes in a song) and time series data (Pi, 2024). The core advancement of the RNN approach over neural network approaches is that it features a feedback loop, which allows the model to pass back information, giving it a temporary short-term memory that can aid in recognising patterns (Pi, 2024).

The second term I define is **Long Short-Term Memory (LSTM)**, an architecture that was invented in order to tackle the limitations of traditional RNNs (Hochreiter and Schmidhuber, 1997). A particular weakness of RNNs is that they suffer from exploding gradients (where, when individual derivatives are large and the final derivative becomes large) and vanishing

gradients (where the number of hidden layers increases to the point that the learning gradient becomes very small – hampering the learning process) (Hamad, 2023). The innovative step between RNNs and LSTMs is the use of memory cells and gating mechanisms, which, through the control of the flow of information, make the capturing of long-term dependencies more effective in sequential data (Hamad, 2023). This architecture, compared to the RNN is able to capture shorter-term dependencies which it can learn from.

Transformers & Self-Attention

The third term I define is **Transformers**, which were introduced by Vaswani et al. (2017). Given the importance of this architecture to the work presented in this thesis, I cover this architecture in more detail than previous architectures, such as RNNs and LSTMs. Transformers are an advancement of NLP research, primarily designed for sequential language data, featuring an encoder-decoder architecture and also utilise a self-attention mechanism (Yıldırım and Asgari-Chenaghlu, 2021). Self-attention allows the model to iterate through every note or token in a given input sequence, and determine the importance of each token in context to the current note (Velardo, 2024a). This self-attention mechanism enables the model to capture longer-term dependencies that models such as RNNs and LSTMs (Yıldırım and Asgari-Chenaghlu, 2021). Additionally, compared to sequential models such as LSTMs, transformers are capable of running notes in parallel which decreases the training bottlenecks (Velardo, 2024a).

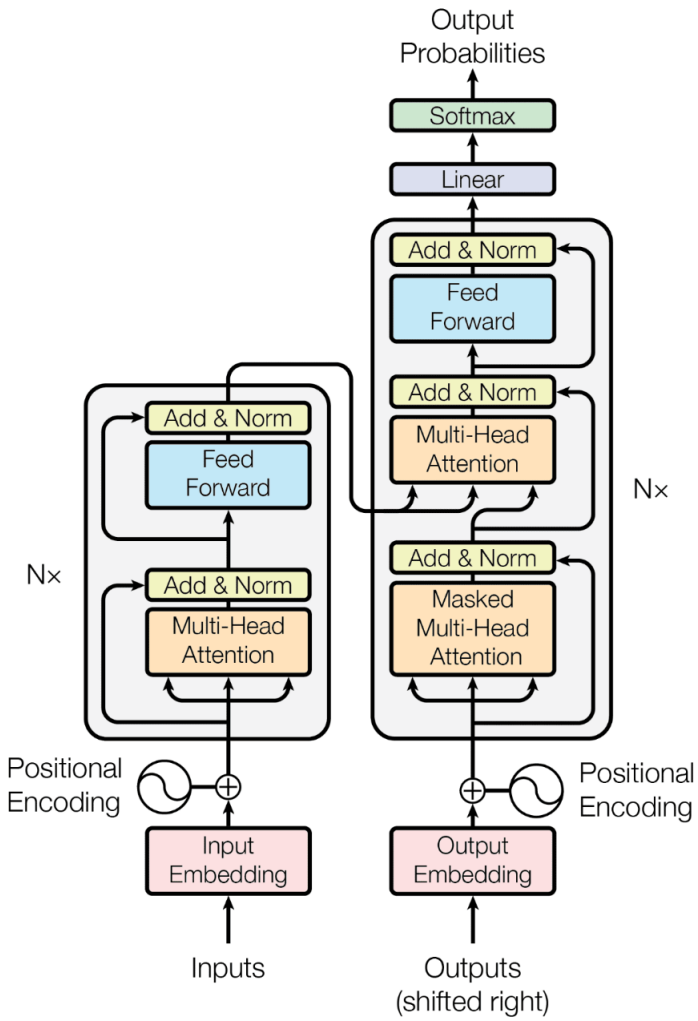

Figure 2.13: An image of the transformer architecture (Vaswani et al., 2017), the left hand portion of the diagram shows the encoder, and the right hand portion the decoder.

The left-hand portion of Fig 2.13 is the encoder, where a tokenised input is taken and used as an input embedding, after which other features are concatenated to provide positional encoding as to the initial input embedding (which does not contain order information). After this, the combined input enters the encoder block, which is a block of n encoder layers, each of which consists of a Multi-Head Attention, a Feed Forward and two Add & Norm blocks. The Multi-Head Attention block runs multiple instances of the attention mechanism, where each head can focus on a different aspect of the input, increasing the computational power and reducing the risk of overfitting (Velardo, 2024a). In the first Add & Norm block, the resulting attention matrices from the Multi-Head Attention block are then combined with the input of the Multi-Head Attention block. This allows the token information for the sequence to have the corresponding attention scores added to them (as the resulting matrices do not include the original tokens). The total information is then normalised so that all features have a mean of 0 and a standard deviation of 1 (Velardo, 2024a). This is passed into the Feed Forward block, which processes each token embedding independently, running them through a ReLu activation function (which is a non-linear transformation of data for increased complexity), which adds nuance to the data. At this point, the encoder layer is repeated n times. The output of this encoder block is a representation which can be decoded into an answer using the decoder portion of the encoder-decoder architecture. However, the representation alone can be passed through a classification head or regression head to perform a classification or regression task, respectively (Velardo, 2024a).

The right-hand portion of Figure 2.13 shows the decoder part of the architecture (Vaswani et al., 2017). If the task the model is being trained for is not a regression or classification task, then a decoder can be trained to perform autoregressive tasks. Autoregressive is a term meaning that a model predicts the future (or future states) based on past values, while many tasks, such as predicting stock price changes, are autoregressive, as are music generation models generally, as they predict each new note autoregressively based on existing notes or a given transition matrix (Velardo, 2024b). With the decoder, an output embedding (the input of the model that you wish to extend, translate or transform) is combined with positional encoding to provide the order of the sequence to each token before being passed into the decoder block. Within the decoder block, there are n decoder layers, each of which includes the following: a Masked Multi-Head Attention block, a Multi-Head Attention block, a Feed Forward block, and three Add & Norm blocks (Vaswani et al., 2017). Most of these blocks perform the same as they do for the encoder, with some key differences.

Firstly, the Masked Multi-Head Attention block prevents each token from attending to future tokens during decoding, addressing the problem of information leakage. In standard self-attention, each token can attend to all other tokens, but in an autoregressive decoder, outputs are generated one token at a time. Allowing the model to access future tokens would give it prior knowledge of the sequence, which can lead to unrealistic predictions. To prevent this, the Masked Multi-Head Attention sets the attention scores of future tokens to *negative infinity*, effectively masking them so that they are not considered during computation.

Another key difference in the decoder is that the Multi-Head Attention block combines the output of the encoder with the normalised output of the Masked Multi-Head Attention block, using them together as input for subsequent computations. The remainder of each decoder layer is identical to the encoder, comprising the Add & Norm blocks and the Feed Forward block. After passing through n decoder layers within a decoder block, the output representation is fed into a Linear layer, producing logits corresponding to the vocabulary size of the input sequence. These logits are then passed through a Softmax function to generate a probability distribution over all tokens, from which either the highest-scoring token or a token exceeding a specified threshold is selected (Velardo, 2024b). The autoregressive model uses these probabilities to generate the sequence one token at a time.

The transformer has been explained in-depth here, as it has applications in many MIR tasks, but most importantly in both the symbolic domain and the audio domain of music generation. This application occurs as both recorded audio and symbolic representations can be preprocessed into tokens, which can be treated as natural language, and the ability of the transformer to capture mid-long range dependencies tends to lead to more cohesive generative music.

Beyond the original transformer architecture, numerous advancements have been made in transformer-based music generation. For example, the pop music transformer uses the REMI (REvamped MIDI-derived events) tokenisation scheme to improve the representation of rhythmic and harmonic structure for expressive pop piano generation (Huang and Yang, 2020). Recent work also explores diffusion-based generative models that condition

on text and semantic features to produce audio outputs, building on ideas from language and image models (Evans et al., 2024). Additionally, large transformer-style music foundation models have been developed that leverage large symbolic datasets and sophisticated tokenisation to improve both generation quality and downstream understanding tasks (Guo and Dixon, 2025). However, as these do not appear as a core part of this thesis, these architectures have not been explained.

2.3.4 Deep Symbolic Music Generation

In recent years, automatic music generation research using machine learning and deep learning has achieved a level of vogue, growing rapidly as a field in the advent of technical progress in architectures, and the increased availability of datasets and cheap computing (Briot and Pachet, 2020; Ji et al., 2020). While many early models used grammar- or rules-based approaches to generate music (Ebcioğlu, 1988; Steedman, 1984; Spangler, 1999), a strength of machine and deep learning approaches for music generation is the generality, where modules can automatically pick up the style of a given corpus without the need to bespoke programming (Briot and Pachet, 2020). This approach is advantageous as machine and deep learning-based approaches are often less fragile than manually coded approaches due to their generality, and the use of such approaches additionally simplifies the process compared to hard-coding analytical approaches (Fiebrink and Caramiaux, 2016). With symbolic music, training data, and the generative output is a symbolic representation of music such as MIDI, sheet music (in musicXML format traditionally), or point sets (a multi-dimensional array of information about notes) (Gao and

Collins, 2023). When working with symbolic domain generation algorithms, often these outputs are handled by a user in a Digital Audio Workstation, as the audio fidelity of the output is not as high as with generative audio (Worrall, 2021; Brown, 2019). As discussed in Section 1.4 (Wilde, 2014; Marks, 2008).

As a machine or deep learning task, symbolic music generation is typically predictive, where features of the next note in a sequence, such as pitch, on-time and duration, are predicted based on the previous notes in a sequence, or a given context (i.e. contemporaneous voices or chords). Contemporary machine and deep learning music research has leveraged: pattern inheritance informed Markov chains which can condition their statistical modelling on repetitive patterns in training data (Collins and Laney, 2017); transformer models that can generate monophonic jazz melodies using basslines, chords and conditional information (Madaghiele et al., 2021); polyphonic pop music generation using beat-based modelling (Huang and Yang, 2020); linear transformer models using Monte Carlo Tree Search as a decoding process to match prediction to emotions (Ferreira et al., 2022); generative adversarial networks, which treat images of symbolic music as training data (Walter et al., 2021); recurrent neural networks/long short term memory models (Dua et al., 2020).

Increasingly, transformer-based architectures are very popular within the symbolic domain, leveraging self-attention mechanisms and autoregressive predictions to explore the effectiveness of generating longer-term structures (Huang et al., 2018; Huang and Yang, 2020; Guo et al., 2022; Ferreira et al., 2022; Madaghiele et al., 2021; Hadjeres and Crestel, 2021). There have been

explorations in supporting different architectures by conditioning them on information about musical structure in order to improve music generation. This includes: researching long-term repetition and phrasal structures (Collins and Laney, 2017; Qin et al., 2022); identifying loop structures as additional conditional information (Han et al., 2022); translational pattern discovery Collins et al. (2010). These types of structures play a role in supporting generation, typically enhancing the musical quality. In some empirical research, pattern-supported Markov models were found to compete with transformer-based architectures (Yin et al., 2022, 2023)

An advancement of transformers, namely the Bidirectional Encoder Representation for Transformers (BERT), has seen use within music generation tasks for in-painting between sections of a composition (Hadjeres and Crestel, 2021), and in generating layers of compositions conditioned on pre-existing material (Guo et al., 2022). These models differ from the base transformer architecture in that it only utilises an encoder, and it is trained to understand the context of data bidirectionally, whereas transformers are trained for sequential tasks.

Beyond language models, Generative Adversarial Networks and Diffusion models are popular in the symbolic domain. However, due to the nature of transformers and their ability to capture longer-form dependencies within a corpus, this thesis focuses on their application.

Melody Generation

Beyond polyphonic music generation, the generation of individual layers of a musical composition (monophonic elements) is an area of research that is particularly appropriate to this research. The strength of many of these learning-based approaches being "generality" (Briot and Pachet, 2020) means that models can be applied to polyphonic and monophonic data with little work required to adapt them. For example, melody generation is a popular topic of research, in particular, where increasingly complex architectures have been shown to generate effective melodic phrases with or without additional conditional information (Herremans et al., 2017; Colombo et al., 2017; Wu et al., 2019, 2020, 2023; Wu and Yang, 2022; Yu et al., 2021; Choi et al., 2021; Madaghiele et al., 2021).

Melody is often described as a sequence of notes that are perceived to be a single unit, recognisable as the voice of a piece of music (Miranda, 2024). Typically, these notes will be higher than contemporaneous voices, and they have a linear style of progression that is often recognisable or singable. The generation of melody can be relatively simple, by producing a monophonic sequence of notes as long as they are in the right range (Brooks et al., 2009). Within video games, the first use of this type of melody generation dates back to the riffology algorithm by Langston (1986), used in *BallBlazer* (Lucasfilm Games, 1985), which generated an endless sequence of notes and included the odd omission of notes for variety.[12]

However, melody generation can be much more complex than this, including additional information to shape the melody around the rest of the

[12]BallBlazer was released before the publication of the academic work by Langston discussing the riffology algorithm.

instrumentation or compositional layers Madaghiele et al. (2021); Pachet et al. (2001); Wu et al. (2020); Choi et al. (2021). These hierarchical and language-based architectures have seen success within research settings, and in a few examples of non-melody layer uses in video games.

Full polyphonic compositions, and melody as a monophonic layer – being the "core" of a musical composition (Miranda, 2024) – are not necessarily the most appropriate approaches to music generation for use in a video game. These ideas will be explored in Chapter 5, which influenced the design of algorithms in Chapters 6 and 7. With this in mind, there is a certain type of musical element that could be generated to reduce repetition, namely: **countermelody**.

Countermelody Generation

A countermelody is defined as another melody that sounds good when played alongside the main melody, often but not always adhering to some basic guidelines (Prudente and Coronel, 2017; Ashton and Colbaugh, 2011). These guidelines include: the use of contrary motion; different rhythms; different notes; and avoidance of parallel motion in phrasing (Prudente and Coronel, 2017; Freedman, 2013). Counterpoint, by comparison, is similarly defined as the "process of placing notes against notes" to construct a polyphonic musical piece from the original monophonic melody (Huang et al., 2019, p.1). This distinction is important, as it can be argued that counterpoint and countermelody are different (Prudente and Coronel, 2017), as their stylistic rules differ, and counterpoint is more often employed in the traditions of Western classical music. However, there are similarities between these definitions

and their compositional approaches, and this division is arguable, especially where computational approaches are concerned.

While melody generation is a popular avenue of research, countermelody generation is an area that receives comparatively less attention (Prudente and Coronel, 2017), and while counterpoint generation by comparison receives more attention than countermelody generation does, the area is still lacking compared to most aspects of MIR research (Padilla and Conklin, 2018; Garay Acevedo, 2004; Ebcioğlu, 1979). Supporting this idea, survey papers on music generation will sometimes omit or fail to include either term, or even both (Ji et al., 2023; Herremans and Chew, 2016).

2.3.5 Expressive Rendering of Symbolic Music

Symbolic music generation can be an effective solution for generating musical layers or countermelodies; however, it is important to consider how we will improve the listenability of generative solutions, as many generative music solutions are often criticised for being "muzak," a term generally used to describe music which is bland and generic. Fortunately, another subfield of computer music research can assist in dealing with this issue. The field of expressive rendering has adequately progressed in the last ten years in modelling expressive music performance onto symbolic (often MIDI) music data (Jeong et al., 2019b,a; Cancino-Chacón, 2018; Worrall et al., 2024). While music generation outside of video games is already exploring the use of expressive training data (Huang et al., 2018) and some modelling of music performance either through individual characteristics related to genre (Malik and Ek, 2017) or as part of the music generation pipeline (Wu and Yang,

2022), expressive rendering has yet to be fully explored in video games. The closest to expressive rendering that video game research has come is in a small study where an early Kungliga Tekniska Högskolan (KTH) ruleset was applied to real-time music synthesis in response to player decisions (Haaland, 2020), and in the doctoral work of Livingstone (2008) who created an early adaptation of their expressive rendering system for a video game engine. However, both systems are outdated and have not seen comparative evaluation in over a decade, making them unsuitable for modern use. Furthermore, without further testing, they do not truly reflect expressive rendering within a game. Unfortunately, as this section will outline, many algorithms are not designed with a wide breadth of applications in mind. Chapter 6 will explore work creating an algorithm to deal with this. However, before a full discussion of the algorithms used for expressive rendering, I will briefly touch upon Music Performance Science, the field informing expressive rendering research.

Music Performance Science

A plethora of research considers how musicians make decisions during performances (Seashore, 1936; Gabrielsson, 1987; Palmer, 1988; Repp, 1990, 1994). While many factors can affect the perceived expression of a musical performance, it is well established that musicians make their performances expressive through the "subtle continuous shaping of musical parameters such as tempo, timing, dynamics and articulation" (Chowdhury and Widmer, 2021, p.1), and so research into expressive performance focuses on these parameters (Kirke and Miranda, 2009; Widmer and Goebl, 2004; Cancino-Chacón et al., 2020).

In the area of music performance science (MPS), there are efforts to mathematically evaluate musical expression (Doherty, 2019) – in particular, how performers use dynamics (Kosta et al., 2016) and timing (Rector, 2021) – often in order to better model timing/micro-timing deviations (Stables et al., 2014) or dynamics (Tan et al., 2020) via machine learning. Consideration of this literature led us to the initial decision to model the general tempo of playing, micro-timing of note start and end times, and dynamics (changes in loudness or velocity) in piano playing, as these are the main factors affecting the expressive qualities in a musical performance (Chowdhury and Widmer, 2021; Doherty, 2019).

Expressive Rendering

Expressive rendering entails the study or modelling of human-like performance parameters, based on but going beyond information contained in a musical score. Computational approaches go back several decades, with rule-based modelling systems emerging in the late 80's/early 90's (Sundberg et al., 1987; Friberg, 1995). Table 2.1 contains a reverse-chronological, non-comprehensive but representative summary of systems used for expressive rendering in the last 20 years: we can see that rules-based models are among the earliest examples of research into expressive rendering (Widmer, 2002; Bresin et al., 2002; Zanon and De Poli, 2003) with artificial neural network models soon following (Bresin, 2000; Friberg et al., 2006). This research leverages MPS to better inform computational modelling of human musical expression.

Table 2.1: Summary of expressive rendering systems. OPD stands for ontimes, pitches, and durations. ANN stands for score annotations.

System	Description	Inference input	Code	Evaluation
CFE and CFE+P (Worrall et al., 2024)[13]	NN Model with 4 or 5 transformers	OPD	Yes	Listening study, metric evaluation of distributions, self-attention mapping
Compose + Embellish (Wu and Yang, 2022)	Generation and performance modelling with transformer	OPD + chord progression + beat-level info	No	Listening study & metrical evaluation of distinct n-grams, beat-level info
3-layer bi-directional LSTM (Shi, 2021)	Three LSTM models with 128 units	beat-wise OPD + score cues	No	Pearson coefficient vs human, schemata discussion
VirtuosoNet (Jeong et al., 2019a)	RNN with hierarchical attention and CVAE	OPD + score cues	∼	Listening study, case study comparison, reconstruction loss

Continued on next page

Table 2.1 Continued

System	Description	Inference input	Code	Evaluation
Graph Neural Network (Jeong et al., 2019b)	Graph gated hierarchical attention model	OPD + score cues	Yes	Listening study, reconstruction loss
Seq2seq + VIB (Gillick et al., 2019)	Seq2Seq model + recurrent variational info bottleneck	OPD + aligned audio	Yes	Listening study, mean square absolute error, KL divergence
Basis Mixer (Cancino-Chacón, 2018)	Neural network + linear models	OPD[14]+ score cues	∼	Cross validation comparison
CVRNN (Fujishima, 2018)	CVRNN with position-dependent conditional inputs	OPD + piano roll around position	No	Listening study

Continued on next page

Table 2.1 Continued

System	Description	Inference input	Code	Evaluation
Maximum Entropy (Moulieras and Pachet, 2016)	Maximum Entropy model	Note metrical onset	No	Listening study, convergence/swing, perceptual validation
CaRo 2.0 (Canazza et al., 2015)	Interactive rule set	OPD + score cues + ANN	No	RENCON
Linear basis mixer (Grachten and Widmer, 2012)	Linear regression	OPD + score cues	No	Goodness-of-fit, analysis of dynamics in recording, and predictions
Kalman Filter (Gu and Raphael, 2012)	Switching Kalman Filter/Gaussian	Ontime + metrical onset	No	Listening study (pairwise), smoothing, improvement
YQX (Flossman et al., 2012)	DBN + rules for articulation	OPD[15]	No	RENCON

Continued on next page

Table 2.1 Continued

System	Description	Inference input	Code	Evaluation
ESP (Grindlay and Helmbold, 2006)	Hierarchical HMMs	OPD + ANN	Yes	Listening study
KTH Perf. Rules (Bresin et al., 2002)	ANN	OPD + KTH rules	Yes	RENCON
SaxEx System (Arcos and De Mántaras, 2001)	Case-based reasoning	OPD[16]+ sound + user input[17]	No	Listening study
KTH Perf. Rules (Sundberg et al., 1987)	Rule-Based	OPD + KTH rules	Yes	Listening study

— Footnotes —

Table 2.1 indicates that early computer systems for expressive music performance (CSEMPs) – unlike contemporary models – are not reliant on the use of score cues as input, focusing mainly on data that can be extracted from notes, such as ontime, pitch, and duration (OPD). Additionally, some models supplement note-level data with extra conditional information using *score annotations*. These computational annotations are often derived from musicological rulesets, and are typically created automatically at runtime (Flossman et al., 2012; Gu and Raphael, 2012; Arcos and De Mántaras, 2001; Cancino-Chacón, 2018). Alternatively, models may incorporate a human-in-the-loop design to achieve similar goals (Hashida et al., 2010, 2006).

The focus on OPD (absent score cues, sometimes including score annotations) for use in expressive performance modelling applies to most early systems such as YQX (Flossmann et al., 2009) and Director Musices/KTH (Bresin et al., 2002), both of which are examples of CSEMPs that performed well at RENCON (Katayose et al., 2012). As the field has evolved, and we move up Table 2.1, additional input features have been included with the aim of improving these models. For example, human-in-the-loop CSEMPs (e.g. (Hashida et al., 2006, 2010)), allow users to annotate music for phrasal information and edit articulation curves. Later machine learning models incorporate a variety of input features for the same purpose, including: score cues (i.e., dynamic and tempo markings) (Jeong et al., 2019a; Flossmann et al., 2009; Canazza et al., 2015); encoded mid- and high-level features such as metrical stress patterns (Cancino-Chacón, 2018); or automatically added score annotations based on musicological analyses (Narmour, 1990; Lerdahl and Jackendoff, 1996; Herremans and Chew, 2016).

As Table 2.1 indicates, the continued development of deep learning architectures and an increased number of input features have continued to push the boundaries of expressive performance modelling in recent years. In particular, the use of long short-term memory (LSTM) models and multivariate Markov models has been used to explore the modelling of expressive note timing in piano mazurkas and string quartets, respectively (Shi, 2021; Stables et al., 2014). Similarly, an LSTM approach has been used in the modelling of genre-dependent note velocity, while requiring only information available in MIDI (similar to OPD) (Malik and Ek, 2017). The incremental development and improvement of deep learning has continued to push the boundaries. CSEMPs have seen continued improvement using, but not limited to; graph neural networks (GNNs) (Jeong et al., 2019b), hierarchical attention recurrent neural networks (HRNNs) with conditional variational autoencoders (CVAEs) (Jeong et al., 2019a), and convolution variable recurrent neural networks (CVRNNs) (Fujishima, 2018).

While the evaluation of CSEMPs has become non-standardised in recent years (see Section 2.3.5), many papers include a listening study to evaluate the quality of a model's performance, and some evaluations are even comparative. In particular, the graph gated HRNN architecture has outperformed the Basis Mixer and the vanilla HRNN, showing the promise of graph gated hierarchical architectures (Jeong et al., 2019a). In this comparison, all models use similar input features, such as; grace notes (Flossmann et al., 2008), dynamics markings (Grachten and Widmer, 2012), and articulation markings (Cancino-Chacón, 2018; Grachten and Krebs, 2014) commonly found in MusicXML files. In addition to listening studies, some research includes

quantitative analysis of the proposed model and human output, to indicate the extent to which model output emulates characteristics of human output (Jeong et al., 2019b,a; Gillick et al., 2019; Moulieras and Pachet, 2016).

The CVRNN model (Fujishima, 2018) is an exception to the trend of the increased utilisation of score cues. Unlike the models compared in Jeong et al. (2019a), the CVRNN emphasises OPD, supplementing it by providing note-level information for all of the notes in a seven-bar window around each note. This approach offers enhanced phrasal context to inform predictions. This aligns closely with the input features of the models introduced in this thesis, with my use of positional embeddings and self-attention mirroring this phrasal-contextual information.

Expressive Music Generation

Beyond the scope of research focused entirely on rendering expressivity onto non-expressive music data, music generation tasks have seen similar application of machine learning techniques, including but not limited to neural networks: seq2seq (Madaghiele et al., 2021), recurrent neural networks (RNNs) (Dua et al., 2020), and Markov chains (Collins and Laney, 2017) in the generation of non-expressive symbolic musical output. In three non-

[13]This model is the architecture introduced in a later chapter of this thesis (Worrall et al., 2024).

[14]This model encodes mid-high level features of the score (i.e., tonal tension) for improved inference (Herremans and Chew, 2016).

[15]This model uses Narmour's IR model; these are basic principles of melodic perception, derived empirically by (Narmour, 1990) to enhance their input.

[16]This model uses the generative theory of tonal music (Lerdahl and Jackendoff, 1996), Narmour's IR (Narmour, 1990), and a proposed analysis (Jazz Theory) to enhance their input.

[17]This model supplements input with user-provided qualitative values across 3 dimensions of preference: tender-aggressive, sad-joyful, calm-restless.

peer-reviewed papers, the transformer neural network architecture has been used to output symbolic notes **and** expressive renderings of those notes, all in one go (Huang et al., 2018; Payne, 2019; Wu and Yang, 2022).[18] As such, these algorithms also represent research into expressive rendering, in scenarios where score cues are not available per se. However, the training data itself is expressive, containing expressive tempo, timing, and dynamic information. Large datasets have been used to train two of these models (e.g., hundreds of hours of piano performances transcribed from YouTube, among other data sets), and the authors acknowledge that there may be shortcomings with regard to the extent to which the algorithms' outputs are original.

Theoretical and empirical work attempting to address this originality issue has found evidence to suggest that users of these algorithms may be listening to or using output consisting of large chunks of training data, often without recognising the problem. This is because the training sets are too large for instances of copying to be identified manually (by ear) (Yin et al., 2021, 2022). As such, without checking properly for originality, we suggest that it might be overly ambitious to use a transformer neural network to generate notes and expressive renderings of those notes all in one go.[19] We see potential, however, in using transformer neural networks to learn and predict just those aspects of expressivity, given quantised start

[18]This has also been termed "direct performance generation" in a peer-reviewed contribution focusing on LSTMs (Oore et al., 2020).

[19]In models that rely on tokenisation, expressive data such as MIDI events where timing and velocity information is included tend to produce a larger token dictionary than inexpressive data, where notes can be represented with fewer tokens and velocities are ignored. If two training sets contain the same pieces, but one is expressive and the other inexpressive, a model trained on the expressive set is more prone to copying chunks of training data. This is because the larger dictionary reduces the relative variety in dependencies between local tokens, making repeated patterns more likely to be memorised.

and end times and pitches of notes as a reduced input, similar to that seen in early CSEMP research (Flossmann et al., 2009; Bresin et al., 2002). The benefits of this reduced set of input features are the increased potential for use within the domains such as DAWs and video game engines, where these algorithms have innovative potential. Furthermore, the transformer architecture's self-attention mechanism may be powerful enough to compensate for the smaller number of input features, by providing positional context to the model's understanding of notes, and potentially surpass the capabilities of non-attention-based architectures, such as RNN or LSTM.

The recent Compose & Embellish model (Wu and Yang, 2022) utilises two transformers based on the Compound Word model (Hsiao et al., 2021). The first "Compose" model generates a lead melody based on a prompt, while the second "Embellishment" model generates an accompaniment (i.e., chords/left hand on a piano), as well as generating deviations in timing and dynamics on the note level for the generated content. While this utilisation of the transformer does tackle elements of performance modelling, it does not tackle deviations in human tempo change, nor the use of a sustain pedal, leaving it lacking in some elements of performance modelling.

Evaluation

Since the discontinuation of RENCON, in 2014, the evaluation of CSEMPs has become non-standardised. Of the papers listed in Table 2.1, 10, including this chapter, use listening studies, and 10 use a variety of metric analyses (goodness-of-fit, loss, divergence, etc.) to assess their models. While only 3 of the models included in Table 2.1 were evaluated at RENCON (Katayose

et al., 2012), the contest/platform enabled models using a variety of architectures, datasets and input features to participate in widespread comparative evaluation. Notable models that performed well at RENCON include YQX (Flossmann et al., 2009), Director Musices (Bresin et al., 2002), Pop-E (Hashida et al., 2006), Mixtract (Hashida et al., 2010) and CaRo 2.0 (Canazza et al., 2015). The models indicated here represent a mix of approaches, including some early OPD-only systems and some human-in-the-loop models.

In recent years, listening studies have been widely used in this field, but they demonstrate a variety of methodologies (Gillick et al., 2019; Fujishima, 2018; Jeong et al., 2019b; Wu and Yang, 2022). Each of these papers includes a listening study with between 11 and 40 participants; however, of these four papers, two use pairwise comparisons per piece, where listeners pick which is better (Gillick et al., 2019; Jeong et al., 2019b), while two compare all systems per piece, where metrics of composition and performance are rated using Likert scales (Fujishima, 2018; Wu and Yang, 2022). In addition, three of these papers use $\approx$ 30 sec excerpts (which (Jeong et al., 2019b) finds is not long enough to judge a performance). Furthermore, none of the 4 papers uses a standardised baseline for their comparison; instead, opting for bespoke or adapted existing algorithms. This variety of evaluation approaches, combined with a historic absence of open-source code, makes "systematic" comparison of models difficult. We propose the adoption of the industry-standard baseline from DAWs to establish consistency across future listening experiments.

2.3.6 Contextualising Music Generation and Expressive Rendering within Video Games

Thus far within this literature review, I covered how music repetition is a large problem within video game audio, and the underlying psychology behind the problem. Furthermore, I have outlined how music generation is a potential solution to this problem (in particular, countermelody generation), and then explored how the musical quality of the generative element can be improved. However, thus far I have yet to dive deeply into the application of generative music within video games. This final section will combine the previous areas to outline existing approaches to video game music generation and performance, and then outline the gaps within the literature that are being tackled within this thesis.

The work of Plut and Pasquier (2020) offers a non-comprehensive history of procedural and generative systems within video games, and applies their taxonomy. This section will now discuss some of these examples and relate them to the evolution of procedural music generation and performance, before identifying the gaps for learning-based approaches. The earliest examples of symbolic generative solutions in a video game date back to the 1980's, with the earliest known example being *BallBlazer's* "riffology algorithm", which generates melodic lines using a rules-based approach that draws from a library of thirty-two predefined eight-note riffs (Langston, 1986; Plut and Pasquier, 2020). This rules-based approach would determine the speed of playback, loudness of sections and whether notes were omitted, or if a rhythmic break was included. *Ballblazer's* use of procedural solutions in non-diegetic, sitting outside the ludo-narrative of the video game, with

a granularity that is phrase- or note-based and linear (as the music that is generated does not adapt to changes in the game state (Plut and Pasquier, 2020). This was shortly followed by *Otocky* (ASCII Corporation, 1987), which differs from *Ballblazer* in that music is a key aspect of the game's design. In *Otocky*, directionality corresponds to different musical notes, which are played when the player presses a button, but is further quantised to match the background beat (Oliva, 2015). In this title, players could improvise melodic lines by being selective about button presses, as an early type of precursor to video games like *Rez* (Sega, 2001). In *Otocky*, you could additionally unlock a music editor to design your own music, as an early precursor to tools such as *Mario Paint* (Nintendo, 1992a). As a procedural solution, Otocky sat within the game's ludo-narrative diegesis, as an adaptive, rules-based approach working in a granularity that can be described as note-level. However, a key difference here is that both systems reduce repetition in different ways: *Ballblazer* creates endless melodic lines and adds performative variation such as note omissions and rhythmic breaks, whereas *Otocky* allows players to control the notes, but controls their performative playback. Neither of these examples utilised modelling of expressive performance, but applies the ideas of generative performance and music to the problem of video game music repetition.

Procedural performance systems are the least common type, with the next earliest examples being Rez, which is similar to Otocky, combining elements of music video games with shooting mechanics, and *Spore* (Maxis, 2008). Spore utilised a generative music system created by Brian Eno, Peter Chilvers, Kent Jolly and Aaron McLeran (McLeran and Jolly, 2008). These

music systems were based on a modified version of PureData. This system was mainly focused on the generation of notes, but included some elements of controlling playback (McLeran and Jolly, 2008). Procedural performance algorithms gained some growing popularity in the early 2010's with video games like *Bit. Trip Runner* (Gaijin Games, 2010), *Child of Eden* (Q Entertainment, 2011), but seemingly the focus of such systems shifted to academic research with projects such as AudiOverdrive (Holtar et al., 2013).

Some theses focus on the application of learning-based approaches to procedural generation (Prechtl, 2016; Cutajar, 2020). Cutajar (2020) will be discussed with systems that are only generating compositions; however, Plut and Pasquier (2020) defines Prechtl (2016) as a performance and composition system. The focus of this work is on generating music with Markov chains that fit dynamically within emotional categories, and the performative aspects account for mostly timbre intensity and velocity or notes, as well as a focus on chord typing and voice leading choices (Prechtl, 2016, p.37). The work of Scirea et al. (2016, p.449) performs a similar function, generating emotional music, and then adding some performative elements to the music through the inclusion of rhythmic and timbre alterations. However, this approach uses an evolutionary algorithm, as opposed to Markov chains. While these systems do add some variety with regard to certain aspects of a performance, similar to Malik and Ek (2017), this falls short of true expressive performance rendering. The closest approximation of expressive modelling to a video game is in Haaland (2020) in which the Director Musices system is added into a small web-based video game, where the player acts as a conductor of the background music (Bresin et al., 2002). Rather than a simple

Mickey-mousing effect,[20] player behaviour is mapped to attributes such as activity, valency and note velocity, which in turn dictates how the music is performed by the computer. This is the first true application of a CSEMP to a video game; however, it is more of a toy than a real application looking to tackle existing problems within the area. This is a gap in research that this thesis investigates.

One of the more common types of procedural approach is composition. This is the generation of notes or musical layers, or whole pieces. Returning to the survey by Plut and Pasquier (2020), *Spore* is one of the earlier examples noted above that does not rely on stochastic randomness, but uses a version of PureData to allow for compositions to evolve in real-time as the player creates monsters and worlds. *The Audience of the Singular* (Plut, 2017) is another project on the list that acts as a DJ simulator with game mechanics in the style of *Guitar Hero* (Harmonix, 2005). *The Audience of the Singular* uses a corpus-based machine learning approach to generate music which attempts to emulate music for video games of the 1980s and 1990s (Plut, 2017). The work of Cutajar (2020) is another thesis that focuses on dynamic music changes in response to game changes, similar to Prechtl (2016) and Scirea et al. (2016). However, the focus of their work is on interpolating between two existing pieces of music to create a more natural cross-fade between areas within a video game. This model also uses a Markov chain-based approach, ignoring the requirement to generate music matching a specific mood, and more on using existing pieces within a video game. This type of music interpolation algorithm is notably anticipated by the doctoral work of Wooller (2007).

[20]Mickey-mousing is a term used to describe a 1-1 synchronisation of movement on screen to musical playback or scoring.

Within the industry, grammar- and learning-based approaches have been adopted by larger projects, such as the use of Impro-visor, a grammar-based music system that is designed for jazz music (Keller et al., 2007). This algorithm was used in *Genesis Noir*, and bass lines are generated in a call and response fashion based on the player's input on a saxophone within the video game (Abel, 2021). Similar to *The Audience of the Singular*, this is more of a toy, offering players a unique experience. Another example of procedural music in video games is the percussion in *Rise of the Tomb Raider* and *Marvel's Avengers*. These projects used the dynamic percussion system invented by Daniel Brown of Intelligent Music Systems, which takes percussion written by the composer and then, in real-time, creates new percussion layers in response to in-game parameters (Brown, 2019; Lamperski, 2015). Playback of these layers, and their corresponding audio samples, is dictated by real-time parameters in the game, such as enemy proximity and whether the player is in combat or not. This use is less of a toy and more functional, as it allows for less repetitive percussion, but also for the video game to offer the player ludic feedback, to better inform their decisions (Stevens et al., 2015).

Moving beyond composition and performance systems, it is made clear in Plut and Pasquier (2020), and by any engagement with contemporary video game titles, that procedural arrangement systems are the most common type. There are a variety of factors that likely cause this to be the case. For example, video games as a domain offer extra technical considerations, such as: the necessity for robust timing systems for the generation and playback of note-level events (Weir et al., 2021); limited infrastructure within the industry to support the use of learning-based approaches (Worrall and

Collins, 2023; Abel, 2021). However, this is not empirically studied and is an area addressed in this research. Furthermore, it is clear that the adoption of learning-based approaches is nascent within the field. Even when considering systems not included in the survey, there are limited applications of learning-based approaches to video game music generation or performance. Within the video game industry currently, some startups are exploring such applications of learning-based approaches, such as Infinite Album and Reactional, but this is still nascent, meaning there is a large gap within the research space that can be addressed within this research.

2.4 Chapter Summary

In this chapter, I have outlined and highlighted the terminology required to understand music within video games, to understand AI and its applications within music generation and I have discussed many of the issues surrounding music generation as a solution to the problem of listener fatigue within video games. In defining the problem area and the issues with potential solutions, I have outlined gaps within the research area that will be addressed in subsequent chapters. Chapter 4 will explore how contemporary titles are tackling the issue of listener fatigue through a comparative analysis of music design approaches over 25 years, and explore which situations within a video game are viable as ideal environments for generative systems. Chapter 5 explores the factors that may impede the use of procedural or generative technologies and informs the creation of any systems in later chapters. This chapter will also compare existing adoption to technology adoption literature in a more localised background area, and is the first empirical study of its kind. Chapter 6 explores the creation of a music performance model which can improve the quality of generative solutions (for later embedding within a video game, which addresses this research gap). Chapter 7 explores the effects that both systems have on listener fatigue during gameplay in a game-based listening study.

3 Methodology

3.1 Introduction

This chapter introduces the methodological approach adopted in this thesis and outlines how this framework has shaped the data collection and analysis processes. Although there are various approaches or *paradigms* that can be used to investigate a research problem, each approach possesses its own unique characteristics and epistemological underpinnings (Cohen et al., 2018). Cohen et al. (2018) explains that these paradigms should not necessarily drive the research, but that the research is driven by the purpose of the investigation, which subsequently implies the use of a specific paradigm. Examples include quantitative approaches, which measure and describe aspects or qualities of the research question through numerical data, or qualitative approaches, which explore opinions, concepts, and experiences through non-numerical data. Given the human-centric and technological nature of the work carried out in this thesis, I have adopted a mixed methods approach (Tashakkori and Teedlie, 2010) that allows for the collection, analysis, and integration of qualitative and quantitative data, which in turn leads to a more comprehensive understanding of the research problem (Creswell, 2019). The rationale for this decision is that developing tools for the video game industry

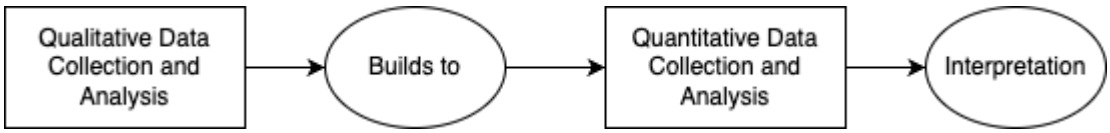

Figure 3.1: The Exploratory Sequential Mixed-Methods Process (Creswell, 2019)

requires the guidance of professional interviews to ensure industry relevance, for which qualitative approaches are more appropriate, while the evaluation of the resulting computational methods requires quantitative approaches to quantify evidence and test for statistical significance.

With regard to research paradigms, Candy (1989) states that all theories can be grouped into three main categories: Positivist, which relies on measurement and reason to develop and test theories about the natural and social worlds; Interpretivist, which assumes reality is a subjective, social construct defined by individual experience; and Critical, which focuses on power imbalances while combining action and theory into praxis. However, Tashakkori and Teedlie (2010) identifies a fourth paradigm, namely Pragmatism, which involves employing a variety of methods to solve real-world problems. Given thepractice-orientedd nature of this research, I have adopted a pragmatic worldview. According to Creswell (2019), this perspective allows researchers to concentrate primarily on the research questions and facilitates the combination of elements from multiple paradigms to create a methodological approach more emblematic of real-world complexities

When considering the structure of this research, Creswell (2019) demonstrates that an exploratory sequential mixed methods design utilises qualitative research to inform and build a subsequent quantitative data analysis phase prior to the final integrated interpretation (see Figure 3.1).

Following Creswell (2019), I outline the full mixed methods approach adopted in this thesis in Figure 3.2. Although the remaining sections of this chapter cover each individual study and its chosen methodology in more detail, Figure 3.2 illustrates the basic framework of how qualitative work directly informs the subsequent quantitative phases. This process commences with a case study (Phase 1 – qualitative) to assess the pragmatic need for music generation in contemporary video games. Simultaneously, a reflexive thematic analysis of interviews with professional composers (Phase 2 – qualitative) is conducted to identify the factors inhibiting the adoption of current technologies and to determine which approaches are most worth exploring in the remainder of the research. The findings of these two studies are synthesised and used to inform the decision to develop an expressive rendering model, which is then tested independently of the video game environment (Phase 3 – quantitative). Once this algorithm has been developed, the model is integrated within an *in situ* video game study to evaluate its efficacy in alleviating repetition and improving the perceived quality of background music (Phase 4 – quantitative).

3.2 Case Study

In Chapter 4, I conduct a case study comparing two versions of a video game to examine how the role of technology in video game music has evolved. This analysis serves to evaluate where, within a modern video game context, there is potential for generative music or AI-powered music solutions. The case study approach was chosen as it facilitates the "detailed and intensive analysis of a particular event, situation, organisation or social unit" (Schoch,

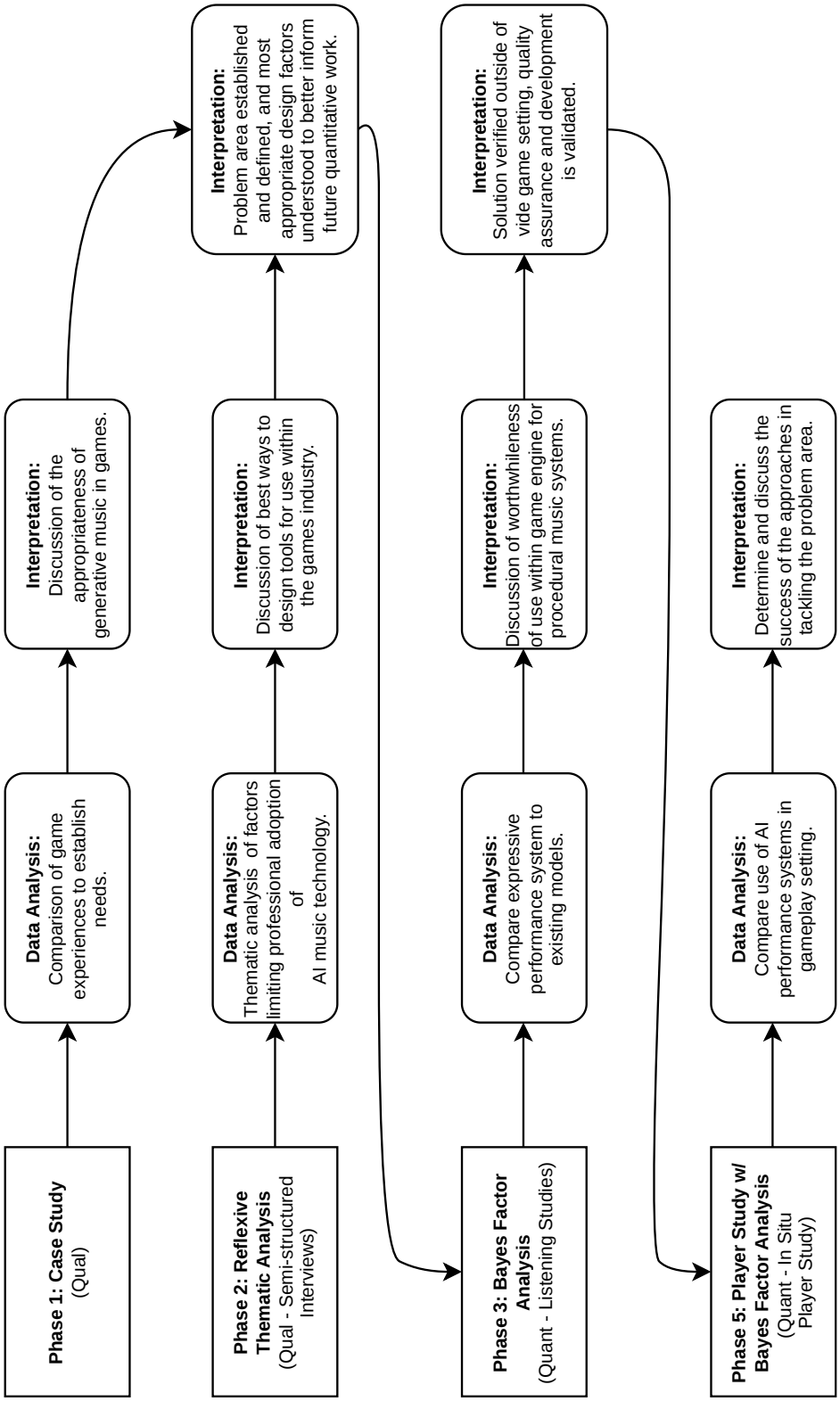

Figure 3.2: The design of the research carried out in this thesis, after Creswell (2019).

2019, p.245). In this instance, the two video games represent the phenomenon being examined. As case studies typically focus on events that exist within "some sort of bounded context" (Miles et al., 2014, p.28), this research adopts a clearly defined scope by comparing two versions of the same title released twenty-three years apart.

An advantage of this approach is that it offers researchers the opportunity to develop a "comprehensive understanding" of the phenomenon and derive "principles and lessons" that may be applied more broadly (Schoch, 2019, p.246). Given that the focus of Chapter 4 is to understand the evolution of music design and isolate specific aspects of game development where music generation may prove beneficial, this is an appropriate methodology.

3.3 Semi-Structured Interviews

In Chapter 5, I employ semi-structured interviews to collect qualitative data for subsequent analysis. Semi-structured interviews are commonly utilised in video game research to explore the thoughts and opinions of developers regarding specific topics, such as experiences with creative AI, for example text-to-image generation (TTIG) models (Vimpari et al., 2023). This approach is also effective for investigating how industry professionals define key terminology (such as "quests," which can vary significantly in meaning between different studios, developers, and titles) (Kristen et al., 2021). Beyond video game research, semi-structured interviews are often used to engage visual artists and AI art communities to further probe experiences of using TTIG models (Buraga, 2022; Ko et al., 2023), and to query sound designers and podcasters about artificially intelligent music and audio tech-

nology (Rime et al., 2023; Kamath et al., 2024). The prevalence of these studies demonstrates the suitability of this methodology for collecting rich qualitative data within creative and technical domains.

3.4 Reflexive Thematic Analysis

In Chapter 5, when analysing qualitative data, I employ a reflexive thematic analysis methodology (Braun and Clarke, 2006, 2019). This approach is similar to other thematic analysis methods, where the researcher takes transcribed data and codes it, labelling snippets representing relevant thoughts before collating these codes to create sub-themes. These sub-themes are then brought together to create a candidate theme that acts as an umbrella for the narrative the sub-themes describe. In contrast to grounded theory, reflexive thematic analysis treats themes as actively constructed by the researcher rather than as emergent properties of the data alone (Braun and Clarke, 2019). The reflexive part of this methodology attempts to address the researcher's subjectivity when the researcher takes the candidate theme back to the raw, uncoded data and checks to ensure that the themes are still representative of the transcribed data. This additional step reduces the likelihood of unintentional bias. Furthermore, I choose reflexive thematic analysis as it allows for deep engagement with the data; knowing that I am the sole analyst, the reflexive aspect provides confidence in the findings in lieu of measures such as Cohen's Kappa for agreement between coders. Further details on the coding procedure and the questionnaire can be found in Chapter 5.

3.5 Listening Studies

In Chapter 6, I conduct two listening studies to evaluate my expressive rendering algorithms (*Cue-Free Express* and *Cue-Free Express + Pedal*). As outlined in the literature review within Worrall et al. (2024) (see Chapter 2), this is the most common approach to evaluation in expressive rendering research. This type of evaluation is often supported by computational analysis, where predicted features or aspects of training data are analysed to gain greater insight into the predictive accuracy of models (see Section 3.8).

With this approach, I present participants with 36 and 63 stimuli (Listening Study 1 and Listening Study 2, respectively) using a within-participants design. Participants rate each stimulus for each condition on four metrics: overall expressivity, timing, tempo/rubato, and dynamics. These ratings are collected using a 7-point Likert scale, and the resulting data are analysed using Bayes Factor analysis—specifically, a Bayesian paired t-test (see Section 3.7).

For these listening studies, surveys are constructed in Qualtrics, where I group renditions of individual pieces on the same page; this allows participants to experience all renditions of a piece within a context where all conditions can be directly compared. The nine pieces are presented in the same order, but the presentation of the systems on each page (4 and 7, respectively) is randomised to mitigate order bias. Figure 3.3 shows the experimental design for Listening Study 1, and Figure 3.4 illustrates the design for Listening Study 2, where the only difference is the number of conditions.

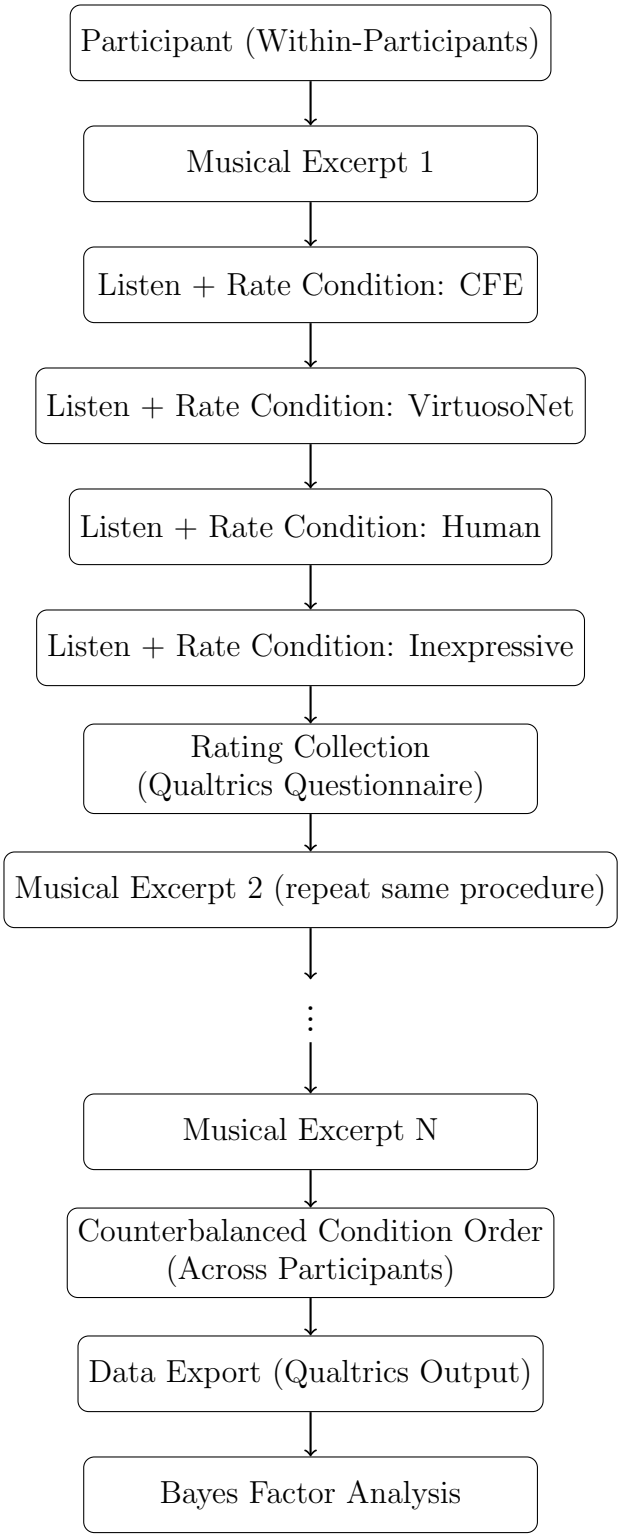

Figure 3.3: Experimental design for Listening Study 1. Participants listened to multiple musical excerpts and rated each excerpt across four conditions (CFE, VirtuosoNet, Human, and Inexpressive). Condition order was counterbalanced across participants.

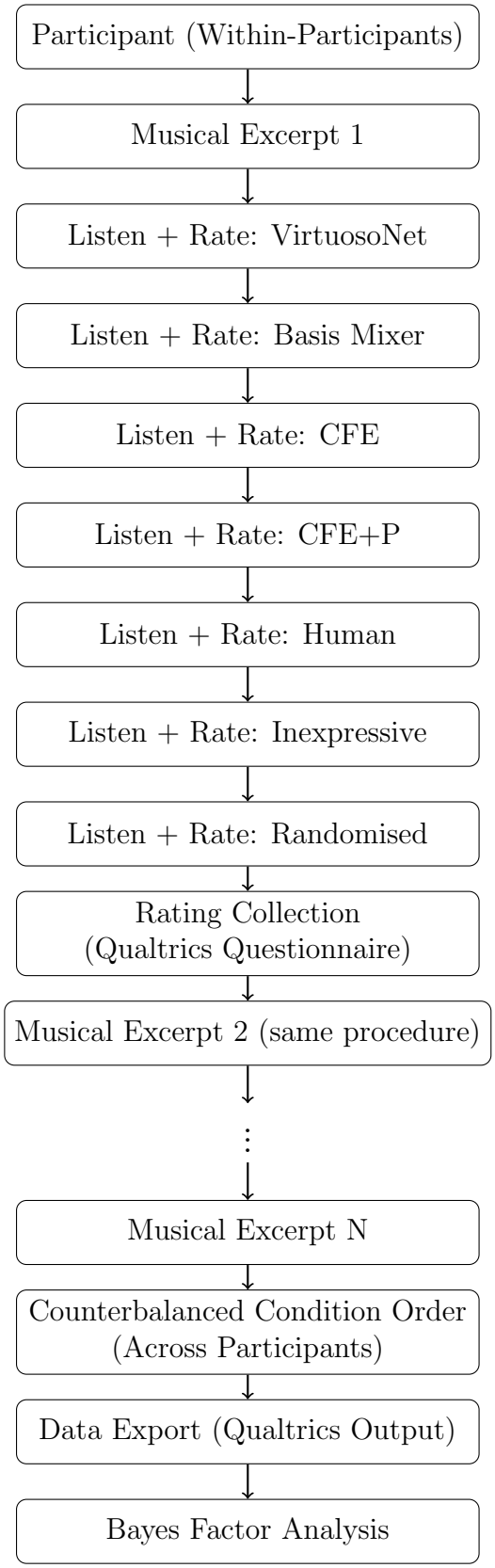

Figure 3.4: Experimental design of Listening Study 2. Participants listened to each musical excerpt rendered under seven conditions (VirtuosoNet, Basis Mixer, CFE, CFE+P, Human, Inexpressive, Randomised). Condition order was counterbalanced across participants.

3.6 Gameplay Study

In Chapter 7, I employ a data collection approach similar to the listening studies; however, participants play a level of a bespoke video game, ensuring their focus is on gameplay rather than solely on musical stimuli. While listening studies are an accepted standard for evaluating expressive rendering, the primary focus of this thesis is the effect of expressive rendering on listener fatigue in players. Consequently, a gameplay study is most appropriate to ensure ecological validity, as it recreates a scenario where the participant's attention is not concentrated exclusively on the music, reflecting a real-world setting.

This method is supported by Cutajar (2020, p.4), who suggests that when studying video game music systems, it is "particularly important to play the game in question and interact with the music system." This allows researchers to understand how the system functions within the context of the play experience. Furthermore, Summers (2016, p.35) notes that while music can be experienced via gameplay recordings, this does not allow the researcher to fully experience the interactivity of the title. I apply this logic to the participants themselves for the same reasons.

The music system used in this study is not interactive; instead, musical performances are pre-rendered. This design choice acknowledges a potential reduction in ecological validity, as pre-rendered assets may still induce listener fatigue in longer real-world play sessions, similar to conventional music design techniques (Weir et al., 2021). However, pre-rendering is necessary to maintain experimental control, as real-time generation can introduce unintended variability. By prioritising control over ecological validity, the system

establishes a rigorous testing environment to explore the effects of expressive variations on the player experience.

Following gameplay, participants rate the music on a 7-point Likert scale for perceived expressive quality, perceived repetitiveness, and the perceived quality of the composition (independent of expressive elements). Data are collected via Qualtrics, following methods common in listening studies. Figure 3.5 illustrates the experimental design, with data analysis after all 20 participants complete the study.

Further details regarding the development of the experimental video game, alongside a discussion of limitations and future directions, are provided in Chapter 7. It is important to note that a ten-minute gameplay session per stimulus may not fully replicate the effects of multiple hours of real-world exposure. This shorter duration is necessary to ensure that participation is feasible while respecting the participants' time.

3.7 Bayes Factor Analysis

In Chapters 6 and 7, I employ Bayes Factor analysis (BFA) to analyse quantitative data resulting from the listening and gameplay studies. BFA is a modern alternative to the traditional frequentist approach to data analysis. This technique offers several advantages over the frequentist hypothesis testing framework (e.g. the F-test behind an ANOVA or standard t-tests), most notably by allowing for the possibility of finding substantial evidence for a lack of difference between two systems. In frequentist hypothesis testing, one can only reject the null hypothesis (H_0) in favour of the alternative hypothesis (H_1), or fail to reject H_0. Critically, the latter is not equivalent to

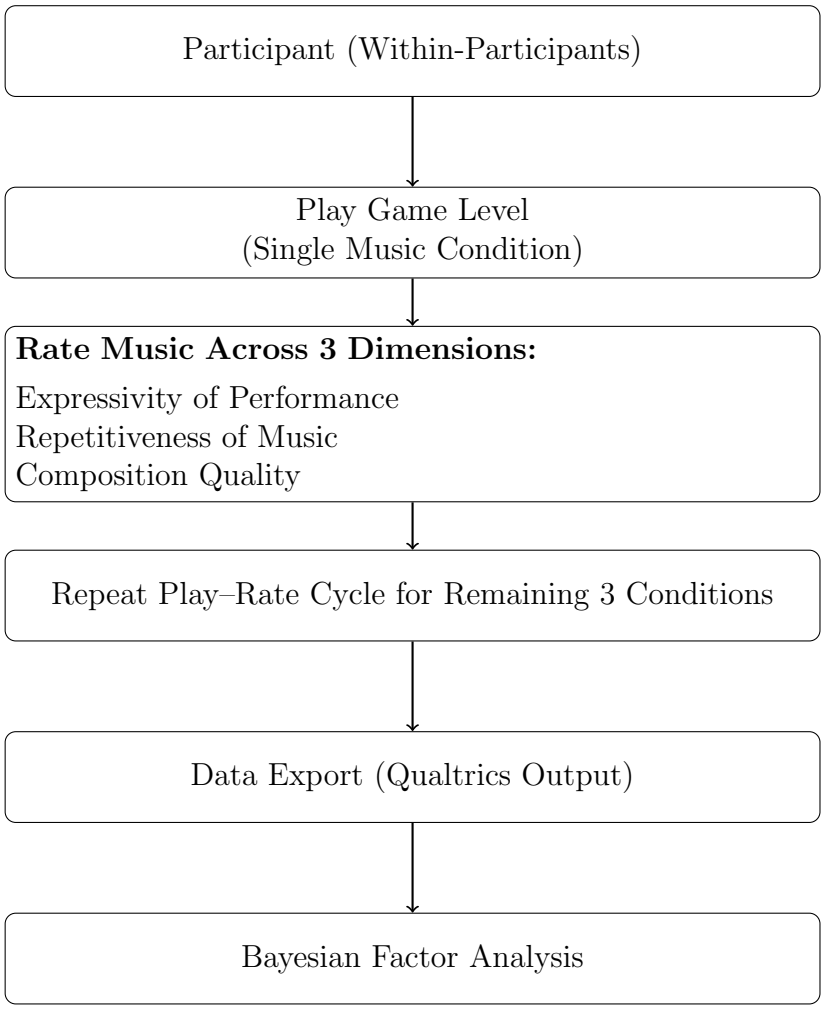

Figure 3.5: Experimental procedure for the gameplay study. Participants played a game level under one music condition, rated the music across three dimensions, and then repeated this play–rate cycle for the remaining conditions. Data were exported for Bayesian Factor Analysis.

finding evidence in favour of H_0 (van Doorn et al., 2020). By comparison, the Bayesian framework can provide evidence in support of either the null or the alternative hypothesis. This framework is gradually being adopted for the comparative evaluation of computational systems (Yin et al., 2023; Barahona-Ríos and Collins, 2022; Worrall et al., 2024), and is particularly relevant to the work presented in Chapters 6 and 7.

In conducting a Bayes Factor analysis, 20 iterations of paired t-tests are performed for each system comparison, with each iteration providing a BF_{10} value. The BF_{10} value represents the Bayesian counterpart to the traditional p-value, quantifying the strength of evidence provided by the data for one hypothesis over another. The mean of these 20 iterations is reported for each hypothesis in the corresponding chapters. Table 3.1 illustrates how these values are typically interpreted regarding the strength of evidence for H_0 or H_1. These thresholds are conventional, similar to the α-value in frequentist statistics, and are based on those established in previous work (Yin et al., 2023). Under the Savage-Dickey density ratio (Dickey and Lientz, 1970; Wagenmakers et al., 2010), the BF_{10} score is calculated by comparing the prior and posterior distributions at the null value (θ_0), as defined by the following equation:

$$\mathrm{BF}_{10} = \Pr(\theta_0|H_0) \,/\, \Pr(\theta_0|\mathrm{data}, H_1) \tag{3.1}$$

For point null hypotheses nested within a continuous alternative, the Bayes factor can be computed using the Savage-Dickey density ratio. In this thesis, BF_{10} is defined by the ratio of the prior density at the null value, $\Pr(\theta_0|H_0)$, to the posterior density after data observation, $\Pr(\theta_0|\mathrm{data}, H_1)$.

Here, the parameter θ_0 represents the point of no difference between the systems being compared. Importantly, these values denote probability densities evaluated at θ_0, rather than probabilities; intuitively, this ratio measures the extent to which the data reduce or increase the plausibility of the null value relative to prior expectations. Because Bayesian estimation utilises Markov Chain Monte Carlo (MCMC) sampling—a stochastic process that can produce minor variations across runs—I perform 20 iterations of each test. Reporting the mean of these iterations ensures a stable estimate of the Bayes Factor and confirms that the findings are not artefacts of a single sampling run.

Furthermore, because I report Bayes Factors in the form of BF_{10}, values greater than 1 indicate increasing evidence in favour of the alternative hypothesis (H_1), whereas values less than 1 indicate increasing evidence in favour of the null hypothesis H_0. Evidence for (H_0) is therefore expressed by Bayes Factors approaching zero. The interpretation thresholds are symmetrical around 1, such that $\mathrm{BF}_{10} = x$ provides the same strength of evidence for H_1 as $\mathrm{BF}_{10} = 1/x$ provides for H_0. Table 3.1 presents the classification scheme used throughout this thesis to describe the strength of this evidence.

BF_{10}	Interpreting
> 100	Extreme evidence for H_1
$30 - 100$	Very strong evidence for H_1
$10 - 30$	Strong evidence for H_1
$3 - 10$	Moderate evidence for H_1
$1 - 3$	Anecdotal evidence for H_1
1	No evidence
$1 - 0.333$	Anecdotal evidence for H_0
$0.333 - 0.1$	Moderate evidence for H_0
$0.1 - 0.033$	Strong evidence for H_0
$0.033 - 0.01$	Very strong evidence for H_0
< 0.01	Extreme evidence for H_0

Table 3.1: Bayes factor interpretation

As a final note on Bayes Factor analysis, classical multiple-comparison corrections, such as the Bonferroni adjustment, are designed to control long-run Type I error rates in frequentist hypothesis testing. Bayesian inference, in contrast, quantifies the plausibility of hypotheses conditional on the observed data and does not rely on fixed significance thresholds or repeated-sampling error control. Consequently, Bonferroni-style corrections are generally unnecessary in Bayesian analyses. Bayesian methods instead mitigate spurious findings through prior distributions, which regularise estimates; the marginal likelihood, which penalises model complexity via an Occam's razor effect; and, where appropriate, hierarchical modelling that partially pools information across comparisons. These mechanisms provide an implicit adjustment for multiplicity while maintaining coherent probabilistic interpretation.

3.8 Computational Analysis and Software Development

A substantial part of the contribution to knowledge made in this thesis consists of the application of programming to develop software solutions to research problems. While the implementation particulars are discussed in depth in the specific results chapters, the overarching technical frameworks are outlined here.

In Chapter 6, I design and train an ensemble of transformer neural networks as a software solution to the problem of creating expressive renditions of non-expressive, quantised MIDI data. In Chapter 7, I adapt an open-source demo video game[1], expanding the level design and integrating both a musical playback system and the previously generated expressive renditions to conduct the experiment. The raw data for this research are made available here, and technical details are discussed in the corresponding chapters. I shall now outline the primary ways in which the application of programming as a software solution was applied to this research.

3.8.1 Data Preprocessing and Model Training

In Chapter 6, I utilise programming to curate and prepare data for training machine learning models. While specific implementation details follow, I first outline the general computational methods applied across my experiments.

[1]The original code can be found here, and it is released under a Creative Commons Zero license stating that release of adapted code is not required. This modified code will be made available following the peer-review process.

Test:Train:Validate Split

Before preprocessing data for model training, it is standard practice to split the dataset into three distinct categories: training, testing, and validation. This process ensures that a deep learning model is accurate and generates generalisable results when applied to unseen data before it is deployed in a production or experimental setting. The necessity for this form of cross-validation long predates deep learning and remains a staple of statistical and machine learning methodologies

Deep learning datasets are generally partitioned as follows: 80% of the data for the training set, 10% for the testing set, and 10% for the validation set. A model is trained on the training set, and the resulting model is subsequently evaluated using the test set. Here, researchers can compare the inferred variables predicted by the model against the ground truth within the test data, simulating the performance of the model on unseen real-world data. Based on these comparisons, the following metrics are recorded: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).[2]

These statistics allow researchers to assess the accuracy and precision of machine learning models by calculating the harmonic mean oftheirs precision and recall scores, where $Precision = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$ and $Recall = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$. The harmonic mean of precision and recall is then used to calculate an F_1 score:

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.2}$$

[2]What constitutes an inferred feature within these categories depends on the context and data type; this is discussed in more detail in the following subsection.

Regarding music and audio data, a potential pitfall is data leakage, where copies or near-copies of training items appear in the validation or testing sets. Consequently, the model may (unbeknownst to the researcher) memorise specific answers rather than learning to identify generalisable patterns. Therefore, rigorous data preprocessing is vital for successful model training.

Performance-Symbolic Beat Mapping

Regarding Chapter 6, I utilise the Aligned Scores and Performances (ASAP) dataset (Foscarin et al., 2020), which consists of 236 musical scores and 1,067 performances of these scores. All of the pieces are Western classical music from 15 different composers, and all are written for and played on piano. Each performance in the ASAP dataset includes an associated annotation file, which I use to calculate a **performance-symbolic beat map** (PSBM) between the beat `ontime` in the symbolic representation of the piece and the beat `onset` in seconds in the performance. Original beat annotations for the ASAP dataset were automatically created (explained in Foscarin et al., 2020).

My general data processing approach and terminology align with established work in the field (Cancino-Chacón, 2018; Jeong et al., 2019a; Widmer, 2002). From the MIDI performance of a piece, I derive the `onset` in seconds that each note begins, the duration in seconds for which it is held (that I call `durSec`), and the `velocity` (dynamic level) of each note. As indicated in Figure 3.6, for any score note (shaded note in top panel), I use the PSBM to *estimate* where that note will begin and end in the performance (empty note in bottom panel), and compare these to the same properties *observed*

in the performance (shaded note in bottom panel), deriving the quantities d' and d''. Using the PSBM again, I can define the start- and end-time differences in units of quarter-note beats (empty note in the top panel), labelled x' and x'', generally, referred to as `ontimeAdj` and `offtimeAdj`. The synchronised audio-symbolic representations, including all the computed properties mentioned above, are stored in JSON format files – one file per MIDI performance.

Before training the algorithms (*Cue-Free Express* and *Cue-Free Express + Pedal*), I verify for each performed note in ASAP whether a score note of the same pitch exists within 0.1 sec of the expected position defined by the PSBM. I calculate a F_1 synchronisation score for each MIDI performance and only admit a performance into the dataset if its F_1-score exceeds .9. For each note in the score MIDI, the annotated beat locations serve to estimate the performance timing. A note of the same pitch found within 0.1 sec of this estimate counts as a true-positive (TP); where multiple candidates exist, the one with the smallest absolute time difference is selected. A missing note counts as a false-negative (FN), and any unmatched performance notes count as false-positives (FP). Additionally, I apply a further filtering of notes if either the `ontimeAdj` or `offtimeAdj` values were outside of the range $(-0.7, 0.7)$ sec. This results in a final dataset of 247,247 notes, adhering to an 80:10:10 train:test:validate split.

While the number of successfully synchronised notes is fewer than the total in the ASAP dataset—likely due to non-standard MIDI tempos—the volume of data is sufficient for this thesis. Further investigation is reserved for future work.

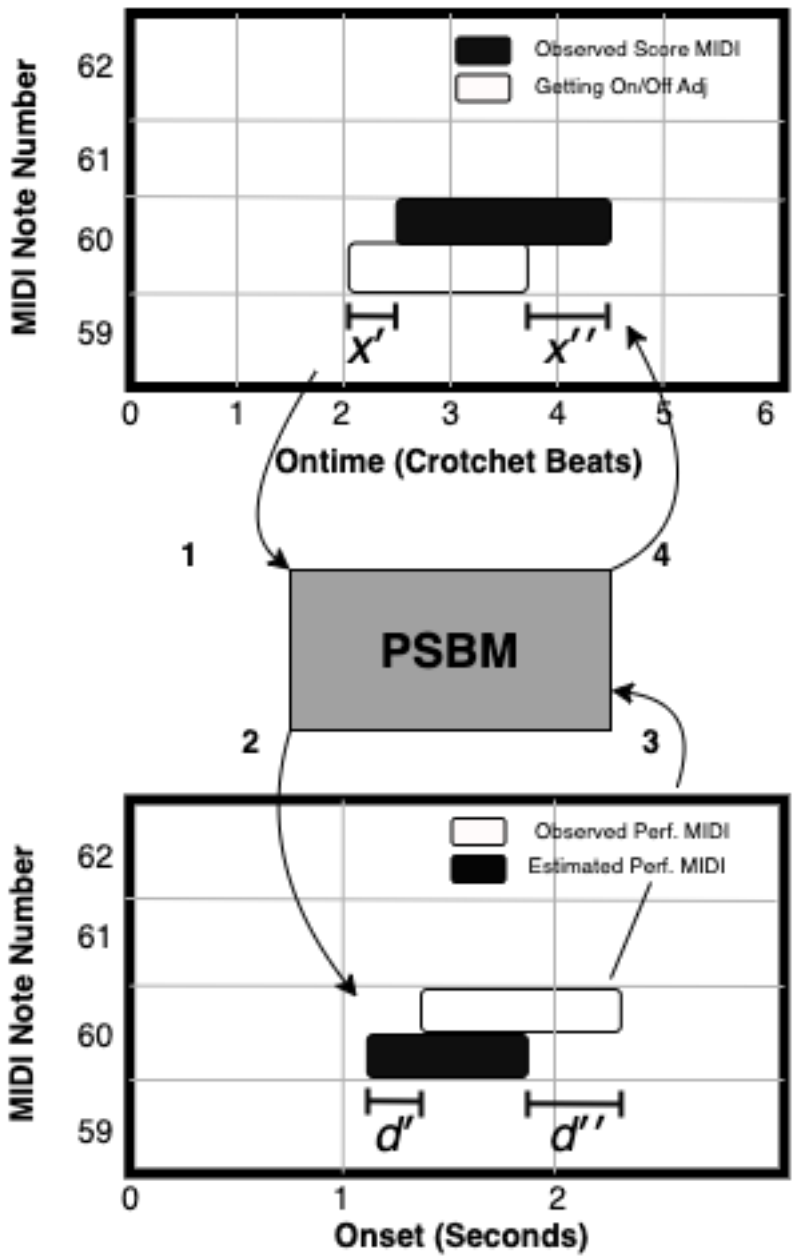

Figure 3.6: 1. A note's `ontime` is extracted from the score; 2. This is combined with the PSBM to estimate the `onset` of the corresponding note in a performance; 3. This estimated `onset` is compared to the actual time at which the note is performed; 4. The difference is combined with the PSBM again to give `ontimeAdj/offtimeAdj`. The values indicated in the figure are exaggerated for sake of readability.

3.8.2 Predicted Feature Analysis

In Chapter 6, I develop bespoke software solutions designed to: 1) quantitatively assess predicted attributes to provide deeper insight into the performance of the neural networks; and 2) elucidate the general distribution and characteristics of the data through statistical visualisations.

Rain Cloud Plots

In Chapter 6, I extend the analysis of the listening study data by extracting the velocity predictions from each computational system alongside those from professional human performances. These are visualised using raincloud plots to examine which model has best captured the generalised distribution of the data. As illustrated in Figure 3.7, the "cloud" (a half-violin plot) represents the probability density of the data, beneath which a boxplot displays the median, interquartile range, and whiskers (with diamonds indicating outliers). Finally, the "rain" consists of the raw data points displayed with vertical jitter to enhance visibility. While an example of this visualisation is provided in Figure 3.7, the specific findings are discussed in depth in Chapter 6, and similar methods are applied to extracted *ontimeAdj* and *offtimeAdj* predictions.

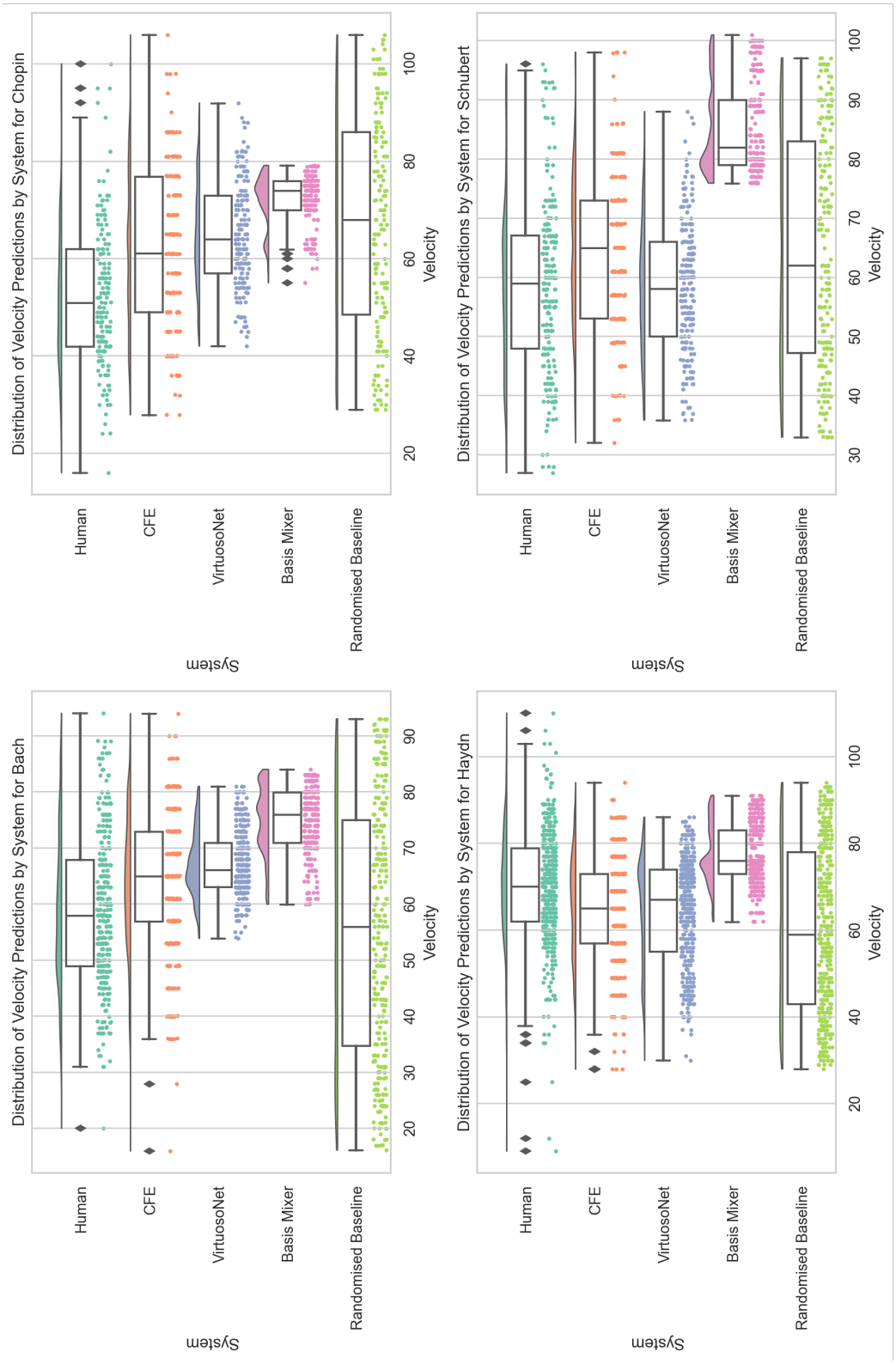

Figure 3.7: Raincloud plots of the distributions of dynamic predictions for each system across four stimuli in Listening Study 2.

3.8.3 Self-Attention Visualisation

In Chapter 6, to further interpret the internal representations of my attention-based models, I develop a script to calculate the self-attention weights for all tokens in the input sequence following the approach of Huang et al. (2018). The script itself extracts the self-attention maps from the Transformer encoder, where self-attention functions as a mechanism that assigns weights to rank the importance of the tokens contextually, and where the total scores for all tokens sum to 1 per layer. For example, an input sequence of 20 tokens yields 20 sets of 20 values summing to 1 for each network layer.

From these data, I can determine if the models are attending to repetitive structures in the music by identifying scores significantly higher than the mean score expected under an assumption of uniform allocation of attention (i.e. $1/n$, where n is the sequence length). To determine the threshold for which notes are being paid a high level of attention by the transformer, I retain a note only if the attention score is greater than three standard deviations above the mean:

$$k = 1/n + 3s \tag{3.3}$$

where s denotes the standard deviation of the attention scores within that layer.

For this analysis, I used a full version of Bach's *Fugue BWV 862* (Bach, 1722a) from the listening studies, as this piece was not included in the training data. A piano roll plot of most of the excerpt is provided in Figure 3.8, with the initial occurrence of the subject and its restatement (the answer/counter subject) highlighted. I focus on these occurrences because:

(1) they constitute thematically important, repetitive material that a Transformer model should ideally recognise; and (2) the restatement appears in the lower voice, which a human performer may choose to emphasise over contemporaneous notes in the upper voice.

In Figure 3.9, I visualise the attention scores for all notes in the subject's restatement that exceed the threshold described in Equation 3.3, connecting them to the notes to which they are paying attention. As with the predicted feature analysis, these results are discussed in depth in Chapter 6.

3.9 Chapter Summary

The work presented in this thesis follows a pragmatist mixed-methods approach, driven by real-world problems that affect the adoption of AI-driven technology within the video game industry. Consequently, the first half of this thesis focuses on collecting exploratory qualitative data using reflexive thematic analysis and a case study approach. These findings inform the development of an AI-driven solution to the practical challenge of rendering expression onto unseen, quantised MIDI data. The predictive performance of this system is evaluated through Bayes Factor analysis of listening study data, which is supplemented by quantitative analysis via bespoke software development. Finally, the algorithms are integrated into a video game for player testing to observe how such systems operate when deployed in a pseudo-real-world setting. The results of this final stage are subjected to a further Bayes Factor analysis to determine the system's impact on player experience.

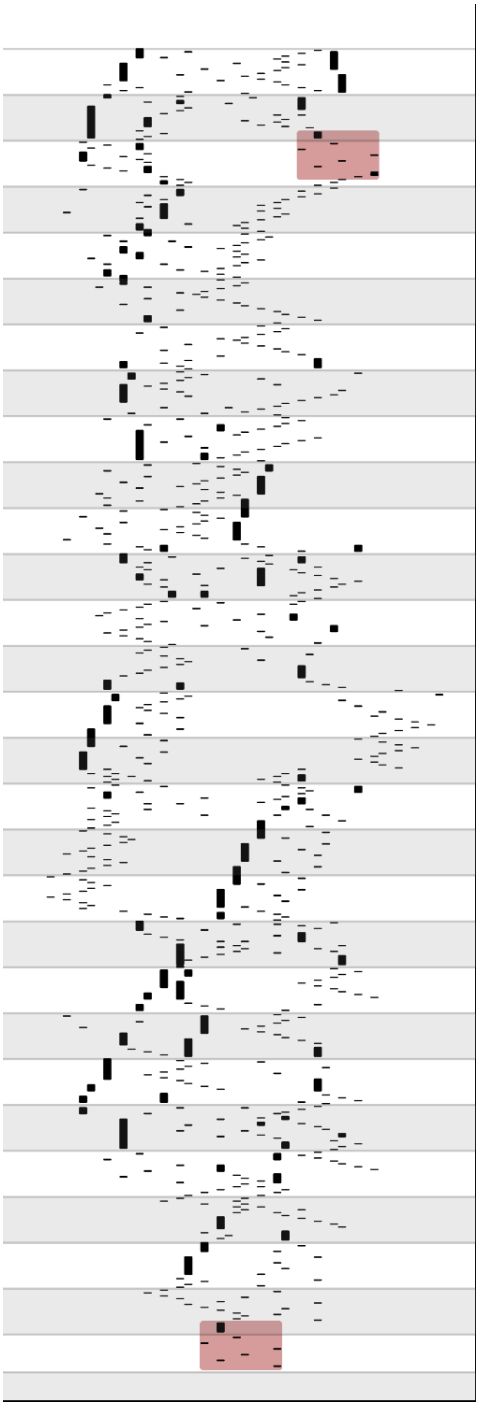

Figure 3.8: A visualisation of the piano roll highlighting the fugue subject and the countersubject in Bach's *Fugue BWV 862* (Bach, 1722a).

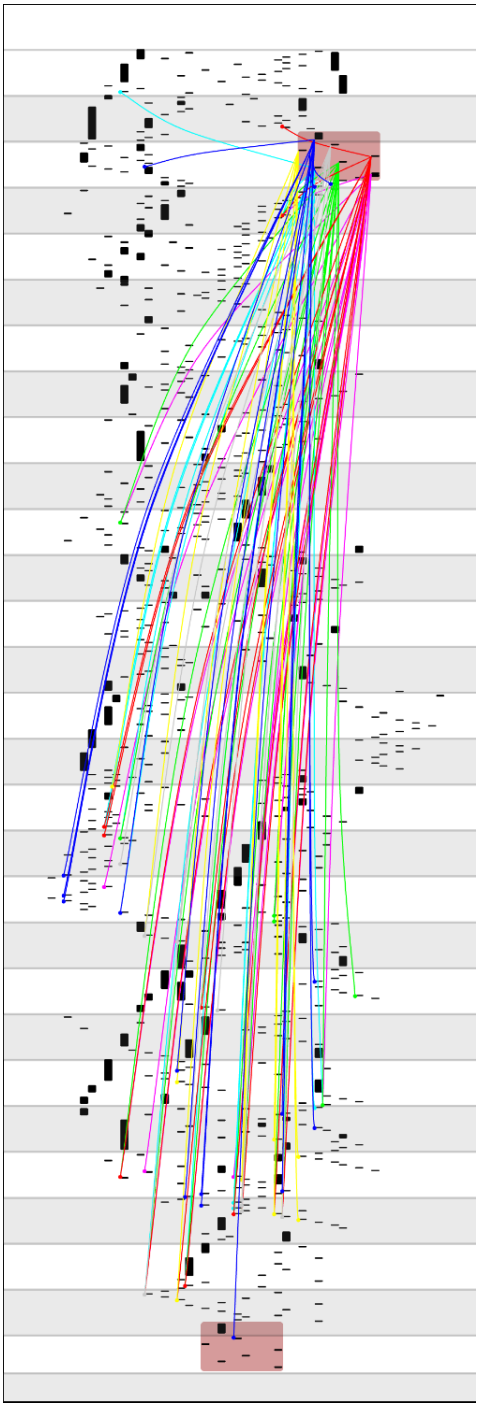

Figure 3.9: A visualisation of the attention that is being paid in layer 6 of the transformer velocity model to previous notes in Bach's *Fugue BWV 862* (Bach, 1722a), originating from the fugue subject in the left hand. Each line colour denotes a different origin note in the restatement in the left hand.

4 The Evolution of Music Systems in RPGs

4.1 Introduction

Chapter 2 motivated and contextualised the use of music generation, AI tools, and machine learning as a solution to listener fatigue in video games. Additionally, it introduced the terminology necessary to discuss and understand the complex behaviours of game music. However, before developing machine learning-driven solutions, it is essential to ground this research in practical implementation. Tracing the evolution of music implementation in video games contextualises how developers currently tackle listener fatigue and how these methods have matured. Furthermore, this comparison highlights specific scenarios in which music generation could effectively alleviate fatigue.

In this chapter, I provide a case study analysing the music implementation and design in an original PlayStation game (*Final Fantasy VII*) and its 2020 remake for PlayStation 4 (*Final Fantasy VII Remake*). I compare three aspects of each title's musical experience: the original soundtracks as a stand-alone experience; the battle music, representing an established problem area in video game music (Phillips, 2014; Armstrong, 2021; Tate, 2021; Worrall, 2021); and the first mission of each game in its entirety, acting as

a microcosm of the broader musical experience. Finally, I discuss how these changes correspond to contemporary developer and composer perspectives regarding listener fatigue, and highlight the potential for generative music to serve as a solution.

4.2 Contextualising Evolving Expectations

The premise that the evolution of music design in video games and the increased awareness of musical repetition reflect the shifting of player expectations was outlined in Chapter 2. Another industry trend that illustrates this evolution is the increasing prevalence of ports, remakes, remasters, and reboots in recent years. Although these terms frequently overlap, for the purposes of this thesis, I define them as follows:[1]

- Ports involve the migration of an existing product to a new platform, typically incorporating quality-of-life updates. For example, the original *Final Fantasy VII* (initially released on the PlayStation in 1997) has been ported to PC on several occasions; the 2013 iteration included features such as a toggle to disable random encounters.;

- Reboots are the restarting of a franchise from a new title, usually with a new story (for example, the reboot of the *Tomb Raider* (Core Design, 1996) franchise with *Tomb Raider* (Crystal Dynamics, 2013), which is the start of a new trilogy);

[1]While I employ these definitions, I acknowledge that they are not exhaustive, as some titles, such as the *Final Fantasy Pixel Remasters* (Square Enix, 2021), defy simple categorisation; in such cases, I contend that the nomenclature functions more as a marketing strategy than a precise technical classification.

- Remasters are the updating of the quality of an existing product by creating a new master version of a file through alteration or enhancement (for example, *Final Fantasy VIII Remastered* (Square Enix, 2019), which includes HD models and increased screen resolution compared to the original 1999 title);

- Remakes are when an original game is made new, in a different form (for example, *Final Fantasy VII Remake*, which is built in a completely different engine, for new platforms, as a vastly different experience to the original title, while only covering a portion of the original plot).

Each of these four categories of video games has become increasingly popular in recent years and attempts to capitalise on existing Intellectual Property. However, each of them remains loyal to the source material and brings the original experience closer to the expectations of a modern audience to varying degrees. These two aspects are explained separately, but visualised together in Figure 4.2.

The primary distinction between these categories is the degree of fidelity to the original source material. The 2013 reboot of *Tomb Raider*, for example, introduces a new narrative providing an origin for Lara Croft; consequently, it maintains low narrative continuity with the original titles. A port, by comparison, preserves the integrity of the original material more strictly than a reboot, as development focuses primarily on quality-of-life enhancements. Similarly, a remaster typically features upgrades to textures, resolutions, and models while remaining loyal to the original narrative. Remakes, however, often incorporate new story elements; they occupy a middle ground, offering less fidelity than remasters or ports, but preserving more than reboots.

The second factor is the degree to which the original material is updated to meet the expectations of a modern audience. In this context, expectations refer to the graphical and audio fidelity players anticipate, alongside certain internalised standards of the gameplay experience. For example, the 2013 reboot of *Tomb Raider* utilised a modern game engine and enhanced voice acting to align the source material with contemporary standards. A port, by comparison, typically retains original models and graphics, offering only minor quality-of-life changes that do little to bridge the gap toward modern expectations. Remasters occupy a middle ground; they often upscale models and textures, improve resolutions, and may include new versions of soundtracks. Consequently, they address modern expectations more effectively than ports but remain constrained compared to reboots. Finally, remakes represent the most significant overhaul, as they are often built from the ground up in a manner similar to reboots, thereby aligning most closely with the demands of current audiences.

In this chapter, I contend that the trend of remaking older video games not only supports the premise of evolving player expectations but also offers a unique methodological device for contextualising phenomena such as listener fatigue. Specifically, the divergence between versions of a title creates a framing device through which researchers can gain insight into the evolving priorities and concerns of developers.

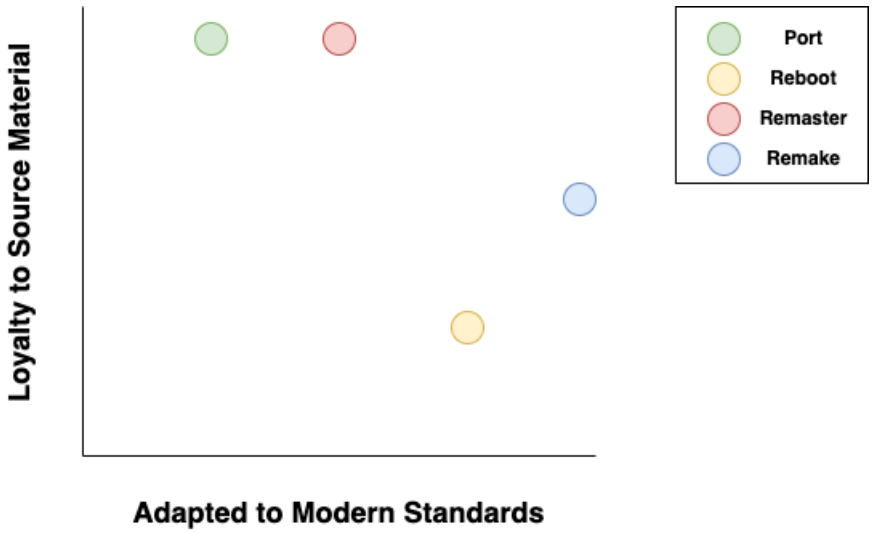

Figure 4.1: A graph, where the x axis denotes low to high adaptation of original material to meet modern game standards, and where the y axis denotes low to high loyalty to original material.

4.3 Methodology

Before comparing these titles, it is necessary to outline the music design of each game to contextualise the subsequent analysis. This involves detailing the techniques employed and situating them against the technology of their respective eras. In this chapter, I evaluate three aspects of the games: the duration, file formats, and track composition of the official soundtracks; the implementation of battle music; and a direct comparison of the opening mission of each game to serve as a microcosm of the macro-musical experience. This chapter concludes by outlining the limitations of existing techniques and highlighting the potential for procedural music systems to augment contemporary approaches, thereby justifying the research undertaken in later chapters.

4.3.1 Contextualising the Music Design of FFVII

Contextualising the music design of *Final Fantasy VII* encompasses understanding how and why the developers chose to store the music as MIDI, and a basic understanding of the game's proprietary engine. These factors are important to consider, as developer decisions affect the music implementation process within the game. This, in turn, ultimately influences how the composer writes and designs the music. Music design for early games on the PlayStation mostly centres on the use of hard cuts and quick fades between songs that fit specific situations or environments (Collins, 2008, p.69). This approach is exemplified in titles such as *Resident Evil* (Capcom, 1996) and *Spyro the Dragon* (Insomniac Games, 1998). In particular, Collins (2008, p.69) states that *Final Fantasy VII* is an exception to the use of recorded audio, relying on MIDI and the onboard synth chip to increase the number of tracks that could be included, explaining that music did not need to "loop endlessly".

Similar to other titles of the same era, *Final Fantasy VII* utilises a transitional music design approach, where the increased number of tracks plays a role in minimising prolonged exposure to individual tracks over the course of thirty to eighty hours of gameplay (see Table 4.1).

	Final Fantasy VII	**Final Fantasy VII Remake**
Main Story	36.5 hours	33 hours
Main Story + Extra	50 hours	41.5 hours
Completionist	82 hours	86 hours

Table 4.1: The total time taken to complete both Final Fantasy VII and Final Fantasy VII Remake, averaged from self-reported data taken from HowLongToBeat.com, taken on 3rd October 2023.

However, this approach offers limited real-time interactivity when compared to the procedural or parallel approaches, as transitions must occur at musically suitable moments to avoid detracting from the player experience. To contextualise this music design choice, the game engine must be understood. Details regarding the proprietary game engine used in Final Fantasy VII are sparse due to a lack of official translated materials; however, Walker (2013) has released an extensive technical breakdown of the game's internal architecture. Walker describes the engine as being split into six modules responsible for different aspects of the experience (Walker, 2013). The kernel acts as the overarching module responsible for managing the entire system, interfacing with the five remaining modules: world map, battle, field, menu, and mini-game. The kernel facilitates transitions between these five states, as illustrated in Figure 4.2 (adapted from Walker, 2013, p.9). Given an architecture predicated on shifting between distinct sub-systems, a solely transitional approach to music design is logical. Furthermore, with the game already stretching the contemporary hardware to its limits, the absence of procedural music design is entirely justified.

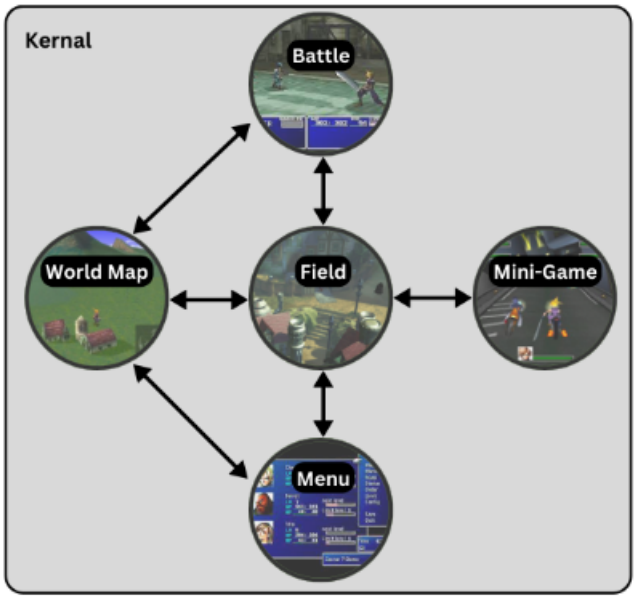

Figure 4.2: Final Fantasy VII's engine modules, and arrows demonstrating possible game state transitions.

4.3.2 Contextualising the Music Design of FFVIIR

Details about *Final Fantasy VII Remake* and its development are more readily available, and official interviews with the developers explain that *Final Fantasy VII Remake* is made in Unreal Engine. Additionally, the Criware audio middleware solution is used for the implementation of adaptive audio (Kamiyama, 2022; Taylor-Kent, 2020a). In contrast to its counterpart, the remake's engine allows the game to be set in a semi-open world with non-instanced real-time action combat. This means that the game's engine did not need to rely on individual modules, which would be transitioned between to run different aspects of the game, but that combat, for example, could take place during exploration, like in many contemporary RPGs. This more contemporary design allows the use of a combination of parallel, transitional, and ornamental music design approaches to "seamlessly transition in

response to in-game conditions" (Kamiyama, 2022). For example, the same melody line present in location music can be used in different situations (such as combat) by dynamically shifting as the game state changes, in turn responding to in-game parameters such as proximity to enemies and health, while retaining elements of the location theme (Taylor-Kent, 2020a,b). These practices are commonplace in contemporary titles and mitigate the repetition of static audio assets by varying when elements cross-fade in and out of a mix, thereby allowing music to respond in a closer approximation of real-time. Given the sophistication of the game engine and the adoption of audio middleware, the use of ornamental, transitional, and parallel forms is technically justified. Furthermore, while the inclusion of procedural or algorithmic elements was technically feasible, the rationale for excluding such techniques remains speculative. However, this omission aligns with the broader lack of procedural music adoption highlighted in Chapter 2.

4.4 Results

4.4.1 A Comparison of OSTs

The *Final Fantasy VII* OST contains approximately four and a half hours of music. Certain tracks are utilised throughout the game, such as "Let the Battles Begin" (Uematsu, 1997e), "Fanfare" (Uematsu, 1997f), and "Shinra Inc." (Uematsu, 1997g), while others are confined to specific narrative sections and, consequently, are heard less frequently. one third of the original game's story. Within the original soundtrack, only 23 tracks (totalling 01:11:00) actually correspond to the narrative arc covered by the *Remake*.

By comparison, *Final Fantasy VII Remake* features 157 tracks equating to approximately 08:35:00. This vast collection covers the main plot points, new narrative content, and roughly an hour of music accessible exclusively through in-game jukeboxes. This duration exceeds that of other contemporary JRPG titles, such as *Persona 5 Royal* (Atlus, 2019), *Tales of Arise* (Bandai Namco Studios, 2021), and *Yakuza: Like a Dragon* (Ryu Ga Gotoku Studio, 2020), which feature soundtracks of 05:32:58, 06:40:15, and 02:43:28 respectively.

The official soundtrack of a video game reflects its broader music design; while the OST presents a linear experience distinct from the dynamic nature of gameplay, the tracks remain representative of the in-game audio landscape. Specifically, the additional tracks included in the *Final Fantasy VII Remake* OST provide context as to which areas the developers identified as requiring musical expansion to avoid repetition. A primary example is the music accessed via in-game jukeboxes, which are typically located within expansive,

exploration-focused areas where players spend considerable time without engaging in combat. The inclusion of jukeboxes in these zones suggests that without combat arrangements to provide variety, the developers harboured concerns regarding listener fatigue. This mechanic is increasingly common in role-playing and open-world titles, as seen in *Grand Theft Auto V* (Rockstar North, 2013), *Final Fantasy XV* (Square Enix, 2016), and *Forza Horizon 5* (Playground Games, 2021). Such systems allow players to autonomously manage their own auditory fatigue within the diegesis of the game world. This is a feature entirely absent from the original 1997 experience.

Furthermore, the length of OSTs compared to the expected runtime of the titless provides some information as to the developers' view of the musical expectations of their players. While the difference in hosting medium (i.e. CD to Blu-ray), and the increased budget for music in video games can account for some of the difference in soundtrack size across console generations, the *Final Fantasy VII Remake* soundtrack **at over eight and a half hours** is much larger than many contemporary video game soundtracks (see the examples above). Although the official soundtrack does not demonstrate the distribution of the tracks within a video game, nor the amount of time a player can expect to be exposed to individual tracks throughout the whole story, it is possible to draw insights from this.

Table 4.1 shows the mean time required to complete the main story, the main story with extra content, and the requirements for a "completionist" run of both *Final Fantasy VII* and *Final Fantasy VII Remake*.[2] A comparison of the data in Table 4.1 demonstrates that the average playtime – at least

[2]This data represents the average of self-reported figures from approximately 8,000 players. Source: HowLongToBeat, "Final Fantasy VII" (2023) and "Final Fantasy VII Remake" (2023). Accessed 21 November 2023.

for the main story and one hundred percent completion – is roughly equivalent across both versions. This is noteworthy because, as previously stated, *Final Fantasy VII Remake* covers roughly one-third of the narrative found in the original title.[3] Despite this narrative compression, the *Remake* offers almost twice the amount of music for the same expected runtime. From this, I infer that the contemporary landscape of musical expectations has shifted significantly; the original asset library is now considered insufficient for covering up to 86 hours of gameplay, even though in 1997 a smaller musical footprint was deemed sufficient.

4.4.2 A Comparison of Battle Music

A distinction between the soundtracks of these titles is that there is an increased number of tracks for battle situations in *Final Fantasy VII Remake*. As outlined in Chapter 2, battle music in JRPGs is a notable problem area for repetition (Armstrong, 2021). Traditionally, in JRPGs, battle music and boss music, although filling similar roles, are considered separate. Boss music is generally more intense and less frequently encountered, whereas battle music is experienced with higher than average frequency (Thompson et al., 2020). The developers increasing the number and variety of tracks for standard battle encounters in *Final Fantasy VII Remake* is not surprising. In *Final Fantasy VII*, most non-boss battles are accompanied by the track "Let the Battles Begin" (Uematsu, 1997e). When considering runtime (see Table 4.1), the use of a single track throughout most combat encounters within the game

[3] I have rounded this figure to a third; while the second disc of the original *Final Fantasy VII* is longer than the first, the substantial additional content in *Final Fantasy VII Remake* complicates a direct linear comparison.

may eventually become fatiguing given the grinding nature of JRPGs (Armstrong, 2021; Gates, 2017). By comparison, *Final Fantasy VII Remake* has five different variations of "Let the Battles Begin" (Uematsu and Shima, 2020a,b,c,d; Uematsu, 2020) which are used in different chapters of the game, as well as additional variations for three boss fights: "Abzu" (Uematsu and Suzuki, 2020a), "Crab Warden" (Uematsu and Suzuki, 2020b), and "Specimen H0512" (Uematsu and Suzuki, 2020d). In addition, *Final Fantasy VII Remake* employs a parallel approach to incorporate battle variations of location music, alongside bespoke arrangements triggered when the player's health is critical.

The first chapter of *Final Fantasy VII Remake* is discussed in greater detail in the following subsection; however, it serves as a pertinent initial example. In this opening segment, combat encounters before entering the reactor are accompanied by "Bombing Mission" (Uematsu and Shima, 2020e), followed by "Let the Battles Begin! - Ex SOLDIER" (Uematsu and Shima, 2020b) which is featured during a single encounter. Subsequently, a battle arrangement of "Mako Reactor 1" (Uematsu and Shima, 2020f) is used for combat encounters inside the reactor ("Mako Reactor 1 - Battle Edit" (Uematsu and Shima, 2020g)). After defeating the boss, "Getaway" (Uematsu and Suzuki, 2020c) plays during all remaining combat encounters until the chapter's end. Including the boss track "Scorpion Sentinel" (Uematsu and Shima, 2020h) this results in five distinct pieces of battle music within the first hour of gameplay.

In the original *Final Fantasy VII*, "Bombing Mission" (Uematsu, 1997b) plays until the player enters the reactor, at which point "Let the Battles Be-

gin!" (Uematsu, 1997a) plays for every battle until the boss encounter. Thereafter, "Bombing Mission" (Uematsu, 1997b) plays during the boss fight and until the player escapes the reactor (even during combat encounters). The original title thus exhibits significantly less musical variety within combat encounters of the opening mission, establishing an expectation for players that standard battle music will persist throughout a play through. By comparison, *Final Fantasy VII* Remake treats "Let the Battle Begin! - Ex SOLDIER" (Uematsu and Shima, 2020b) as a singular thematic event, effectively a nod to returning fans, while contemporary parallel approaches provide a more dynamic shape to the musical experience.

In Chapter 2 of *Final Fantasy VII Remake*, there is a version of the original battle music titled "Let the Battles Begin! - Break Through!" (Uematsu and Shima, 2020a), which is used throughout most of the level after encountering Aerith. In this section, the developers foreshadow the Whispers' role as primary antagonists by playing a short segment of the original boss theme "Fight On!" (Uematsu, 1997d) (later renamed to "Those Who Fight Further") (Thompson et al., 2020). Throughout Chapter 2, the battle music is dynamically mixed using a parallel approach: a full arrangement accompanies combat, while a reduced arrangement comprising percussion and lower register instrumentation plays during sections of exploration between encounters (Thompson et al., 2020).

During combat, the leitmotif intentionally avoids a musical climax, informing the player that the conflict remains unresolved (Thompson et al., 2020). Conversely, the stripped-back exploration arrangements maintain the driving pace of the battle music while providing the player with necessary

breathing room. This dynamic mixing persists until the final encounter in Sector 8, where the full arrangement returns and the leitmotif finally reaches its climax. Although this encounter is not a formal boss fight, reaching this musical resolution bestows a sense of finality upon the chapter's conclusion.

The music design exhibited in both chapters utilises dynamic mixes that facilitate significantly faster response times than the strictly transitional approach of the original title. This emphasis on parallel mixing aligns with psychological research suggesting that faster, more complex rhythms contribute to the induction of emotional arousal through rhythmic entrainment (Trost and Vuilleumier, 2013, pp.213-225). While the original title induced these affective changes by transitioning to a discrete battle track, *Final Fantasy VII Remake* achieves this transition more subtly through the parallel mixing of stems.

Interviews with Square Enix developers confirm this was a conscious architectural decision (Walden, 2020; Taylor-Kent, 2020a). For example, Music Supervisor Keiji Kawamori explains that while transitions are occasionally employed for battle music, the audio team focused on incorporating active layers to location music using a parallel approach (Walden, 2020). Kawamori notes that location music often features three distinct variants: combat, non-combat (exploration), and crisis; the latter is triggered when the player's health is critically low to notify the player and elicit a heightened physiological state (Walden, 2020). This is expanded upon in a *PlayStation Official Magazine - UK* exposé, in which Co-Director Naoki Hamaguchi emphasises the technological necessity of "moving into the future," describing a system capable of maintaining a consistent melody line while ensuring the arrange-

ment becomes "more urgent and louder when you get into a fight" (Taylor-Kent, 2020a, p.11). These statements corroborate the focus on parallel music design in *Final Fantasy VII Remake* as a means of enhancing player immersion and arousal (Walden, 2020).

These examples of music design in the early missions of *Final Fantasy VII Remake* support the premise that developers are increasingly cognizant of listener fatigue as a significant challenge for battle music. Furthermore, this analysis demonstrates how contemporary music design can positively impact the player experience. The developers mitigate repetition by limiting the frequency with which a player is exposed to a static arrangement throughout the first few chapters of the game (typically representing 02:00:00–03:00:00 of playtime). Additionally, the implementation of a parallel approach not only enhances interactivity, thus offering a closer approximation of real-time musical response to events, but also varies the arrangement's texture to further diminish fatigue.

This has similarities to other modern action RPG titles such as *Rise of the Tomb Raider* (Crystal Dynamics, 2015), which combines parallel music design with procedural generation, where drums of a specific level of rhythmic intensity are generated, and the appropriate tonal material is cross-faded to sync up. This system supports play by informing the player musically of the state of play (exploration, combat, and stealth). Furthermore, while I argue that players have learnt to be more critical of the video games they play, as the medium evolves, and current technology becomes more familiar, it is important to understand that early experiences within a video game can play a role in reinforcing or subverting player expectations. By moving away from

the long-standing paradigm of battle encounters as separate musico-spatial events, towards dynamically responsive arrangements of location music, the developers reinforce the modern expectations of the player (Armstrong, 2021, p.1).

Beyond regular battle music, boss battle music also speaks to the evolving expectations of audiences. In the original *Final Fantasy VII*, only a few select bosses have multiple tracks across separate phases of the fight; in contemporary RPGs, however, it is not unexpected for the majority of main story bosses to have multiple phases accompanied by separate arrangements. Although this does not occur in every boss encounter in *Final Fantasy VII Remake*, "Scorpion Sentinel," (Uematsu and Shima, 2020h) the boss music from Chapter 1 is a pertinent example of phased boss music. The Scorpion Sentinel fight comprises three phases that synchronise with distinct gameplay stages. These sections are demarcated by cutscenes, while the music escalates in intensity to match the increasingly aggressive behaviour of the boss as its health drops below specific thresholds. This multi-stage approach to boss music is utilised throughout contemporary titles, such as the Ludwig, the Accursed/Holy Blade encounter in *Bloodborne* (FromSoftware, 2015); the fight against Baldur in *God of War* (Sony Santa Monica, 2018); and the Tiamat boss battle in *Stranger of Paradise: Final Fantasy Origin* (Team Ninja, 2022). Given the design decisions highlighted in this section and Hamaguchi's objective to adopt more advanced technology, it can be inferred that these musical choices are made to align *Final Fantasy VII Remake* with contemporary player expectations in modern RPGs (Taylor-Kent, 2020a).

4.4.3 A Comparison of Bombing Mission

Final Fantasy VII and *Final Fantasy VII Remake* both use multiple pieces of music to support their cinematic opening missions, using music design approaches that are better suited to their respective engines. For example, the proprietary engine of *Final Fantasy VII*, which transitions between modules to render various aspects of the game, uses a transitional approach to music design (Walker, 2013). By comparison, *Final Fantasy VII Remake* uses a combination of parallel, ornamental, and transitional music design in the opening mission. Figure 4.3 shows a timeline for events that take place in the opening mission and shows how each track that appears in each version synchronises with the gameplay sections. A smaller box within a larger box denotes the use of either another version of a piece of music (where layers are mixed in/out) or another piece of music (which is transitioned to and from) in the same section, while white spaces between coloured boxes signify a purposeful lack of music.

In *Final Fantasy VII*, the game begins with a cutscene over which "Opening - Bombing Mission" (Uematsu, 1997b) plays. As the cutscene ends, the "Bombing Mission" (Uematsu, 1997b) part of the track is introduced, aligning with the player taking control of Cloud. This piece loops throughout all of the exploration and combat encounters until the player enters the reactor. From this point, "Mako Reactor 1" (Uematsu, 1997c) plays during exploration but transitions to "Let the Battles Begin!" (Uematsu, 1997a) when the player enters random encounters. Upon finishing a combat encounter, "Victory Fanfare" (Uematsu, 1997f) plays, and then "Mako Reactor" (Uematsu, 1997c) resumes until the next battle. This continues until the player

plants the bomb at the core of the reactor (over which "Mako Reactor" (Uematsu, 1997c) plays). This action initiates the Scorpion Sentinel boss battle. "Bombing Mission" (Uematsu, 1997b) accompanies the boss battle and continues to play until either the player leaves the reactor within the allotted time post-boss battle, which ends the mission successfully, or the timer runs out, blowing up the reactor and putting the player into a game-over state, whereby they can choose to restart the mission.

Final Fantasy VII Remake, much like with the original, opens with an impressive cutscene. Over this cutscene, a live recorded and fully orchestrated version of the first half of the original "Opening - Bombing Mission" (Uematsu, 1997b) track plays which is called "Midgar, City of Mako" (Uematsu et al., 2020b). As with the original, when the cutscene ends and the player is given control over Cloud, "Bombing Mission" begins. After a few tutorial fights, the player gets a small expositional cutscene, and a sparser arrangement of "Bombing Mission" (Uematsu and Shima, 2020e) plays, which makes room in the mix for dialogue. When the player moves to continue the mission, the full arrangement of "Bombing Mission" (Uematsu and Shima, 2020e) fades back in and continues to play until the player enters the reactor.

Unlike in the original title, there is a cutscene and an extremely brief segment of gameplay with no accompanying music before the player is rewarded with a modern version of the original battle music "Let the Battles Begin! - Ex SOLDIER" (Uematsu and Shima, 2020b) during a single combat encounter, before fading back to silence before another cutscene. In this cutscene, the player sees flashbacks of Nibelheim and hears a short section of "Anxiety" (Uematsu and Nakamura, 2020), musically foreshadowing that

there is something wrong with Cloud's memories. After another moment of musical silence, the location theme "Mako Reactor 1" (Uematsu and Shima, 2020f) is introduced and loops as the player explores the reactor. While exploring the reactor, during combat, the music cross-fades into a more intense variant titled "Mako Reactor 1 - Battle Edit" (Uematsu and Shima, 2020g). When combat is over, the battle variant cross-fades back into the original exploration variant. This continues until the player plants the bomb in the reactor, and Cloud has a vision and sees Sephiroth, while a short piano arrangement of "Trail of Blood" (Uematsu and Makino, 2020) plays, musically foreshadowing later story beats and danger.

After the short cutscene, the Scorpion Sentinel boss battle begins, accompanied by "Scorpion Sentinel" (Uematsu and Shima, 2020h). Unlike with the original game, this piece of music is specifically designed for the boss fight and responds dynamically to the boss's health gauge. When the boss's health is below the thresholds of 75% and 50%, short transitional cutscenes play, and after which new sections of the "Scorpion Sentinel" (Uematsu and Shima, 2020h) piece play with different arrangements. After defeating the boss, the situational music "Getaway" (Uematsu and Suzuki, 2020c) plays to match the frantic pace of escaping from a reactor in meltdown. During this section, the situation music is prioritised over the previous battle music during any combat encounters. This prioritisation continues until the end of the mission. As the player makes their way to the exit, a cutscene is triggered by the player moving up a ladder, and a cutscene plays where Cloud saves Jessie while a more rhythmic, staccato section of "Getaway" (Uematsu and Suzuki, 2020c) plays, heightening the excitement of the player. After this

section, the normal "Getaway" (Uematsu and Suzuki, 2020c) loop resumes using a more transitional approach rather than parallel cross-fades. During the player's escape, another cutscene and a new section of "Getaway" (Uematsu and Suzuki, 2020c) are triggered when encountering enemies to increase tension, after which tension is mitigated by Barrett singing a rendition of the "Victory Fanfare" (Uematsu and Ishikawa, 2020) over the top of "Getaway" (Uematsu and Suzuki, 2020c), demonstrating the ornamental approach to music design. The music system then returns to the regular version of "Getaway" (Uematsu and Suzuki, 2020c), which continues through to the end of the chapter. Upon leaving the reactor, a final celebratory section of "Getaway" (Uematsu and Suzuki, 2020c) is triggered, followed by a short cutscene featuring "Shinra's Theme" (Uematsu et al., 2020a), which musically foreshadows the militaristic threat of Shinra. The player exits the reactor as it explodes in earnest, as "Getaway" (Uematsu and Suzuki, 2020c) plays out the chapter's end.

Final Fantasy VII Remake using a combination of parallel, transitional, and ornamental approaches, does not innovate but remains true to contemporary practices. However, there are some noteworthy observations that are present when comparing this experience to the original. As Figure 4.3 demonstrates, during the course of the first mission in *Final Fantasy VII*, the player will encounter four musical pieces, namely: "Opening - Bombing Mission" (Uematsu, 1997b), "Mako Reactor" (Uematsu, 1997c), "Let the Battles Begin!" (Uematsu, 1997a), and "Victory Fanfare" (Uematsu, 1997f). By comparison, in *Final Fantasy VII Remake* players experience six full pieces of music, two additional arrangements of existing pieces, four short

pieces that musically foreshadow later story beats, and three short segments of musical silence. I argue that these opening segments can be treated as microcosms of the larger macro-musical experience, where design choices in the first hour of play reflect the music design of the whole game. The choice to include additional music in this mission plays a significant role in reducing the likelihood of listener fatigue in the first mission, while also affecting players' expectations going forward.

Similarly, the inclusion of alternative edits of "Mako Reactor 1" and "Bombing Mission" imply an importance to arrangement/shape in maintaining theme, while also alleviating musical repetition. This is implied, as although the main melodies for specific areas continue across arrangements, the shape of the arrangement changes, introducing musical variety, and reducing the likelihood of listener fatigue. Furthermore, the use of the four instances of musical foreshadowing ("Victory Fanfare" (Uematsu and Ishikawa, 2020), "Shinra's Theme" (Uematsu et al., 2020a), "Trails of Blood" (Uematsu and Makino, 2020), and "Anxiety" (Uematsu and Nakamura, 2020)) not only give nods to later story beats; they act as palate cleansers. These tracks are encountered after longer segments of gameplay, where the player will have been exposed to the same musical content repeatedly (albeit in differing arrangements). These shorter tracks break up the experience musically, and the three instances of musical silence play a similar role in mitigating.

By comparison, in *Final Fantasy VII*'s opening mission, while the player hears two different pieces during combat, they are quickly introduced to the paradigm where an individual piece of battle music will make up a substantial portion of the next thirty-six to eighty-two hours (see Table 4.1).

Beyond the effects of these music design choices on listener fatigue, the choices made here reflect how developers view changes in player expectations. The music design choices in *Final Fantasy VII Remake* reflect those seen in modern action RPG titles. This supports my suggestion that *Final Fantasy VII Remake*'s developers are not only moving the game's experience, but also the music design closer in line with contemporary titles. They are doing this by moving away from the more repetitive music design of the original, while looking to provide an experience closer in quality and style to what is expected by modern audiences.

4.5 Discussion

How music design for video games and the expectations of players regarding music design have evolved in the last twenty-three years are of central interest to researchers of video game audio and procedural music, as well as to video game composers and music designers in the video game industry. To my knowledge, before the paper that informs this chapter, little work has been done to analyse the evolution of music design through the comparative lens of an original video game and its remake[4] (Glynn, 2023; Thompson et al., 2020). In this chapter, I discuss the engines used for Final Fantasy VII and Final Fantasy VII Remake and use this knowledge to contextualise the differences in music design, specifically when it comes to battle music, a notable problem area for listener fatigue in RPGs.

[4]There has, however, been work comparing the same games from a music composition/experience perspective.

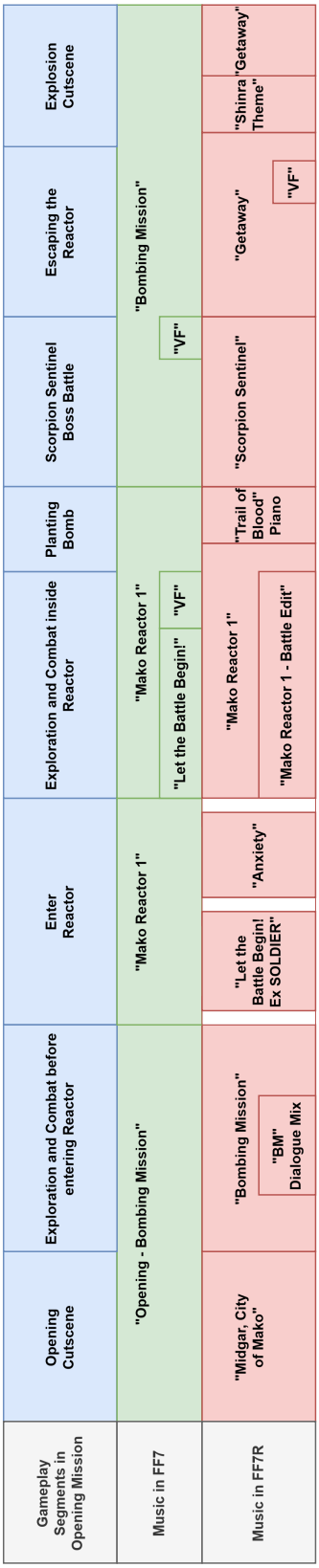

Figure 4.3: A timeline of the music in the first mission of both Final Fantasy VII and Final Fantasy VII Remake, and how this music aligns with sections of gameplay. VF denotes Victory Fanfare, and empty slots denote silence.

I now discuss my findings with relevance to my first research objective: "**To evaluate the evolution of music design and implementation in video games, and establish whether there is space for generative music in video games.**"

Concerning the first part of my research objective, I find that music design and implementation have changed significantly in the twenty-three years between the release of these titles. Little can be learned from the framing device of a remade video game to inform us about how the expectations of actual players have changed. However, the evidence presented demonstrates that the player experience for *Final Fantasy VII* and *Final Fantasy VII Remake* differ significantly, and the video game developers have moved not only the gameplay experience, but also the music design of the game closer to that of contemporary titles. For example, the inclusion of in-game systems that allow players to control music playback (i.e. jukeboxes) and contemporary approaches to music design reflect that developers think that players' expectations and tastes have evolved in the intervening 23 years. Furthermore, in the opening missions of *Final Fantasy VII Remake*, the full official soundtrack and battle music in the first two chapters of the game, I find that the phenomenon of listener fatigue has *likely* been carefully considered by the developers. This is supported by the inclusion of elements to reduce listener fatigue, such as: additional pieces of music; new arrangements of location music for battle and crisis game states; shorter fragments of musical foreshadowing to act as breaks in longer repetitive segments; and sections of varying intensity to match enemy health in boss battles. Furthermore, similarly to how *Final Fantasy VII* has a larger-than-average soundtrack for an

original PlayStation game, *Final Fantasy VII Remake* has a soundtrack that is far larger than the average soundtrack for a video game in 2020. This reflects the idea that the original number of tracks is not sufficient to support a contemporary gameplay experience. While in the original title, the addition of a larger soundtrack seemed enough from a developer perspective to combat music repetition, the developers, twenty-three years later, have put much more consideration into dealing with this issue of listener fatigue. Furthermore, they have done this while also bringing the game experience closer to that of contemporary RPG titles.

With regard to the second half of my research objective, while solutions may not exist to fully deal with the problem of musical repetition, as technology evolves, I find that composers should be prepared to work alongside it to make the best experience possible for the players. The musical landscape of video games has clearly changed in the intervening twenty-three years, and the adoption of modern music design techniques has made for a stronger experience in *Final Fantasy VII Remake*. As games continue to grow in length, it may become increasingly necessary to adopt procedural, AI-powered, or even generative approaches to further mitigate musical repetition. The developers of *Final Fantasy VII Remake* have determined that it is necessary to include jukeboxes to allow players to remedy their own fatigue within the diegesis of the game world, even though a much larger than normal soundtrack has been utilised. This indicates that there is a room for generative or procedural music techniques in modern video games, as while *Final Fantasy VII Remake* does not use any procedural music techniques, such techniques could enable developers to avoid licensing real-world music for use in car

radios or jukeboxes, and also avoid having to build in these types of systems if generative elements could be used to augment existing human-composed music. I find in this analysis that the inclusion of such technology could lead to a stronger experience in future remakes, especially if video games continue to move toward potentially "endless" live service models (Wilson, 2019; Worrall and Collins, 2023).

4.6 Chapter Summary

Consequently, through the comparison of the original title and its remake, I demonstrate that despite technological evolutions to cope with the increasing scope of modern titles, there is distinct room for generative music systems and AI-powered music tools in video games. In doing so, I address the first research objective of this thesis: "**To evaluate the evolution of music design and implementation in video games, and establish whether there is space for generative music in video games.**"

One identified use case for generative music solutions in video games is during prolonged segments of exploration, such as the jukebox feature in *Final Fantasy VII Remake*, or similar systems such as car stereos in *Final Fantasy XV* and *Grand Theft Auto V*. A second example is battle music, which players encounter with considerably high frequency. Furthermore, I observe that there is no adoption of procedural techniques in *Final Fantasy VII Remake*—a finding that reflects broader industry trends. The adoption rates for procedural music remain low as of 2025, and prior to the work in this thesis, there was a lack of empirical research aimed at understanding the underlying causes. This leads directly to the next chapter, in which I investigate the factors affecting why procedural and artificially intelligent music technology exhibit lower adoption rates than visual or mechanics-focused procedural systems. This transition addresses the second objective of this thesis: "**To evaluate and potentially impact the workflow, programming knowledge, and opinions of video game composers to inform tool design and use cases of procedural music generation.**"

5

Investigating Factors Affecting the Adoption of Intelligent Music Technology

5.1 Introduction

Chapter 2 established the problem of listener fatigue in games and outlined that procedural music generation has not been widely adopted, even though it offers a potential solution. Chapter 4 examined an example of a modern top-selling role-playing video game and highlighted gaps in the music system where generative or AI-powered music could be useful. However, before developing machine learning-driven music tools for video games, it is imperative to understand why there is a lack of adoption regarding procedural music technology and how this may apply to artificially intelligent music technology (AIMT).

As discussed in Chapter 2, video games have long been considered a medium that is uniquely suited to music generation, due to their interactive, unpredictable, and non-linear nature (Collins, 2009). However, in contrast to visual procedural content generation (PCG), and despite procedural music generation (PMG) being proposed as a means to empower composers through assistive use cases (Plut and Pasquier, 2020), the generation of music and sound has seen a much more limited scope of application within the video

game industry. This is notable, as PCG is already utilised to overcome repetition-based fatigue associated with visual assets in video games (Greuter and Nash, 2014). Furthermore, machine learning approaches have been used to speed up creative workflows for animators (Bosque, 2022; Miladi, 2024). Meanwhile, in music for video games, it is common for ∼4 hours of music to be repeated across ∼100 hours of gameplay (Plut and Pasquier, 2020; Worrall, 2021).

AI-driven music technologies ("music AI") outside of video games are rapidly improving in their abilities to perform creative tasks, such as composition (Collins and Laney, 2017; Huang et al., 2018), musical in-filling (Guo et al., 2022), expressive rendering (Jeong et al., 2019a; Worrall et al., 2024), mastering (Sterne and Razlogova, 2019), and mixing (Moffat, 2021). Furthermore, music AI has been shown to assist novice music creators with creative tasks (Louie et al., 2020). The breadth of applications highlights numerous ways music AI could support professional composers. Yet, with the exception of a few largely rule-based examples and specific implementations such as the *Dynamic Percussion System* (Brown, 2019; Lamperski, 2015), AI technology remains largely unadopted in the video games industry.

Although existing literature suggests theories as to why procedural music systems have not been widely adopted, there is a lack of empirical evidence. Industry practitioners and researchers highlight several limitations inhibiting procedural music/sound in video games, including: the need for robust timing systems and more audio programmers (Weir et al., 2021); the resource intensiveness of modern titles (Plut and Pasquier, 2020); inconsistency in generative quality (Weir, 2017); and a lack of human nuance or expression

(Worrall et al., 2024). Furthermore, discourses within online forums and social media reveal a narrative of aversion to creative-generative AI, citing ethical and legal concerns (Collins and Laney, 2017; Flick and Worrall, 2022; Yin et al., 2021). However, while recent co-creative AI literature has assessed how audio professionals view AIMT in podcasting, sound design, and general composition (Rime et al., 2023; Kamath et al., 2024; Knotts and Collins, 2020), there has been a notable absence of empirical research specifically investigating video game music professionals.

In this chapter, I tackle my second research objective "**To evaluate and potentially impact the workflow, programming knowledge, and opinions of video game composers to inform tool design and use cases of procedural music generation.**" To address this, I establish a foundation of empirical data by interviewing eleven professional video game composers to interrogate their thoughts on AIMT (used hereafter to describe both PMG and music AI collectively). This approach aims to elucidate the relationship between emerging AIMT and professional creatives, while exploring the potential impacts of the tools introduced in later chapters. Additionally, I present the results of a reflexive thematic analysis (RTA, (Braun and Clarke, 2006, 2013)) to identify recurring themes and sub-themes, addressing the following two questions:

1 How do composers perceive AIMT, and how does this inform future research?

2 What can I learn about the professional workflow and technical knowledge to inform future research and design of intelligent music tools?

These questions are selected for their direct alignment with the second research objective of this thesis. The findings presented here have large-scale impacts on the remaining chapters of this thesis and provide essential context for the continued development of my work.

5.2 Background

Chapter 2 examined the limited adoption of procedural music within video games and identified its potential utility in mitigating listener fatigue. However, to provide a robust analytical framework for the findings of this study, it is necessary to examine how professional perspectives align with established theoretical models of technology acceptance. By grounding the subsequent analysis in these principles, we can better categorise the specific barriers to innovation within the video game music industry.

5.2.1 Technology Acceptance

Research on technology acceptance originated the 1980s with the Technology Acceptance Model (TAM), which posits that two primary factors in technology acceptance are perceived ease of use and perceived usefulness - although external variables such as social influence also play a role (Davis, 1989). This was later expanded in subsequent literature: one study found that specific determinants affecting perceived ease of use evolve over time, specifically control, intrinsic motivation, and emotion (Venkatesh, 2000); another proposed an updated model (TAM2) demonstrating that social influence processes (subjective norm, voluntariness of use and image), and cognitive in-

strumental processes (job relevance, output quality, result demonstrability and perceived ease of use) significantly influence user acceptance (Venkatesh and Davis, 2000).

A unified model of technology acceptance (UMTA) has also been proposed and empirically validated, synthesising eight competing technology acceptance models (Venkatesh et al., 2003). This research identifies four constructs that directly determine user acceptance and usage behaviour: performance expectancy, effort expectancy, social influence and facilitating conditions, where factors such as gender, age, experience and voluntariness of use could impact each of the four constructs to some degree (Venkatesh et al., 2003). Consequently, it is understood that technology is more likely to be accepted if it is intuitive, performs reliably, and aligns with positive public opinion.

5.2.2 Contextualising PCG Acceptance

PCG is a well-documented area of research, where rules-based systems are used to generate worlds (McKendrick, 2017; Carrier, 2018), geometry (Sander van Rossen, 2020), world history (Grinblat, 2018), and determine resource/enemy placement throughout generative levels (Bucklew and Grinblat, 2019). PCG has seen extensive practical applications in the video game industry and is a particularly popular technique for independent studios, serving as a "technical strategy for generating content despite limited production resources" (D'Errico, 2015, p.197). PCG has also seen a wide range of applications in triple-A (AAA) development, as demonstrated by the generation of dungeons in *Bloodborne*, weapons in *Borderlands 3* and the world in *Far Cry 5*.

In these examples, PCG is employed to accelerate development and reduce costs (Amato, 2017; Barriga, 2019; Carrier, 2018) while maintaining high-fidelity quality (Shaker et al., 2016; Watkins, 2016; Amato, 2017). The application of PCG has led to unending video game experiences for players and is a cornerstone of live-service games (LSG) (Wilson, 2019). LSGs are characterised by content delivered via a continuous revenue model over a game's lifespan, which necessitates a constant supply of new assets, sometimes including music. Examples of LSGs include massively multiplayer online video games such as *Destiny 2* (Bungie, 2017), *World of Warcraft* (Blizzard Entertainment, 2004), and *Final Fantasy XIV* (Square Enix, 2010), as well as multiplayer online battle arenas such as *League of Legends* (Riot Games, 2009).

Although widely adopted, PCG is often critiqued for a perceived lack of diversity and polish in its output; this phenomenon is famously described by Compton (2016) as the production of "10,000 bland bowls of porridge." While PCG offers distinct creative and economic advantages, a lack of generative variety can lead to negative player appraisal and adversely affect retention, particularly when environments become monotonous to traverse. This directly relates to the social influence factor within technology acceptance models, whereby the reception of the player-base can dictate the perceived social acceptability of a technology, subsequently influencing developer adoption through public discourse.

Considering PCG application in relation to the unified model of technology acceptance (Venkatesh et al., 2003), we can observe that PCG tools provide substantial value to studios (suggesting high performance expectancy).

While the development of proprietary internal tools represents a significant initial investment (high effort expectancy), the ubiquity of these techniques and the wealth of available learning resources mitigate this hurdle. Furthermore, these tools ultimately reduce the effort expectancy for level designers—the primary end-users. As a result, PCG has achieved widespread adoption despite aesthetic concerns regarding generative variety. This normalisation of the technology within the industry has minimised negative social pressures from peers, ensuring that PCG remains a standard professional technology provided the output maintains a baseline threshold of fidelity (Venkatesh et al., 2003).

5.2.3 Contextualising PMG Acceptance

The impact and benefits of Procedural Music Generation (PMG) are similar to those of visual PCG, allowing video game developers to computationally generate, perform or transform music in real-time settings within video games (Wooller et al., 2005; Plut and Pasquier, 2020). Given the notably restrictive data allowances for audio in video games (Marks, 2008; Wilde, 2014; Stevens and Raybould, 2015), and the increasing demand for variation (Plut and Pasquier, 2020; Worrall, 2021) and interactivity (Phillips, 2014; Stevens and Raybould, 2015), there is a compelling theoretical justification for the adoption of PMG in the video game industry.

Successful industry implementations of PMG often utilise approaches that do not require composers to significantly alter their existing workflows or engage in technical tasks beyond their standard remit (Brown, 2019; Lamperski, 2015; Weir, 2017), thereby reducing effort expectancy (Venkatesh et al.,

2003). As an example of successful symbolic music generation in industry, the Dynamic Percussion System used in *Rise of the Tomb Raider* employs machine learning to generate real-time drum patterns from symbolic MIDI data provided by the composer. By requiring only standard assets, the system avoids adding cumbersome steps to the creative process, which significantly increases the likelihood of professional acceptance (Brown, 2019; Lamperski, 2015). Furthermore, the real-time resampling of this generative percussion ensures that the audio fidelity remains indistinguishable from pre-recorded assets—a crucial factor in satisfying the performance expectancy requirements of high-end titles.

Similarly, for *No Man's Sky* (Hello Games, 2016), Paul Weir developed the audio domain music generation tool (PULSE). He intentionally allowed the contributing band, 65daysofstatic, to compose an album using their traditional methods, provided they delivered the material in stems for generative manipulation (Weir, 2017). This strategy limits the impact of the technology on the band and pushes the work of adapting material post-ideation onto the developer, or in this case, Weir himself. Weir, however, in advocating for the tool, increases the likelihood that the band will work with it by opting to have their creative process unaltered. This means that the input for the generative model is of a high audio fidelity/quality as it was created using typical practices, which in turn increases the output quality and improves technology acceptance.

In cases where specialist resources are unavailable, developers have leveraged open-source algorithms, such as the jazz-solo generator Impro-visor (Keller et al., 2007), to bridge the gap and overcome the lack of access

to machine learning specialists in independent studios (Abel, 2021). This reduces effort expectancy in another dimension by using functional open-source code that already has a GUI rather than being left at the command line (Venkatesh et al., 2003). Developing an algorithm from scratch would, of course, be a much higher effort expectancy, but the easy-to-use tool lowered the barrier even further for the developers, increasing the likelihood of adoption. While Plut & Pasquier outline a more comprehensive list of PMG systems in video games, it is clear that compared to PCG, PMG has seen much less application in the video game industry, giving the impression of less acceptance (Plut and Pasquier, 2020).

Outside of the industry, academic research focused on automatic music generation for video games has seen more interest in recent years, often focussing on rules-based approaches or less computationally taxing approaches such as Markov chains to decrease latency and computational costs in real time (Engels et al., 2015; Shapiro and Huber, 2021), which in turn increases performance expectancy and lowers effort expectancy compared to deep learning approaches (Venkatesh et al., 2003). However, there is a disconnect between industry and academia, so while research has been addressing game-specific use cases such as transition generation (Cutajar, 2020), generating music that adapts to gameplay (Hutchings and McCormack, 2020), using Gaussian mixtures to control the melodic shape in generative music (Xiang and Guo, 2021) or generating music to match non-player character relationships (Washburn and Khosmood, 2020), PMG is largely ignored in industry.

5.2.4 PMG Acceptance in Video Games

There are a variety of reasons that researchers and practitioners have identified as potential reasons for the lack of procedural music tools in the video game industry. The first is a lack of experienced developers to build or support PMG or machine learning-driven music tools in the video game industry (Weir et al., 2021), especially when video games have real-time computations and resources to consider (Plut and Pasquier, 2020). This means that potential users (video game developers) have a lower performance expectation and a higher effort expectation, as there is no support for using these tools and using them without careful consideration can damage the gameplay experience (Venkatesh et al., 2003). Compare this to non-video games related music AI tools, for instance, such as the Google Magenta Suite (Roberts et al., 2019) and in-painting tools for Ableton (Hadjeres, 2021; Guo et al., 2022), which have more support for the development of tools and also do not have to consider real-time video game rendering and frame drops.

Another reason why the adoption of AIMT may be low is the inconsistency in the quality of the generative output (Compton, 2016; Weir, 2017) or the lack of human nuance in computer-generated music (Worrall et al., 2024). This is supported by anecdotal evidence, such as composer discussions on social media, which outline the low quality and inconsistency of generative output as a factor against using these tools. This resonates with the unified model of technology acceptance, as low quality, non-nuanced output designates low performance expectancy, which will lower acceptance levels. A final reason that could potentially be lessening the adoption of this technology is the ethical and legal concerns around creative AI (including music

AI), where generative art tools (e.g., Midjourney) and LLM (e.g., ChatGPT) have caused a stir in the creative community, for a variety of reasons based on copyright infringement concerns (Collins and Laney, 2017; Yin et al., 2021), and concerns from creatives regarding misrepresentation and ownership (Flick and Worrall, 2022). These tie into the social influence aspect of the unified model of technology acceptance, as discourse surrounding generative art/music is volatile, and with such discourse comes social pressures to abstain from their use. These factors could explain the lack of adoption of AIMT in video games, but there is a clear lack of empirical evidence that outlines how composers (and creatives) feel about these developments. Without looking to understand the end users of these tools, it is unlikely that they will be adopted. Highlighting the thoughts and feelings of such users, as I go on to do, could shape future tool design for increased technology acceptance and fill this gap in the literature.

5.2.5 AIMT Acceptance

Research in the field of AI co-creation (the study of how humans and AI interact together) has shown that AI tools can support novices during composing tasks (Louie et al., 2020); however, research into the acceptance of AIMT for music professionals has yet to be fully explored.

Tsiros & Palladini investigate AI-assisted music production and propose a framework for how to design AI tools to be human-centric, noting the importance of reducing the following risks for users: AI making sub-optimal decisions; AI affecting engineer authority and control; AI forcing extreme change on existing workflow (Tsiros and Palladini, 2020). Meanwhile, Vanka

et al. (2023) studies the opinions and thoughts of mixing engineers about how they use AI mixing tools and finds that amateurs, semi-professionals and professionals use intelligent mixing tools for different purposes. They find that professionals are in favour of AI mixing tools and that they use these tools as a way to speed up their workflow and experiment creatively. However, Vanka et al. (2023) state that this is ultimately due to individual differences and that it is important that these tools integrate seamlessly with existing workflows, have a fine balance of control and automation, and become context-sensitive (as generic output tends not to be of interest to users) (Vanka et al., 2023). The above begins to provide a baseline for understanding AIMT acceptance, but does not tackle the realm of video game music, nor of professional game composers, which this chapter seeks to address. Unfortunately, until this thesis, there has been no research in this particular area for video game composers.

Returning to the task of music generation and moving beyond academia, while the video game industry is considered the ideal platform for the inclusion of music generation (Collins, 2008), most growth in generative music technology has occurred outside of the video game industry, where higher rates of adoption have occurred. This is particularly evident as digital audio workstation developers (such as Apple (Logic X), Steinberg (Cubase 14) and Avid (Pro Tools)) have begun to implement AI-based music technology in an attempt to remain competitive (Apple Newsroom, 2024; Avid, 2023). The analysis of market trends supports this, highlighting that 70% of digital audio workstations will include some type of AI-driven music technology in the near future (Fortune Business Insights, 2024).

Not all examples of production grade tools are created by these companies; however, notable examples of the application of research as tools for music making include in-painting algorithms for Digital Audio Workstations such as Ableton (Hadjeres and Crestel, 2021; Guo et al., 2022) and Cubase (Tchemeube et al., 2023), which levy enhanced transformer models such as the Bidirectional Encoder Representations from Transformers (BERT) to create new layers in a piece, conditioning generative output on existing layers. Other examples of stand-alone applications include Magenta Suite, a set of free symbolic music tools that use a variety of architectures such as RNN and LSTM, created by Google (Roberts et al., 2019), and Text2Sample,[1], an audio domain tool from Samplab that generates a sound file based on a provided text prompt.

Furthermore, generative music technology has been developed by start-ups offering generative solutions across different sectors. For example, AIVA[2] use a variety of symbolic models trained on out-of-copyright data (such as that of Western classical music), as well as bespoke datasets. AIVA's tools come built-in to a browser-based DAW, offering easy access to generative music tools without the need for users to install packages, libraries or software, decreasing effort expectancy. Another example of a generative music start-ups includes Suno[3] and Udio,[4] which both utilise Large-Language Models (LLMs) that allow users with no musical training to generate songs based with prompts.[5]

[1]https://samplab.com/text-to-sample
[2]https://www.aiva.ai/
[3]https://suno.com/
[4]https://www.udio.com/
[5]It is important to note, given the stance of this thesis, that these companies are alleged to have gathered massive amounts of data in a way that potentially infringes copyright.

Start-ups like Infinite Album[6] leverage AI to tackle issues such as the generation of copyright-free music (using ethical training data) to supplement live streaming of games when video games have copyrighted material as background music. This use of generative solutions mitigates legal issues introduced by live streaming on platforms such as Twitch.

Beyond generative music tools, many technology companies also employ researchers to create AI music algorithms to tackle issues such as source separation and transcription, such as TikTok/ByteDance, Meta/Facebook, and Google/Alphabet. However, these types of algorithms are beyond the remit of this thesis, and as such have been left out. Furthermore, while emerging literature draws on existing technology acceptance models to inform research exploring how embedded music AI tools are accepted and viewed by composers during composition tasks, and whether aspects such as LLM-powered user interfaces, server-client file transfer delays for model inference, and whether ease-of-use drastically impact their acceptance by participants (Collins et al., under review), this work is nascent and builds upon knowledge presented in this chapter.

[6]https://www.infinitealbum.io/

5.3 Method

5.3.1 Recruitment and Demographics

I used social media platforms to recruit 11 professional video game composers for semi-structured interviews, where the initial interviewees played a role in helping to suggest secondary interviewees through snowball sampling. As an indication of the level of expertise of the participants, all participants have worked on at least one published video game. Figure 5.1 shows that of the 11 participants, seven identify as male, three identify as female, and one identifies as non-binary. This is a more fair representation than in the industry itself, where 84% of the industry is identified as male (Schmidt, 2021).

In terms of experiences that might inform their responses, all participants had multiple years of experience and participants self-report their levels within the industry in the following manner: two at the AAA level; one at the Midcore level (typically small medium enterprises that produce professional video games on smaller budgets than AAA); and eight at the indie level. Furthermore, five of the participants have worked or currently work in adjacent roles in the video game industry.

5.3.2 Data Collection

Participants took part in semi-structured interviews, an approach that offered the flexibility to explore emergent topics during interviews, while providing a degree of structure as outlined in Chapter 3. In this study, participants were asked ten questions ranging from topics such as their level of experience,

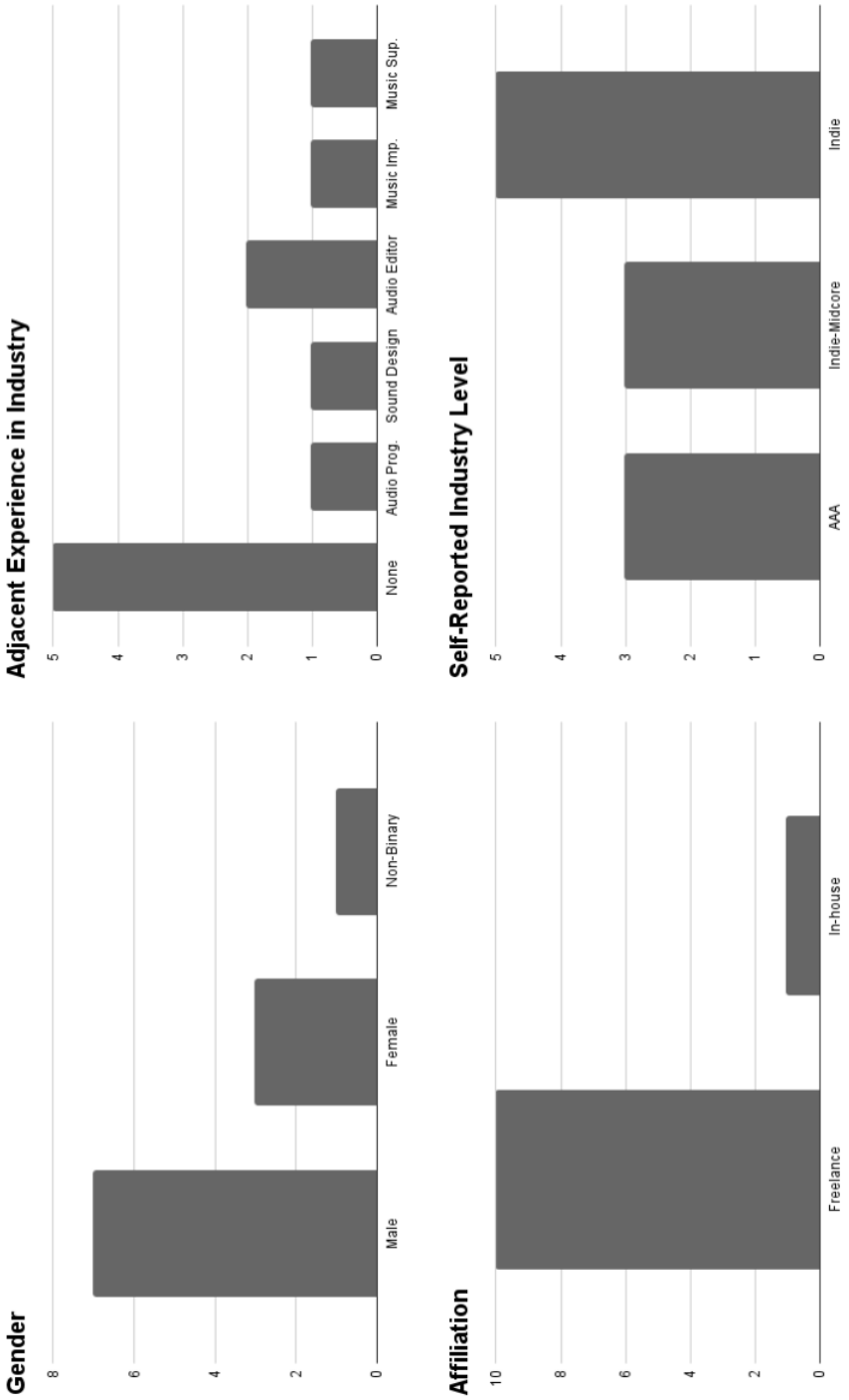

Figure 5.1: The demographic information for all participants, including: gender (TL), Experience (TR), Affiliation (BL), and Industry Level (BR).

specific considerations relating to their role, thoughts on their workflow, programming knowledge, and then their opinions on a range of existing music AI tools. These questions were designed to introduce participants to a range of different music AI tools likely to be of differing levels of controversy, to elicit their responses and gauge differences in their responses based on what kind of role the tool plays (i.e. composition, mixing, mastering, humanising). Figure 5.2 shows the full list of definite questions asked during interviews, while the transcripts in the appendices include additional questions explored in real-time, based on the answers of the participants. These interviews were conducted via Zoom, which allowed non-UK participants, and the resulting ∼13 hours of audio recordings from the eleven interviews were transcribed and anonymised by hand. For replicability, all data collected is made available here.

5.3.3 Data Analysis

Once the data were transcribed and anonymised, a reflexive thematic analysis of the collected data was performed. The advantages of this approach were that it offered a reflexive aspect to reduce researcher bias as outlined in Chapter 3. An inductive approach was used during the initial coding of the data, whereby the data were allowed to determine the themes, rather than researchers approaching the data with preconceived notions and theories, such as when using Grounded Theory (Braun and Clarke, 2013). This approach was used to explore the participants' ideas more deeply, rather than in an attempt to support existing theory (such as the TAM and UMTA). A latent approach was used when interpreting the codes, in which the subtext

Semi-Structured Interview Questions

1. How long have you been working in video game music? If you have experience in music for films or linear media, then how long have you done that?

2. What specific considerations do you have in mind when working on game music?

3. Do you have any concerns when it comes to how long a player may be listening to your music during extended gameplay segments and repetition?

4. Please describe your workflow when it comes to composing music for games.

   Prompt:
   - What macro-tasks would you say makes up the process of creating a piece of music? I.e. drafting, composing, mixing, mastering etc.
   - What smaller micro-tasks would you say that these larger tasks can be broken up into when describing your workflow?
   - Do you spend time writing various versions of musical sections due to the nonlinear nature of video games?

5. How much if any experience in programming (coding) do you have? And if you tend to have a hand in how your music is implemented, please give details on the process.

6. Do you have any opinions or concerns relating to procedural music systems in games? Please give details. And your thoughts on Music AI?

7. If you look at the list of things that AI can be used for in Music (on the attached document),what do you think about each of these ideas/demos?
   a. Melody Generation
   b. Texture/Pad Generation
   c. Chord Generation
   d. Musical In-Painting
   e. Automatic Expression for Musical Passages
   f. Generating Small Ensembles to support melody
   g. Generating your music in different styles or genres?

8. Is there a specific aspect of music composition for which you would be most open to AI assistance?

9. What do you think about the usability of the interface designs for the tools shown on the attached document?

10. Would you be willing to participate in another study in the future where you get to try out a music AI tool so we can assess how they impact workflow?

Figure 5.2: Semi-structured interview questions used in the interviews with video game composers.

of the data was analysed to find the underlying meaning, rather than relying only on stated opinions/thoughts (Braun and Clarke, 2013). This allowed for a deeper understanding of the interviewees and allowed for the linking of different ideas together more holistically.

Coding Procedure

The first stage of the reflexive thematic analysis involved transcribing the interview recordings and anonymising them by removing names or other identifying features such as employers and notable projects. Following transcription, the data were coded in Google Sheets as an alternative to tools such as NVivo. This process involved a number of passes through the data, in the first of which any excerpts of particular interest to the research questions were annotated with short descriptions. Subsequent passes involved re-examining the data for the passages that match the initial codes. This process could involve renaming the initial codes to better fit the data. Table 5.1 shows a selection of excerpts and their associated initial codes.

While the typical next step in a coding procedure would be the grouping of codes into a sub-theme, my initial number of codes totalled over 1000 ($n = 1195$). These codes were too nuanced to form a cohesive, manageable selection of sub-themes, so as an interim next step, these codes were grouped into broader codes before moving on to sub-themes. This left a more reasonable number of codes ($n = 250$) to group into sub-themes ($n = 30$). Table 5.2 shows a selection of the initial codes, broader codes and initial sub-themes.

The next step of this coding procedure was to establish a tree structure of sub-themes to candidate themes, where the candidate theme describes the

No.	Comment	Code 1	Code 2	Code 3
1	Maybe like if you wrote a melody, right. And you know, it's going to do these variations, I'd love to establish that melody before we rip it apart.	Repetition is important	AI/PMG destroys their music	
2	I'm a fan of arpeggiators and using things if they fit the style.	Acceptance linked to appropriateness	Associating AI with existing technology	
3	A huge learning curve that would potentially put me off of using it because I haven't got the time, just like the time that I would lose. Trying to learn it is time that I could have spent doing the tasks that I needed it for.	Learning curve	Time to learn	
4	A theme holds significance in its repetition. So having something that is constrained enough to be identifiable as the source material but manipulatable so that it's not repetitious is valuable.	Repetition is important	AI could be valuable	Identifying source material
5	A variation of that that could be more useful would be a really well-designed synthesizer or something that has like great sounds, great feel or whatever that is one note.	Associating AI with existing technology	Great sounds, feel and design	Tone/Timbre generators accepted

Table 5.1: Example of initial coding applied to transcribed interview excerpts.

Codes	Broader Codes	Sub-Themes
AI is a useful tool	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
An interesting tool	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
Another piece of a tool kit	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
AI can be part of a tool kit	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
AI is a tool in the toolbox	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
AI tools can be useful	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
Finding a creative, unique way to use AI	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
Fits with existing tools	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
Never going to turn down a tool	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
You can reject AI output or use it selectively	AI can be part of a tool kit	AI can be a benefit to the compositional workflow
AI can potentially save time	AI can save time	AI can be a benefit to the compositional workflow
AI can save time	AI can save time	AI can be a benefit to the compositional workflow

Table 5.2: Grouping of initial codes into broader codes and sub-themes during the thematic analysis process.

narrative of the connected sub-themes. Table 5.3 demonstrates a selection of candidate themes and related sub-themes that were explored in the coding process. Given the reflective element of this reflexive thematic analysis, the final step of this process was to then return these themes to the raw data and assess whether they represent the raw data in the absence of the context provided by the codes. This step ensures that candidate themes match well to the raw, uncoded data and limits researcher bias in situations where measuring inter-rater reliability is not possible with Cohen's Kappa.

Candidate Theme	**Relevant Sub-Theme**
Composer Benefit	AI can be a benefit to compositional workflows AI can benefit composers creatively AI can improve Accessibility in music composition Music AI could benefit the games industry
Lack of Support	Lack of standardised technical support/knowledge across industry Music AI is less Necessary if you have assistants All composers are different, and they have non-standardised needs Music AI needs to easily fit into existing workflows without too much hassle/setup
Composers' Concerns	Bad AI music can damage experiences Ethical concerns about Music AI Music AI could negatively impact the process of composing Music AI does not generate original, human-quality music Music AI introduces concerns about job security Music AI introduces concerns about musical ownership and representation of musical self Overarching technical concerns for Music AI in games

Table 5.3: Candidate themes and relevant sub-themes

5.4 Results

As a result of the RTA, I identified 1195 codes, which can be grouped into 250 broader codes. These broader codes can then be grouped into 30 sub-themes, which fit under five thematic umbrellas (with some overlap). Three of the themes create a narrative that addresses the first question of this chapter, while two address the second question (see Section. 5.1). These themes are outlined below, and the coverage of the themes across the initial 1195 codes can be seen in Fig 5.3. The thirty sub-themes and the thematic groupings are explained in greater detail in Table 5.4). As is common in reflexive thematic analysis, I now outline the narrative of each theme, connecting the sub-themes together into a cohesive whole, while supporting each with direct quotes from the raw data.

5.4.1 Benefits of Music AI

The first overarching theme identified in the data explains that composers can mainly see and are excited by the benefits to workflow and creativity that could come with adopting music AI. Participants construct the benefits of music AI in a few ways. Firstly, by focusing on what they could perceive as workflow benefits, where AIMT is part of a toolbox that saves composers time and supports them in completing tasks efficiently.

> *A really useful way for the AI to help out [...] You can get it to finish or at least get started on the tasks that I'm having trouble with, which in this case would be the fine-tuning.[P4: Indie-Midcore Male]*

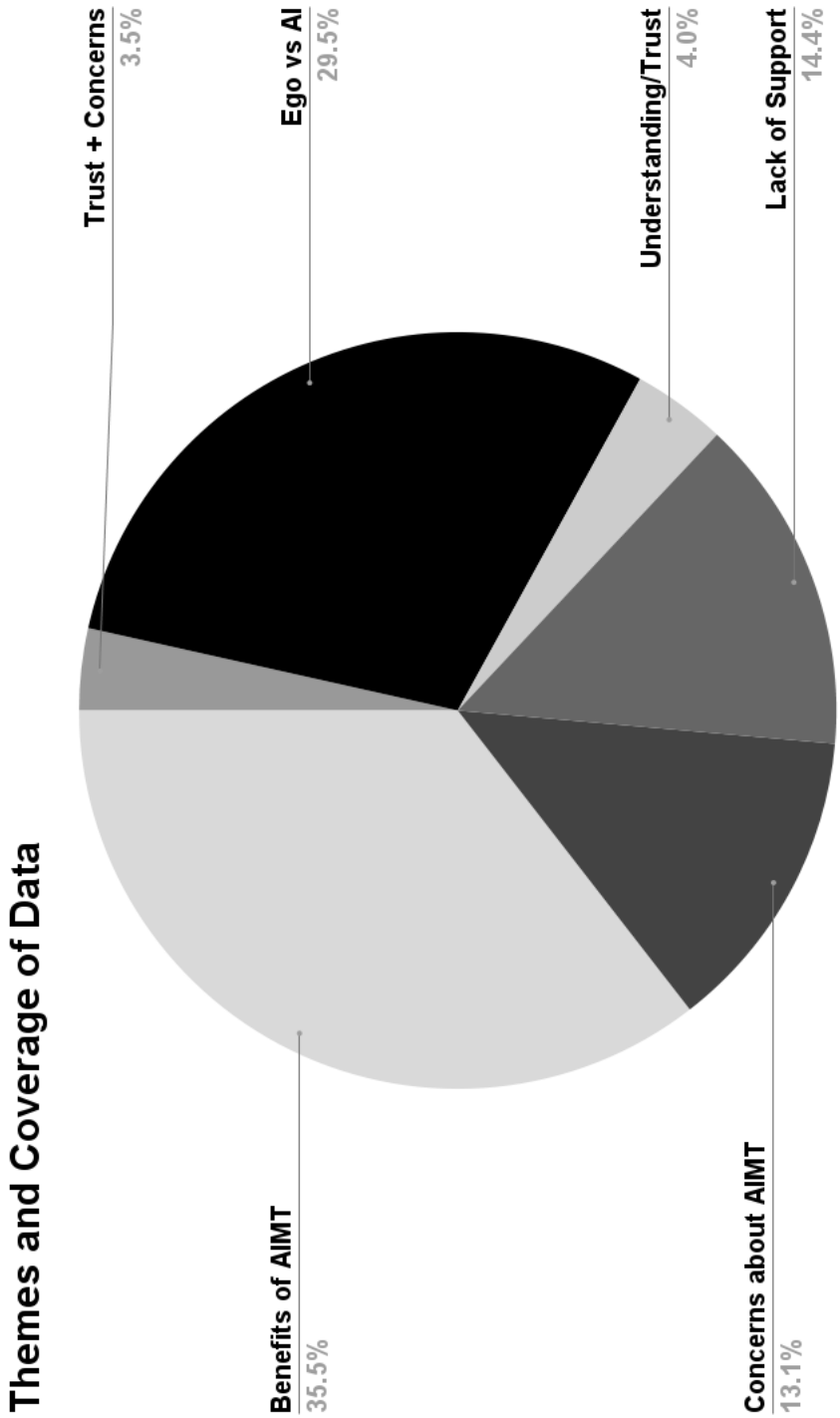

Figure 5.3: The percentage of coverage over individual codes that each theme represents.

Sub-themes	$n = 1195$	Theme
AI can benefit workflow	70	Benefits
AI can benefit creativity	60	Benefits
AI can improve accessibility	11	Benefits
AI could benefit the industry	16	Benefits
AI hard to visualise/understand	28	UT
Tools need to communicates musically	15	UT
Familiarity = Acceptance	1	UT
Curious about AI capability	4	UT
Ethical Concerns about music AI	42	Concerns + UT
Bad AI can damage player experience	3	Concerns
AI could change composition process	78	Concerns
AI music is not original/human quality	191	Concerns
Concerns about job security	11	Concerns
Concerns about representation/ownership	83	Concerns
Technical concerns for music AI in games	58	Concerns
Lack of standard support/knowledge	100	Support
Assistants vs AI	6	Support
All composers needs are different	17	Support
music AI needs to fit existing workflows	27	Support
Devs and composers communicate differently	26	Support
Generating less melodic music is less problematic	31	Ego
Deep-rooted psychological need to control	63	Ego
Desire for non-compositional AI	33	Ego
music AI clashes with the creative ego	133	Ego
Quality of music AI is not good enough	10	Ego
music AI is not well suited to music	3	Ego
If it benefits the game/players	29	Ego
Humanity in Music	16	Ego
AI is not worth the time/effort/cost	34	Ego

Table 5.4: The sub-themes identified in the data, their count across the data set and the thematic groupings that they represent. UT is a shortening of Understanding/Trust.

> *[F]rankly, except to make the composer's job easier, right? [...] the more things I can automate, the more creative I can be, because then I'm just focused on creativity. [P7: AAA Male]*

Participants constructed a second sub-narrative of AIMT as a creative benefit, where it performs the role of prompting composers who are stuck with writer's block, rather than finishing the music for them, or where it is used for musical exploration that leads to new, novel and unexpected musical ideas in their work.

> *I think that could be fun to use. I think sometimes if you're a bit creatively stuck it could be a really good prompt, the same way that writers use prompts to write, you know,...and that will kind of be a similar thing. [P6: AAA Female]*

Furthermore, participants constructed this theme by adding in additional thoughts about how AIMT can be a benefit to accessibility for novices (supporting existing research into musical co-creation as a tool for novices (Louie et al., 2020)). They also explore how AIMT could be a potential solution to legal issues around licensed music that is used during the streaming of video games, an issue being explored by Music AI startup Infinite Album.

> *Yeah, that would be cool. I think that's another example of it making it more accessible. Cause like you say, depending on music theory knowledge, you may not even be aware that inversion exists or what inversions are or how to create them and stuff like that. [P6: AAA Female]*

5.4.2 Complex Concerns about AIMT

The second overarching theme identified in the data indicates that composers have multifaceted, deeply complex concerns about music AI that go beyond the obvious concerns about job security. Participants construct a narrative theme of the complex concerns about AIMT, the first subset of which relate to concerns about the computational implications of music AI use in video games, and inconsistent musical quality and lack of human nuance in the output, all of which can potentially have a severely negative impact on the gameplay experience, and subsequently on players as a whole.

> *You lose the composer's tricks like all the flurries...but from my own standpoint, I don't think it would be something that I would personally use, because everybody's writing style is very different. [P1: Indie Male]*

The second subset of the concerns of composers about AIMT form a narrative about how such technology could have ethical and legal implications surrounding, but not limited to: job security; musical ownership; misrepresentation (of a composer's quality or competence if an AI makes a mistake); how AI could potentially disrupt the way composers work; and damage their enjoyment of the music creation process.

> *At the end of the day, I think it would be something I wouldn't particularly be comfortable using because you could definitely lose a lot of the composer's identity...because they might write bass very differently from an AI or they might do things very differently. So again, I think the composition might lose a lot of its identity. [P1: Indie Male]*

> *[Regarding AI voice] there are severe legal issues with the idea of trying to market something based on the voices of existing people. Because fundamentally, you're going to... run into potential fraudulent cases. [P8: AAA Male]*

5.4.3 The Clash between Ego and AI

The third overarching theme identified in the data describes an inherent clash between creative ego and AIMT. Participants construct the theme of the clash between ego and AI in a variety of sub-themes that outline a compelling narrative, whereby composers disliked how generative AI steps on their toes in composition tasks, but had a contrasting desire for AI to handle non-composing tasks. When combined with the sub-theme suggesting that participants found AI use for less complex and less melodic music to be more suitable than when used in melodic or interactive music, this suggests that the clash between the creative ego and AI not only exists, but could be measured by how closely the AIMT infringes on the tasks with which the individual identifies. In other words, a mix engineer may dislike a mixing AI more than a composing or sketching AI, because although each task is arguably as nuanced and complex as the others, their self-identity and in some way their self-worth is wrapped in their vocation. In response to being asked what they wish an AI could be used for:

> *[H]aving some feedback on mixes and they're [the AI] being like, ah, yes, clashing frequencies, like we suggest cutting here, boosting that and like, I never learned mixing that's the one thing that I wish I had some professional training. [P9: Indie Non-Binary]*

Furthermore, this lack of openness to specific vocationally aligned AIMT seems to scale with the importance of the musical element being generated for a composer. For example, the majority of composers were not open to the generation of melody, going so far as to describe melody as "king" [P1: Indie Male], "important," [P7: AAA Male], "a communication from a human being to another human being," [P7: AAA Male], "[having] soul" [P11 Indie Female] and as "the soul" [P10: Indie Male]. In comparison, participants were more open to chord generation (although some mentioned that generating musical inversions can change the whole feeling of a piece), and even more were open to bass, percussion or pad (texture) generation to support composition. As such, this implies that the less ego-centric the element being generated, the less likely the composer is to find an issue with the generation of the element.

> *I love to establish that melody before we rip it apart. Yeah. I would say just so it's in there somewhere because maybe that's just a sense of pride like, you wrote this melody and you really want it to be there and they want it to stand out... Cause sometimes coming up with just that melody between you and the client can be so special and important that you don't want ever bury it under iterations... Yeah, chords are chords but a melody has that soul, you know? [P11: Indie Female]*

> *No, not interested... you know, melody is as simple as just any string of notes. It doesn't matter what the string of notes is. That can be a melody is it a good melody? How do I discern that? Like how did, how would a machine discern good from bad? I don't know that anyone knows what makes a good melody... I would never want to rely on a machine to give me options because either I've heard it before a million times, because all it's going to do is regenerate the same kind of thing. I'm looking for something no one else has found... so I don't see that as being useful at all. Waste of time. [P7: AAA Male]*

The clash between ego and AI is further supported by the way that participants described a desire for bespoke tools with lots of finer controls as part of the interface of AIMT, which allow the users improved and more nuanced output, and in their concerns about musical ownership and misrepresentation (through lack of quality, or musical mistakes).

> *[The key to] feeling confident in those choices is being able to quickly scrub through them and audition them and make sure that I'm okay with that combination. And then in some cases almost being able to disallow certain combinations which then gets a lot more complicated to figure out. [P10: Indie Male]*

A second part of this theme's narrative is identified in the sub-themes of AI not being suited to music at all, and that there is humanity in art (whereby the participants were expressing that art and music are human expressions and pointless when carried out by an AI). This further supports

the clash between creative ego and AI, as computers have been shown to perform almost as well as humans for some tasks (e.g., Worrall et al. (2024)), yet creatives fully believe that computers are incapable of participating in something in which they find value.

> *[There's a] camp that…art is a human-only expression and therefore a computer can never do it. [P10: Indie Male]*

> *When do we cease to bring the humanity to whatever art we're making by allowing machines to make the art for us? I mean, what's the point of art in that case? [A]rt is arguably a communication from a human being to another human being… it's not, let the computer tell us what's good. [P7: AAA Male]*

5.4.4 No Standardisation of Support/Understanding

The fourth overarching theme identified in the data indicates that there is a lack of standardisation in the technical support provided within the industry to support composers, especially given how practices vary from studio to studio. This is especially true for AIMT, as there is already a clear need for more audio programmers in the industry to support audio teams, but AIMT also requires machine learning or AI audio specialists as part of this support structure. This theme also highlights the lack of standardisation of technical knowledge and language among composers, which has previously been identified around the term "procedural" (Plut and Pasquier, 2020). This lack of standardisation inhibits composers' ability to understand AIMT clearly, which in turn lowers their trust and understanding of it.

> *[Audio programming] is not really canonised as a field anyways. Like talking to the audio programmers that I've met, it's like, it wasn't even a dedicated profession. [P10: Indie Male]*

Beyond the support needed to allow composers to easily experiment with AIMT, there are further operational support requirements that are created when using AIMT. For example, if an algorithm generates symbolic music such as MIDI, then the game needs to either include samples to render this music in real-time, or the composer needs to book studios/musicians to make the AI-generated music match the quality of the pre-recorded assets. While the former allows for real-time generation, the second, if the tool is used early enough, could form part of the normal recording process, but if not, this then provides additional support needs for the user and disallows real-time generation in-game. This observation resonates with industry professionals and researchers, who discuss the lack of audio programmers/audio-focused AIMT developers to support the development and use of procedural audio tools in industry Weir et al. (2021).

> *If you need to do things like live recordings...you're probably going to need to send some music to live players [musicians]...if that's a requirement, then you're going to need to hire some players [musicians]. [P3: Indie Male]*

Additionally, a sub-theme of "human assistance" emerges that supports the disparity in support structure between the levels in industry (in the composers' own studios and not in video game studios). In this sub-theme, composers at the AAA level are less likely to be open to using AIMT, as many

of their tasks are already automated through human assistants (which they see as ideal, and not requiring a learning curve or disruption that accompanies new technology). Furthermore, the sub-themes of "technical concerns", "no standardisation in understanding/language", and "the disparity between the language of video game developers and composers" form a narrative that demonstrate the hurdles keeping composers (and creatives in general) from learning about AIMT, as they are not only facing a lack of support, but a lack of cohesive language to use to ask and learn about new technology.

5.4.5 Lack of Understanding and Trust in AIMT

The fifth and final overarching theme suggests that the technical nature of AIMT and the lack of consistency in its attendant terminology make it difficult for creative individuals to gain confidence and proficiency in this domain. The distinct way that participants construct this theme is rooted in their ethical concerns (often stemming from a lack of understanding of how AI models are trained), combined with their difficulty visualising what AIMT can do. This is further impacted by the lack of musical language used in existing tools. All this ferments a distrust in AI, especially when considering the lack of consistency in output quality, and the current AI-negative narrative present on social media, meaning creatives can have a bias against using AI tools.

> *That one's a trickier one because it's hard to kind of envisage what that would actually, how that would function or sort of what it would sound like. [P6: AAA Female]*

If we relate this distrust and lack of understanding back to the UMTA

(Venkatesh et al., 2003), then we can see how not being able to understand a tool leads to distrust, and then this distrust leads to negative bias impacting social influence, and lowering the likelihood of AIMT being accepted. The effects of audio-domain language versus technology-driven language have since been explored in Kamath et al. (2024), further validating this theory.

5.5 Discussion

How the perception and understanding of music AI impacts the acceptance and use of AIMT within the video game industry and among professional composers is of central interest to researchers of music AI and music informatics, as well as in multiple application domains, such as AI musical co-creation and video game audio. To my knowledge, prior to this paper, there is no empirical research looking into professional composers' opinions on AIMT, nor into what we can learn about their technical knowledge, to better inform future design and research in AIMT. Some music AI research involves asking professional composers for their opinions of generative output (Oore et al., 2020); however, more often than not, music AI is evaluated by music students (due to their accessibility and musical knowledge) (Jeong et al., 2019a,b; Worrall et al., 2024). However, there have been issues with the reliability of participants at this level in musical tasks (Worrall et al., 2024). As such, in this study, I chose to interview professional composers with higher levels of experience.

In this chapter, I outline the five identified themes, and now I discuss their relationship to my initial research questions:

1 How do composers perceive about AIMT, and how does this inform future research?

2 What can we learn about professional workflow, technical knowledge and tool use to inform future intelligent music tool research/design?

With regards to Research Question 1 from this study, I find that while composers can see some benefits to music AI (mainly assisting to prompt creativity and speed up their workflow), two main factors are holding back AIMT acceptance among professionals. The first is that they have complex and multifaceted concerns regarding AIMT, limited not only to the quality of generative output, but including: ethical concerns about training data; societal concerns about misuse by video game developers; concerns about misrepresentation by and ownership of musical material generated based on their own work; concerns for job security; concerns for technical constraints at run time in video games; and worries about how their workflow will be affected.

The second factor holding back the acceptance and use of AIMT is an inherent clash between the creative ego and AI, which scales depending on how closely the tool infringes upon tasks with which the creative individual identifies (in this case, compositional AI tools were rejected more than mixing and humanising tools). The ego in this case desires very fine-tuned control over AIMT that are non-composition focused, and do not infringe upon the music composition process. Furthermore, the adversity of reactions to generative music is linked to three other aspects: firstly, the complexity of the genre of music being generated, with ambient soundscape generation viewed as more acceptable than melodic classical music generation; secondly, the medium for which the music is intended, with video game composers viewing games as more challenging to generate for due to their adaptive and interactive nature when compared to what participants described as the relative simplicity of "meditation music" and the formulaic nature of "trailer music"; thirdly, the

extent to which the material being generated was melodic, as participants described melodies as the most important component of a track and as central to the connection between the developer and composer, leading to lower acceptance of melody generation than of chord or bass generation. In addition to these desires, the ego does not believe that AI can recreate human experience in some cases, and believes that the output of AI will always be generic and unnuanced.

My findings resonate with many of the opinions held by practitioners and theorists, such as the inconsistency in quality of output (Weir, 2017)), originality of generative output (Collins and Laney, 2017; Yin et al., 2021), resource-intensiveness of modern video games (Plut and Pasquier, 2020), the need for robust timing systems (Weir et al., 2021), and ethical and legal considerations around misuse of creative AI (Flick and Worrall, 2022).

The participants' concerns about the originality and quality of output, resource-intensiveness of AIMT, and misrepresentation (in musical style but also of themselves as a composer) in my findings demonstrate that composers have low performance expectancy when it comes to AIMT. Additionally, the concerns I find about ethical and societal misuse and job security build a picture of negative social influence, creating bias against AIMT. Lastly, concerns about their workflow being disrupted and their enjoyment of the process being reduced show that my participants have high effort expectancy. These factors likely lead to lower acceptance of AIMT. My findings extend beyond these ideas, demonstrating that, in addition to the concerns described about AIMT, existing tools are likely not being designed in a way that considers the inherent clash between AIMT and creative ego. By designing around this

consideration, researchers and developers can ameliorate the negative impact that social factors play in reducing acceptance, making them less abrasive to creatives, and increasing technology acceptance for AIMT.

With regard to research question 2 of this study, I find that the use of AIMT is being stymied by three separate factors. The first is a lack of standardisation in support for music professionals within video game studios. While all composers' needs differ, support varies not only based on the level of the video game studio (i.e. AAA/midcore/indie), but across studios at the same level. This lack of standardisation in support mirrors the concerns of practitioners about the lack of audio programmers to build and support tool use in the video game industry (Weir et al., 2021). Furthermore, this lack of standardisation increases effort expectancy for composers, as the support structure is not in place across the industry to help the less technical creatives utilise new technology,especially when machine learning is involved. This aligns with technology acceptance theories such as UMTA (Venkatesh et al., 2003).

The level at which a composer is working also seems to affect AIMT acceptance. AAA video game composers have assistants to handle tasks, whereas indie video game composers do not, and as such, the latter were more likely to be accepting of AI tools that can support their needs that an assistant would meet if they could afford one. In this example, AIMT reduces their personal effort expectancy in their role and frees up time for creative work, whereas at AAA companies, human assistants are handling these tasks already, meaning AAA video game composers are less open to these assistive technologies as effort expectancy is not lowered.

A second factor to consider is language. Composers want tools to fit easily into their workflow while providing high-quality, original and stylistically appropriate output; these tools need to be designed around the clash between ego and AI that is described in this Chapter. Furthermore, participants articulated a desire for tools to communicate "in their language", as there is a disparity between how video game developers and composers communicate, and tools often use quantitative scales as a method of communicating musical features, which differs from the qualitative language that composers may use. This difference in the way that tools and users communicate is disruptive to workflow and requires experimentation from users, increasing effort expectancy, but also reducing perceived performance expectancy if the user misunderstands the language the tool is using. This likely leads to lower acceptance of AIMT (Venkatesh et al., 2003).

Additionally, there is a lack of standardisation in the understanding of and trust in AIMT among composers. This differs from the lack of standardisation in support, as this theme relates more to the idea of how tool developers design and communicate ideas to the end user. Composers are often non-technical; as such, they find AIMT use hard to visualise, especially when it comes to the training of models on data and how models can work with their own provided music. This, in turn, leads to a lack of trust in AIMT, as it is hard for users to trust what they do not understand. This lack of general understanding of AIMT is likely to lower acceptance of technology, as it increases effort expectation and lowers performance expectation, as composers do not fully understand how AIMT is designed to work (Venkatesh et al., 2003). Furthermore, this problem becomes more complicated because

there is a lack of consistent terminology within the industry (i.e. procedural as noted by (Plut and Pasquier, 2020; Weir et al., 2021)). This lack of consistent language makes it hard for composers/participants to communicate their concerns or problems in a clear way. A final thought on tool design is that often composers relate AIMT to non-AI-driven tools that already exist, and when doing so, they are more positively disposed toward AIMT. This demonstrates the potential that better understanding can play in increasing trust in AIMT, and also in reducing the negative impact of social factors such as external opinions on AIMT adoption (Venkatesh et al., 2003).

The implications of these findings are that researchers and developers should be designing AIMT to perform para-compositional tasks that can support composers creatively (e.g., expressive rendering, mixing or mastering), or add to existing music while guaranteeing consistent high-quality and original output in the style of the user, all while fitting easily into a composer's workflow. That is, if researchers wish their tool to be used commercially by professionals.

Additionally, in order to increase trust and decrease negative social influence, developers should be very clear about: what data they require from the composer in order to generate in their style; if the model trains on the composer's data. Furthermore, AIMT should communicate in more qualitative language and be very accessible for users with non-technical backgrounds. Finally, developers would benefit from ongoing communications with creative users during tool development to increase familiarity with the tool, which will likely lead to increased trust.

5.5.1 Limitations

Researcher Bias

A weakness of this research is that I alone conducted the RTA. Without a second researcher to conduct the data analysis, I am unable to use Cohen's Kappa to measure agreement in the codes/themes. This means that there is likely some researcher bias affecting the results of this study. However, as outlined in Chapter 3 due to the nature of RTA, the role of researcher bias on the results is expected to some degree in their identification of codes and themes (with *identification of codes and themes* being used to show that the researcher plays an active part in the process of creating themes and codes, when compared to the commonly used *emerging themes* of other qualitative approaches – which disregard the active role that researcher bias plays in the creation of themes). Furthermore, RTA as an approach does help to mitigate this issue somewhat, as candidate themes (themes that have not been fully tested against the data yet) are compared against the un-coded original data, allowing the opportunity to review the themes against unbiased data. This ensures that they explain the data and reduce the likelihood of bias.

Participant Bias

Music AI is a controversial topic among music-making communities, especially in the advent of tools such as Midjourney and ChatGPT, the misuse of which is negatively impacting creative industries and academia. The participants I managed to recruit could still be biased against the use of AI in music making, but they are perhaps less biased and more open-minded to the possibility than those who would not consider participating.

Ethnocentricism

It is notable that the majority of participants in this study are based in Europe and North America. While this may partially reflect the broader demographic distribution of video game composers, it nevertheless places limitations on the extent to which the findings can be generalised, particularly in relation to under-represented regions such as East Asia, including Japan. Future research could consider extending this methodological approach to non-Western contexts in order to further explore whether and how these findings may translate across a more geographically and culturally diverse range of video game composers.

5.6 Chapter Summary

In this chapter, I address the second research objective of this PhD, namely: "**To evaluate and potentially impact the workflow, programming knowledge, and opinions of video game composers in order to inform the design and use cases of procedural music generation tools.**" This objective is intentionally exploratory in nature, reflecting the limited existing empirical research that foregrounds composers' and more broadly video game audio professionals' perspectives on AI-driven music tools.

The findings presented here primarily address composers' opinions, expectations, and technical understanding of AI music technologies, rather than demonstrating measurable changes to workflow in situ. While extensive qualitative data relating to workflow practices were collected during the study, the

scope of this chapter focuses on attitudes, perceived risks, and acceptance criteria, as reported in Worrall and Collins (2023). As such, the evidence supports an indirect rather than immediate impact on workflow: by identifying points of friction, misunderstanding, and resistance, the study informs how future tools might be designed to better align with existing professional practices.

Importantly, the results indicate that composers' concerns extend beyond technical capability to include issues of creative identity, authorship, and trust, which are not always considered in the development of AI music tools. These findings suggest that any meaningful impact on workflow is contingent on first addressing these human and cultural factors. In this sense, the contribution of this chapter lies in establishing design constraints and priorities—such as expressivity, consistency, and additive rather than replacement-based tool behaviour—that must be met before workflow integration can realistically occur.

These insights directly inform subsequent chapters of the thesis by motivating a focus on expressive rendering rather than melody generation, and on systems that augment rather than replace compositional practice. This decision is motivated by evidence that expressive rendering is a worthwhile area to explore, as demonstrated by participant responses in Figure 5.4. While the study does not claim to demonstrate long-term changes in professional workflows, it provides an empirical foundation for future work that could evaluate such impacts through longitudinal deployment or industry-based case studies.

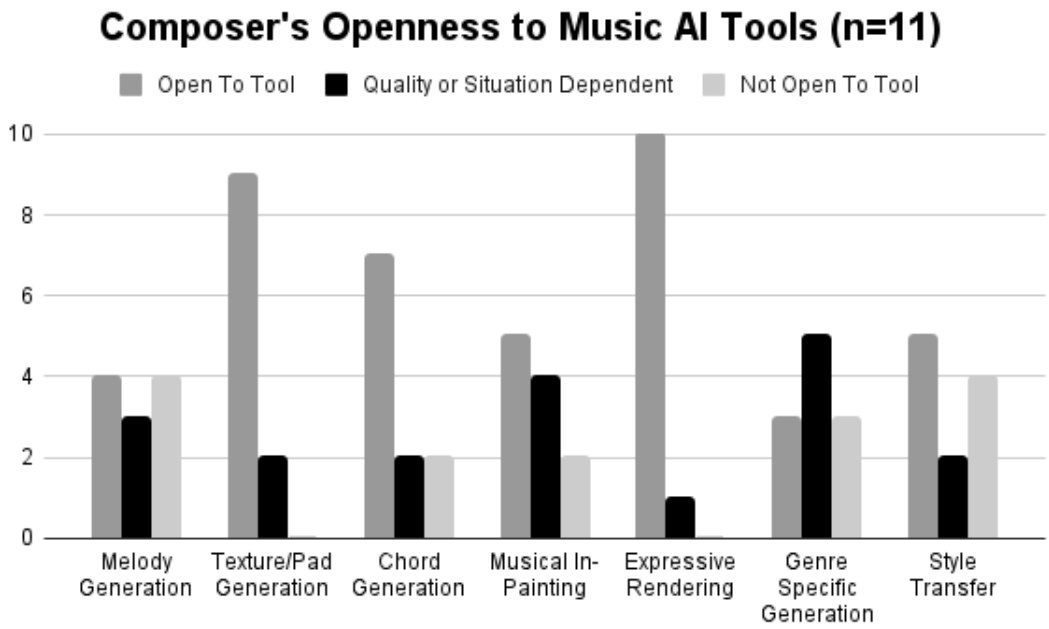

Figure 5.4: A bar chart showing the participants openness to different AI-driven technologies.

6

Modelling Musical Expression With Minimal Input Features

6.1 Introduction

In Chapter 4, I outline that procedural music generation in video games is limited in its adoption, but that the reasons are not empirically understood. In Chapter 5, I explore factors affecting the adoption of artificially intelligent music technology (AIMT), and find that, in particular, generative/procedural music lacks human nuance and expression and that tools are not designed to be easy to use with regard to their required inputs. In this chapter, I introduce and evaluate Cue-Free Express (CFE) and Cue-Free Express + Pedal (CFEP), two algorithms that, given the pitch and quantised start- and end-times of notes, are capable of emulating aspects of expressive human performance to improve the output of generative music systems and MIDI. I compare these algorithms in relation to the current SOTA in expressive rendering, as available in the "wild".[1]

As discussed in Chapter 2, the art of musical performance resides in being able to take a score – which, in addition to the notes themselves, often

[1]My use of the term wild here refers to the fact that, as outlined in Chapter 2, a majority of algorithms are not open-source, and have not been made readily available or easy to train.

includes **expressive instructions** or **score cues** regarding the tempo, articulation, and dynamics of notes – and play it for an audience in a manner that is deemed novel yet within cultural and stylistic norms "creative" (Williamon et al., 2006). One could also say that composers use score cues to indicate to performers how to bring out the character of a piece – in line with the composer's artistic vision. In Western tonal music, scores from the Romantic era tend to contain the most detailed score cues, whereas those from other periods, styles, or cultures may contain fewer, different, or no such expressive cues. Irrespective of the presence of score cues, music is rendered expressively in the hands of professional performers.

Beyond score cues as directives for performers, there are further annotations that can be added to scores to describe the musical information on a deeper level, thus improving the way they can be understood/computed. These include, but are not limited to, annotations for metre, phrase information, or musicological analyses (Narmour, 1990; Lerdahl and Jackendoff, 1996; Herremans and Chew, 2016).

Research spanning several decades has explored computational analysis and generation of expressive performances (e.g., Widmer, 2002; Grindlay and Helmbold, 2006; Grachten and Widmer, 2012; Katayose et al., 2012; Cancino-Chacón, 2018), with more recent efforts leveraging neural networks and deep learning (Jeong et al., 2019a,b; Wu and Yang, 2022). For example, the VirtuosoNet (Jeong et al., 2019a) algorithm takes basic score data (notes) and additional score cues and creates deviations in tempo, note timing, articulation (sometimes including pedalling) and dynamics (note velocity) that imitate human performance. For comparison, the Basis Mixer (Cancino-

Chacón, 2018) encodes high-level aspects of the score as a type of score annotation (for example, tonal tension calculations (Herremans and Chew, 2016) to supplement encoded score cues (e.g., *dynamic markings*) and low-level note information).

While architectures and input features become more complex, there is a lack of standardised evaluation of expressive rendering algorithms following the discontinuation of the Performance Rendering Contest (RENCON) in 2014 (Katayose et al., 2012). This is in contrast to other components of automatic music generation, where there is systematic evaluation through initiatives such as the AI Music Generation Challenge. Additionally, while some machine learning-driven music generation techniques have been used in video games (Brown, 2019; Abel, 2021), there is very little research looking at the application of expressive rendering in a video game setting, with the only real attempt being more of a toy than a real expressive rendering system (Haaland, 2020).

Furthermore, while symbolic music generation algorithms are adopted in professional Digital Audio Workstations (DAWs) such as Ableton (Hadjeres and Crestel, 2021; Guo et al., 2022) and Cubase (Tchemeube et al., 2023), algorithms for the expressive rendering of inexpressive music data have yet to be integrated in DAWs. Rather, DAW users can choose to add uniformly distributed randomised deviations to musical parameters such as loudness and timing, to provide variation (Logic Pro X's "Humanise") or create their own rule-sets to improve upon randomness (Cubase's logical editor). In Logic Pro X, this "Humanise" functionality, while better than nothing, ironically constitutes a pale imitation of human expressivity. These industry standards

are generally overlooked in comparative evaluations (Jeong et al., 2019a)–(Katayose et al., 2012).

As previously stated, in this chapter, I introduce two transformer-based expressive rendering algorithms, Cue-Free Express (CFE) and Cue-Free Express + Pedal (CFE+P), which require only the start and end times and pitches of notes as input.[2] Although the application of the transformer architecture to music tasks is not novel in and of itself (Huang et al., 2018; Huang and Yang, 2020; Wu and Yang, 2022), this is the first use of an ensemble of transformers to tackle expressive rendering in isolation.

Where contemporary expressive rendering models such as VirtuosoNet and Basis Mixer rely on, or at least utilise score cues or additional annotations to better inform performance deviations, these cues are not always available in real-world settings such as video games or DAWs. Cue-Free Express (CFE) and Cue-Free Express + Pedal (CFE+P) are designed to generate expressive performances directly from the MIDI note sequence alone, without requiring structural or expressive annotations. This cue-free approach enables more immediate integration into applications where annotated scores are unavailable or impractical, while still aiming to capture human-like expressive nuances in timing and velocity.

I compare my systems with the aforementioned "Humanise" function, which I propose should be a baseline for future research in this area, as it is an industry standard and does not require any additional input beyond that found in MIDI. Additionally, I compare these systems with two other deep learning models, Basis Mixer (Cancino-Chacón, 2018) and Virtu-

[2]Henceforth, I refer to this information as OPD, short for ontimes, pitches (represented using MIDI note numbers – MNN), and durations.

osoNet (Jeong et al., 2019a), as well as real human performances (Foscarin et al., 2020). While other existing models could be compared in studies to follow, I must consider the time that listening studies take to complete, and thus choose to be representative of the existing open-source systems by choosing the Basis Mixer and VirtuosoNet. I choose these models as both have been shown to perform well in this domain, and because they utilise fundamentally different architectures (a Long Short Term Memory + Feed Forward Neural Network and a Hierarchical Recurrent Neural Network, respectively). The use of the phrase "in the wild" in the earlier description of these systems indicates that Basis Mixer and VirtuosoNet's source code was not usable for retraining the models on the same data as my models (Foscarin et al., 2020), but they could be used in their pre-trained forms, and so I did not want to omit them from consideration. Due to this difference in training data, however, the description "in the wild" is a better characterisation than "systematic" of the comparative evaluation I provide in this chapter.

The remaining sections of this chapter are structured as follows. First, I discuss the aims of my machine learning approach and how the dataset is prepared. Second, I introduce Cue-Free Express (CFE) and Cue-Free Express + Pedal (CFE+P). Third, I report the results of two comparative listening studies and the results of their Bayes factor analysis (BFA) (van Doorn et al., 2020). Fourth, I report the results of quantitative analyses of predicted velocity and timing values, as well as an exploration and visualisation of the architecture's self-attention mechanism. Lastly, I discuss my findings and their implications, the limitations of this work, and possible future directions for research in this area, relating this work back to my research questions.

6.2 Dataset and Learning Aims

The dataset used for this chapter is the Aligned Scores and Performances (ASAP) dataset (Foscarin et al., 2020), which consists of 236 musical scores and 1,067 performances of these same scores. All of the pieces are Western classical music from 15 different composers and all are written for and played on piano. Each performance in the ASAP data set has an associated annotation file, which I use to calculate a **performance-symbolic beat map** (PSBM) between the beat `ontime` in the symbolic representation of the piece and the beat `onset` in seconds in the performance. Original beat annotations for the ASAP dataset were automatically created (explained in Foscarin et al., 2020).

My general data processing approach and terminology are similar to other work in the field (Cancino-Chacón, 2018; Jeong et al., 2019a; Widmer, 2002). From the MIDI performance of a piece, I derive the `onset` in seconds that each note begins, the duration in seconds for which it is held (which I call `durSec`), and the `velocity` (dynamic level) of each note. As indicated in Figure 6.1, for any score note (shaded note in top panel), I can use the PSBM to *estimate* where that note will begin and end in the performance (empty note in bottom panel), and compare these to the same properties *observed* in the performance (shaded note in bottom panel), deriving the quantities d' and d''. Using the PSBM once more, I can define the start- and end-time differences in units of quarter-note beats (empty note in top panel), labelled x' and x'', generally, referred to as `ontimeAdj` and `offtimeAdj`. The synchronised audio-symbolic representations, including all the computed properties mentioned, are stored in JSON format files – one file per MIDI

performance.

Based on existing MPS literature and observations of my first listening study (see Section 6.4, I developed models aiming to predict dynamics, tempo, timing, and sustain pedal use.

- **Dynamics.** Given OPD note data, predict the `velocity` value of each note;
- **Tempo.** Given OPD note data and one existing tempo in BPM, predict the beat-wise tempo values up to and including the end of the note data;
- **Timing.** Given OPD note data, predicted velocity data and associated beat-wise tempo values, predict the `ontimeAdj` and `offtimeAdj` values of each note;
- **Sustain pedal.** Given OPD note data, predict beat-wise pedal values (our CFE+P model only).

6.2.1 Applicability of this Dataset to Video Games

The choice to model expressive performance using a piano dataset is made for two distinct reasons: firstly, piano performance datasets are more widely available to use in research than for other instruments, in particular where MIDI recordings are concerned; secondly, while a piano features aspects that do not apply to other instruments, such as pedal use, the strength of this instrument is that many aspects of performance can be generalised. The way in which sounds are generated from a piano is by the hammer attached to the key striking a string to generate sound. The same can be said of

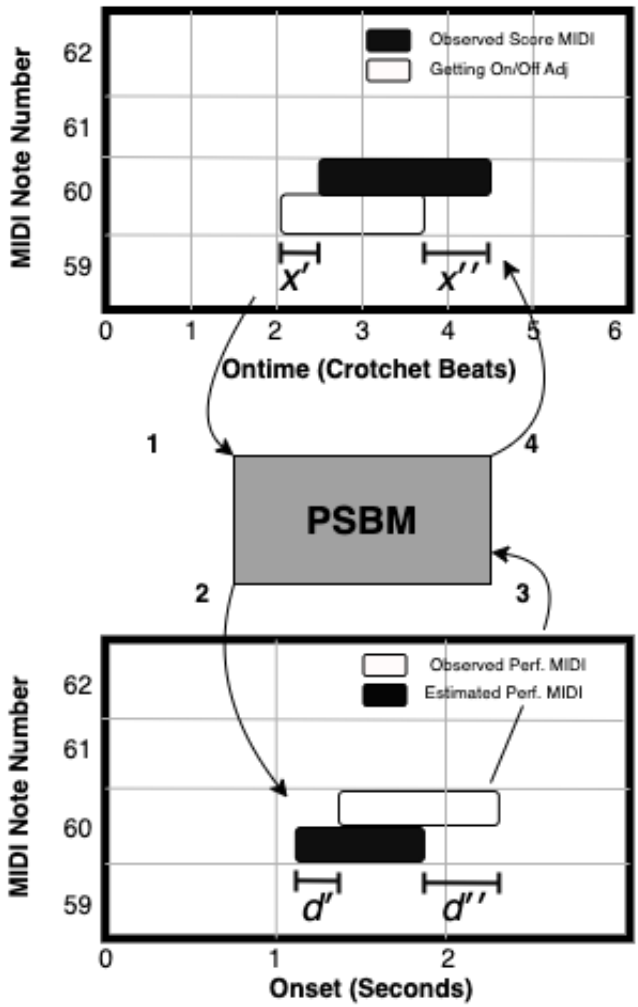

Figure 6.1: 1. A note's `ontime` is extracted from the score; 2. This is combined with the PSBM to estimate the `onset` of the corresponding note in a performance; 3. This estimated `onset` is compared to the actual time at which the note is performed; 4. The difference is combined with the PSBM again to give `ontimeAdj/offtimeAdj`. The values indicated in the figure are exaggerated for the sake of readability.

using a plectrum for a bass guitar, pizzicato for strings and sticks for playing drums. While the expressive nuance across instruments may differ, a piano-based model is far more generalisable than a string dataset (due to the aforementioned mechanics of playing). Additionally, the piano is a very common instrument in video game music, with generative research using piano arrangements of video game music as training data.

6.3 Method

6.3.1 Data Preparation

Before training any models, for each note performed in ASAP, I first determine if there was a score note of the same pitch within 0.1 sec of where I would expect it to occur, given the PSBM. I calculate an F_1 synchronisation score for each MIDI performance and only admit a performance into the dataset if its F_1-score exceeds .9. For each note in the score MIDI, I use the annotated beat locations to derive an estimate of where that note should occur in a performance of the piece. On searching the performance, if I find a note of the same pitch within 0.1 sec of my estimate, then this counts as a true-positive (TP) result (when there are multiple candidates, the one with the smallest absolute time difference between estimated and performed time is selected); otherwise (no note found in the performance) it counts as a false-negative (FN); if, once the score MIDI has been iterated over, a note from the performance MIDI remains unmatched, this counts as a false-positive (FP). The F_1 synchronisation score is then calculated in the ordinary way: $F_1 = 2PR/(P + R)$, where $P = \text{TP}/(\text{TP} + \text{FP})$ and $R = \text{TP}/(\text{TP} + \text{FN})$.

Additionally, I applied a further filtering of notes if either the `ontimeAdj` or `offtimeAdj` values were outside of the range $(-0.7, 0.7)$ sec. This leaves 247,247 notes for training the models, and I adhere to a train:test:validate split of 80:10:10. While I note that the number of successfully synchronised notes is fewer than the total number of notes in the ASAP dataset, I suspect that some MIDI encodings of performances having non-standard tempos may be responsible for synchronisation failures. However, I am satisfied with the amount of data available, and leave further investigation to future work.

6.3.2 Architecture

The architecture for my initial CFE model consists of an ensemble of four transformer models (Vaswani et al., 2017; Huang et al., 2018) (one for velocity, one for tempo fluctuations, one for note start-time adjustment, and one for note end-time adjustment). This initial model is then augmented for my second listening study, to include a fifth transformer that predicts sustain pedal usage (also indicated in Figure 6.2). Henceforth, I refer to this second model as CFE+P.

In this architecture, I use the transformer encoder only, as the predictive task is not auto-regressive, and I directly input the conditioning features and predict the target feature, meaning there is no application of cross-attention or causal masking. The right-hand side of Figure 6.2 shows my architecture in more detail: they have an initial embedding layer for pitch, which converts the discretised pitches into learnable embeddings, whereas in the original transformer paper (Vaswani et al., 2017) the absolute positional embedding takes the positional indices and dot product with sinusoidal vectors.

Subsequently, these models use the remaining input features (e.g., ontime and duration) to provide positional information to the model, where the value or position is not discrete. For the ontimeAdj and offtimeAdj models, I also provide the predicted velocity values as additional positional information. These embeddings are then converted into a vector of the same size as the initial pitch embedding, and the summed information is passed through the transformer encoder, which captures the dependencies and relationships between elements in the sequence. The model output y_t is a certain predicted attribute (e.g., velocity) of the note at time t, based on the given attributes x_t (e.g. OPD).

The transformer architecture is chosen for this chapter because the self-attention mechanism, in combination with positional context, gives the model the potential to understand the relationship between tokens, within the max sequence length parameter of the model. My premise is that with this architecture's focus on using only OPD as input, self-attention may help it to overcome the lack of additional input features, such as score cues (e.g., crescendo marks).[3]

First in the inference pipeline, `OPD` are used as input for the *Dynamics* model. Using these features, the model predicts the velocity of each note in the given sequence, where the predicted attribute is an integer value of 0-31, with each value representing four values in the range of 0-127 typically used for velocity within MIDI encodings or Digital Audio Workstation representations.

Second, the *Timing* model is comprised of two separate transformer

[3]All of the models in this architecture make probabilistic predictions, using categorical sampling at a temperature of 0.4.

encoders: one that predicts adjustments to the note's quantised ontime (`ontimeAdj`); and another that predicts adjustments to the note's quantised offtime (`offtimeAdj`). This model takes the `OPD` and predicted `velocity` values as input, and predicts an ontimeAdj and offtimeAdj for each note in the sequence. In these cases, the predicted attribute is a float representing a micro-adjustment to the quantised start- or end-time of a note, respectively, in units of quarter notes.

Third, the *Tempo* model analyses the `OPD` information for all notes in the piece, and calculates a set of features per beat, namely: `note density, min ontime, max ontime, mean ontime, min MNN, max MNN, mean MNN, min duration, max duration, mean duration`. Using these input features, it predicts the proportional change in tempo from one beat to the next. I label this $\alpha\Delta t$, where $\alpha = 0.25$ is a weighting that affects the amount I allow the tempo to change proportionally from one beat to the next.

With the information provided, I then calculate the new expressive start time x according to Eq. 6.1 and the expressive duration z according to Eq. 6.2, where O is the expected start time and D is the expected duration according to the tempo fluctuations, x' and x'' are `ontimeAdj` and `offtimeAdj`, respectively, and $\beta = 0.25$ is a weighting that affects the amount that timing variations can shift notes either side of the underlying beat. The weights α, β were chosen by varying these parameters and listening to the resulting output until they sounded expressive but not too extreme.

$$x = O + \beta x' \tag{6.1}$$

$$z = O + D + \beta x'' - x \tag{6.2}$$

For CFE+P the last step in the inference pipeline involves using the same beat-level input features employed in tempo modelling to predict the given position of the sustain pedal at the time of each beat. For this model, the predicted attribute is an integer within the range of 0-3, where 0 denotes complete pedal depression (application of sustain), and 1-3 represent varying degrees of pedal being raised.[4] I do not model sustain pedal use in this project initially, for the CFE architecture, because I do not want these modelling efforts and results to be too confined to one specific instrumental category (e.g. keyboard instruments with a sustain pedal). The downside of this decision, however, is that the software I use to turn expressive MIDI data into audio accesses a qualitatively different sample bank when the sustain pedal is down, and this may lead to slightly more realistic or favourable-sounding excerpts for the VirtuosoNet and human categories in Listening Study 1. As such, I introduce the modelling of the sustain pedal in Listening Study 2, which comes with a reduction in instrument agnosticism, but, as shown in later sections, improves ratings.

[4]The idea here is that with varying degrees of prediction, varying degrees of sustain could be applied; however, further investigation shows that sustain application in a Digital Audio Workstation is binary, as such all three non-depressed states equate to 0% sustain.

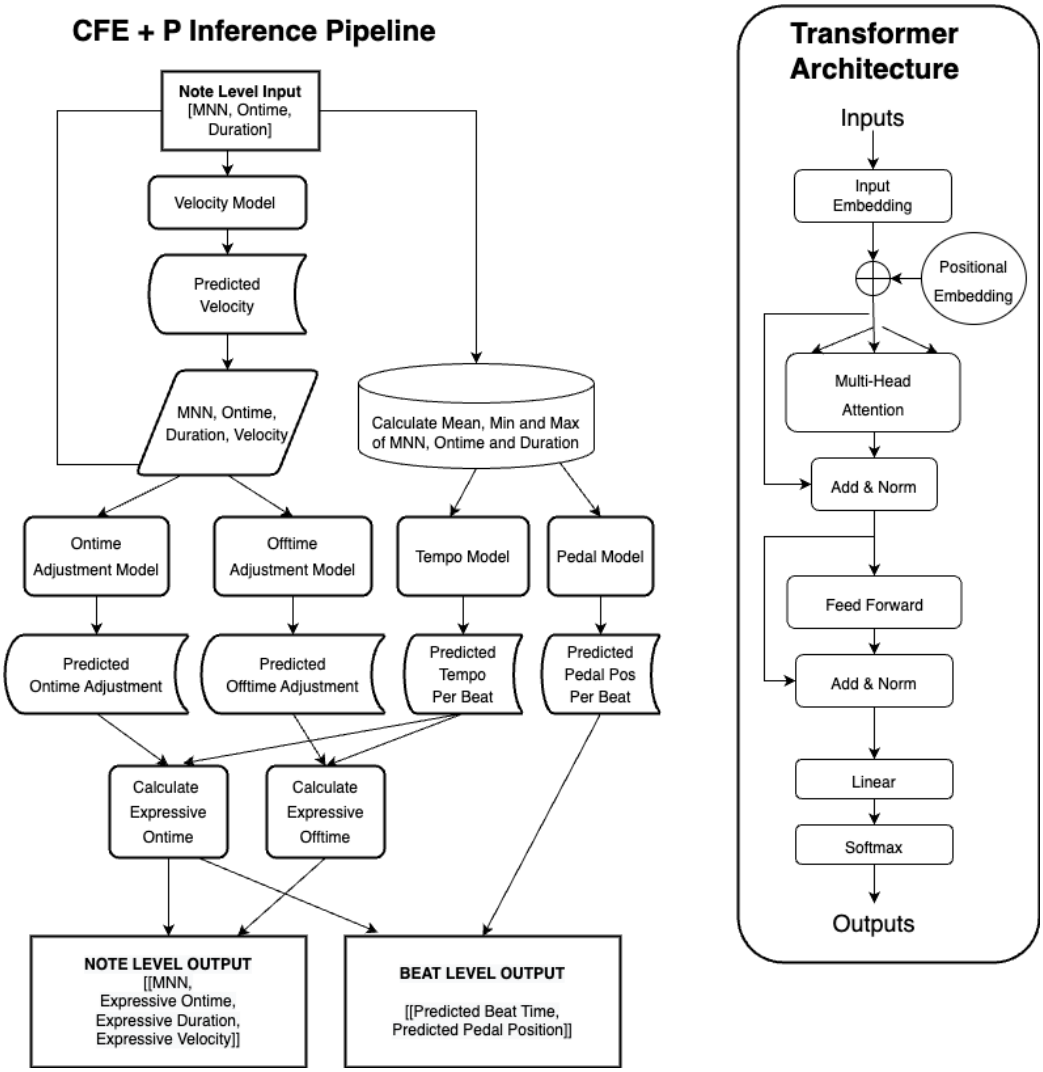

Figure 6.2: The inference pipeline of Cue-free Express (CFE) and Cue-free Express + Pedal (CFE+P) models (left); and a diagram after indicating the transformer architecture used (right).

6.3.3 Training

All models are trained in a supervised fashion, with the aim of minimising the loss of predictions (Jeong et al., 2019b; Shi, 2021). The specifications of the three note-level transformers and the two beat-level models can be seen in Table 6.1. All are trained with the Adam optimiser (Kingma and Ba, 2014) with a learning rate = 0.0001. Loss scores are tracked during training, and the final scores are shown in Table 6.2 as reported in similar work Jeong et al. (2019a). Table 6.1 shows the specific parameters of the models in the CFE and CFE+P architectures.

Model	Parameter	Value
Note-level	Model Dimension	256
Note-level	Number of Heads	4
Note-level	Number of Layers	6
Note-level	Batch Size	32
Note-level	Sequence Length	255
Beat-level	Model Dimension	64
Beat-level	Number of Heads	4
Beat-level	Number of Layers	3
Beat-level	Batch Size	12
Beat-level	Sequence Length	255

Table 6.1: The specifications of all models within the CFE & CFE+P architectures.

I evaluate the note-wise models using L^2 loss (mean squared error, MSE), because it is appropriate for the loss ratings to be scaled depending on the distance between the prediction and the target, rather than reporting the prediction as equally incorrect for all values except the target. For example, the difference between a target and the nearest values can be difficult to dis-

cern audibly with regard to velocity, whereas a note being played as loudly as possible will be much easier to notice. I evaluate the beat-wise models using L^1 (mean absolute error, MAE) and Cross Entropy (CE) loss, respectively. I do this as I feel that deviations in tempo do not need to be penalised as strongly as those in note-wise models, and pedal prediction is a classification task, meaning Cross Entropy loss is appropriate.

Model	Loss Type	Loss/Error
Velocity	MSE	0.650
OntimeAdj	MSE	1.200
OfftimeAdj	MSE	0.098
Tempo	MAE	0.029
Pedal	CE	0.441

Table 6.2: Final loss ratings for each model during training.

6.4 Experiments

I assess the model's (CFE and CFE+P) through a comparative evaluation of expressive renderings of music data, taking place over two listening experiments. The first study includes data for nine participants, and the second expanded version of the first study includes data for 13 participants. The experimental design, hypotheses, and results of each study are detailed below. For replicability, all data collected is made available here.

6.4.1 Listening Study 1

Experimental Design

Two computational expressive rendering algorithms (VirtuosoNet (Jeong et al., 2019a), my CFE model) and two comparison points (Human Performer and Inexpressive MIDI) are evaluated using a within-participants design involving nine pieces of music, giving rise to 36 stimuli.

Participants are undergraduate or postgraduate music students at the University of York, and the study received ethical approval from the University's Computer Science Ethics Board. Nineteen participants completed the study and received £15 in compensation for one hour of their time. Four participants' data are removed for completing the questions in less time than it took to listen to all the music excerpts, and six for rating *completely inexpressive* output as highly expressive (rating of four or higher on a Likert scale of 1-7) two or more times, as this implied a lack of attention, leaving nine participants' data for analysis. While there is an argument to be made that these participants were potentially rating the musical expressivity high

because of the composition itself, in the participant information sheets and before the study, they were informed to rate based only on the expressive qualities present in the performance. In order to ensure fair comparison, the data for these six participants were not included.

The participants rate excerpts along the following dimensions: overall expressiveness of the performance; use of dynamics (variation in loudness); use of timing and articulation; use of tempo change or "rubato". A free-text box is also provided per piece for participants to record their thoughts and reactions. Participants are not aware of the provenance of any excerpt with respect to the underlying system to avoid biases. Furthermore, it is explained to participants that the lowest ratings should be reserved for excerpts that sound inexpressive and robotic, and the highest ratings should be reserved for excerpts that sound expressive or human-like.

The nine excerpts used as stimuli for this study consist of: Bach Fugues BWV862 (Bach, 1722a), 885 (Bach, 1742a) & 893 (Bach, 1742c); Bach Preludes BWV865 (Bach, 1722b) & 892 (Bach, 1742b); Chopin Ballade D47 no.3 mvt.1 (Chopin, 1841); Haydn Keyboard Sonata Hob XVI nr.32 mvt.1 (Haydn, 1776); Schubert Impromptu D899 no.2. mvt.2 (Schubert, 1827). These pieces represent a range of musical periods (Baroque, Classical, Romantic) and time signatures (9-8, 4-4, 3-4, 6-8). This piece selection and the test:train:validate split happens pseudo-randomly to keep the tests fair, using a seed for replicability. In keeping with previous comparative evaluations of expressive rendering systems (e.g., Wu and Yang, 2022; Fujishima, 2018), I take matching sections from the human performance data and the MIDI scores, and give VirtuosoNet and CFE the necessary input (MusicXML and

MIDI, respectively) to render the same selections. Many seeds were tested, and this one gave the widest spread of pieces, composers and classical periods to ensure the fairest testing possible.

All of the MIDI is processed using the same Steinway Piano preset in GarageBand, and is faded in and out using the same amounts of time within Audacity to ensure consistency. Based on the existing literature, I opt for all excerpts to be 20-60 sec long (due to feedback that 10-15 sec is too short a time to assess music, Malik and Ek, 2017).

Hypotheses

As I use a BFA for data analysis, I am able to state hypotheses about finding evidence in favour of the null hypothesis (i.e., no difference between systems), unlike in frequentist hypothesis testing. My hypotheses are as follows:

1. CFE will receive significantly higher ratings on expressivity than the Inexpressive baseline;

2. VirtuosoNet will receive significantly higher ratings on expressive timing compared to CFE;

3. Human Performer will receive significantly higher ratings on tempo/rubato than VirtuosoNet;

4. Human Performer will receive significantly higher ratings on tempo/rubato than CFE;

5. Human Performer will receive significantly higher ratings on expressivity than VirtuosoNet;

6. Human Performer will receive significantly higher ratings on expressivity than CFE;

7. There will be no significant difference in ratings for expressive tempo between CFE and VirtuosoNet.

Results

Figure 6.3 indicates that CFE was rated higher than the inexpressive baseline, and that VirtuosoNet is evaluated on par with the human performers across all four metrics. The outcomes of the BFA are as follows:

1. Extreme evidence ($BF_{10} > 100$) that CFE receives significantly higher ratings on expressivity than the expressionless baseline;

2. Extreme evidence ($BF_{10} > 100$) that VirtuosoNet receives significantly higher ratings on expressive timing than CFE;

3. Strong evidence ($BF_{10} = 0.0565$) that Human Performer did not receive significantly higher ratings on expressive tempo than VirtuosoNet;

4. Extreme evidence ($BF_{10} > 100$) that Human Performer receives significantly higher ratings on expressive tempo than CFE;

5. Strong evidence ($BF_{10} = 0.0624$) that Human Performer did not receive significantly higher ratings on expressivity than VirtuosoNet;

6. Extreme evidence ($BF_{10} > 100$) that Human Performer receives significantly higher ratings on expressivity than CFE;

7. Extreme evidence ($BF_{10} > 100$) that there was a difference in expressive tempo use between VirtuosoNet and CFE, with the former outperforming the latter.

Listening Study 1 Discussion

The main findings of Listening Study 1 are that: my model (CFE) performs better than the *inexpressive* baseline; and VirtuosoNet performs well compared to Human Performer and CFE. Interesting qualitative responses highlight some thoughts on: how participants engage with listening studies based on their own performance experience; the nature (and emphasis) of fugue subject during performances; the use of pedal in performances being a potential detriment (even though ratings indicate otherwise in this case).

- "my opinion [of each audio stimuli] would have differed had I had experience playing [the pieces the excerpts came from]...I judged the ones I have played much more critically – probably because I knew what the score demanded" [Participant 16, verbal comment on no specific system];

- "all [system's renditions of a single piece] seem to think the fugue's subject has to [be] emphasised and be dynamic" [Participant 4];

- "[The final performance of Schubert's D899 no.2] is rather muffled with too much pedal" [Participant 4, commenting on VN.

While the runtime of both models is comparable and VirtuosoNet is open source, my model (CFE) is better suited for integration within a video game engine, despite performing less well in expressive rendering benchmarks. First, VirtuosoNet requires MusicXML files as input, which may include performance context features that are typically unavailable in the MIDI data

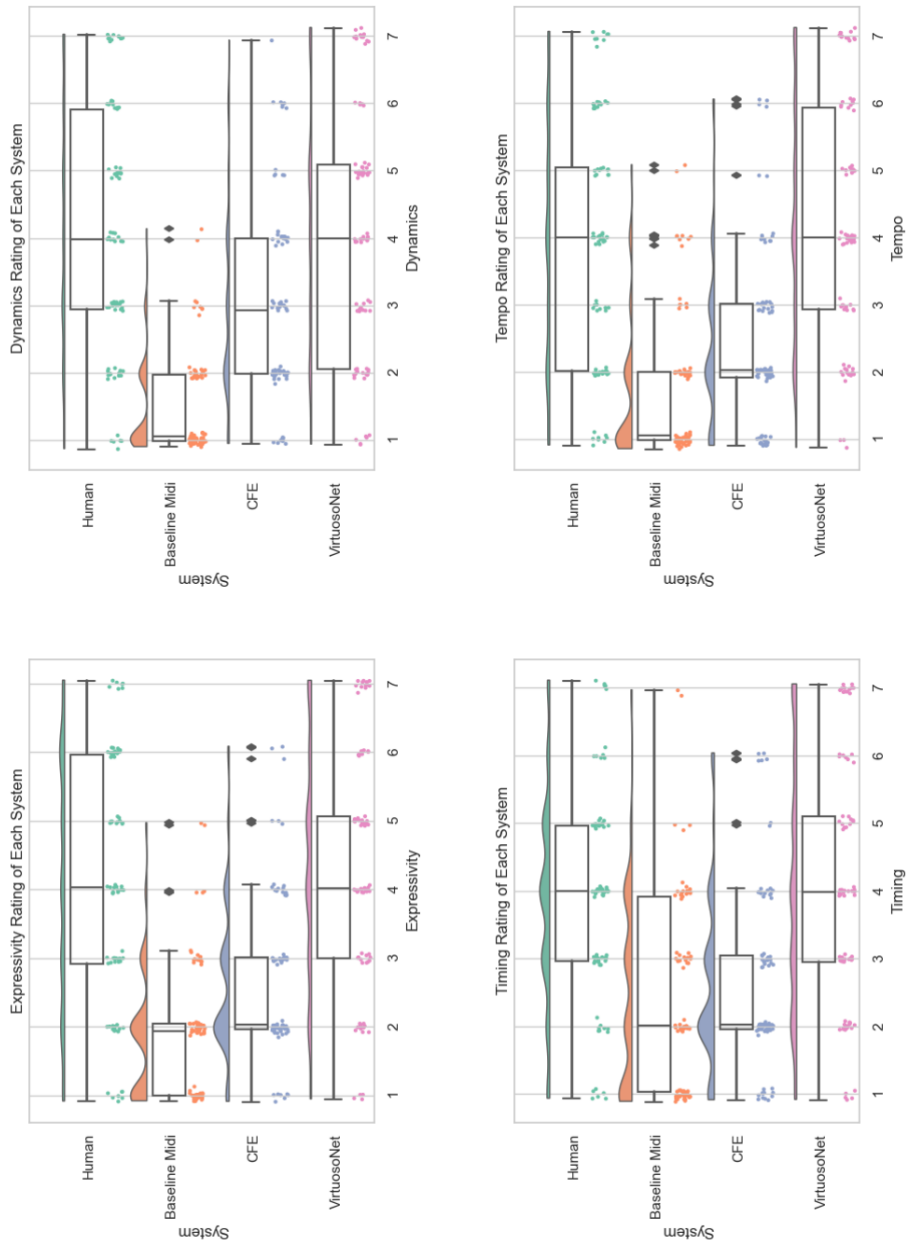

Figure 6.3: Raincloud plots of my results for Listening Study 1; expressivity, dynamics, timing and tempo ratings.Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side.

used by video game composers and audio middleware. Although VN can function without these features, it performs best when they are present, making it less practical for real-time video game applications. Second, VirtuosoNet primarily models short, bar-level expressive gestures, whereas CFE can capture longer-span performance patterns across multiple bars, better supporting continuous, gameplay-adaptive expressive rendering.

6.4.2 Listening Study 2

Based on feedback received concerning Listening Study 1, I decide to conduct a second listening study that includes two new systems. These systems are: the Basis Mixer (BM, Cancino-Chacón, 2018), another score cue informed algorithm that uses a feedforward neural network architecture; and a Randomised MIDI (RM) system (described more fully below). RM is intended to offer a stronger baseline than *Inexpressive*, while BM offers another computational model for comparison.

Furthermore, based on observations of VirtuosoNet's performance in Listening Study 1, I train a fifth transformer model to generate sustain pedal predictions. I do this as the use of sustain pedal automation in GarageBand appears to access qualitatively different samples on the Steinway Piano preset, and VirtuosoNet's incorporation of sustain pedal predictions could be driving a larger difference in ratings between CFE and VirtuosoNet in Listening Study 1 than would otherwise be the case. As such, I include CFE+P (a variant of the original model which includes pedal use prediction) in Listening Study 2, which enables me to shed some light on this matter.

For Listening Study 2, the same musical excerpts are used as in Listen-

ing Study 1, so the examples for CFE, VirtuosoNet, Human Performer and *Inexpressive* remain the same. I prepared additional audio stimuli for the three new systems: CFE+P, RM, and BM. The BM system uses the publicly available implementation that can be found "in the wild",[5] which is trained on the Vienna 4x22 data set. This data set is smaller than the full ASAP data set, but of a comparable size to the amount of data that I use for my models based on the previously mentioned synchronisation F_1 scores. The RM system is based on code I wrote to simulate Logic Pro X's "humanise function" (referred to technically as MIDI Transform presets). This is the industry standard, non-machine learning solution that composers use when working in Logic Pro X, whereby random, uniformly distributed numbers in a specifiable range are added to or subtracted from the velocity, ontime, and duration of inexpressive note data, to replicate what Logic Pro X users would expect from "humanising" their MIDI.[6]

In this case, the natural, scientific approach would be to parameterise the RM system using the ranges observed in CFE's output, but an RM system parameterised in this way sounds too chaotic, so I reduce the ranges RM uses until the output sounds less chaotic but not too metronomic or inexpressive. While Cubase's logical editor provides an alternative, potentially superior baseline for comparison, I choose to use "humanise" as the second baseline because: my model purposely requires only inputs seen in MIDI data and is designed to replace this function in particular (based on interviews seen in Chapter 5); the use of the logical editor requires direct input from users

[5] `https://github.com/CPJKU/basismixer`

[6] I read the following documentation (`https://support.apple.com/guide/logicpro/midi-transform-window-presets-lgcp215831be/mac`) and studied distributions derived from applying the MIDI Transform presets to toy examples of 1000 notes, in order to arrive at my findings and simulation.

which for the experiment would be different depending on who is conducting the experiment; the use of the logical editor is analogous to how score cues and markings are potential impediments to creative workflow when using a DAW, even if they may increase the expressive quality of MIDI playback.

In terms of runtime cost, there is no published benchmark or comparison for the existing models. In this case, RM, being purely random and not based on any deep learning architecture, has the lowest computational cost. BM's model uses relatively simple feed-forward and recurrent networks and incurs the lowest computational overhead, making it feasible for real-time use. VN, by contrast, uses hierarchical RNNs with attention and a variational encoder, which increases inference cost and typically suits offline rendering more than real-time gameplay. My models (CFE & CFE+P) lie between these extremes: although transformer layers are more computationally demanding than simple RNNs, their parallelisability and independence from extensive score annotations make CFE better suited for integration into video game engines with low-latency requirements.

Experimental Design

For the second listening study, I use the same experimental design and recruitment methods as the previous study, and the same exclusion criteria, but scaled up to allow for participants listening to almost twice as many excerpts. For example, in Listening Study 2, a participant's data are only excluded if they rated **four or more** *Inexpressive* stimuli higher than 4 for expressivity. This exclusion criterion excludes data for seven participants, leaving only the data of 13 participants for analysis.

Hypotheses

For Listening Study 2, my hypotheses are as follows:

1. CFE will receive significantly higher ratings in expressivity than RM;

2. CFE+P will receive significantly higher ratings in expressivity than CFE;

3. There will be no significant difference in ratings for tempo/rubato for BM and CFE;

4. BM will receive significantly higher ratings for timing/articulation than CFE;

5. CFE will receive significantly higher ratings in dynamics than RM;

6. CFE will receive significantly higher ratings in dynamics than BM.

Results

The results can be seen in Figure 6.4, with the BFA results as follows:

1. Anecdotal evidence ($BF_{10} = 0.267$) that CFE did not receive significantly higher ratings in expressivity than RM;

2. Moderate evidence ($BF_{10} = 4.160$) that CFE+P receives significantly higher ratings in expressivity than CFE;

3. Anecdotal evidence ($BF_{10} = 0.860$) that there is a significant difference in ratings for tempo/rubato for BM and CFE;

4. Strong evidence ($BF_{10} = 0.0779$) that BM did not receive significantly higher ratings for timing/articulation than CFE;

5. Strong evidence ($BF_{10} = 0.0956$) that CFE did not receive significantly higher ratings in dynamics than RM;

6. Extreme evidence ($BF_{10} > 100$) that CFE receives significantly higher ratings in dynamics than BM.

Based on the results above, I decided to conduct a post-hoc analysis to determine if the CFE+P model rates significantly higher than RM (since the original CFE did not). The results of this post-hoc analysis show strong evidence ($BF_{10} = 30.0$) that *CFE+P receives significantly higher ratings for expressivity than RM.*

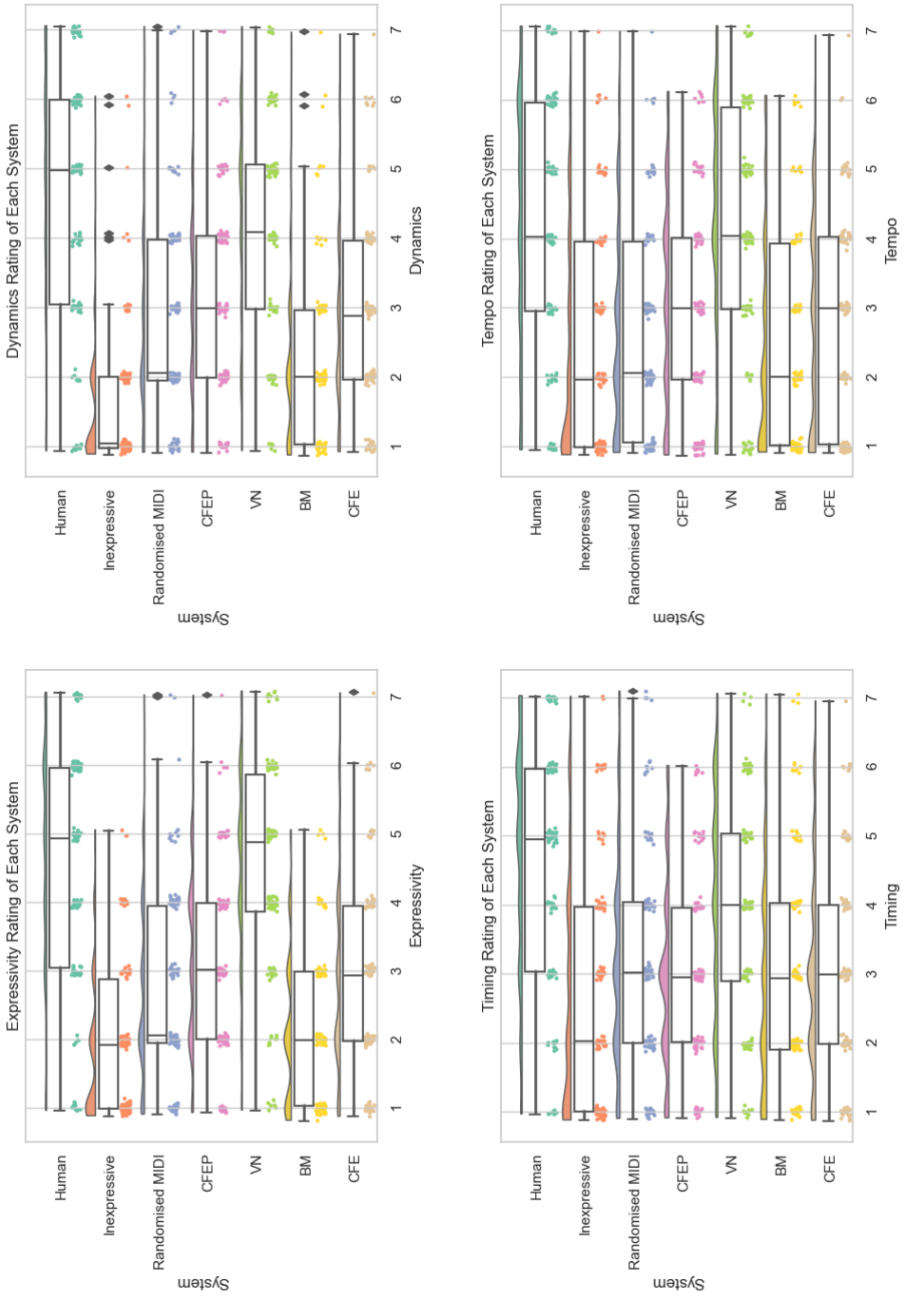

Figure 6.4: Raincloud plots of my results for Listening Study 2; expressivity, dynamics, timing and tempo ratings. The scores are integers between 1 and 7. Jittering of data means that there may be slight discrepencies between scores of the same value, i.e. if they sit around the same number but on either side.

6.4.3 Listening Study 2 Discussion

The results of this study indicate that my model, which includes sustain pedal prediction (CFE+P) rates higher than BM (Cancino-Chacón, 2018), CFE (the non-pedal using model), and both the RM and *Inexpressive* baselines for expressivity, dynamics and tempo. RM is evaluated better than BM and *Inexpressive* for expressivity and dynamics, but performs equally to BM and *Inexpressive* for tempo. Furthermore, RM performs better than *Inexpressive*, and equally to BM, CFE and CFE+P for timing. The timing result is unexpected, considering the random distribution of expressive values used by this system. Additionally, as with Listening Study 1, VirtuosoNet (Jeong et al., 2019a) is evaluated almost on par with Human Performer across the 30-sec excerpts, the difference being slightly more distinguishable than found previously (see Figure 6.3).

Qualitative data from free-text boxes in Listening Study 2 reveal: a participant senses that RM's distribution of velocity values is random and arbitrary compared to a human; longer excerpt lengths can affect the ratings positively; pedal use could be masking perception, making it harder for participants to discern expressive intent from timing mistakes.

- "Some of the [RM] performances have wide dynamic ranges, but are not necessarily expressive as a result of this because the louder/softer notes are clearly arbitrary" [Participant 7];
- "The midi [*Inexpressive* system] is far too perfect with its timing, I hate it" [Participant 15];

- "I felt as though some of these extracts were too short to get a true measure of their expression – had these gone on for longer, I would probably have considered some of them to be more expressive because there would have been more room for contrast" [Participant 7, commenting on no specific system];
- "It is quite hard to tell which is good because a lot of them cover their mistakes with the pedal" [Participant 15, commenting on no specific system].

6.4.4 Quantitative Analysis of Predictive Distributions

Predicted Velocity Distribution Analysis

In addition to listening studies, I extract the velocity values for all of the notes in four of the outputs from each system (one piece per composer), in order to compare the distribution of each system's predictions to the distributions evident in the human performance data. Throughout the remainder of this section, I refer to CFE and CFE+P collectively as CFE, since their velocity predictions are the same. I use the velocity model here, as each model in the experiment predicts velocity, and it is a stand-alone feature that is easily extracted from all the systems' expressively rendered output and from the human performance MIDI.

Figure 6.5 shows these distributions as raincloud plots, where the cloud (a half-violin plot) denotes the distribution of the data, under which a boxplot shows the min, max, median and inter-quartile range for the data (with diamonds showing outliers). Finally, under the boxplot is the "rain", which is the raw data with vertical jittering for visibility.

When comparing the half-violin plots, CFE is the most similar to the human performances, compared to the randomised baseline, VirtuosoNet and Basis Mixer. VirtuosoNet's general distribution is closer to both CFE and human for the Haydn and Schubert pieces, but not for Bach or Chopin, even though it performed almost on par with the human performances in the listening study. The CFE boxplots for three of the four pieces (Bach, Schubert, and Haydn) show that CFE has the most similar min, max, and inter-quartile range to the human performances in these cases.

When interpreting the jittered raw data (the rain) for CFE, it is notably stacked into vertical lines compared to the other systems. This is because when predicting a note's velocity based on the input OPD, the model predicts 0-31 (where each of these numbers is multiplied by 4 to become the standardised velocity value used in MIDI (0-127).

All of the above suggest that while the transformer-based architecture used here did not outperform VirtuosoNet in the listening studies, it has better learned the distributions that occur in the training data, or at least as far as the velocity data is concerned in this limited example.

Contextualising Timing Adjustment Distribution Analysis

Due to the inherent nature of performance timing, in which global/local tempo changes, and individual note timing adjustments themselves are used together to render a performance as expressive, it is difficult to separate these aspects in a way that makes them comparable across systems. Fortunately, during data preprocessing to train CFE, I created the PSBM, in which I estimate note-wise ontime and offtime adjustments for every performance in

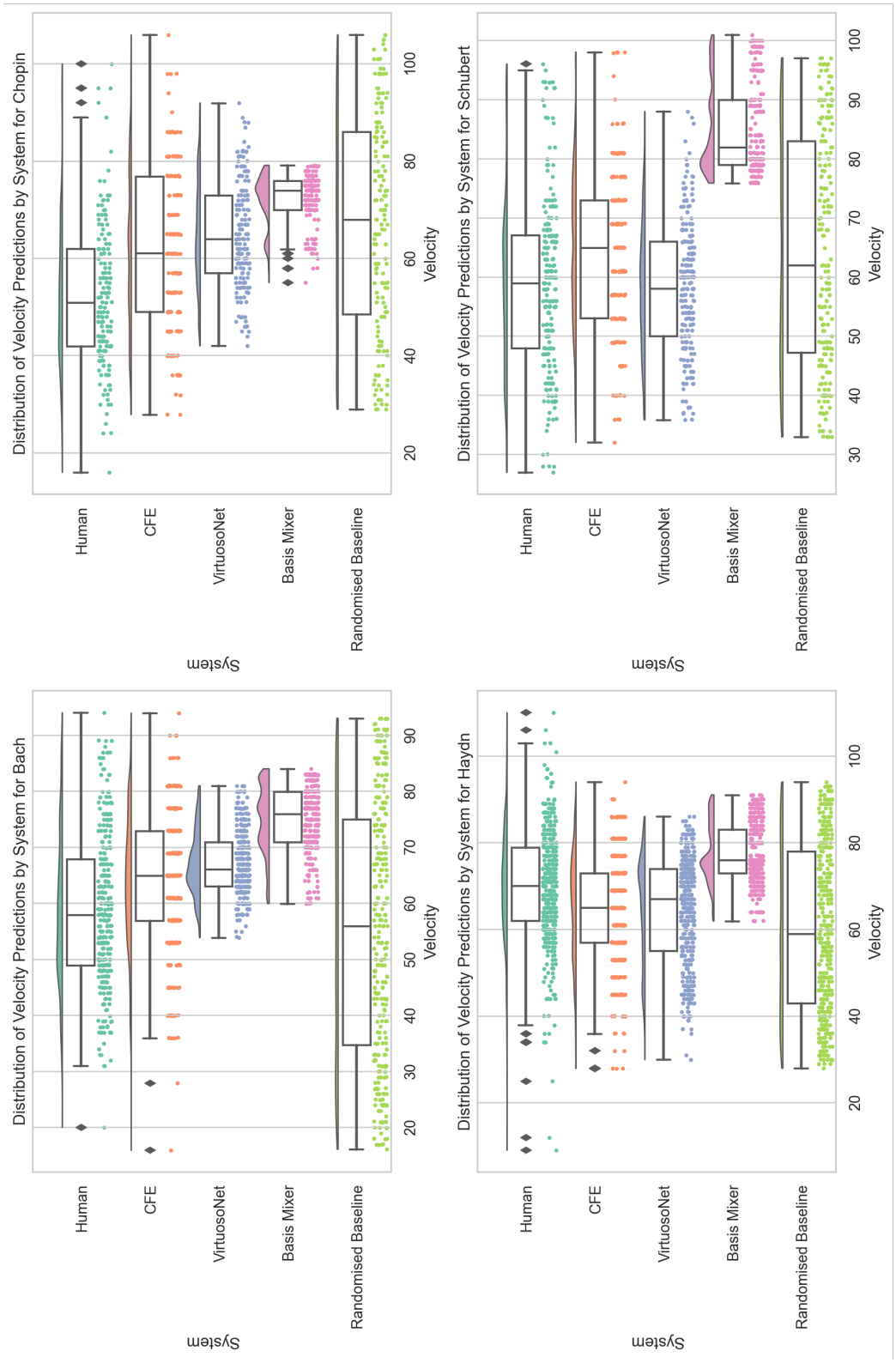

Figure 6.5: Raincloud plots of the distributions of dynamic predictions for each systems across four stimuli.

the dataset. This means that I can compare the estimated note-wise ontime and offtime adjustment values of the human performances to those predicted by CFE. Furthermore, the randomised baseline only makes adjustments to note timings and not to beat-wise tempo, so these can be included in a comparison. Unfortunately, it was not possible to separate note-level timing changes from expressive beat-wise tempo adjustments for the Basis Mixer and VirtuosoNet outputs, however, so these models are omitted from the following analyses.

As in the velocity distribution analysis above, I compare the extracted ontime and offtime adjustment values of CFE and the randomised baseline to those extracted from the human performances of the same pieces. In this example, rather than using shorter excerpts that are the same as the listening study stimuli, I use the same pieces, but their full-length versions. I do because (1) the PSBMs are made for the full performances, and (2) to see how CFE performs across full pieces (containing more notes than shorter excerpts).

Ontime Adjustment Distribution Results

Figure 6.6 shows the distribution of ontime adjustment values for each note across the same pieces, as in the velocity distribution results, but all the notes in the full piece rather than in the shorter excerpt. For this analysis, each plot shows a single, pseudo-randomly selected performance from each composer for each compared system. This was done because the dataset is heavily skewed towards a few composers, and including all performances would have produced an uneven and cluttered visualisation. Selecting one representative

piece per performer provides a clearer and more interpretable illustration of the data. Figure 6.6 shows that the CFE models have learnt the general distribution for the Chopin piece well, but that the randomised baseline is actually closer to the human distributions for both Bach and Haydn (where the curve of the distribution is more pronounced). For the fourth excerpt (Schubert), my model has a median of 0.0 beats, whereas the median of human performance is 1.0 beats. Concerning the human performance of Schubert, whereas CFE is trained on an amalgamation of many performances in the ASAP dataset and it makes sense that the median is 0.0 beats as it would not have characteristics of an individual performer, the human performance on this plot is only representative of one performance from the dataset, so will be heavily affected by that one performer's expressive interpretation of the sheet music they were given, which brings the median of offtime adjustments to 1.0 beats.

This component of CFE has the highest loss ratings (see Table 6.2), which is reflected in these plots containing some extreme outliers: this will appear in expressive renderings as the model plays some notes too late, leading to perceivable, disjointed timing.

Offtime Adjustment Distribution Results

Figure 6.7 shows the distribution for offtime adjustment values for each note across each piece in the raincloud plots. This shows a wide range of values across human performances that are expected due to individual performer differences. By comparison, my offtime adjustment model has learned to sit around the median 0.0 beats (similar to the human performances of Bach

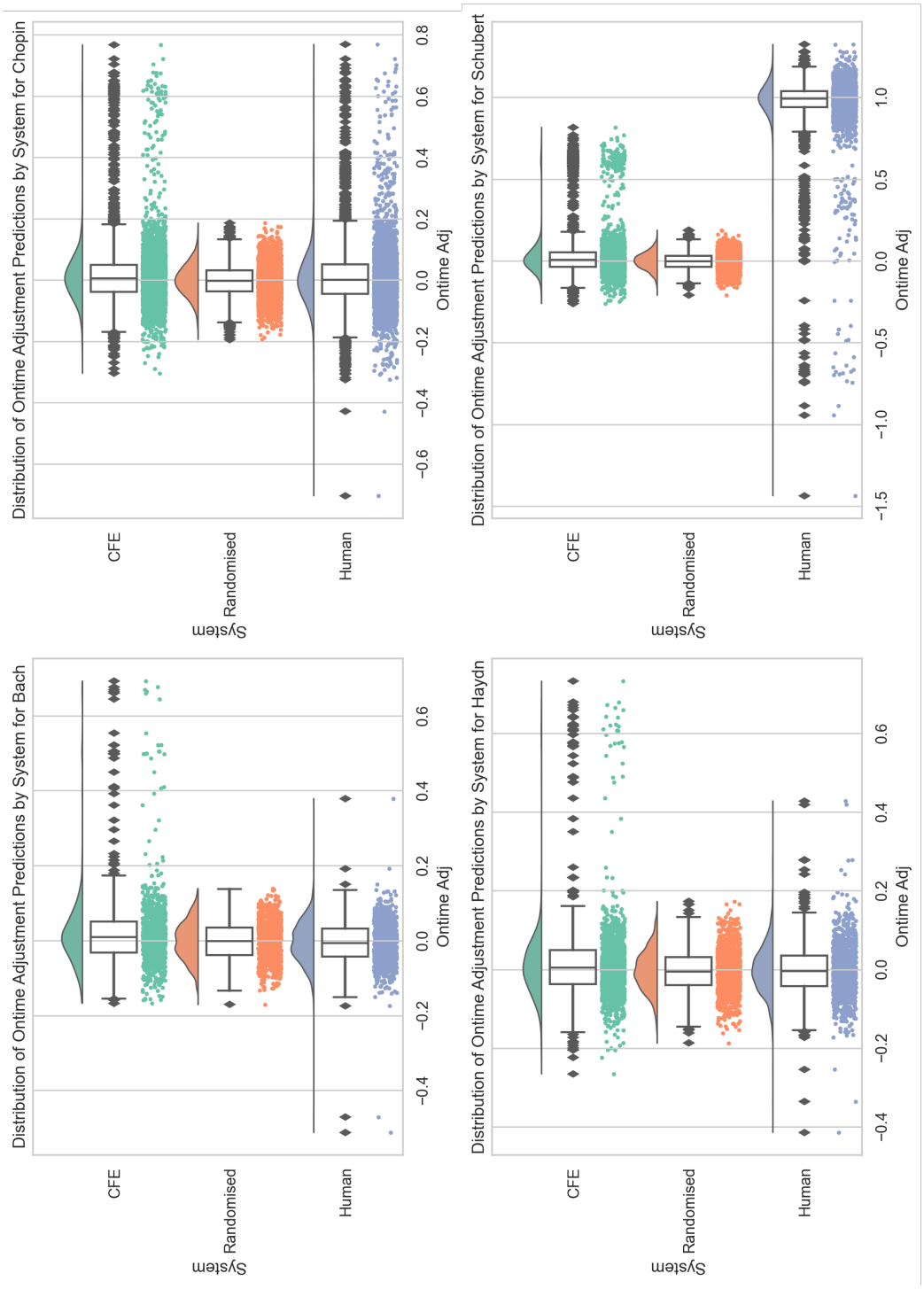

Figure 6.6: Raincloud plots of the distributions of ontime adjustments made to each note in a piece for four stimuli.

and Haydn), and it has learned a predisposition for shortening notes, apparent in the bimodal distributions with the smaller mode at a value less than zero beats. It is worth bearing in mind that when the sustain pedal is suppressed, a performer can release keys earlier than notated without them being perceived as short, which may affect these expressive choices.

Quantitative Analysis Discussion

From analyses of ontime, offtime and velocity prediction, I can see that while these three targets have been preprocessed within the training data in the same way (i.e. with the same model parameters, and in a predictable range of 0-31, where each number represents a specific element within a given possible range), that the velocity model has most successfully learnt the shape of the data more effectively than the timing models. This could be for two reasons: firstly, the addition of predicted velocity as an input feature in the timing models may be skewing the learning rather than empowering it as I hoped; secondly, and more likely, is that the size of the dataset is too small and too filled with outliers (which is more realistically an issue with timing as timing changes are much more granular than the velocity predictions).

Figures 6.6 and 6.7 in particular demonstrate a large number of outliers not only in CFE's predictions but also in the training data from the human performances. Additionally, other factors present in the data that support this concern include: one of the four performances being examined having a mean time closer to 1.0 than 0.0; Figure 6.7 being pruned due to such a large range of outliers; the above issues with synchronisation as outlined in Sec 6.2. The further investigation – and potential solving – of these issues is

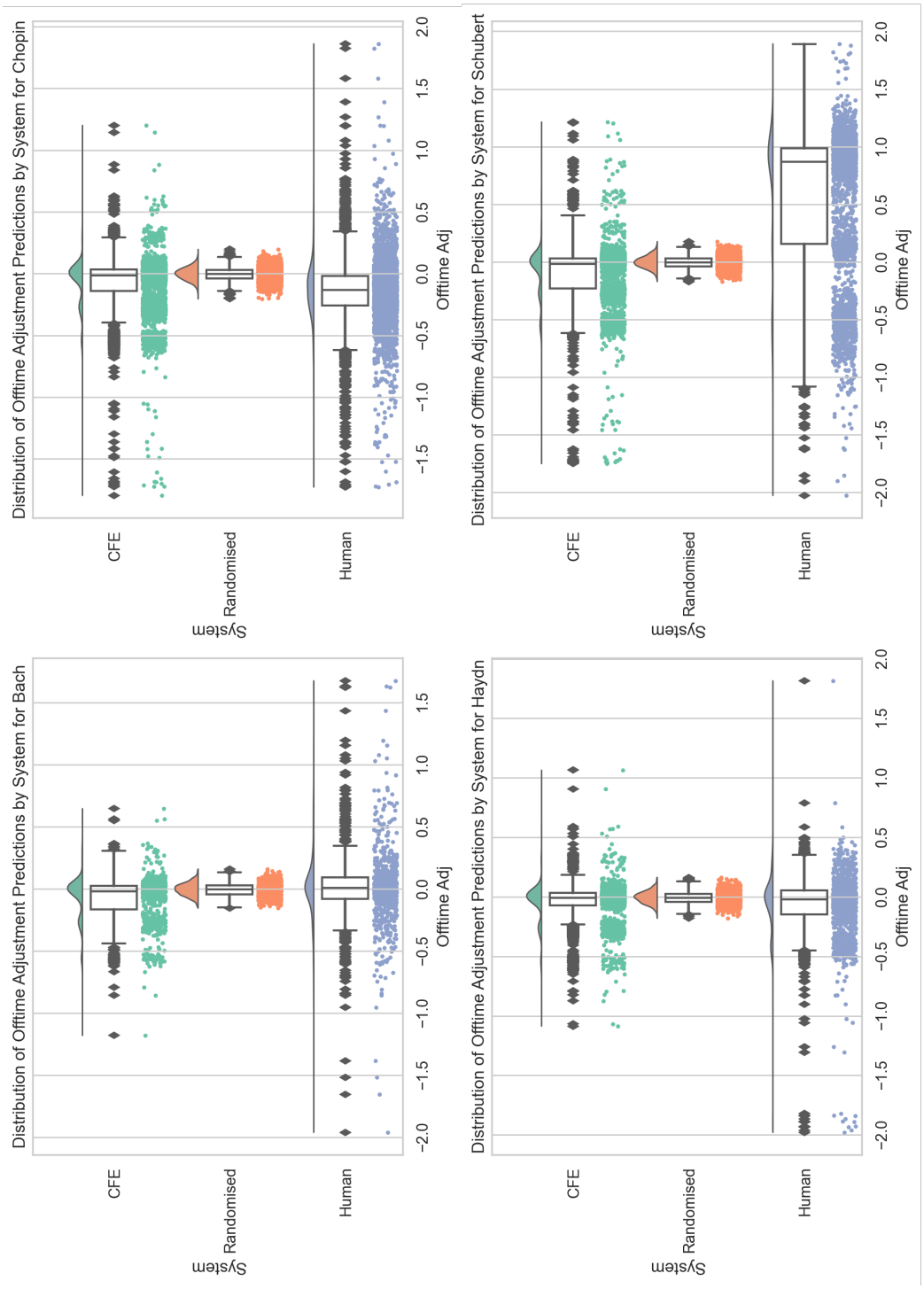

Figure 6.7: Raincloud plots of the pruned distributions of offtime adjustments made to each note in a piece for four stimuli.

beyond the scope of this work, and I leave it to future research to build upon this, but this analysis does show that at least concerning velocity modelling, the transformer architecture is powerful enough to capture the overall shape of the data quite well (even if input features are limited to OPD only).

6.4.5 Attention Mapping

The necessary technical details for this analysis are located in Chapter 3. In this analysis, I use a full version of Bach's Fugue BWV 862 (Bach, 1722a) from my listening studies. A piano roll plot of most of the piece is given in Figure 6.8), with the initial occurrence of the subject and its restatement towards the fugue's end highlighted in red. I focus on the subject and these two occurrences in particular because (1) they constitute thematically important (repetitive) material that one hopes a transformer model would recognise, and (2) the restatement is in the bottom voice, which a human performer may choose to play more strongly than the top voice. In Figure 6.9, I visualise the attention scores for all notes in the subject's restatement that score above the threshold described in Chapter 3, connecting them to the note/s to which they are paying attention, in a manner similar to (Huang et al., 2018).

In layer 1 (whose attention values do not feature in Figure 6.9), it seems the largest attention values associated with each note are directed back towards the first five notes of the excerpt. In other words, this layer does not appear to have learned about local/global structures in note sequences: to predict the velocity of the current note, it relies on properties of the first five notes.

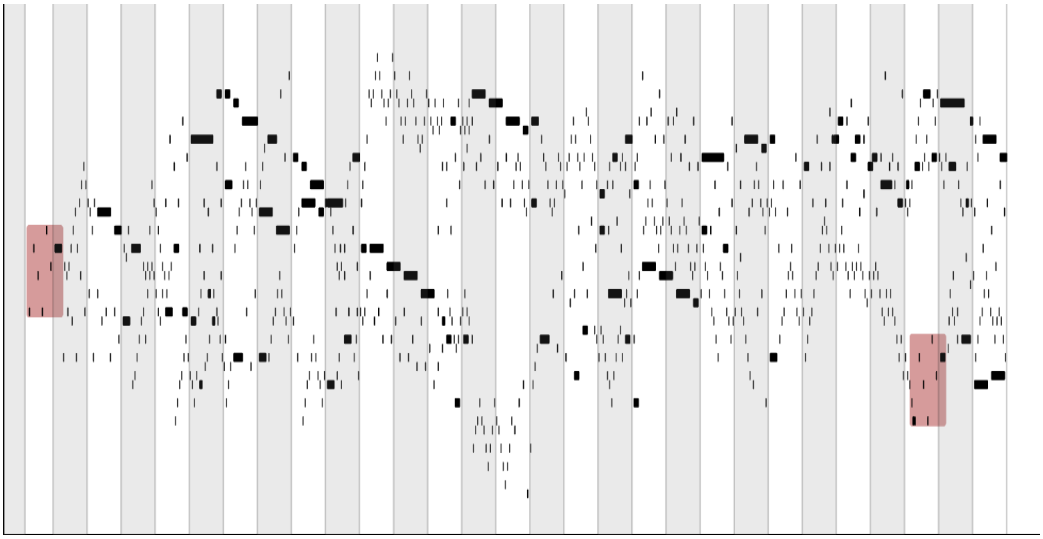

Figure 6.8: A smaller visualisation of the piano roll highlighting the fugue subject in Bach's BWV 862 (Bach, 1722a). A larger version appears in Chapter 3.

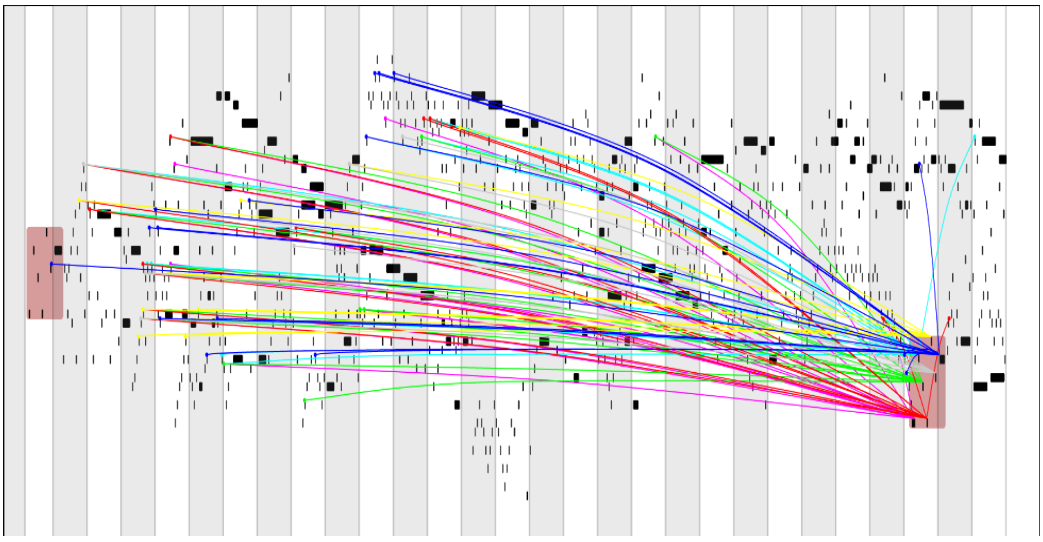

Figure 6.9: A visualisation of the attention that is being paid in layer 6 of my transformer velocity model to previous notes in Bach's BWV 862 (Bach, 1722a), originating from the fugue subject in the left hand. Each line colour denotes a different origin note in the restatement in the left hand. A larger version appears in Chapter 3.

In layer 6 (whose attention values are indicated in Figure 6.9), it is quite common for the largest attention values associated with a given note to be directed towards earlier notes that are near the registral peaks of voice lines in the piece. This can be seen in the way that lines emanate from the seven oblongs associated with the fugue subject's restatement, and go to oblongs associated with small collections of notes earlier in the piece. In contrast to layer 1, this layer appears to have learnt a strategy in which successful velocity prediction involves looking back to small groups of notes in different sections of the piece.

From a musicological point of view, the excerpt plotted comprises the beginning of the fugue (seven-note subject highlighted in the left red box in Figure 6.9), up until the restatement of the subject in the lowest voice in bar 27 (seven notes highlighted in the right red box). If the velocity transformer model had really learned something profound about musical structure, then the largest attention values associated with notes belonging to the subject's restatement ought to implicate notes associated with earlier occurrences of the subject or answer. Studying the lines in Figure 6.9, I can see that only one of the notes in the restatement looks back to a note from the initial fugue subject (where it looks to the note directly preceding itself in the earlier pattern). Apart from this, the transformer does not appear to have learned anything profound about musical structure, at least as demonstrated through the self-attention maps. This suggests that while a transformer-based model achieves well-rated results in expressive rendering tasks, more research is required to establish whether such architectures are capable of learning about more profound aspects of musical structure.

Attention Mapping Discussion

A potential factor affecting the velocity model's ability to detect musical repetition in the fugue subject is the maximum sequence length used during training, which was 255 tokens (see Table 6.1). While the model performed successfully on shorter excerpts in the listening studies, capturing a segment containing the fugue subject may require a longer sequence, as the repetition occurs 27 bars apart—likely exceeding the trained sequence length. By the sixth layer, the counter-subject begins to attend to the initial subject, as shown in Fig. 6.9, but it is unclear whether this reflects a true learned relationship or is coincidental. Training a model with additional layers or a longer maximum sequence length may clarify this, but such investigations are beyond the scope of the present work and are left to future research.

While further work is needed to improve the model's sensitivity to long-term musical repetitions (as far as fugue subject and counter subject are concerned), it does audibly capture structurally significant moments such as phrase endings. However, this section specifically examines fugue subject and countersubject mapping and does not reflect the model's performance on other aspects of musical structure relevant to expressive rendering.

6.5 General Discussion

How expert human performers imbue music with expression, and how computational models of expressive performance emulate this process, are topics of interest to the academic fields of music performance science (MPS), music-cognitive science, and music informatics research, as well as in multiple application domains, such as musical co-creation with AI, and video game audio. While research into the creation of expressive rendering algorithms is ongoing, unlike music generation research, there has been no commercial application of these algorithms in professional music-making software, even if there is a desire for such a tool (see Chapter 5). Furthermore, while these algorithms continue to yield improving results, the industry standard, non-machine learning solution that is available in DAWs has not been adopted in research to serve as a basis of comparison. This lack of a standardised baseline complicates comparative evaluation between experiments, and this issue is compounded by a lack of systematic evaluation within the field (since the discontinuation of RENCON). Further complicating the problem is a lack of available open-source models "in the wild." Understandably, these factors make it difficult to conduct a fair comparison between models, in experimental settings with high internal validity. This is especially true when considering that many models utilise wildly different data sets, architectures, and require varying input features.

Not only is there an academic-scientific imperative to establish a more ecologically valid methodology for comparison between expressive rendering algorithms in future work, but to consider the many potential applications for expressive rendering algorithms, which could innovate and support cre-

ative workflows. OPD-focused models (such as the ones introduced in this chapter), by requiring minimal input features, can be applied in situations where a composer is working with a piano roll in a DAW, and intends the DAW export to represent some or all of the final expressive musical artefact. The application of these algorithms can support creative workflows, as it can be cumbersome and an impediment to have to draw in tempo, timing, and velocity information for each bar/note (Roberts et al., 2019), or where a tool like Cubase's logical editor requires the devising of rulesets by the user. Furthermore, while it is not explored in this chapter, it is possible to implement a system with minimal input requirements into a video game engine in order to improve procedural/generative music systems.

As demonstrated in Chapter 2, models that do not use score cues (those aimed at computational models) often require human annotations (Hashida et al., 2010, 2006; Grindlay and Helmbold, 2006; Bresin et al., 2002). This requirement makes them inaccessible to specific domain applications, such as inside of a video game engine, where a human cannot interact with and annotate files at runtime, and as mentioned above, human-in-the-loop interactions can be cumbersome. Although CSEMPs using human annotations do not necessitate being fully annotated to function, this does add an extra step which affects potential users due to the need to use extra software compared to models that use score annotations that are computed automatically, thus not affecting users and increasing the barrier to entry for video game composers to utilise such tools (Weir et al., 2021; Worrall and Collins, 2023; Weir, 2017). While human annotation is not a large impediment in a DAW setting, the focus of this research is to introduce a model that can take lim-

ited input and be useful immediately when applied to generative symbolic music in a video game engine, or in a DAW at the hands of a composer. This informed the focus of CFE/CFE+P on low-level features for note-wise models, and mid-level features for beat-wise models. Additional steps such as annotations or adding score cues, also decrease the perceived ease-of-use when thinking back to technology acceptance models presented in Chapter 5.

With such motivation in mind, the first algorithm I present in this chapter is called Cue-Free Express (CFE). This model consists of an ensemble of transformer neural networks focused on tempo, timing, and dynamics, and requires only minimal input that can be found in MIDI data for the maximum breadth of application. In this chapter, I pose the question of whether this cue-free model can outperform an inexpressive baseline (see Listening Study 1), and I find that CFE does receive significantly higher ratings for expressivity compared to the inexpressive baseline, but it is short of the cue-based system VirtusosoNet (Jeong et al., 2019a) and also the professional human performers. I use non-parametric Bayes factor analyses to test my hypotheses (van Doorn et al., 2020), which is an improvement upon frequentist hypothesis testing (allowing for testing of meaningful non-differences between systems), and could be of benefit in future research. Based on these results, I then augment the initial architecture with a fifth model that predicts sustain pedal use (CFE+P), and pose the questions: can either of my models outperform the industry standard, non-machine learning baseline; and how do these OPD-focused models compare to existing neural network models (Cancino-Chacón, 2018; Jeong et al., 2019a)? In Listening Study 2, I find that while my original CFE model only outperforms the inexpressive baseline, CFE+P

outperforms the industry standard, the inexpressive baseline, my initial CFE model, and performs on par with the cue-informed feedforward neural network Basis Mixer as it is found "in the wild". This improvement is not enough to close the gap between my efforts and VirtuosoNet/human performers, but it provides valuable insight into the advantage of including pedal predictions in expressive rendering algorithms. I also provide evidence to demonstrate the potential advantages of architectures leveraging self-attention for modelling expressive performance tasks over architectures such as feedforward neural networks.

Thus, I re-initiate the interesting and application-relevant challenge of generating expressive renderings *using only the pitch and quantised start and end-times of notes as input for prediction* and compare my systems to those models that are available for comparison "in the wild" and the industry standard baseline, and finish by outlining some limitations and ideas for future work in this domain.

6.5.1 Limitations

A limitation of this research is that there are three reasons that the Basis Mixer may not have scored higher than CFE+P: a lack of pedal predictions in the Basis Mixer (as it scored similarly to CFE); the MusicXML files used for these studies may contain fewer cues that are necessary for better prediction of basis functions (the supplementary method used to perform expressive renditions using the basis mixer); the Vienna 4x22 dataset may not be large or diverse enough to compete with my selection from the ASAP dataset (even though they are approximately equal once I filter for synchronised data).

Another limitation is that I cannot synchronise all the quantised expressive representations present in the ASAP dataset. The dataset contains separate quantised (e.g., MusicXML and MIDI from score) and expressive (e.g., MIDI from human performer) representations, and then (semi-automatically-calculated) beat annotation files that should make it possible to bridge or synchronise the two. However, I find that some of these annotation files lead to predictions of performed notes given the score notes that do not exist within a reasonable time threshold, and so I work only with the subset of pieces for which synchronisation F_1 values are above .9. Addressing this limitation may lead to future modelling improvements, because I would be able to work with a larger training dataset. This also means that, while a broad range of periods, keys, and time signatures are used for musical excerpts in this chapter, all the pieces do have quite similar tempi (allegro). This means that I cannot guarantee that these findings will generalise well to more extreme tempi.

Another limitation is that VirtuosoNet's (Jeong et al., 2019a) output never predicts a sustain pedal value of less than 20 (on a scale of 0 for sustain pedal up and 1-127 for the pedal being marginally through totally down). Therefore, in this chapter, I systematically map each instance of sustain pedal ¡= 20 to 0 before exporting MIDI to audio for the listening studies. Otherwise, the software interprets VirtuosoNet's pedal as always being down to some degree, and all notes are sustained for as long as the excerpt lasts. This does not sound like a successful expressive performance, and it is something that requires attention in future work. Neural networks often fail to predict rare but musically significant events (e.g., in the tails of

distributions that they learn), which on this occasion leads to spurious pedal predictions. This change in output likely caused it to rate higher than it would otherwise have, which I felt was important to represent VirtuosoNet fairly in my listening studies.

A limitation relating to pedal modelling and ratings in this research is that only two systems use pedal models. This makes for a difficult inference of whether scores are impacted by too much pedalling, disproportionate pedalling, or the style of pedal use used (i.e., pedal used for Bach, which would realistically not be played with a pedal by a professional).

Another shortcoming of this research is that the velocity mapping of MIDI files can be different depending on the instrument on which they were performed/recorded (Bresin et al., 2002). While the VirtuosoNet model used in this study and my models (CFE and CFE+P) are trained on the same data (ASAP, which are recorded and mapped on a Disklavier), the version of the Basis Mixer used in this study is trained on the Vienna 4x22 dataset, which was recorded on an SE290 system. This means that there could be some differences in output dynamic predictions across models.

6.5.2 Future work

As well as considering deep learning approaches to cue-free expressive rendering, it is important to comparatively evaluate against non-deep learning approaches, such as linear regression models. I would like to explore this in future work, to more fully establish whether deep learning approaches provide any advantage over non-deep learning approaches in this domain, as similar research into comparing deep and non-deep learning approaches for

music generation has found that the performance gap between them is small and not statistically significant (Yin et al., 2021, 2022).

Music performance science literature says surprisingly little about the order in which performers make decisions regarding dynamics, timing, tempo and pedalling, and the way in which those decisions are made. This is not surprising, given that the process is highly complex and likely differs between performers, whereas in my computational approach, I have had to decide on a fixed order. Multiple cognitive models exist in this domain and Parncutt and McPherson (2002, pp.135–138) for example states that "when musicians speak of sight-reading, not all of them have the same activity in mind", and that performers would automatically "take care of the decoding of pitch and duration information", while other processes may be more conceptually driven, "interfacing with what a performer has learnt and stored in long-term memory". This suggests that expressive choices relating to cues are based on learned behaviour. While expert performers may also employ look-ahead sight-reading strategies and anticipate structurally salient moments during performance, such anticipatory behaviour itself arises from long-term training and internalised stylistic knowledge, rather than from immediate decoding alone. However, the minimal model of music reading suggests that verbal and music notation are combined and transcoded in the production of instrumental playing instead of being understood separately (Schön et al., 2002).

Wolf (1976) supports this idea of a combined understanding by stating that performers use cues and recognisable patterns in music to guess aspects of the performance/note content and condense down all the infor-

mation into manageable blocks. These theories suggest that decisions are not necessarily made in a specific order based on note-level information, but rather by looking at larger patterns of note content and the *score cues* encompassing passages, i.e., how players arc their tempo for phrases bearing deccelerando/accelerando markings (Gabrielsson, 1987).

With this lack of findings and cohesion in the MPS literature, for this paper, I use an order of models that made sense for inference. Namely, an ensemble of transformer neural networks where the dynamics of notes are predicted first, conditioning timing, and, separately, tempo and pedal predictions (see Figure 6.2). It would be interesting to explore different network architecture configurations in future, with a view to determining the role this plays in affecting the success of expressive rendering (i.e. timing/tempo models preceding dynamics and pedalling or vice versa).

In conducting this research, I made initial explorations in using Gaussian-uniform transformations during inference, to increase the frequency with which my transformer models predict tokens from the tails of distributions – to "take more risks" or utilise stronger contrasts with regards to expressive rendering decisions. This was motivated by how human performers use (and listeners perceive) contrast in music performance, and take risks such as extreme changes in dynamics or tempo, as opposed to making changes that can be quantified as being drawn from near a distribution's mean.

As an example of "risk taking" and a final, more high-level thought, I return to the Bach fugue (BWv 885) (Bach, 1742a) excerpt from my listening studies. The fugue subject is present in the top or soprano part at the beginning of the excerpt, and again in the tenor (or lower) part towards the end

of the excerpt. The human performer emphasises its appearance with dynamic values that are greater than those of contemporaneous, higher-pitched notes – which could be said to go against a general observation of expressive performance that the highest notes in a texture should be played louder, typically because this is where the "melody" resides.[7] This is a relatively mild risk for a human performer to take – to go against the general observation or rule of expressive performance and instead adhere to a stylistically appropriate rule that the appearance of a thematically important element or repetition occurrence ought to be highlighted, even if it occurs in a lower voice. Considering the dynamic values of the VirtuosoNet output, I see that the occurrence is quieter than the contemporaneous higher notes.[8] VirtuosoNet appears to have learnt the general rule (top-of-texture notes should be played loudly) but presumably not the rule that would be more stylistically appropriate in this scenario (emphasise thematically important note collections). My models also adhere to this same generalised rule.

Existing work on pattern discovery is sufficiently advanced that repetitive elements, such as fugue subjects, can be automatically extracted from a musical score (Collins et al., 2013, 2016; Giraud et al., 2015). As such, future work may consider how such score cue and score marking information could be automatically derived and incorporated within a neural network-based expressive rendering system, rather than manually annotated, in order to further advance performance modelling. Crucially, such approaches would aim to generate cue-like information similar to score cues directly from learnt distributions over MIDI files, allowing expressive rendering to benefit from

[7]This is Human 01_02 in the Supplementary Audio.
[8]This is VirtuosoNet 05_02 in the Supplementary Audio.

higher-level structural awareness without requiring pre-marked score representations or manual XML annotation, as assumed by current cue-based systems. Although this work adds to converging evidence that automatic expressive rendering shows promising results for short excerpts, there remains substantial scope for improvement before such systems can produce full-length performances rated comparably to professional human interpretations. This is likely due, in part, to human performers' sensitivity to, and articulation of, hierarchical and repetitive structures unfolding over longer musical timescales (Dai et al., 2022).

6.6 Chapter Summary

In conclusion, in this chapter, I introduce the first transformer-based CSEMP models and evidence that an ensemble of transformers using self-attention and positional embeddings for tokens in a sequence can be used to expressively render inexpressive MIDI data, *using only the pitch and quantised start and end times of notes as input for prediction.* These results demonstrate that this model can outperform both the industry standard and inexpressive baselines, while performing on par with an existing FFNN model. Additionally, I demonstrate that with small edits to the pedal predictions, an existing model (VirtuosoNet) can perform on par with human performances over 30-sec excerpts. Furthermore, I propose that future research use a model based on the "Humanise" function from Logic Pro X as the industry standard to set a precedent for fair and valid comparison across experiments.

The algorithms developed in this chapter are capable of addressing a factor affecting the adoption of music generation in video games (namely, lack of expression or nuance – see Chapter 5). It does this while also not replacing the human in the loop, and following the guidelines set out in Chapter 5. Now for the final step in this thesis, to embed this technology into a video game and see how it affects player experiences.

7

Evaluating the Effects of AI Music Performance on Video Game Music

7.1 Introduction

The overarching thesis of this research is that listener fatigue is a continually growing problem in video game music and that artificially intelligent music technology can be used to alleviate it. This was established in Chapter 4 which demonstrated the presence of listener fatigue in modern video games from a developer's viewpoint. Furthermore, Chapter 5 explored the human-computer interaction (HCI) affordances required for AI to be acceptable to video game music creators. These findings informed the development of the technology in Chapter 6, which demonstrated that AI can enhance the expressive quality of MIDI by introducing variations in micro-timing, tempo, pedal use, and velocity.

To fully engage with this thesis, it is vital to test whether expressive rendering technologies impact listener fatigue *in situ* rather than within a restricted listening study context, where participants are often focused exclusively on music outside of a gameplay environment. In this chapter, my best-performing algorithm (Cue-Free Express + Pedal) is implemented within an RPG. This system is compared against others to evaluate whether:

1. Expressive modelling perceptibly affects the perceived expressive quality of background music when players are focused on gameplay;

2. Expressive modelling mitigates the perception of musical repetition by introducing micro-variations to repeating phrases;

3. Expressive modelling impacts the perceived compositional quality of pre-composed music – a concern raised by composers in Chapter 5.

In line with the initial experiments shown in Chapter 6, my model is compared against a non-expressive baseline, an industry standard baseline (Logic Pro's "humanise"), and VirtuosoNet, a hierarchical recurrent neural network that performed best compared to professional performers in Chapter 6. Although there is no scientific expectation that any of these systems will have a specific or directional effect, these systems constitute a range of approaches present within both literature and industry, and are the basis for a wide range of potential results.

The motivation for this study stems from Chapter 5, where many generative music technologies were perceived as threatening to a composer's vocational identity. Since that initial research, public perception of generative AI has grown increasingly fraught due to the training of models on copyright-protected materials and a sentiment among musicians that AI startups treat music as a "solvable" business problem rather than a creative process (Collins et al., under review). While generating entirely new musical content is a viable technical solution to fatigue, it risks clashing with the professional identity of the composer.

Consequently, this research focuses on technology designed to function within existing compositional workflows. My *Cue-Free Express& Pedal* model requires only note-wise pitch, on-time, and duration values—data available in most formats regardless of the engine. Unlike expressive score cues found in XML files, this model works with raw MIDI data. This chapter investigates whether such technology can provide sufficient variation to mitigate fatigue without overlapping with or undermining the composer's creative authority.

From a ludomusicological perspective, an algorithm capable of generating variations should logically impact listener fatigue. Standard non-computational approaches (transitional and parallel design) alleviate fatigue by changing the arrangement of music at scale. I define these as changes to the **macro-arrangement**. Macro-arrangement involves the addition or subtraction of musical material (layers or full tracks) on a scale larger than the typically note-by-note basis provided by procedural approaches. These changes typically impact phrases, bars, or measures to respond to in-game variables in a closer approximation of real-time (Stevens and Raybould, 2015). However, most applications of the parallel approach do not use AI to generate layers; they tend to re-mix existing music stems which, as explored in Chapter 4, are often insufficient for highly extended gameplay (Wilson, 2019). While the human lifespan is limited, the core issue remains that approximately 01:11:00 to 04:30:00 of music is insufficient for up to 86:00:00 of gameplay if repetition is to be avoided. The issue with generative solutions, however, is that the creation of new material is potentially problematic for professional creators, as it can alter the identity of their music, impact perceived quality, and overlap with their vocational identity (Worrall and Collins, 2023).

However, a comparison between music-making workflows and those of sound designers reveals a distinct alternative. While sound variations can be created using neural networks, as seen in Barahona-Ríos and Collins (2021), Barahona-Ríos and Pauletto (2020), and Donahue et al. (2019), it is significantly more common for sound designers to avoid repetition by introducing pseudo-random micro-variations in timing or pitch between layers of a sound effect (Vachon, 2009). Although a sound effect is a shorter sonic experience than a musical loop, there is a strong argument that these types of micro-variations can mitigate listener fatigue within music in a manner similar to sound design, provided they are sensibly implemented.

Expressive rendering has not yet been explored empirically in a video game context, nor has its potential to alleviate fatigue. By providing small-scale changes to micro-timing and velocity, expressive rendering has the potential to shift how specific notes sit within a mix. As such, I posit that this technology is capable of altering the **micro-arrangement** of music in a way that parallels modern sound design strategies.

Stepping back from expressive rendering, existing procedural music systems often generate streams of notes, thereby affecting a composition's **macro-arrangement**. The changes presented by an expressive rendering algorithm, by contrast, impact the arrangement to a much smaller degree. Expressive rendering operates entirely at the micro-arrangement level, as the fundamental note content remains the same; therefore, the tonality, style, and motifs of the piece remain consistent. To use a concrete example from the fugue analysis in Chapter 6: if the notes of a bass line are rendered louder than the contemporaneous melody, they typically "cut through the mix." This would

not change the musical notes within the macro-arrangement, nor would it likely affect the tonality as all the note content would remain consistent, yet it would significantly alter how the notes are perceived by the listener. This research addresses the unexplored gap of whether such micro-arrangement variety can successfully reduce the perception of repetition.

While Curtis Roads' nine timescales of musical structure provide a useful framework for understanding perceptual organisation, the concepts of macro- and micro-arrangement introduced here describe interaction-driven modes of change specific to video game music systems (Roads, 2004). These terms do not replace Roads' taxonomy, but instead align with it at different perceptual levels. Macro-arrangement corresponds most closely to Roads' **Macro** and **Meso** timescales, insofar as it involves phrase-level and layer-level restructuring over seconds to minutes (Roads, 2004). Micro-arrangement, by contrast, operates primarily at the level of the **Sound Object** and the upper **Micro** scale, shaping the expressive timing and dynamics of individual note events without altering higher-level musical form (Roads, 2004).

7.2 Learning Aims and Reasoning

Extant research outside the domain of video games has demonstrated that expressive rendering can enhance the expressive quality of music, thereby increasing listener enjoyment. However, as outlined in Chapter 6, participants in these studies are often music students who are predisposed to notice subtle performance nuances. It is vital to acknowledge that the active listening practised by musicians differs significantly from the peripheral listening inherent to gameplay, where the player's primary focus is diverted. This shift in attention necessitates an exploration into how such algorithms affect the perception of musical expressive quality within an interactive environment. Furthermore, expressive rendering literature rarely incorporates non-musician participants, an omission that lacks ecological validity given that the majority of the gaming population likely does not possess formal musical training.

Additionally, as identified in my interviews, composers express concern that AI could negatively affect the perceived quality of a composition rather than merely the performance. While generative systems might introduce erroneous notes, the changes proposed in this research occur on a micro-arrangement level, reducing the risk of tonal dissonance. Nevertheless, there is an argument that deviations in note timing could potentially disrupt rhythmic phrasing or cause clashes. Determining whether micro-timing shifts inadvertently alter the perceived compositional intent is, therefore, a key avenue of this investigation.

Finally, it remains an open question whether expressive rendering can impact the perception of repetition (listener fatigue) within a video game, as the existing corpus of expressive rendering literature has largely ignored the implications of looped audio. With this in mind, there are three main learning objectives for this chapter:

- To measure the degree to which expressive modelling improves the quality of background music when participants are focused on gameplay as opposed to solely musical stimuli.
- To measure the degree to which expressive modelling affects player perception of musical repetition.
- To measure the degree to which expressive modelling affects the players' perception of the quality of the compositional quality of the music used in the experiment.

7.3 Methods

7.3.1 Creating a Video Game as a Testing Environment

Game Design Choices

Chapter 4 identified RPGs and live-service games as genres characterised by significantly longer-than-average playtimes. While the specific genre used in a small-scale experiment might seem secondary given the artificially constrained duration of the study, an RPG remains the most ecologically valid choice for investigating listener fatigue. However, it must be acknowledged that this environment cannot be fully ecological; a localised experiment cannot replicate the multi-hour immersion of a commercial title, as participant time is naturally limited. Consequently, this study creates a functional facsimile of a gameplay experience. This is a necessary methodological compromise discussed further in the Limitations & Future Work section.

Recalling the definition of RPGs from Chapter 1, many narrative and structural elements outlined by NeverKnowsBest (2024) such as complex character creation and branching dialogue are beyond the scope of this research. Nevertheless, the core mechanical requirements, including weapon statistics, and combat determined by statistical probability (e.g. damage variance), are both achievable and representative of the genre (NeverKnowsBest, 2024).

Subgenres of the RPG offer varying mechanics: Action RPGs (ARPGs) prioritise real-time combat, whereas Japanese RPGs (JRPGs) often utilise turn-based systems. Despite these differences, both subgenres rely on simi-

lar statistical backbones for combat outcomes. Furthermore, while Massively Multiplayer Online RPGs (MMORPGs) feature specific social and networking requirements, these are extraneous to the current study and have been excluded to maintain focus on the auditory-ludic relationship.

The within-participants nature of the experimental design further informed these choices. To avoid participant fatigue and time-intensive repetition across conditions, the game design focuses on a concise, high-intensity gameplay loop. Players explore a small area and engage in real-time combat within a 2D environment that emulates retro aesthetics; this approach is not only computationally efficient but also allows for rapid iteration.

The choice of real-time Action RPG combat is particularly strategic. Findings from Chapter 6 suggested that active focus leads to higher discernment regarding expressive renderings. By utilising an Action RPG model, players are prevented from pausing mid-combat—as they might in a turn-based JRPG—thereby ensuring their primary cognitive load remains dedicated to gameplay rather than passive music criticism. Theoretically, this maintains the music's role as background stimuli, allowing for a more authentic test of how expressive rendering impacts fatigue during active play.

Figure 7.1: A screenshot of gameplay seen in a 2D *Legend of Zelda* (Nintendo, 1986) clone, made in PyGame, produced by Clear Code. Zelda (and this clone) is an Action RPG where players participate in combat in real-time rather than instanced battles.

PyGame Programming

The majority of commercial video games are developed using Unreal Engine or Unity. Although these engines support Python scripting through various plugins, their significant computational overhead can negatively impact performance when running concurrent deep learning models. By comparison, PyGame functions as a wrapper for the SDL2 C++ library, providing a lightweight environment that facilitates faster iterative design within a native Python workflow. Furthermore, the availability of comprehensive open-source templates, specifically those emulating the retro RPG styles appropriate for this research (see Figs 7.2 and 7.3) allowed for the rapid development of a functional testing environment without the unnecessary complexity of a 3D engine.

Figure 7.2: A scene from the inspiration for the video game in this study, namely *The Legend of Zelda: A Link to the Past* (Nintendo, 1992b).

This research utilises an existing open-source template. Fig 7.2 shows a screenshot of *The Legend of Zelda: A Link to the Past* (Nintendo, 1992b), which incorporates AI-controlled monsters, multiple biomes (distinct visual regions), and player movement and combat mechanics including player attacks, and varied weapon and magic skills.[1] I subsequently modified this project to include player death animations, dedicated user interface screens for controls, a score system, and player respawning. The latter was implemented to ensure that the player is reset to the level's starting point upon death, thereby maintaining the integrity of the gameplay loop and ensuring consistent exposure to the musical stimuli for the required amount of time.

[1]This demo was created by the YouTube channel Clear Code, and the code is here.

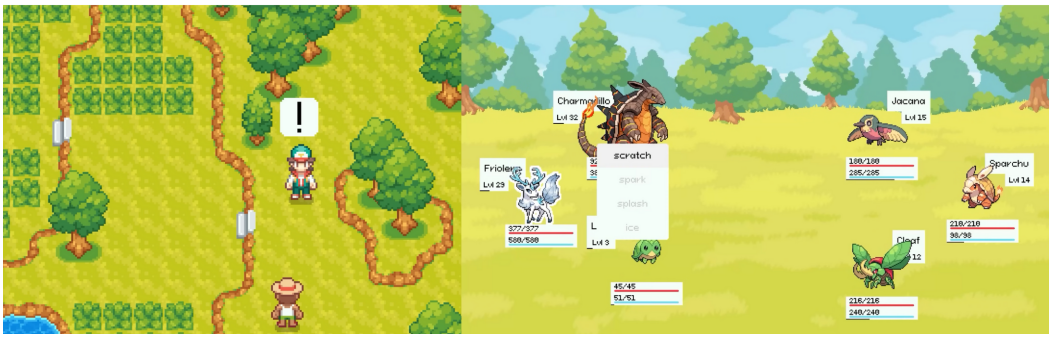

Figure 7.3: A screenshot of gameplay seen in a 2D *Pokemon* (Game Freak, 1996) clone, made in PyGame, produced by Clear Code. Pokemon (and this clone) is an RPG where players explore the world in real-time (left), but battles take place in instances (right).

Although other RPG templates are available in PyGame—such as a *Pokemon* clone featuring instanced, turn-based combat, which is more reminiscent of legacy titles (e.g., *Final Fantasy VII* and *Chrono Trigger* (Square, 1995)). While such systems remain a popular stylistic choice in contemporary releases (such as *South Park: The Stick of Truth* (Obsidian Entertainment, 2014) or *Persona 5 Royal*), the real-time Action RPG framework was deemed more appropriate for preventing the aforementioned cognitive breaks that turn-based systems allow.

Fig 7.3 illustrates an example of instanced combat (right) and a discrete exploration phase (left), reminiscent of the kernel system used in *Final Fantasy VII*. RPGs utilising turn-based, instanced combat frequently exacerbate listener fatigue due to their structural design. While the transition between gameplay states is intended to break up the experience, it often interrupts ludic flow and frequently reintroduces the same battle theme from its starting point. Additionally, whilst contemporary titles employ distinct music for combat instances, failing to adhere to this standard in a study

might break immersion; conversely, this is less of a concern for the real-time combat model utilised here. This decision permits players to remain within the uninterrupted flow of a primary gameplay loop while experiencing music continuously, maximising exposure to the musical assets within a condensed experimental timeframe.

7.3.2 Implementation of AI Music Technology

Two primary methodologies were considered for integrating music AI elements into the video game. The first involved real-time inference specifically, the prediction of performance qualities and MIDI rendering during loading screens, followed by MIDI playback via a custom-built sampler within the game engine (Option A). The second approach involved running inference and conducting quality checks on musical stimuli prior to the study, subsequently "pre-baking" these assets within a Digital Audio Workstation (DAW) and implementing them as standard loops (Option B).

Option A represents a more technologically integrated deployment of expressive rendering, paralleling the use of dynamic systems in commercial titles such as *Rise of the Tomb Raider's* Dynamic Percussion System (Brown, 2019; Lamperski, 2015), the implementation of expressive rendering in research discussed by Haaland (2020), or the model for expressive rendering in video game engines proposed Livingstone (2008). However, this approach introduces significant technical fragility. Any prediction errors could adversely affect the participant experience, introducing unintended variables into the experiment. Furthermore, real-time inference during loading screens reduces the total exposure time to the musical loop, which is particularly problematic

for recruitment when experimental sessions risk exceeding one hour. Finally, such fragility highlights an inherent challenge in procedural music systems: the necessity for robust timing mechanisms to prevent detrimental impacts on the player experience (Weir et al., 2021).

By comparison, the primary limitation of Option B is that it is not a scalable deployment solution for listener fatigue in a commercial context. In a production environment, including numerous pre-baked variations would significantly increase the audio data footprint, which is a common constraint for developers. However, within an experimental framework, Option B is the most rigorous choice. It facilitates the controlled testing of performance renderings on the perception of repetition, composition, and expressivity without the confounding risks of system instability. This methodology aligns with contemporary music generation research that utilises pre-rendered assets to evaluate generative music during gameplay (Tchemeube et al., 2023).

Consequently, this research utilises Option B. Expressive renderings were generated prior to participant recruitment, and MIDI performances were rendered within *Cubase 14* (Steinberg Media Technologies, 2024) using the *Intimate Grand Piano* sample library (Spitfire Audio, 2018).[2]. This approach ensures that all tracks maintain uniform volume and audio fidelity, thereby isolating the performance variations as the primary independent variable. Audio stimuli are available for review on the Open Science Framework (OSF) project page.[3].

[2]https://labs.spitfireaudio.com/packs/intimate-grand-piano

[3]To ensure access to files in the event of OSF crashing, they can also be found alongside all raw data here

Musical Stimuli

Each experimental condition utilised a different expressive rendering of a fixed selection of bars from the track "Secret of the Forest" (Mitsuda, 1995) by Yasunori Mitsuda, originally composed for *Chrono Trigger*. This excerpt was selected because its aesthetic profile aligns with the retro visual style of the testing environment, thereby enhancing the study's ecological validity. Furthermore, both the CFE+P and VirtuosoNet models were trained on Western classical piano repertoires, rendering this material an appropriate stylistic proxy.

While "Secret of the Forest" (Mitsuda, 1995) is not strictly a piano composition, nor does it originate from the Western classical tradition, early Japanese video game music is widely recognised for its roots in progressive rock, which is a genre itself heavily influenced by Western classical structures and idioms (Collins, 2008; Tate, 2021). Consequently, the harmonic language and phrasing of the excerpt are sufficiently compatible with the training datasets of the evaluated models to ensure reliable expressive rendering.

7.4 Experiment

I evaluate CFE+P and the other three systems (an inexpressive baseline, a randomised baseline, and VirtuosoNet) through a comparative evaluation that takes place over a single study. The experimental design, hypotheses, and results of the study are detailed below. For replicability, all data collected are made available here.

7.4.1 Experimental Design

The study employed a within-participants design wherein all twenty participants experienced all four conditions for ten minutes each. Each gameplay session was followed by a five-minute silent period to allow participants to rest their ears and complete the relevant surveys in Qualtrics. The order of these conditions was randomised for each participant to negate potential order effects and biases. Prior to the study, all participants reviewed the information sheet, provided informed consent, and received a three-letter code to facilitate data removal post-anonymisation if requested.

To ensure data consistency, participants completed a preliminary test question. During this phase, they: were introduced to specific musical terminology; rated an excerpt from a previous study across the chosen metrics to familiarise themselves with the Likert scales; and were encouraged to ask clarifying questions before the formal stimuli sessions began.

Two computational expressive rendering algorithms (VirtuosoNet (Jeong et al., 2019a) and the proposed CFE+P model) were evaluated against two comparison points: randomised ("humanised") MIDI and inexpressive (non-expressive) MIDI.

Participants were recruited from the undergraduate, postgraduate, and staff populations at the University of York. The study received ethical approval from the Department of Computer Science Ethics Board, and participants received £20 in compensation for one hour of their time. Of the twenty participants, seven self-identified as musicians or music creators. Their experience levels varied: two had one to three years of experience; one had three to five years; two had five to ten years; and two had ten or more years of experience.

The participants rated each condition along three dimensions:

1. Overall expressivity of the performance;
2. Overall perceived level of musical repetition;
3. Overall quality of the musical composition.

A free-text box was also provided for qualitative feedback. To avoid bias against artificial intelligence, participants remained unaware of the provenance or generation method of any audio stimuli until after their participation was complete.

7.4.2 Hypotheses

Utilising Bayesian data analysis allows this research to quantify evidence in favour of the null hypothesis (H_0), whereas frequentist testing can only fail to reject it. Based on literature and the findings from Chapter 6, the following hypotheses are proposed:

1. The Inexpressive baseline will receive significantly lower ratings in Perceived Expressive Quality and Perceived Musical Quality than all other systems;

2. VirtuosoNet will receive significantly higher ratings in Perceived Expressive Quality compared to CFE;

3. There will be no difference in ratings between CFE+P and VirtuosoNet in Perceived Musical Quality or Perceived Repetitiveness;

4. The Inexpressive baseline will receive significantly lower ratings for Perceived Repetitiveness than all other systems;

5. CFE+P will receive significantly higher ratings in Perceived Expressive Quality and Perceived Musical Quality than the Randomised baseline;

6. VirtuosoNet will receive significantly higher ratings in Perceived Expressive Quality and Perceived Musical Quality than the Randomised baseline.

Notably, while these hypotheses provide a directional framework for analysis, the primary value of this research lies in its exploratory nature rather than the definitive results. This chapter serves as an initial investigation into the perceptual impact of expressive rendering within the complex cognitive environment of active gameplay, where listener attention is inherently divided.

7.4.3 Results

The results of this study are illustrated in Figures 7.4, 7.5, and 7.6. Figure 7.4 indicates that the CFE+P model was rated on par with all other systems regarding the perceived expressive quality of the musical performance. While CFE+P exhibited a broader distributional range in participant scores, the interquartile ranges and means across all four systems were approximately equivalent. This finding is noteworthy; in previous listening studies, CFE+P and VirtuosoNet consistently outperformed both baselines in the expressivity metric. Although comparable scores between CFE+P and the randomised baseline were anticipated based on prior results, it is surprising that VirtuosoNet performed only on par with the inexpressive systems within a video game setting.

Free-text responses from participants expand upon these findings, supporting the notion that the cognitive load of gameplay may diminish the listener's sensitivity to fine-grained expressive nuances that are otherwise apparent in focused listening contexts.

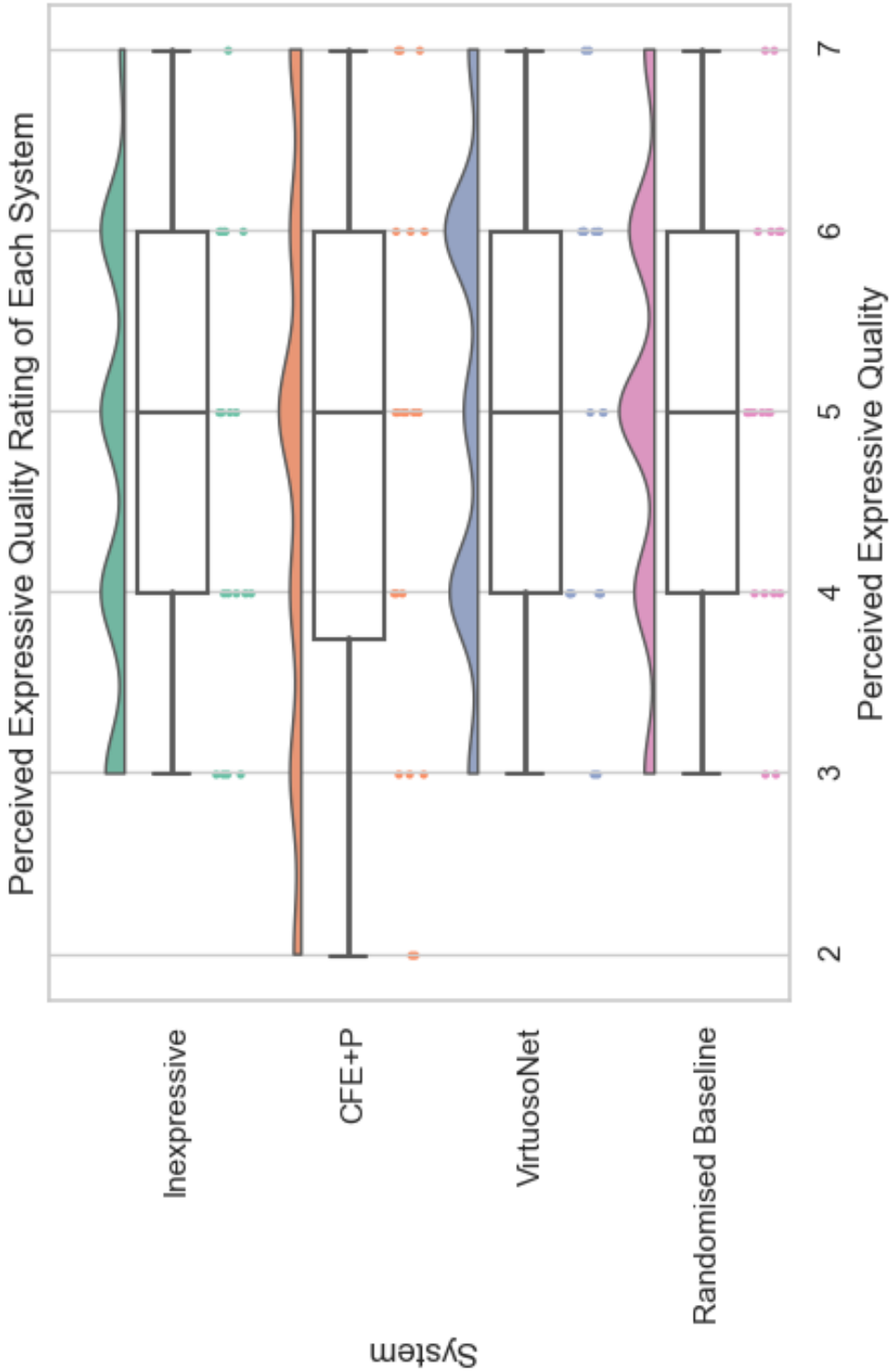

Figure 7.4: Raincloud plots of the results for the Gameplay Study, denoting ratings for Perceived Expressivity of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side.

Figure 7.5 illustrates how participants perceived musical repetition during the ten-minute gameplay segments. Scores were recorded on a Likert scale where 1 represented "extremely repetitive" and 7 indicated "not repetitive". In this metric, the CFE+P model yielded the least favourable results, with its interquartile range sitting at the lowest point of 2 out of 7. VirtuosoNet and the randomised baseline performed relatively similarly in terms of mean and interquartile range, though VirtuosoNet exhibited a wider variance. Interestingly, the inexpressive baseline achieved a higher mean score, suggesting that it was perceived as the least repetitive system in this specific context.

Figure 7.6 indicates that performance style and system variation significantly impact the perceived quality of the musical composition itself. While non-musician participants may experience difficulty in distinguishing between compositional structure and performance nuance, the data suggests that the CFE+P model was rated least favourably in this metric. The inexpressive baseline and VirtuosoNet performed comparably, exhibiting similar mean scores and upper quartile ranges, whilst the randomised baseline yielded an identical mean score.

These results suggest a potential "penalty" for highly varied expressive models in a gameplay context: where performance deviations are perceived as rhythmic or harmonic inconsistencies, participants may attribute these flaws to the composition rather than the rendering system.

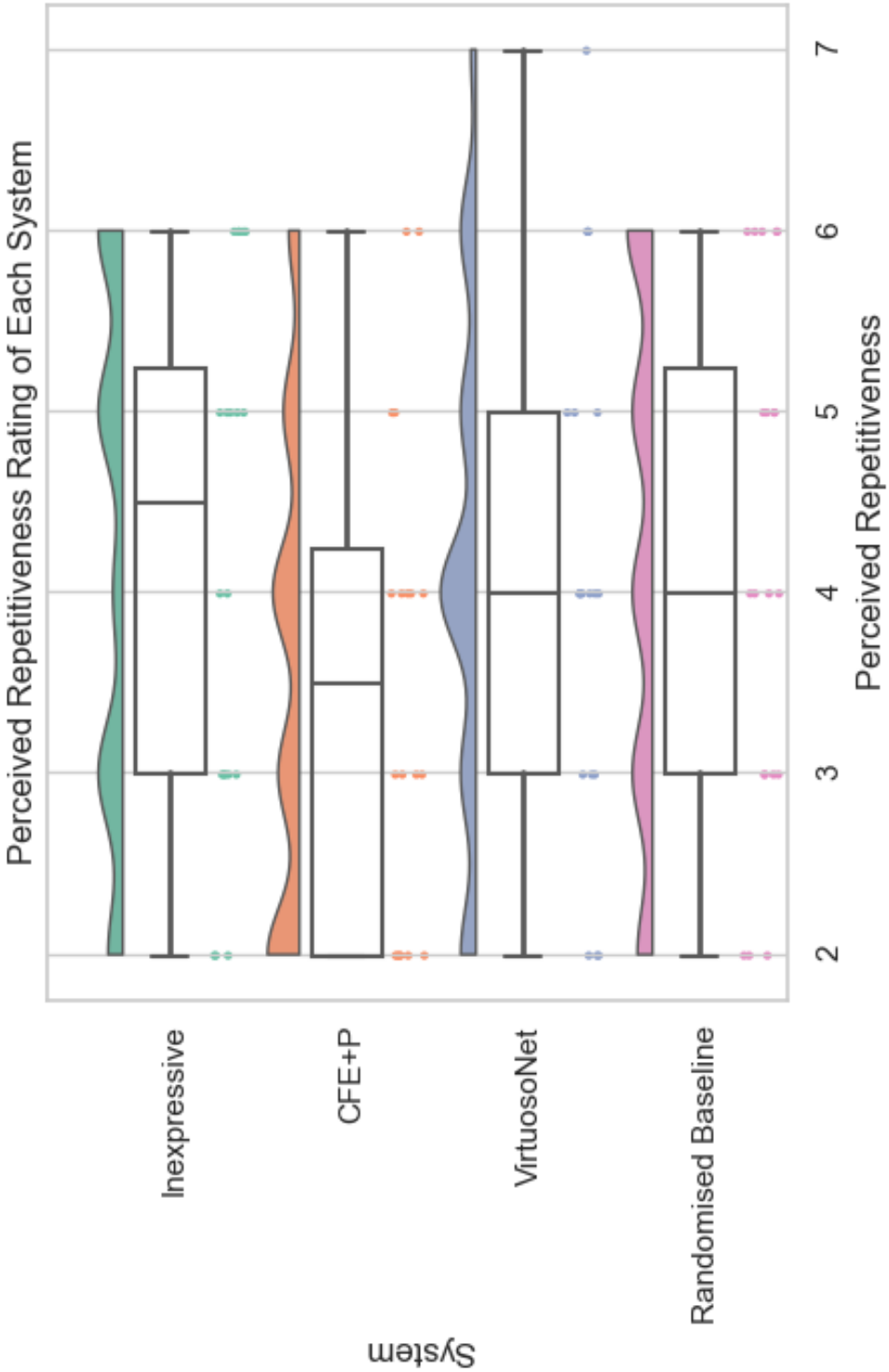

Figure 7.5: Raincloud plots of the results for the Gameplay Study denoting ratings for Perceived Repetitiveness of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side.

The outcomes of the Bayesian Factor Analysis (BFA) are detailed below, utilising multiple pairwise comparisons to test each hypothesis.

1. The Inexpressive baseline will receive significantly lower ratings in Perceived Expressivity and Perceived Musical Quality than all other systems;

    - A: Moderate evidence ($BF_{10} = 0.232$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Expressive Quality than CFE+P.
    - B: Anecdotal evidence ($BF_{10} = 0.610$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Expressive Quality than VirtuosoNet.
    - C: Anecdotal evidence ($BF_{10} = 0.750$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Expressive Quality than the Randomised baseline.
    - D: Moderate evidence ($BF_{10} = 0.120$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Musical Quality than CFE+P.
    - E: Anecdotal evidence ($BF_{10} = 0.889$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Musical Quality than VirtuosoNet.
    - F: Anecdotal evidence ($BF_{10} = 0.358$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings in Perceived Musical Quality than the Randomised baseline.

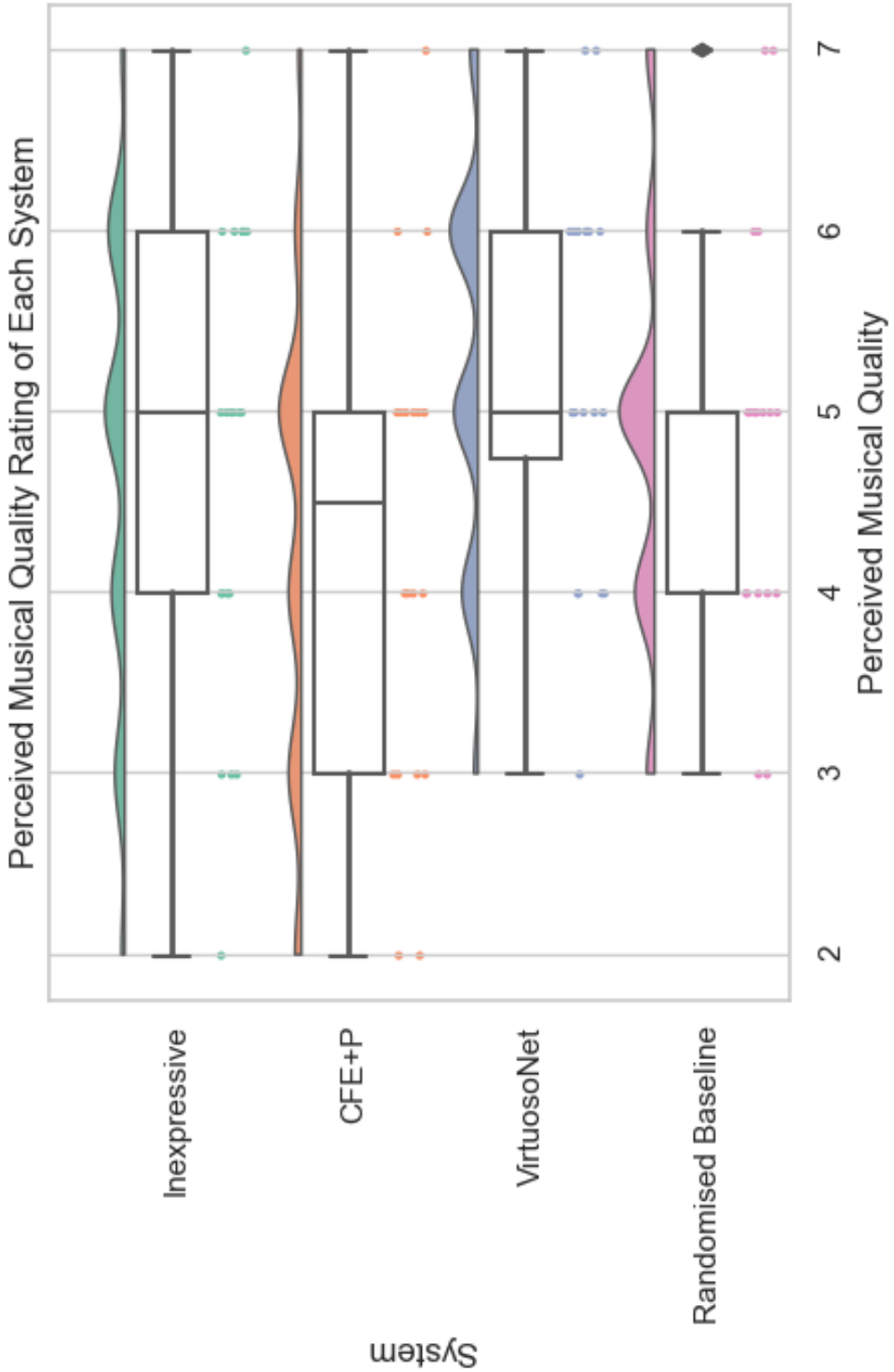

Figure 7.6: Raincloud plots of the results for the Gameplay Study denoting ratings for Perceived Musical Quality of stimuli across each condition/system. Jittering of data means that there may be slight discrepancies between scores of the same value, i.e. if they sit around the same number but on either side.

2. VirtuosoNet will receive significantly higher ratings in Perceived Expressivity compared to CFE+P;
   - G: Anecdotal evidence ($BF_{10} = 0.830$) for H^0 indicating that VirtuosoNet did not receive significantly higher ratings in Perceived Expressivity than CFE+P
3. There will be no difference in ratings between CFE+P and VirtuosoNet in Perceived Musical Quality or Perceived Repetitiveness;
   - H: Moderate evidence ($BF_{10} = 3.489$) for H^1 indicating that there was a difference in the ratings in Perceived Musical Quality received by CFE+P and VirtuosoNet.
   - I: Anecdotal evidence ($BF_{10} = 0.535$) for H^0 indicating that there was no difference in ratings in Perceived Repetitiveness received for CFE+P and VirtuosoNet
4. The Inexpressive baseline will receive significantly lower ratings for Perceived Repetitiveness than all other systems;
   - J: Strong evidence ($BF_{10} = 0.089$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings for Perceived Repetitiveness than CFE+P.
   - K: Moderate evidence ($BF_{10} = 0.207$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings for Perceived Repetitiveness than VirtuosoNet.
   - L: Moderate evidence ($BF_{10} = 0.232$) for H^0 indicating that the Inexpressive baseline did not receive significantly lower ratings for Perceived Repetitiveness than the Randomised baseline.

5. CFE+P will receive significantly higher ratings in Perceived Expressivity and Perceived Musical Quality than the Randomised baseline;

   - M: Moderate evidence ($BF_{10} = 0.142$) for H^0 indicating that CFE+P did not receive significantly higher ratings for Perceived Expressive Quality than the Randomised baseline.
   - N: Moderate evidence ($BF_{10} = 0.107$) for H^0 indicating that CFE+P did not receive significantly higher ratings for Perceived Musical Quality than the Randomised baseline.

6. VirtuosoNet will receive significantly higher ratings in Perceived Expressive Quality and Perceived Musical Quality than the Randomised baseline.

   - O: Moderate evidence ($BF_{10} = 0.232$) for H^0 indicating that VirtuosoNet did not receive significantly higher ratings for Perceived Expressive Quality than the Randomised baseline.
   - P: Anecdotal evidence ($BF_{10} = 0.732$) for H^0 indicating that VirtuosoNet did not receive significantly higher ratings for Perceived Musical Quality than the Randomised baseline.

7.4.4 Experiment Discussion

The discrepancy between the current findings and those established in Chapter 6 highlights the profound impact of cognitive load on musical perception. While previous listening studies afforded participants the mental bandwidth to discern micro-timing and velocity shifts, the active requirements of combat and navigation in the Action-RPG appear to have *flattened* these distinctions for many players. This suggests that whilst expressive rendering algorithms can significantly improve the aesthetic quality of music in a focused setting, their efficacy is mediated by the nature of the player's engagement with the game.

Furthermore, the qualitative data reveals a clear divergence based on musical expertise. For non-musicians, the "inexpressive" baseline was often indistinguishable from the "expressive" AI models, suggesting a threshold of perception that is rarely crossed during gameplay. However, for those with musical training, the inexpressive version was not merely "bland" but actively distracting, as evidenced by Participant 17's comment about "tuning out" the music to maintain focus. This implies that for a subset of the gaming population (i.e., the musically literate), expressive rendering may not be a luxury, but a necessity to prevent the "uncanny valley" of robotic performance from breaking immersion.

Finally, the unexpected "penalty" regarding compositional quality (Figure 7.6) warrants further scrutiny. It appears that when CFE+P or VirtuosoNet introduced performances with high variance, players did not necessarily attribute this to an expressive performance, but rather to a flawed composition. This suggests that within the rigid, grid-based logic of a retro

2D game, highly fluid expression may be perceived as a rhythmic error rather than a stylistic choice, potentially clashing with the visual rhythm of the game's frame rate and animations.

The following responses from the qualitative free-text boxes illustrate these varying levels of discernment:

> "[The Inexpressive baseline] feels very similar to the two previous pieces I have listened to, and it is hard to say this one is any different so I gave the same/similar scores." [Participant 2].

> "[The Inexpressive baseline] sounded the same as D [the Randomised baseline] - I couldn't tell much, if any difference." [Participant 7].

> "[The Inexpressive baseline] felt less expressive than C and D which made it feel more repetitive. I didn't enjoy this version as much because of that. It was still a good piece of music but I started to tune it out after a while so I could focus on what I was doing in the game." [Participant 17].

Beyond the boxplots, the distributional density of the raincloud plots reveals that VirtuosoNet and the randomised baseline received more ratings in the higher values than the inexpressive baseline, which exhibited a more uniform distribution across the full range. This suggests that whilst expressive rendering may offer subtle improvements, they do not currently reach the threshold of statistical significance within an active gameplay context (as evidenced by the BFA results).

Figure 7.5 demonstrates that expressive models significantly influenced the perceived repetition of the music. While the CFE+P model yielded lower scores—indicating higher perceived repetition—a specific performance characteristic may have influenced this outcome. Unlike the other systems, which maintained relatively consistent tempi, CFE+P introduced a significant ritardando, effectively halving the tempo in the third section. This deviation likely functioned as a "perceptual anchor," allowing players to easily identify the loop's structure even if they were not consciously attending to the melodic content. Qualitative feedback supports this notion:

> "[In CFE+P's performance] the music felt a lot slower that time and it made it feel more repetitive." [Participant 8].

> "[CFE+P's performance] felt very emotional and the slowing down and then the tempo picking back up made me pay attention to it more." [Participant 15].

> "I enjoyed the ones with variation the most, especially different tones and volumes, even if the music was repetitive" [Participant 15].

A critical caveat regarding this metric is the implementation method (Option B), which utilised static pre-rendered tracks. Whilst these tracks contain internal expressive variation, the lack of real-time inference means that every iteration of the loop is identical. Ludomusicology literature emphasises that players are highly adept at recognising repeating patterns within game soundscapes (Vachon, 2009; Stevens and Raybould, 2015). If players can

identify micro-repetitions in sound effects, they are likely to discern identical expressive "gestures" in a repeating musical loop. It is probable that in a system where every loop was rendered with novel expressive variations in real-time, the perception of repetition would significantly decrease as it would not be noticeably repeating.

A final consideration regarding the perception of repetition is that the audible "errors" or exaggerations in the CFE+P performance may have served as negative anchors, inadvertently highlighting the loop's recurrence. While the system intended to provide human-like nuance, over-emphasised gestures or mistimed notes acted as distinctive markers that drew the player's attention away from the gameplay and toward the musical structure. This suggests that salient performance flaws do not merely lower aesthetic quality; they increase the cognitive ease with which a player identifies a repeating pattern. This idea is directly supported by a free-text response from a participant:

> "[In CFE+P's performance] some of the expression was a little bit overdone, and sounded like mistakes (notes played a bit too late) which was drawing my attention away from the game. These patterns resulted in an increased perception of repetitiveness (since my attention was focused on these moments in the musical track)." [Participant 1].

The perceived quality of the composition (Figure 7.6) appeared largely resilient to expressive rendering, with the notable exception of the CFE+P model. Despite the composition remaining identical across all conditions, CFE+P received the lowest ratings. This suggests that conspicuous performance errors such as the aforementioned tempo shifts or mistimed notes func-

tioned as negative anchors. For some participants, the distinction between performance and composition was blurred; they interpreted "amateurish" execution as a flaw in the music's writing rather than its delivery.

> "[With CFE+P's performance] the slower pace made it feel quite melancholic and sadder than the other levels. A lot more emotive than the previous levels, but I'm not sure if that made it sound more fitting for the gameplay. I would argue slightly less fitting. Partly the way the music was played, slightly out of rhythm, made it feel more amateurish, as if played with less confidence - the piece slowed down and sped up in certain areas that sounded dissonant." [Participant 20].

7.5 General Discussion

The impact of expressive rendering and music AI on player perception is a subject of significant interest in both academic and professional settings. Within research domains, this work addresses core concerns of ludomusicologists, music AI specialists, and music informatics researchers. For industry professionals including developers, composers, and audio programmers, the technology presented in this research offers a dual potential: streamlining the development pipeline and facilitating novel, dynamic player experiences.

To my knowledge, this study represents the first empirical investigation into the application of "authentic" expressive rendering technology within a functional gameplay environment. I distinguish this as "authentic" because previous attempts to integrate expressive modelling, such as Haaland (2020), utilised the technology as a ludic novelty, essentially an "audio toy" creating a "mickey-mousing" effect rather than exploring its efficacy as a solution to established design challenges such as listener fatigue.[4] By comparison, while the framework proposed by Livingstone (2008) more closely resembles the architectural goals presented in this chapter, it remains largely hypothetical and falls short of conducting empirical research within an active video game setting. This fundamentally places the work presented in this chapter between these two examples of expressive rendering work within the domain of video games.

[4]Mickey-mousing is a film scoring technique wherein musical cues precisely mimic specific on-screen actions. In Haaland (2020), expressive rendering is employed in this manner, as players directly manipulate rendering parameters via their in-game avatars.

In this chapter, I evaluated multiple systems (including the CFE+P model introduced in Chapter 6) within a bespoke Action RPG environment. Through a study involving twenty participants, I interrogated the impact of expressive rendering on player perception by addressing the following research questions:

RQ1 To measure the degree to which expressive modelling improves the quality of background music when participants are focused on gameplay as opposed to solely musical stimuli.

RQ2 To measure the degree to which expressive modelling affects player perception of musical repetition.

RQ3 To measure the degree to which expressive modelling affects the players' perception of the quality of the composition used in the experiment.

With regard to RQ1, the quantitative findings suggest that expressive modelling did not significantly alter the overall perceived expressive quality during active gameplay. However, qualitative feedback from musically literate participants suggests that performances were perceived as more melancholic, confident, or intense depending on the specific model and its performance characteristics. It is important to note that this experiment was limited to a single mood within the Action RPG genre; consequently, generalising these findings would require further cross-genre studies using varied musical styles and longer exposure periods (see further section on Limitations & Further Work).

Whilst the quantitative data showed no statistically significant shift, the qualitative commentary suggests that expressive rendering did influence the

"feel" of the music for certain players. Furthermore, isolated analysis of the seven musically experienced participants revealed a more pronounced preference for rendered music over the non-rendered baselines compared to the non-musician cohort. This implies that the current model predictions may be too subtle for a general audience during the high cognitive load of gameplay. Future research might explore exaggerated or "hyperbolic" expressive rendering to break through the sensory threshold of active play.

Notably, some participants perceived the structural nuances in both CFE+P and VirtuosoNet as being indicative of human intent. However, while CFE+P demonstrated an ability to model long-term dependencies in Chapter 6, it failed to sufficiently account for repeating musical structures within this context, specifically the fugue subject and counter-subject (Worrall et al., 2024). In contrast, VirtuosoNet's hierarchical approach explicitly models measure-level features to ensure consistent long-term performance (Jeong et al., 2019a). This suggests that CFE+P requires further iteration to match the structural consistency of SOTA algorithms like VirtuosoNet.

Finally, it is critical to observe that in several instances, the expressive rendering caused active distraction. While distraction is not a direct metric of expressive quality, music that detracts from the player's focus is generally considered a failure within the video game industry. This indicates that significant refinement is required before such technologies can be reliably deployed in commercial environments.

With regard to RQ2, this study explored whether micro-arrangement via expressive performance variations could mitigate the perception of repetition. While this approach aligns with established game audio literature (where micro-variation is a pivotal tool for alleviating repetition in sound effect design (Vachon, 2009; Forss, 2016)), the experimental data indicate that expressive timing did not yield a similar benefit for musical loops.

Counter-intuitively, the inexpressive baseline was rated as favourably as the sophisticated expressive rendering systems. While quantitative analysis failed to show any expressive system performing statistically better than the control, qualitative feedback suggests a more complex relationship: one, in fact, where expressive rendering may actually render repetition *more* noticeable. When specific phrases or "gestures" differ audibly from the surrounding material, they act as salient markers that draw listener attention, paradoxically making the loop's recurrence easier to identify. Consequently, within the constraints of this study, expressive rendering was not found to reduce the perception of musical repetition.

Further research is essential to determine whether this "cognitive anchoring" effect persists under different conditions, such as the use of more structurally consistent expressive models, alternative performance strategies, or real-time inference (Option A) where every loop iteration is unique. Additionally, longer gameplay scenarios that are more representative of commercial play experiences may be necessary to observe whether the perceived benefits of expression only emerge after the initial novelty of the salient "anchors" has faded.

With regard to RQ3, the findings indicate that expressive rendering possesses the potential to influence the perceived quality of the underlying musical composition. This result directly aligns with the concerns raised by professional composers in Chapter 5, where participants expressed apprehension regarding the potential for AI-driven musical errors to damage their music and, by extension, their musical legacy (Worrall and Collins, 2023). The "amateurish" or "dissonant" perception of certain AI-rendered passages (as noted in Figure 7.6) suggests that players do not always distinguish between the quality of the writing and the quality of the performance.

While these findings are not yet statistically definitive, they represent the first empirical investigation of this effect within a video game context and provide a clear impetus for future research. Further work is essential to explore whether these perceptions persist in systems capable of real-time inference and audio re-sampling—approaches that more closely emulate emerging commercial solutions (Brown, 2019; Lamperski, 2015). If expressive rendering is to be adopted by industry professionals, the technology must demonstrate that it can preserve the composer's intent without introducing distracting artefacts that reflect poorly on the composition itself.

7.5.1 Limitations and Future Work

A primary limitation of this study involves the potential for participant bias regarding the provenance of the musical stimuli. If participants become aware, or even suspect, that a performance is AI-generated, there is a significant risk of skewed results. In this study, qualitative feedback indicated that several participants correctly guessed the involvement of artificial intel-

ligence. This is a critical factor, as recent research has consistently demonstrated an "attribution bias" wherein music labelled as human-generated is rated more favourably than identical material attributed to AI (Shank et al., 2023; Grassini and Koivisto, 2024; Ansani et al., 2025; White et al., 2025; Wu and Holmes, 2026). Furthermore, the varying levels of musical expertise among participants likely influenced both the sensitivity of the ratings and the understanding of the specific terminology used in the metrics.

A second limitation arises from the heterogeneous nature of the participant group, which complicates direct comparisons between active gameplay and passive listening contexts. Had the cohort consisted entirely of musically experienced individuals, in turn matching the demographic of Chapter 6, then clearer conclusions could be drawn regarding how the gameplay environment specifically modulates expressive perception. The inclusion of non-musicians introduced the phenomenon of "immersion-driven inattention"; two participants explicitly noted that they "zoned out" or ceased attending to the music as they became focused on the game mechanics. While this provides high ecological validity by reflecting how most players experience games, it does diminish the resolution of the experimental data. Future research should consider bifurcating samples into expert and non-expert groups to better quantify how musical literacy affects the "immersion-perception" trade-off:

> "[I] liked this one but can't tell why, [I] got too immersed [in the game] to listen." [Participant 5].

> "[The Randomised baseline] felt quite repetitive, I stopped paying attention to it quite fast." [Participant 14].

A third limitation of this research concerns its longitudinal ecological validity. In a commercial Action RPG, a selection of three-minute tracks might be experienced over dozens of hours of gameplay. In contrast, participants in this study were exposed to audio for only ten minutes per stimulus, with tracks ranging from 1:35 to 2:09 in length. Consequently, participants heard approximately five repetitions of each piece, which is considerably fewer than would occur in a standard commercial setting. While this limitation is a pragmatic necessity as requiring multi-hour contributions from participants would be logistically unfeasible, it necessitates caution when interpreting or extrapolating the results.

The perception of expressive "gestures" as anchors or the degree of listener fatigue may evolve significantly over a longer temporal scale. As such, any observed affective qualities such as melancholy, confidence, intensity, or dissonance should be viewed as context-specific responses to the tested material rather than as generalisable emotional constants across the wider video game industry. Future research could address this gap by applying similar methodologies to longer play sessions, perhaps through remote "at-home" studies, to more closely reflect the sustained exposure characteristic of the video game medium.

A final limitation of this research is its restricted scope; the study was confined to a single musical composition within a bespoke Action RPG environment featuring real-time combat. While the findings provide valuable insights within this specific context, a substantially broader research programme would be required to establish generalisability across varied video game genres or musical styles. Future work should specifically investigate

whether similar perceptual patterns emerge in different ludic contexts, such as slower-paced strategy games or abstract puzzle games, and across diverse musical idioms. Evaluating these variables across varied gameplay environments and extended play durations would offer a more comprehensive understanding of how expressive rendering functions within commercial video game ecosystems.

Furthermore, as established in the general discussion, a critical avenue for future research lies in contrasting real-time inference (Option A) with the pre-rendered ("pre-baked") asset approach (Option B) employed here. Investigating systems that perform inference during gameplay would allow for a more ecologically valid assessment of how expressive rendering impacts perception, as real-time generation is a prerequisite for most commercial applications. Crucially, real-time inference enables an infinite variety of musical iterations; this may prove to be the definitive solution for alleviating listener fatigue, as it eliminates the "perceptual anchors" created by identical, static loops.

7.6 Chapter Summary

In conclusion, this chapter provides the first empirical investigation into expressive rendering within a functional video game environment, evaluating how performance nuance influences perceptions of repetition, composition quality, and expressivity during active gameplay. Furthermore, the study addressed a core objective of this thesis: **"to evaluate how AI-driven music systems impact the broader perception of listenability and listener fatigue"**. Across all metrics, Bayesian analyses consistently yielded evidence in favour of the null hypothesis, indicating that neither the CFE+P nor VirtuosoNet models significantly outperformed the inexpressive or randomised baselines.

Crucially, the findings reveal that static expressive variation may paradoxically increase the awareness of repetition. Rather than mitigating listener fatigue, unique expressive "gestures" appeared to function as perceptual anchors, drawing the player's attention to the loop's recurrence. This suggests that without real-time inference to provide continuous variety, expressive rendering may inadvertently highlight the mechanical nature of game audio, thereby limiting its efficacy in improving long-term listenability. The discrepancy between these results and the focused listening studies in Chapter 6 underscores the profound impact of task focus and participant expertise on musical perception.

Despite the lack of quantitative gains in this specific context, this chapter demonstrates a successful technical integration of "human-in-the-loop" AI that respects the composer's structural intent – a key requirement identified in Chapter 5.

Furthermore, this work establishes a foundational methodology for future research into real-time, generative expressive systems as a solution to fatigue. I now conclude this thesis by synthesising the overarching findings of this research programme.

8

Coda: Conclusions and Future Work

8.1 Answering the Research Question

Detailed conclusions for the technical and empirical components of this thesis—Chapters 4, 5, 6, and 7—are provided in Sections 4.5, 5.5, 6.5, and 7.5 respectively. In this final chapter, I synthesise these findings and contributions to address the primary research objectives established in Chapter 1, and identify future avenues of research to further advance the field.

As outlined in Chapter 1, the overarching research question guiding this work is: *How can machine learning be used to responsibly and acceptably augment video game music to reduce the likelihood of listener fatigue in RPGs?* Section 1.3 deconstructs this question into four specific research objectives (ROs) that have been addressed throughout this thesis:

1. RO1: To evaluate the evolution of music design and implementation in video games, and establish whether there is space for generative music in video games.

2. RO2: To evaluate and potentially impact the workflow, programming knowledge and opinions of video game composers to inform tool design and use cases of procedural music generation.

3. RO3: To explore the degree that the quality of procedural music for video games can be improved by modelling expressive musical performance.

4. RO4: To evaluate how an AI music system impacts perception of listenability, and listener fatigue in role-playing video games.

8.1.1 RO1: Evaluating the Evolution of Music Design

In Chapter 2, I provided a comprehensive overview of video game audio and ludomusicology, highlighting a distinct lack of computer music methodologies within current game music systems. By categorising existing artificial intelligence systems into expressive rendering and music generation, this chapter established the theoretical foundation for the subsequent content chapters. In Chapter 4, I utilised a comparative case study to evaluate how music design has evolved over a twenty-three-year period, assessing the musicological approach of a modern video game to identify potential "use cases" for music AI. This analysis revealed that there is a significant opportunity for computational approaches to mitigate repetition-based fatigue, particularly in gameplay segments where musical arrangement remains static despite shifting game states. Furthermore, the scarcity of existing implementations suggests that significant mitigating factors are obstructing the adoption of music AI. These findings provided the primary motivation for this thesis, highlighting the necessity for exploratory work into the limitations and practical applications of AI-driven video game music systems.

Significance

The significance of this research lies in its status as the first comparative analysis of music design between an original title and its modern remake. This novel comparative framework has already demonstrated pedagogical value, having been adopted into the ludomusicology curriculum at Royal Holloway, University of London. Additionally, this work is among the first to empirically explore the requirement for AI-driven music within contemporary game design. By framing AI as a functional musical tool through a pragmatist lens, this research aligns with and extends existing literature identifying listener fatigue as a persistent challenge in audio design (O'Donnell, 2002; Vachon, 2009; Forss, 2016; Armstrong, 2021). While select industry leaders have begun leveraging machine learning solutions (Brown, 2019; Lamperski, 2015; Weir, 2015), such applications remain exploratory. *This thesis, however, establishes a clear mandate for the evolution of music integration and design methods, specifically advocating for the adoption of AI to mitigate long-standing systemic issues such as listener fatigue.*

8.1.2 RO2: Interviewing Composers to Inform Design

In Chapter 5, I conducted semi-structured interviews with professional video game composers to explore the sociological, technological, and legal factors that may impede the use of AI-based technology in video game music. Through Reflexive Thematic Analysis (Braun and Clarke, 2006, 2019) of eleven interviews totalling 13 hours of data, 1199 codes, 250 broader codes, 30 sub-themes and five key themes were established, namely:

1. Benefits of Music AI,
2. Complex Concerns about Music AI,
3. The Clash between Creative Ego and AI,
4. No Standardisation of Support and Understanding,
5. Lack of Understanding and Trust in AI Music Technology.

These findings represent a primary contribution as the first study to explore professional video game composer sentiment regarding AI-driven music technology, paralleling similar inquiries into the perspectives of mix engineers and producers (Vanka et al., 2023; Tsiros and Palladini, 2020) and non-game composers (Knotts and Collins, 2020). Themes 1–3 address individual-level concerns, while Themes 4–5 identify systemic barriers. Analysis revealed that while composers recognise workflow benefits in an increasingly competitive industry, their concerns about AI are multifaceted. Beyond job security, participants highlighted nuanced anxieties regarding ownership, authorship, and the lack of legal frameworks for copyright. Furthermore, the data suggests that not all musical tasks are perceived as equal; certain elements, such

as melody, are intrinsically tied to creative identity, whereas utility-based elements like basslines or percussion are viewed as more acceptable candidates for automation.

The systemic themes (4–5) revealed a profound lack of standardised technical support, particularly for freelancers who lack access to the machine learning specialists required to implement modern algorithms (Abel, 2021). This is compounded by a systemic lack of trust and technical understanding, suggesting that education is a prerequisite for artistic empowerment in the age of AI.Three key findings from this RO directly shaped the trajectory of this thesis:

1. The generation of core melodic content is perceived as a threat to compositional legacy, particularly in the context of recent tools like ChatGPT, Suno, and Udio;

2. Generative music often lacks the expressive quality of human-composed assets, creating a perceived "quality gap" that could alienate players;

3. Expressive rendering emerged as a high-utility, low-threat application of AI that could address both composer workflows and the issue of repetition in video game audio.

Significance

The significance of this work lies in its status as the first exploratory investigation into the psychological, professional, and conceptual frameworks through which video game composers perceive AI-driven music technology.

This is vital for ensuring that the technical advancements presented in subsequent chapters are aligned with industry requirements and grounded in the pragmatist philosophy of addressing real-world problems. Methodologically, this research is significant for its rigorous application of Reflexive Thematic Analysis to the field of Music Information Retrieval (MIR), a domain that frequently prioritises quantitative metrics over the lived experiences of practitioners.

The findings are theoretically consistent with established Technology Acceptance Models, such as Davis (1989) and Venkatesh (2000), as well as Rogers' Diffusion of Innovations model (Rogers, 2003). Specifically, this research demonstrates that AI-driven technology in video game audio suffers from high effort expectancy and a perceived threshold of low-quality output, both of which severely limit adoption. Furthermore, this study provides a granular understanding of the idiosyncratic factors such as creative ego and reputational risk that impede innovation among professional creators, offering a roadmap for more empathetic and effective tool design in the future.

8.1.3 RO3: Enhancing Music Playback with AI

In Chapter 6, I addressed concerns regarding expressive quality by developing a novel algorithmic architecture comprising five Transformer encoders designed to imbue raw MIDI with expressive performance qualities. This approach improves the playback quality of MIDI (regardless of its generative provenance) by introducing micro-variations through expressive predictions. This research involved the development of an architecture capable of operating with the minimal input features typically found in broad musical datasets (i.e. note-level pitch, ontime and duration) to model aspects of expressive musical performance such as note-wise velocity, micro-timing adjustments to ontime and duration, and beat-wise sustain pedal prediction, and tempo adjustment. Furthermore, this work benchmarked the quality of this expressive rendering against State of the Art models (Jeong et al., 2019a; Cancino-Chacón, 2018), as well as industry-standard tools such as Logic's Humanise function. The results indicate that while the proposed model is not statistically competitive with fully informed models that rely on score cues and performance markings to support their inference, it outperforms existing industry standard tools. Crucially, it is capable of application in areas where annotation-heavy models are not viable, such as the dynamic environment of video games, where fully annotated score cues are not readily available.

Significance

The significance of this work lies in its status as the first application of transformer encoders to the task of expressive rendering. This demonstrates the efficacy of self-attention-driven networks for complex musical tasks, validat-

ing a shift away from traditional recurrent architectures in this domain which parallels work in music generation literature (Huang et al., 2018; Huang and Yang, 2020). This research also provides a proof-of-concept for how the minimal musical features inherently present in video game engines can be leveraged to inform a sophisticated architecture with promising results. By bridging the gap between high-end performance modelling and the practical constraints of game development, these findings offer a scalable method to improve the expressive quality of both human-authored and generative MIDI within real-time interactive environments.

8.1.4 RO4: Evaluating AI Performers In-Game

In Chapter 7, I integrated the developed expressive rendering systems into a Python-based RPG environment and conducted a comprehensive player study to evaluate participants' perception of musical repetition, expressive quality, and musical quality during active gameplay. The findings indicate that expressive rendering algorithms such as VirtuosoNet (Jeong et al., 2019a) and CFE+P (presented in Chapter 6) have the potential to impact all three variables, albeit in complex and sometimes contradictory ways. While ratings for expressive quality were not significantly impacted by the use of these algorithms under the high cognitive load of gameplay, the research uncovered a dualistic effect on perceived repetition: expressive variation can either mitigate listener fatigue or, paradoxically, exacerbate it by introducing salient, easily identifiable patterns that function as "perceptual anchors" during loops.

Furthermore, there was evidence suggesting that expressive rendering can influence a participant's appraisal of a composition's musical quality—either by enhancing the emotional delivery or by introducing distracting artefacts that reflect poorly on the composition itself. However, the results were less definitive than those observed in previous listening studies. This discrepancy underscores the need for future work to explore how real-time inference might impact results by providing infinite variation, versus the static approach of pre-baking musical assets. Finally, the research findings suggest that participant expertise is a critical confounding variable; future studies should bifurcate samples into expert and non-musical participants to better isolate how musical literacy influences the perception of algorithmic nuance.

Significance

The significance of this work lies in its status as the foundational exploration of expressive rendering within the video game medium, marking the first concerted effort to apply performance modelling to the systemic issue of listener fatigue. Although the quantitative findings were not definitive at this stage, they identify a high-potential research trajectory that addresses current industry shifts. As evidenced by the 2025 GameSoundCon survey (Schmidt, 2025), AI integration remains relatively rare in video game audio workflows but is demonstrating a clear upward trend. This suggests that expressive rendering is an emerging field with significant opportunities for commercial and artistic impact.

Global technological trends already highlight the appetite for AI-driven performance, from Yamaha's AI-human duets to the sustained success of virtual avatars in large-scale productions like ABBA Voyage. These precedents demonstrate how AI can facilitate novel experiences that bypass human constraints of time, physical endurance, and consistency. When applied to the ludic context, expressive rendering offers a dual benefit: it addresses pre-existing problems such as listener fatigue while enabling novel experiences where every playthrough features a unique, nuanced performancethat potentially adapts to the player's emotional state or in-game choices. Critically, this approach enhances the player experience without compromising the creative agency or job security of the original composer. As such, this research provides a vital framework for video game companies seeking to leverage AI responsibly, even as Chapter 7 acknowledges the necessity of further iterative research to optimise these strategies.

8.2 Future Work and Limitations

Future research should prioritise the application of machine learning and deep learning to functional, in-game music tasks, an area that remains underdeveloped compared to general-purpose music informatics. A critical frontier for this work is the implementation of real-time inference within commercial engines such as Unreal and Unity, where the impact of latency on system performance must be rigorously evaluated. While existing literature outlines the effectiveness of non-real-time generation (where composers can manually curate output), the perceptual efficacy of in-game inference remains largely untested. As the results of Chapter 5 indicate, professional concerns regarding the "untestable" nature of generative systems persist. To achieve real-world impact within the industry, extensive technical validation and stress-testing of real-time models are required to improve the professional optics and trust surrounding these technologies.

A technical limitation of this thesis is the absence of Large Language Models (LLMs), Bi-directional Encoder Representations from Transformers (BERT), or diffusion-based architectures. This was a deliberate choice to focus on Transformers, which constituted the state-of-the-art for sequence-based music tasks during the primary development phase of this PhD. Given the rapid acceleration of generative AI since this research commenced, it is vital to assess the continued relevance of these findings. While this thesis does not empirically evaluate foundation models, many of its core insights are architecturally agnostic. They concern the fundamental interaction between generative systems and the practical constraints of game development, including latency, controllability, and integration within established workflows.

Specifically, the findings regarding expressive rendering and listener fatigue remain relevant regardless of the underlying model class. Even as newer architectures demonstrate increased generative capacity, systemic challenges regarding authorial control, predictability, and professional acceptance remain. Consequently, the contributions of this thesis should be viewed not as an evaluation of a specific algorithm, but as an exploration of how expressive variation and creative constraints interact within a ludic context.

Future work could apply contemporary architectures to tasks such as real-time mixing or prompt-based content generation. LLMs, in particular, offer a robust mechanism for guiding high-quality generation through natural language prompts, potentially assisting in automated quality assurance. Similarly, the mask-based conditioning of BERT models could be applied to generate countermelodies or chord progressions that maintain stylistic consistency with a human-authored input.

Another limitation of this research is its ethnocentric scope and sample size. The participants in Chapters 5, 6, and 7 were predominantly British or European. This limits the cross-cultural transferability of the findings, particularly when considering the distinct musical and ludic traditions of Eastern cultures. Furthermore, this rapid evolution means that composer sentiment is a moving target. When the research in Chapter 5 was conducted, the now ubiquitous tools such as Suno, Udio, and ChatGPT had not yet disrupted the public consciousness. As professionals become more technologically literate, their anxieties and expectations evolve. This underscores the need for regular reviews of human factors in AI research, to ensure that ethical and professional frameworks keep pace with technological acceleration.

8.3 Concluding Remarks

As this thesis draws to a close, it is evident within our increasingly technologically literate landscape that AI adoption is accelerating across almost every facet of modern life. LLMs have become both commonplace tools, and points of contentionin classrooms, healthcare, entertainment, and music. In the context of video games, this transition represents a significant opportunity, provided the technology is deployed responsibly. This is particularly true for live-service RPGs, where musical repetition is an enduring systemic challenge; in such environments, it is not uncommon for fewer than ten hours of music to be stretched across a thousand hours of gameplay. Currently, standard music design methods are ill-equipped to address this disparity, making the evolution and adoption of AI-driven technology a functional necessity. Furthermore, a thriving ecosystem of start-ups is already exploring AI-driven music technology across Digital Audio Workstations, video games, and live streaming platforms. While copyright laws and regulations continue to struggle with the complexities of data mining and model training, there is profound interest from companies of all scales in creating ethical AI-driven music technology, bolstered by non-profit initiatives such as Fairly Trained, which seek to certify trustworthy companies and their datasets.

While systemic concerns regarding the misuse of music AI persist, the trajectory of the industry suggests that the integration of AI music technology is inevitable, with major technology industry players such as Steinberg, Apple, and Presonus increasingly investing in AI engineering talent. However, the longevity of these tools remains a subject of debate, with some industry observers predicting an eventual contraction of the “AI bubble.”

In conclusion, as a musician, machine learning engineer, and entrepreneur, I remain curious to see how the global music community will ultimately assimilate artificially intelligent music technology. I am equally fascinated to see if music AI will become a permanent fixture of the music-making workflow, mirroring the journey of sample libraries, which moved from controversial outsiders to industry standards, or if the field will be significantly curtailed as legislation evolves to handle copyright. Ultimately, this thesis serves as a foundational step toward a future where AI and human creativity coexist as a refined, responsible, and authentic partnership.

References

Jeremy Abel. Genesis Noir: Bringing Jazz Improvisation To Gaming, 2021. URL https://www.youtube.com/watch?v=enccMn3LObU.

Adam Alpern. Techniques for algorithmic composition of music, 1995. URL https://api.semanticscholar.org/CorpusID:18258525.

Alba Amato. Procedural Content Generation in the Game Industry. In Oliver Korn and Newton Lee, editors, *Game Dynamics: Best Practices in Procedural and Dynamic Game Content Generation*, pages 15–25. Springer International Publishing, Cham, 2017. ISBN 978-3-319-53088-8. doi: 10.1007/978-3-319-53088-8_2. URL https://doi.org/10.1007/978-3-319-53088-8_2.

Andrew Glassner. *Interactive Storytelling: Techniques for 21st Century Fiction*. CRC Press, Boca Raton, Florida, 2004. URL https://www.routledge.com/Interactive-Storytelling-Techniques-for-21st-Century-Fiction/Glassner/p/book/9781568812212.

Alessandro Ansani, Friederike Koehler, Lisa Giombini, Matias Hämäläinen, Chen Meng, Marco Marini, and Suvi Saarikallio. Ai performer bias: Listeners like music less when they think it was performed by an ai. *Empirical Studies of the Arts*, 43(2):1137–1161, 2025.

Apple Newsroom. Logic Pro takes music-making to the next level with new AI features, 2024. URL https://www.apple.com/uk/newsroom/2024/05/logic-pro-takes-music-making-to-the-next-level-with-new-ai-features/.

Josep Lluís Arcos and Ramon López De Mántaras. An Interactive Case-Based Reasoning Approach for Generating Expressive Music. *Applied Intelligence*, 14(1):115–129, January 2001. ISSN 1573-7497. doi: 10.1023/A:1008311209823. URL https://doi.org/10.1023/A:1008311209823.

Stephen Armstrong. Sounding the Grind. *Journal of Sound and Music in Games*, 2(2):1–21, April 2021. ISSN 2578-3432. doi: 10.1525/jsmg.2021.2.2.1. URL https://online.ucpress.edu/jsmg/article/2/2/1/116914/Sounding-the-GrindMusicospatial-Stasis-in-JRPG.

ASCII Corporation. Otocky, 1987. Video game.

Bob Ashton and Nancy Colbaugh. *You Can Teach Yourself to Compose Music*. Mel Bay Publications, 2011.

Atlus. Persona 5 Royal, 2019. Video game.

Avid. Artificial intelligence: AI a key focus of Avid innovation, 2023. URL https://www.avid.com/resource-center/avid-and-the-future-of-ai.

Johann Sebastian Bach. Fugue in G minor, BWV 862, 1722a. The Well-Tempered Clavier, Book 1.

Johann Sebastian Bach. Prelude in A minor, BWV 865, 1722b. The Well-Tempered Clavier, Book 1.

Johann Sebastian Bach. Fugue in A-flat major, BWV 885, 1742a. The Well-Tempered Clavier, Book 2.

Johann Sebastian Bach. Prelude in B major, BWV 892, 1742b. The Well-Tempered Clavier, Book 2.

Johann Sebastian Bach. Fugue in B major, BWV 893, 1742c. The Well-Tempered Clavier, Book 2.

Bandai Namco Studios. Tales of Arise, 2021. Video game.

Adrián Barahona-Ríos and Tom Collins. SpecSinGAN: Sound effect variation synthesis using single-image GANs. In *Proceedings of the Sound and Music Computing Conference*, pages 302–309, Saint-Étienne, France, 2022.

Adrián Barahona-Ríos and Tom Collins. SpecSinGAN: Sound Effect Variation Synthesis Using Single-Image GANs. https://arxiv.org/abs/2110.07311, 2021. Accessed: 2021-11-26.

Adrián Barahona-Ríos and Sandra Pauletto. Synthesising Knocking Sound Effects Using Conditional WaveGAN. In *17th Sound and Music Computing Conference, Online*, 2020.

Nicolas A. Barriga. A short introduction to procedural content generation algorithms for videogames. *International Journal on Artificial Intelligence Tools*, 28(02):1930001, 2019. Publisher: World Scientific.

D. E. Berlyne. Novelty, complexity, and hedonic value. *Perception & Psychophysics*, 8(5):279–286, September 1970. ISSN 1532-5962. doi: 10.3758/BF03212593. URL https://doi.org/10.3758/BF03212593.

D. E. Berlyne. Aesthetics and Psychobiology. *Journal of Aesthetics and Art Criticism*, 31(4):553–553, 1971. doi: 10.2307/429334. Publisher: Wiley-Blackwell.

Bethesda Softworks. The Elder Scrolls II: Daggerfall, 1996. Video game.

BioWare. Mass Effect, 2007. Video game.

Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

Blizzard Entertainment. World of Warcraft, 2004. Video game.

Blizzard Entertainment. Diablo Immortal, 2022. Video game.

Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *ISMIR*, pages 255–261. New York City, 2016.

Margaret A Boden. *Artificial intelligence: A very short introduction*. Oxford University Press, 2018.

Margaret A Boden and Ernest A Edmonds. What is generative art? *Digital Creativity*, 20(1-2):21–46, 2009.

Gabriel Bosque. Cinematic Animation in RPGs: Motion Capture and Emotion Recognition, 2022.

Niels Böttcher. Current problems and future possibilities of procedural audio in computer games. *Journal of Gaming & Virtual Worlds*, 5(3):215–234, 2013.

Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, January 2006. ISSN 1478-0887. doi: 10.1191/1478088706qp063oa. URL https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa. Publisher: Routledge _eprint: https://www.tandfonline.com/doi/pdf/10.1191/1478088706qp063oa.

Virginia Braun and Victoria Clarke. *Successful Qualitative Research*. Sage Publications Ltd, London: UK, 2013. URL `https://uk.sagepub.com/en-gb/eur/successful-qualitative-research/book233059`.

Virginia Braun and Victoria Clarke. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health*, 11(4):589–597, 2019.

Roberto Bresin. *Virtual virtuosity: studies in automatic music performance*. PhD Thesis, KTH, Stockholm, 2000.

Roberto Bresin, Anders Friberg, and Johan Sundberg. Director Musices : The KTH Performance Rules System. *MUS*, 2002(63 (2002-MUS-046)): 43–48, July 2002.

Jean-Pierre Briot and François Pachet. Deep learning for music generation: challenges and directions. *Neural Computing and Applications*, 32(4):981–993, 2020.

Frederick P Brooks, Albert L Hopkins, Peter G Neumann, and William V Wright. An experiment in musical composition. *IRE Transactions on Electronic Computers*, (3):175–182, 2009.

Daniel Brown. The Dynamic Percussion System: A procedural music engine for video games., 2019.

B Bucklew and J Grinblat. Math for Game Developers: End-to-End Procedural Generation in Caves of Qud, 2019. URL `https://www.youtube.com/watch?v=jV-DZqdKlnE&t=581s`.

Bungie. Destiny 2, 2017. Video game; later self-published by Bungie from 2019.

Alexandra-Petronela Buraga. The Emergence of the Type-Generated AI Art Community: A Netnographic and Content Analysis Approach, 2022.

J Peter Burkholder, Donald Jay Grout, and Claude V Palisca. *A history of western music: Tenth international student edition*. WW Norton & Company, 2019.

Sergio Canazza, Giovanni De Poli, and Antonio Rodà. Caro 2.0: an interactive system for expressive music rendering. *Advances in Human-Computer Interaction*, 2015, 2015.

Carlos Cancino-Chacón. *Computational Modeling of Expressive Music Performance with Linear and Non-linear Basis Function Models*. PhD Thesis, Johannes Kepler University, Linz, 2018.

Carlos Cancino-Chacón, Silvan Peter, Shreyan Chowdhury, Anna Aljanaki, and Gerhard Widmer. On the characterization of expressive performance in classical music: First results of the con espressione game. *arXiv preprint arXiv:2008.02194*, 2020.

Philip C Candy. Constructivism and the study of self-direction in adult learning. *Studies in the Education of Adults*, 21(2):95–116, 1989.

Capcom. Resident Evil, 1996. Video game.

Etienne Carrier. Procedural World Generation of Far Cry 5, 2018. URL `https://www.youtube.com/watch?v=JBp8zvLVsgg`.

Kyoyun Choi, Jonggwon Park, Wan Heo, Sungwook Jeon, and Jonghun Park. Chord conditioned melody generation with transformer based decoders. *IEEE Access*, 9:42071–42080, 2021.

Frédéric Chopin. Ballade No. 3 in A-flat major, Op. 47, 1841. Mvt. 1.

Shreyan Chowdhury and Gerhard Widmer. Towards explaining expressive qualities in piano recordings: Transfer of explanatory features via acoustic domain adaptation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 561–565. IEEE, 2021.

Miguel Civit, Javier Civit-Masot, Francisco Cuadrado, and Maria J Escalona. A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends. *Expert Systems with Applications*, 209:118190, 2022.

Louis Cohen, Lawrence Manion, and Keith Morrison. *Research methods in education*. routledge, 2018.

Karen Collins. *Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design*. MIT Press, 2008.

Kc Collins. An Introduction to Procedural Music in Video Games. *Contemporary Music Review*, 28:5–15, February 2009. doi: 10.1080/07494460802663983.

Nick Collins. Origins of algorithmic thinking in music. In Roger T. Dean and Alex McLean, editors, *The Oxford Handbook of Algorithmic Music*, pages 1–24. Oxford University Press, 2018. doi: 10.5281/zenodo.1200221.

Tom Collins and Robin Laney. Computer-Generated Stylistic Compositions with Long-Term Repetitive and Phrasal Structure. *Journal of Creative Music Systems*, 1(2), March 2017. doi: 10.5920/JCMS.2017.02. URL `https://www.jcms.org.uk/article/id/510/`. Number: 2.

Tom Collins, Jeremy Thurlow, Robin Laney, Alistair Willis, and Paul H. Garthwaite. A comparative evaluation of algorithms for discovering translational patterns in baroque keyboard works. In J. Stephen Downie and Remco Veltkamp, editors, *Proceedings of the International Society for Music Information Retrieval Conference*, pages 3–8, Utrecht, The Netherlands, 2010.

Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.

Tom Collins, Andreas Arzt, Harald Frostel, and Gerhard Widmer. Using geometric symbolic fingerprinting to discover distinctive patterns in polyphonic music corpora. In *Computational Music Analysis*, pages 445–474. Springer, 2016.

Tom Collins, Kyle Worrall, Chenyu Gao, Jemily Rime, Mark Hanslip, and Ashay Dave. Perspectives on roles and rewards in new cocreative systems for music-making. *Interacting with Computers*, December under review.

Florian Colombo, Alexander Seeholzer, and Wulfram Gerstner. Deep artificial composer: A creative neural network model for automated melody generation. In *Computational Intelligence in Music, Sound, Art and Design: 6th International Conference, EvoMUSART 2017, Amsterdam, The Netherlands, April 19–21, 2017, Proceedings 6*, pages 81–96. Springer, 2017.

Kate Compton. So you want to build a generator..., 2016. URL `https://galaxykate0.tumblr.com/post/139774965871/so-you-want-to-build-a-generator`.

Core Design. Tomb Raider, 1996. Video game.

John Creswell. Qualitative, quantitative, and mixed methods approaches, 2019.

John B Crozier. Verbal and exploratory responses to sound sequences varying in uncertainty level. *Studies in the new experimental aesthetics*, pages 27–90, 1974.

Crystal Dynamics. Tomb Raider, 2013. Video game.

Crystal Dynamics. Rise of the Tomb Raider, 2015. Video game.

Crystal Dynamics. Marvel's Avengers, 2020. Video game.

Custom Market Insights. Global Game Music Market 2024-2033, December 2023. URL `https://www.custommarketinsights.com/report/game-music-market/`.

Simon Cutajar. *Automatic Generation of Dynamic Musical Transitions in Computer Games*. PhD Thesis, The Open University, 2020.

Shuqi Dai, Huiran Yu, and Roger B Dannenberg. What is missing in deep music generation? A study of repetition and structure in popular music. In *The Proceedings of the 23rd International Society for Music Information Retrieval*, 2022.

Dambuster Studios. Dead Island 2. Video game, 2023. PlayStation 4, PlayStation 5, Xbox One, Xbox Series X/S, Windows, released April 21, 2023.

Fred D. Davis. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3):319–340, 1989. ISSN 0276-7783. doi: 10.2307/249008. URL `https://www.jstor.org/stable/249008`. Publisher: Management Information Systems Research Center, University of Minnesota.

Raul Paiva de Oliveira and Tiago Fernandes Tavares. Impact of algorithmic composition on player immersion in computer games: A case study using markov chains. *Revista Música Hodie*, 18(1):61–73, 2018.

Emmanuel Deruty, Maarten Grachten, Stefan Lattner, Javier Nistal, and Cyran Aouameur. On the development and practice of ai technology for contemporary popular music production. *Transactions of the International Society for Music Information Retrieval*, 5(1):35–50, 2022.

James M. Dickey and B. P. Lientz. The Weighted Likelihood Ratio, Sharp Hypotheses about Chances, the Order of a Markov Chain. *The Annals of Mathematical Statistics*, 41(1):214–226, 1970. doi: 10.1214/aoms/1177697203.

Eamon O Doherty. *Measuring Expressive Music Performances: a Performance Science Model using Symbolic Approximation*. PhD thesis, Technological University Dublin, 2019.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial Audio Synthesis, February 2019. URL `http://arxiv.org/abs/1802.04208`. arXiv:1802.04208 [cs].

Paul Doornbusch. Composers' views on mapping in algorithmic composition. *Organised Sound*, 7(2):145–156, 2002. doi: 10.1017/S1355771802002066.

Mohit Dua, Rohit Yadav, Divya Mamgai, and Sonali Brodiya. An Improved RNN-LSTM based Novel Approach for Sheet Music Generation. *Procedia Computer Science*, 171:465–474, 2020.

Vikram Dutta. *THE REPETITION OF VIDEO GAME MUSIC, ITS IMPACT ON VIDEO GAME ENJOYMENT, AND HOW BEST TO MANAGE IT*. PhD, University of Huddersfield, Huddersfield, 2020.

Michael D'Errico. Worlds of Sound: Indie Games, Proceduralism, and the Aesthetics of Emergence. *Music, Sound, and the Moving Image*, 9(2):191–206, December 2015. ISSN 1753-0768, 1753-0776. doi: 10.3828/msmi.2015.13. URL `https://online.liverpooluniversitypress.co.uk/doi/10.3828/msmi.2015.13`.

Kemal Ebcioğlu. *Strict counterpoint: A case study in musical composition by computers*. Middle East Technical University (Turkey), 1979.

Kemal Ebcioğlu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3):43–51, 1988.

Henry Ellington, Eric Addinall, and Fred Percival. *A handbook of game design*. Koran Page, London, UK, 1982. URL `https://scholar.google.com/scholar_lookup?title=A+handbook+of+game+design&author=H.+Ellington&author=E.+Addinall&author=F.+Percival&publication_year=1982`.

Steve Engels, Tiffany Tong, and Fabian Chan. Automatic real-time music generation for games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 11, 2015. Issue: 1.

Brian Eno. Generative music. `https://www.inmotionmagazine.com/eno1.html`, 1996. Accessed: 2026-04-18.

Elijah Alixtair L Estolas, Agatha Faith V Malimban, Jeremy T Nicasio, Jyra S Rivera, May Florence D San Pablo, and Toru L Takahashi. Automatic beatmap generating rhythm game using music information retrieval with machine learning for genre detection. In *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–6. IEEE, 2020.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Long-form music generation with latent diffusion. *arXiv preprint arXiv:2404.10301*, 2024.

Andy Farnell. *Designing Sound | The MIT Press*. MIT Press, 2009. ISBN 9780262014410 688. URL `https://mitpress.mit.edu/books/designing-sound`. Publisher: The MIT Press.

Feral Cat Den. Genesis Noir, 2021. Video game.

Lucas N Ferreira, Lili Mou, Jim Whitehead, and Levi HS Lelis. Controlling perceived emotion in symbolic music generation with monte carlo tree search. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, pages 163–170, 2022.

Rebecca Fiebrink and Baptiste Caramiaux. The machine learning algorithm as creative musical tool. *arXiv preprint arXiv:1611.00379*, 2016.

Catherine Flick and Kyle Worrall. The Ethics of Creative AI. In Craig Vear and Fabrizio Poltronieri, editors, *The Language of Creative AI: Practices, Aesthetics and Structures*, Springer Series on Cultural Computing, pages 73–91. Springer International Publishing, Cham, 2022. ISBN 978-3-031-10960-7. doi: 10.1007/978-3-031-10960-7_5. URL `https://doi.org/10.1007/978-3-031-10960-7_5`.

S. Flossman, M. Grachten, and G. Widmer. Expressive Performance Rendering with Probabilistic Models. In *Guide to computing for expressive music performance*, pages 75–98. Springer, 2012.

Sebastian Flossmann, Maarten Grachten, and Gerhard Widmer. Experimentally investigating the use of score features for computational models of expressive timing. In *Proceedings of the 10th international conference on music perception and cognition*, pages 1–6, 2008.

Sebastian Flossmann, Maarten Grachten, and Gerhard Widmer. EXPRESSIVE PERFORMANCE RENDERING: INTRODUCING PERFORMANCE CONTEXT. In *Proceedings of the 6th Sound and Music Computing Conference*, pages 155–160, Porto, Portugal, 2009.

David Forss. *Reducing Repetition in Game Sound : Utilizing Frequency Manipulation to Create Variations of Footstep Sound Assets*. 2016. URL `http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-52842`.

Fortune Business Insights. Digital Audio Workstation Market Revenue, Size | Growth., 2024. URL `https://www.fortunebusinessinsights.com/industry-reports/digital-audio-workstation-market-100150`.

Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai. ASAP: a dataset of aligned scores and performances for piano transcription. In *Proceedings of the International Society for Music Information Retrieval Conference Information Retrieval*, pages 534–541, 2020.

Nathan Fradet, Jean-Pierre Briot, Fabien Chhel, Amal Seghrouchni, and Nicolas Gutowski. Miditok: A python package for midi file tokenization. 11 2021.

Barbara Freedman. *Teaching music through composition: A curriculum using technology*. Oxford University Press, 2013.

A Friberg, R Bresin, and J. Sundberg. Overview of the kth rule system for musical performance. advances in cognitive psychology. *Advances in Cognitive Psychology*, 2(2):145–161, 2006.

Anders Friberg. *A Quantitative Rule System for Musical Expression*. PhD thesis, KTH Royal Institute of Technology, 1995.

FromSoftware. Bloodborne, 2015. Video game.

FromSoftware. Elden Ring, 2022. Video game.

Akira Maezawa Kazuhiko Yamamoto Takuya Fujishima. Rendering music performance with interpretation variations using conditional variational rnn. In *Proceedings of the International Conference for Music Information Retrieval*, pages 855–861, 2018.

Alf Gabrielsson. Once again: The theme from Mozart's piano sonata in A major (K.331). In *Action and Perception in Rhythm and Music*, pages 81–104. Royal Swedish Academy of Music, Stockholm, 1987.

Gaijin Games. Bit.Trip Runner, 2010. Video game.

Game Freak. Pokémon Blue Version, 1996. Video game.

Chenyu Gao and Tom Collins. The pendular graph: Visualising hierarchical repetitive structure in point-set representations of the pop909 music dataset. In *International Conference on Human-Computer Interaction*, pages 3–16. Springer, 2023.

Andres Garay Acevedo. Fugue composition with counterpoint melody generation using genetic algorithms. In *International symposium on computer music modeling and retrieval*, pages 96–106. Springer, 2004.

Richard Garriott. Ultima: The First Age of Darkness, 1981. Video game.

Sarah Gates. Enjoying the grind: Musical encouragement of repetitive action in final fantasy x. Conference presentation, Society for Music Theory, November 2017. Presented at the Society for Music Theory Annual Meeting.

Marvin Gaye and Tammi Terrell. Sad Wedding. Track on: Dead Island 2 Original Soundtrack [CD], 2023. Track from *Dead Island 2 Original Soundtrack*.

Gearbox Software. Borderlands 3, 2019. Video game.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-Audio Generation using Instruction Guided Latent Diffusion Model. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 3590–3598, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3612348. URL `https://dl.acm.org/doi/10.1145/3581783.3612348`.

William Gibbons and Steven Reale. *Music in the Role-Playing Game*, volume 2. New York: Routledge, 2020.

Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. Learning to Groove with Inverse Sequence Transformations. In *Proceedings of the International Conference on Machine Learning*, pages 2269–2279, 2019.

Mathieu Giraud, Richard Groult, Emmanuel Leguy, and Florence Levé. Computational fugue analysis. *Computer Music Journal*, 39(2):77–96, 2015.

James Denis Mc Glynn. The 'cinematic promise' of video game music: Stylistic convergence, current-generation remakes, and the case of final fantasy vii. *Journal of Sound and Music in Games*, 4(4):108–138, 2023.

Maarten Grachten and Florian Krebs. An assessment of learned score features for modeling expressive dynamics in music. *IEEE Transactions on Multimedia*, 16(5):1211–1218, 2014.

Maarten Grachten and Gerhard Widmer. Linear Basis Models for Prediction and Analysis of Musical Expression. *Journal of New Music Research*, 41 (4):311–322, 2012.

Simone Grassini and Mika Koivisto. Understanding how personality traits, experiences, and attitudes shape negative bias toward ai-generated artworks. *Scientific Reports*, 14(1):4113, 2024.

Stefan Greuter and Adam Nash. Game Asset Repetition. In *Proceedings of the 2014 Conference on Interactive Entertainment*, IE2014, pages 1–5, New York, NY, USA, December 2014. Association for Computing Machinery. ISBN 978-1-4503-2790-9. doi: 10.1145/2677758.2677782. URL `https://doi.org/10.1145/2677758.2677782`.

Mark Grimshaw, Craig Lindley, and Lennart Nacke. Sound and immersion in the first-person shooter: Mixed measurement of the player's sonic experience. In *Audio Mostly-a conference on interaction with sound*. www. audiomostly. com, 2008. URL `https://www.diva-portal.org/smash/record.jsf?pid=diva2:835992`.

Mark Grimshaw, John Charlton, and Richard Jagger. First-person shooters: Immersion and attention. *Eludamos: Journal for Computer Game Culture*, 5(1):29–44, 2011. URL `https://eludamos.org/index.php/eludamos/article/view/vol5no1-3`.

Mark Grimshaw, Siu-Lan Tan, and Scott D. Lipscomb. Playing with sound: The role of music and sound effects in gaming. In *The psychology of music in multimedia*, pages 289–314. Oxford University Press, New York, NY, US, 2013. ISBN 978-0-19-960815-7. doi: 10.1093/acprof:oso/9780199608157.003.0013.

Jason Grinblat. Procedurally Generating History in Caves of Qud, 2018. URL `https://www.youtube.com/watch?v=H0sLa1y3BW4`.

Grinding Gear Games. Path of Exile, 2013. Video game.

Graham Grindlay and D. Helmbold. Modeling, analyzing, and synthesizing expressive piano performance with graphical models. *Machine Learning*, 65:361–387, 2006. doi: 10.1007/s10994-006-8751-3.

Yupeng Gu and Christopher Raphael. Modeling Piano Interpretation Using Switching Kalman Filter. In *Proceedings of the International Society for Music Information Retrieval*, page 6, 2012.

Rui Guo, Ivor Simpson, Chris Kiefer, Thor Magnusson, and Dorien Herremans. MusIAC: An Extensible Generative Framework for Music Infilling Applications with Multi-level Control. In Tiago Martins, Nereida Rodríguez-Fernández, and Sérgio M. Rebelo, editors, *Artificial Intelligence in Music, Sound, Art and Design*, Lecture Notes in Computer Science, pages 341–356, Cham, 2022. Springer International Publishing. ISBN 978-3-031-03789-4. doi: 10.1007/978-3-031-03789-4_22.

Zixun Guo and Simon Dixon. Moonbeam: A midi foundation model using both absolute and relative music attributes. *arXiv preprint arXiv:2505.15559*, 2025.

Guy Somberg. Game Audio Programming - Guy Somberg, November 2018. URL `https://www.youtube.com/watch?v=jY3tPM1oNyU`.

Lundh Haaland. *The Player as a Conductor : Utilizing an Expressive Performance System to Create an Interactive Video Game Soundtrack*. 2020. URL `http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-281324`.

Gaëtan Hadjeres. The Piano Inpainting Application - Gaëtan Hadjeres | Etincelle #14, 2021. URL `https://www.youtube.com/watch?v=mgM6eBBfZqg`.

Gaëtan Hadjeres and Léopold Crestel. The Piano Inpainting Application. *ArXiv*, July 2021. URL `https://www.semanticscholar.org/paper/1306680402070eecd37da6bb0d531017166c5f80`.

Rebeen Hamad. What is LSTM? Introduction to Long Short-Term Memory, December 2023. URL `https://medium.com/@rebeen.jaff/what-is-lstm-introduction-to-long-short-term-memory-66bd3855b9ce`.

Sangjun Han, Hyeongrae Ihm, Moontae Lee, and Woohyung Lim. Symbolic music loop generation with neural discrete representations. In *Ismir 2022 Hybrid Conference*, 2022.

David J Hargreaves. The effects of repetition on liking for music. *Journal of research in Music Education*, 32(1):35–47, 1984.

Harmonix. Guitar Hero, 2005. Video game.

Mitsuyo Hashida, Noriko Nagata, and Haruhiro Katayose. Pop-E: a performance rendering system for the ensemble music that considered group expression. In *Proceedings of the 9th International Conference on Music Perception and Cognition*, pages 526–534, 2006.

Mitsuyo Hashida, Shunji Tanaka, Takashi Baba, and Haruhiro Katayose. Mixtract: An Environment for Designing Musical Phrase Expression. In *Proceedings of Sound and Music Computing*, pages 21–24, 2010.

Joseph Haydn. Keyboard Sonata in B minor, Hob. XVI:32, 1776. Mvt. 1.

Stephen A Hedges. Dice music in the eighteenth century. *Music & Letters*, 59(2):180–187, 1978.

Hello Games. No Man's Sky, 2016. Video game.

Dorien Herremans and Elaine Chew. Tension ribbons: Quantifying and visualising tonal tension. 2016.

Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A functional taxonomy of music generation systems. *ACM Computing Surveys (CSUR)*, 50(5):1–30, 2017.

L Highams and JI Olszewska. Markov-chain-based musical creative intelligent agent passing successfully the turing test. In *2023 IEEE 27th International Conference on Intelligent Engineering Systems (INES)*, pages 000215–000220. IEEE, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Nicolai Holtar et al. Audioverdrive: Exploring the use of audio-based interaction for game design. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 565–568, Daejeon, Korea, 2013.

Yuanbo Hou, Qiaoqiao Ren, Huizhong Zhang, Andrew Mitchell, Francesco Aletta, Jian Kang, and Dick Botteldooren. Ai-based soundscape analysis: Jointly identifying sound sources and predicting annoyance. *The Journal of the Acoustical Society of America*, 154(5):3145–3157, 2023.

Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):178–186, May 2021. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v35i1.16091. URL https://ojs.aaai.org/index.php/AAAI/article/view/16091.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music Transformer. *arXiv:1809.04281 [cs, eess, stat]*, December 2018. URL http://arxiv.org/abs/1809.04281. arXiv: 1809.04281.

Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. Counterpoint by convolution. *arXiv preprint arXiv:1903.07227*, 2019.

Yu-Siang Huang and Yi-Hsuan Yang. Pop Music Transformer: Beat-based modeling and generation of expressive Pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.

Sam Hughes. HADES Special with Audio Director Darren Korb - Supergiant Games, 2020. URL https://www.thesoundarchitect.co.uk/tsap-s04e01/.

Javier Nistal Hurle, Stefan Lattner, and Gael Richard. Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks. In *21 st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

Patrick Edward Hutchings and Jon McCormack. Adaptive Music Composition for Games. *IEEE Transactions on Games*, 12(3):270–280, September 2020. ISSN 2475-1510. doi: 10.1109/TG.2019.2921979. Conference Name: IEEE Transactions on Games.

Insomniac Games. Spyro the Dragon, 1998. Video game.

Sho Iwamoto. Epic AND Interactive Music in 'Final Fantasy XV', 2017. URL https://www.gdcvault.com/play/1023971/Epic-AND-Interactive-Music-in.

Sejal Jaiswal. Multilayer perceptrons in machine learning: A comprehensive guide, April 2025. URL `https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning`. Accessed: 2025-09-03.

Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. VirtuosoNet: A HIERARCHICAL RNN-BASED SYSTEM FOR MODELING. In *International Society for Music Information Retrieval (ISMIR)*, pages 908–915, 2019a.

Dasaem Jeong, Taegyun Kwon, Yoojin Kim, and Juhan Nam. Graph neural network for music score data and modeling expressive piano performance. In *International Conference on Machine Learning*, pages 3060–3070. PMLR, 2019b.

Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

Shulei Ji, Xinyu Yang, and Jing Luo. A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges. *ACM Computing Surveys*, 56(1):1–39, 2023.

Bijue Jia, Jiancheng Lv, and Dayiheng Liu. Deep learning-based automatic downbeat tracking: a brief review. *Multimedia Systems*, 25(6):617–638, 2019.

Kristine Jørgensen. On transdiegetic sounds in computer games. *Northern Lights: Film & Media Studies Yearbook*, 5(1):105–117, September 2007. ISSN 1601-829X, 2040-0586. doi: 10.1386/nl.5.1.105_1. URL `https://intellectdiscover.com/content/journals/10.1386/nl.5.1.105_1`. Publisher: Intellect.

Purnima Kamath, Fabio Morreale, Priambudi Lintang Bagaskara, Yize Wei, and Suranga Nanayakkara. Sound Designer-Generative AI Interactions: Towards Designing Creative Support Tools for Professional Sound Designers. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17, Honolulu HI USA, May 2024. ACM. ISBN 9798400703300. doi: 10.1145/3613904.3642040. URL `https://dl.acm.org/doi/10.1145/3613904.3642040`.

Marius Kaminskas and Francesco Ricci. Contextual music information retrieval and recommendation: State of the art and challenges. *Computer Science Review*, 6(2-3):89–119, 2012.

Daiki Kamiyama. How CRI Middleware's HCA codec aids the development of Final Fantasy and other Square Enix titles, June 2022. URL https://automaton-media.com/en/interviews/20220609-13173/.

Haruhiro Katayose, Mitsuyo Hashida, Giovanni De Poli, and Keiji Hirata. On Evaluating Systems for Generating Expressive Music Performance: the Rencon Experience. *Journal of New Music Research*, 41(4):299–310, December 2012. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2012.745579. URL http://www.tandfonline.com/doi/abs/10.1080/09298215.2012.745579.

Robert M Keller, David R Morrison, and Harvey Mudd College. A Grammatical Approach to Automatic Improvisation. In *Proceedings of the 4th Sound and Music Computing Conference*, pages 330–337, Greece, 2007.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. https://arxiv.org/abs/1412.6980, 2014. Accessed: 2019-04-09.

Alexis Kirke and Eduardo Reck Miranda. A survey of computer systems for expressive music performance. *ACM Computing Surveys (CSUR)*, 42(1): 1–41, 2009.

Anssi Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 6, pages 3089–3092. IEEE, 1999.

Shelly Knotts and Nick Collins. A survey on the uptake of music ai software. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 499–504, 2020.

Hyung-Kwon Ko, Gwanmo Park, Hyeon Jeon, Jaemin Jo, Juho Kim, and Jinwook Seo. Large-scale text-to-image generation models for visual artists' creative works. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*, pages 919–933, 2023.

Athanasios Vasileios Kokkinakis, Simon Demediuk, Isabelle Nölle, Oluseyi Olarewaju, Sagarika Patra, Justus Robertson, Peter York, Alan Pedrassoli Chitayat, Alistair Coates, Daniel Slawson, Peter Hughes, Nicolas Hardie, Ben Kirman, Jonathan Hook, Anders Drachen, Marian Ursu, and Florian Block. Dax: Data-driven audience experiences in esports. In *Proceedings of IMX International Conference on Interactive Media Experiences*, pages 94–105. ACM, 2020.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020. URL `https://ieeexplore.ieee.org/abstract/document/9229505/`. Publisher: IEEE.

Katerina Kosta, Rafael Ramírez, Oscar F. Bandtlow, and Elaine Chew. Mapping between dynamic markings and performed loudness: a machine learning approach. *Journal of Mathematics and Music*, 10(2):149–172, 2016.

K. Yu Kristen, Matthew Guzdial, and Nathan Sturtevant. The definition-context-purpose paradigm and other insights from industry professionals about the definition of a quest. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 17, pages 107–114, 2021. Issue: 1.

David Kushner. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*. Random House, New York, 2003. ISBN 978-0-375-50524-6.

Philip Lamperski. Real-time Procedural Percussion Scoring in 'Tomb Raider's' Stealth Combat, 2015. URL `https://www.gdcvault.com/play/1023215/Real-time-Procedural-Percussion-Scoring`.

Peter S Langston. Eedie & Eddie on the Wire An Experiment in Music Generation. In *Summer USENIX Technical Conference and Exhibition Proceedings*, pages 13–27, 1986.

Fred Lerdahl and Ray S. Jackendoff. *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT Press, 1996. ISBN 978-0-262-26091-6. Google-Books-ID: 6HGiEW33lucC.

Matthew Lightbound and Ryan Williams. Elevating diegetic music: Dead island 2's implementation approaches to in-world music. In *Game Audio Symposium 2025*. Game Audio Symposium, 2025. URL `https://gameaudiosymposium.com/talks`. Presented at the Game Audio Symposium, Leeds School of Arts, Leeds Beckett University, UK.

Lele Liu and Emmanouil Benetos. From audio to music notation. *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*, pages 693–714, 2021.

Steven R Livingstone. *Changing musical emotion through score and performance with a computational rule system*. University of Queensland, 2008.

Ryan Louie, Andy Coenen, Cheng Zhi Huang, Michael Terry, and Carrie J. Cai. Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Honolulu HI USA, April 2020. ACM. ISBN 978-1-4503-6708-0. doi: 10.1145/3313831.3376739. URL https://dl.acm.org/doi/10.1145/3313831.3376739.

Lucasfilm Games. Ballblazer, 1985. Video game.

Vincenzo Madaghiele, Pasquale Lisena, and Raphaël Troncy. Mingus: melodic improvisation neural generator using seq2seq. In *Proceedings of the 22nd International Society of Music Information Retrieval Conference*, pages 412–419, 2021.

Iman Malik and Carl Henrik Ek. Neural Translation of Musical Style. https://arxiv.org/abs/1708.03535, 2017. Accessed: 2020-04-10.

Aaron Marks. *The Complete Guide to Game Audio: For Composers, Musicians, Sound Designers, Game Developers.* Focal Press, 2008.

John Maurer. The History of Algorithmic Composition, 1999. URL https://ccrma.stanford.edu/~blackrse/algorithm.html.

Maxis. Spore, 2008. Video game.

Innes McKendrick. Continuous World Generation in No Man's Sky, 2017. URL https://www.youtube.com/watch?v=sCRzxEEcO2Y.

Aaron McLeran and Kent Jolly. Procedural Music in SPORE — gdcvault.com. https://www.gdcvault.com/play/323/Procedural-Music-in, 2008. [Accessed 15-03-2025].

Dimitris Menexopoulos, Pedro Pestana, and Joshua Reiss. The State of the Art in Procedural Audio. *J. Audio Eng. Soc.*, 71(12):825–847, December 2023. ISSN 15494950. doi: 10.17743/jaes.2022.0108. URL https://www.aes.org/e-lib/browse.cfm?elib=22346.

David Meredith. *Computational music analysis*, volume 62. Springer, 2016.

Riley Miladi. Empowering artists through machine learning: Balancing creativity and control. Keynote presentation at the iGGi Conference 2024, September 2024. URL https://www.iggi2024.org/speakers. 15:15 GMT.

Matthew B. Miles, A. Michael Huberman, and Johnny Saldana. *Qualitative Data Analysis: A Methods Sourcebook*. Thousand Oaks: Sage, Canada, fourth edition, 2014.

Marvin Minsky and Seymour A Papert. *Perceptrons*. MIT press, 1969.

C Miranda. What is melody in music?, 2024. URL `https://moises.ai/blog/tips/what-is-melody-in-music/`.

Yasunori Mitsuda. Secret of the Forest. Track on: Chrono Trigger Original Soundtrack [CD], 1995. Tokyo: Square.

David Moffat. AI Music Mixing Systems. *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*, pages 345–375, 2021. Publisher: Springer.

Simon Moulieras and François Pachet. Maximum entropy models for generation of expressive music. `http://arxiv.org/abs/1610.03606`, October 2016. Accessed: 2022-10-12.

Isa Muhammad. 65% of game studios are working on live service games, February 2024. URL `https://www.pocketgamer.biz/65-of-game-studios-are-working-on-live-service-games/`.

Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.

Eugene Narmour. *The analysis and cognition of basic melodic structures: The implication-realization model*. University of Chicago Press, Chicago: USA, 1990.

Naughty Dog. Crash Bandicoot, 1996. Video game.

Naughty Dog. Uncharted 4: A Thief's End, 2016. Video game.

NeverKnowsBest. The Entire History of RPGs, March 2024. URL `https://www.youtube.com/watch?v=8o3i10OuMFQ`.

Gerhard Nierhaus. *Algorithmic composition: paradigms of automated music generation*. Springer, 2009.

Nintendo. The Legend of Zelda, 1986. Video game.

Nintendo. Mario Paint, 1992a. Video game.

Nintendo. The Legend of Zelda: A Link to the Past, 1992b. Video game.

Nintendo. The Legend of Zelda: Ocarina of Time, 1998. Video game.

Javier Nistal, Marco Pasini, and Stefan Lattner. Improving musical accompaniment co-creation via diffusion transformers. In *Audio Imagination: NeurIPS 2024 Workshop AI-Driven Speech, Music, and Sound Generation*.

Obsidian Entertainment. South Park: The Stick of Truth, 2014. Video game.

Martin O'Donnell. Producing Audio for Halo, 2002. URL `https://www.gamasutra.com/view/feature/131896/producing_audio_for_halo.php`.

Costantino Oliva. Reassessing otocky: a forgotten classic in digital games history, 2015.

Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, 2020. Publisher: Springer.

Mark G. Orr and Stellan Ohlsson. The Relationship Between Musical Complexity and Liking in Jazz and Bluegrass. *Psychology of Music*, 29(2): 108–127, October 2001. ISSN 0305-7356, 1741-3087. doi: 10.1177/0305735601292002. URL `http://journals.sagepub.com/doi/10.1177/0305735601292002`.

François Pachet, Pierre Roy, Gabriele Barbieri, and Sony CSL Paris. Finite-length markov processes with constraints. *transition*, 6(1/3), 2001.

Victor Padilla and Darrell Conklin. Generation of two-voice imitative counterpoint from statistical models. *International Journal of Interactive Multimedia and Artificial Intelligence ...*, 2018.

Caroline Palmer. *Timing in Skilled Piano Performance*. PhD, Cornell University, 1988.

Richard Parncutt and Gary McPherson. *The Science and Psychology of Music Performance: Creative Strategies for Teaching and Learning*. Oxford University Press, April 2002. ISBN 978-0-19-535017-3.

Christine McLeavey Payne. MuseNet. `https://openai.com/blog/musenet/`, 2019. Accessed: 2020-04-09.

Winifred Phillips. *A Composer's Guide to Game Music*. MIT Press, 2014.

Wendy Pi. An Introduction to Recurrent Neural Networks (RNNs), April 2024. URL https://medium.com/@researchgraph/an-introduction-to-recurrent-neural-networks-rnns-802fcfee3098.

Playground Games. Forza Horizon 5, 2021. Video game.

Cale Plut. The Audience of the Singular, 2017. Video game.

Cale Plut and Philippe Pasquier. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing*, 33:100337, March 2020. ISSN 1875-9521. doi: 10.1016/j.entcom.2019.100337. URL https://www.sciencedirect.com/science/article/pii/S1875952119300795.

Anthony Prechtl. *Adaptive Music Generation for Computer Games*. phd, The Open University, February 2016. URL http://oro.open.ac.uk/45340/.

Luke Prudente and Andrei Coronel. Towards automated counter-melody generation for monophonic melodies. In *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, pages 197–202, 2017.

Q Entertainment. Child of Eden, 2011. Video game.

Yang Qin, Huiming Xie, Shuxue Ding, Benying Tan, Yujie Li, Bin Zhao, and Mao Ye. Bar transformer: a hierarchical model for learning long-term structure and generating impressive pop music. *Applied Intelligence*, August 2022. ISSN 1573-7497. doi: 10.1007/s10489-022-04049-3. URL https://doi.org/10.1007/s10489-022-04049-3.

Adhika Sigit Ramanto and Nur Ulfa Maulidevi. Markov chain based procedural music generator with user chosen mood compatibility. *International Journal of Asia Digital Art and Design*, 21(1):19–24, 2017.

Michael Rector. Historical Trends in Expressive Timing Strategies: Chopin's Etude, Op. 25 no. 1. *Empirical Musicology Review*, 15(3-4):176–201, 2021.

Bruno H. Repp. Patterns of expressive timing in performances of a Beethoven minuet by nineteen famous pianists. *The Journal of the Acoustical Society of America*, 88(2):622–641, 1990. doi: https://doi.org/10.1121/1.399766.

Bruno H. Repp. On Determining the Basic Tempo of an Expressive Music Performance. *Psychology of Music*, 22(2):157–167, 1994. doi: 10.1177/0305735694222005.

Jemily Rime, Alan Archer-Boyd, and Tom Collins. How Will You Pod? Implications of Creators' Perspectives for Designing Innovative Podcasting Tools. *ACM Trans. Multimedia Comput. Commun. Appl.*, 20(3):69:1–69:25, October 2023. ISSN 1551-6857. doi: 10.1145/3625099. URL https://dl.acm.org/doi/10.1145/3625099.

Charles Ringer, Sondess Missaoui, Victoria J. Hodge, Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Alvaro Caceres Munoz, Oluseji Olarewaju, Marian Ursu, Ben Kirman, Jonathan Hook, Florian Block, Anders Drachen, and James Alfred Walker. Time to die 2: Improved in-game death prediction in dota 2. *Machine Learning with Applications*, 12(June 2023):100466, 2023.

Riot Games. League of Legends, 2009. Video game.

Xochitl Bott Rivera. These video games have won Grammys for their soundtracks, February 2024. URL https://www.deseret.com/2024/2/7/24065143/video-games-win-grammys-soundtracks/. Section: Tech.

Curtis Roads. *The computer music tutorial*. MIT press, 1996.

Curtis Roads. *Microsound*. MIT Press, 2004.

Adam Roberts, Jesse Engel, Yotam Mann, Jon Gillick, Claire Kayacik, Signe Nørly, Monica Dinculescu, Carey Radebaugh, Curtis Hawthorne, and Douglas Eck. Magenta studio: Augmenting creativity with deep learning in ableton live. 2019.

Rebecca Roberts. Fear of the unknown: Music and sound design in psychological horror games. In *Music In Video Games*, pages 138–150. Routledge, 2014. URL https://api.taylorfrancis.com/content/chapters/edit/download?identifierName=doi&identifierValue=10.4324/9781315882697-7&type=chapterpdf.

Rockstar North. Grand Theft Auto V, 2013. Video game.

Rockstar San Diego. Red Dead Redemption, 2010. Video game.

Everett M. Rogers. *Diffusion of Innovations, 5th Edition*. Simon and Schuster, August 2003. ISBN 978-0-7432-5823-4. Google-Books-ID: 9U1K5LjUOwEC.

Joroen Rogier. Better Adaptive Music for Starship Troopers, November 2022. URL https://www.youtube.com/watch?v=9K2ISYe14no.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Jeffrey Rousseau. Report: 95% of studios are working on or aim to release a live service game, February 2024. URL https://www.gamesindustry.biz/report-95-of-studios-are-working-on-or-aim-to-release-a-live-service-game.

Philip A Russell. Relationships between judgements of the complexity, pleasingness and interestingness of music. *Current Psychology*, 2:195–201, 1982.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2020.

Ryu Ga Gotoku Studio. Yakuza: Like a Dragon, 2020. Video game.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 44(1.2):206–226, 2000. doi: 10.1147/rd.441.0206.

Sander van Rossen. Tools Summit: Geometry in Milliseconds: Real-Time Constructive Solid Geometry, 2020. URL https://www.gdcvault.com/play/1026755/Tools-Summit-Geometry-in-Milliseconds.

Timothy A. Sanders and Paul Cairns. Time perception, immersion and music in videogames. September 2010. doi: 10.14236/ewic/HCI2010.21. URL https://scienceopen.com/document?vid=9d9ed094-7bef-4e9f-978d-d3c433f8873b.

E Glenn Schellenberg, Isabelle Peretz, and Sandrine Vieillard. Liking for happy-and sad-sounding music: Effects of exposure. *Cognition & Emotion*, 22(2):218–237, 2008.

Brian Schmidt. Game Audio Industry Survey 2021, October 2021. URL https://www.gamesoundcon.com/post/game-audio-industry-survey-2021.

Brian Schmidt. Game Music and Sound Design Jobs Data Analysis - GameSoundCon 2025, aug 2025. URL https://www.gamesoundcon.com/post/gamesoundcon-game-audio-industry-survey-2025. Accessed: 2026-04-20.

Elena N Schneider, Corinna Bernarding, Alexander L Francis, Benjamin WY Hornsby, and Daniel J Strauss. A quantitative model of listening related fatigue. In *2019 9th international IEEE/EMBS conference on neural engineering (NER)*, pages 619–622. IEEE, 2019.

Kurt Schoch. Case Study Research. In Gary Buckholder, Kimberley Cox, Linda Crawford, and John Hitchcock, editors, *Research Design and Methods: An Applied Guide for the Scholar-Practitioner*, pages 245–258. Sage Publications, Inc., August 2019. URL `https://us.sagepub.com/en-us/nam/research-design-and-methods/book262895`.

Daniele Schön, Jean Luc Anton, Muriel Roth, and Mireille Besson. An fMRI study of music sight-reading. *NeuroReport*, 13(17):2285–2289, 2002. ISSN 0959-4965.

Franz Schubert. Impromptu in E-flat major, Op. 90, No. 2, D. 899, 1827. Mvt. 2.

Elliot Schwartz and Daniel Godfrey. *Music Since 1945: Issues, Materials, and Literature.* Schirmer Books, New York, NY, US, 1993.

Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. Metacompose: A compositional evolutionary music composer. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design: 5th International Conference, EvoMUSART 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings 5*, pages 202–217. Springer, 2016.

Carl Seashore. *Objective Analysis of Music Peformance.* University of Iowa Press, Iowa City, IA, 1936.

Sega. Rez, 2001. Video game.

Scott Selfon. Interactive and Adaptive Audio. In Todd M Fay, Scott Selfon, and Todor J Fay, editors, *DirectX 9 Audio Exposed Interactive Audio Development*, pages 55–74. Wordware Publishing, Plano, TX, 2003. URL `https://blackwells.co.uk/bookshop/product/DirectX-9-Audio-Exposed-by-Todd-M-Fay-Scott-Selfon-Todor-J-Fay/9781556222887`. ISBN: 9781556222887.

Hriday Shah and Mugdha Oberoi. Effect of Video Game Music on Hand Dexterity Performance in Young Adults. *Int J Cur Res Rev| Vol*, 13(02): 39, 2021.

Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural content generation in games*. Springer, 2016.

Daniel B Shank, Courtney Stefanik, Cassidy Stuhlsatz, Kaelyn Kacirek, and Amy M Belfi. Ai composer bias: Listeners like music less when they think it was composed by an ai. *Journal of Experimental Psychology: Applied*, 29(3):676, 2023.

Ilana Shapiro and Mark Huber. Markov Chains for Computer Music Generation. *jhummath*, 11(2):167–195, July 2021. ISSN 21598118. doi: 10.5642/jhummath.202102.08. URL `http://scholarship.claremont.edu/jhm/vol11/iss2/8/`.

Zhengshan Shi. Computational analysis and modelling of expressive timing in Chopin mazurkas. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, pages 650–656, 2021.

Maximilien Simard-Poirier. Out of sight: musique concrète in Resident Evil 7 | Audiokinetic Blog, April 2017. URL `https://blog.audiokinetic.com/en/out_of_sight_musique_concrete_in_resident_evil_7/`.

J David Smith and Robert J Melara. Aesthetic preference and syntactic prototypicality in music:'tis the gift to be simple. *Cognition*, 34(3):279–298, 1990.

Sony Santa Monica. God of War, 2018. Video game.

Randall Richard Spangler. *Rule-based analysis and generation of music*. California Institute of Technology, 1999.

Spitfire Audio. Labs intimate grand piano, 2018. URL `https://labs.spitfireaudio.com/packs/intimate-grand-piano`. Virtual Instrument Sample Library.

Square. Chrono Trigger, 1995. Video game.

Square Enix. Final Fantasy XIV, 2010. Video game, MMORPG; original release with ongoing expansions.

Square Enix. Final Fantasy XV, 2016. Video game.

Square Enix. Final Fantasy VIII Remastered, 2019. Video game.

Square Enix. Final Fantasy VII Remake, 2020. Video game.

Square Enix. Final Fantasy Pixel Remasters, 2021. Video game collection.

Square Enix. Final Fantasy VII Rebirth, 2024. Video game.

SquareSoft. Final Fantasy VII, 1997. Video game.

SquareSoft. Final Fantasy IX, 2000. Video game.

Ryan Stables, Satoshi Endo, and Alan Wing. Multi-Player microtiming humanisation using a multivariate Markov model. In *Proceedings of the 17th International Conference on Digital Audio Effects*, pages 1–6, 2014.

Mark J Steedman. A generative grammar for jazz chord sequences. *Music Perception*, 2(1):52–77, 1984.

Steinberg Media Technologies. Cubase 14, 2024. URL `https://www.steinberg.net/cubase/`.

Jaakko Stenros. The Game Definition Game: A Review. *Games and Culture*, 12(6):499–520, September 2017. ISSN 1555-4120. doi: 10.1177/1555412016655679. URL `https://doi.org/10.1177/1555412016655679`. Publisher: SAGE Publications.

Jonathan Sterne and Elena Razlogova. Machine learning in context, or learning from landr: artificial intelligence and the platformization of music mastering. *Social Media+ Society*, 5(2):2056305119847525, 2019. Publisher: SAGE Publications Sage UK: London, England.

Richard Stevens. The Inherent Conflicts of Musical Interactivity in Video Games. In Melanie Fritsch and Tim Summers, editors, *The Cambridge Companion to Video Game Music*, pages 74–93. Cambridge University Press, 1 edition, April 2021. ISBN 978-1-108-67028-9 978-1-108-47302-6 978-1-108-46089-7. doi: 10.1017/9781108670289.007. URL `https://www.cambridge.org/core/product/identifier/9781108670289%23CN-bp-5/type/book_part`.

Richard Stevens and Dave Raybould. *Game Audio Implementation: A Practical Guide Using the Unreal Engine.* Focal Press, 2015.

Richard Stevens, Dave Raybould, and Danny McDermott. Extreme Ninjas Use Windows, Not Doors: Addressing Video Game Fidelity through Ludo-Narrative Music in the Stealth Genre. In *Audio Engineering Society Conference: 56th International Conference: Audio for Games*. Audio Engineering Society, 2015. URL `https://www.aes.org/e-lib/browse.cfm?elib=17591`.

Feng Su and Chris Joslin. Procedural Sound Generation for Soft Bodies in Video Games. In *Motion, Interaction and Games*, pages 1–12, Newcastle upon Tyne United Kingdom, October 2019. ACM. ISBN 978-1-4503-6994-7. doi: 10.1145/3359566.3360068. URL `https://dl.acm.org/doi/10.1145/3359566.3360068`.

Tim Summers. *Understanding Video Game Music*. Cambridge University Press, 2016.

Tim Summers. Mother/earthbound zero and the power of the naïve aesthetic: no crying until the ending. In *Music in the Role-Playing Game*, pages 35–53. Routledge, 2019.

Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

Johan Sundberg, Lars Frydén, LG Bodin, and Anders Friberg. Performance rules for computer controlled performance of contemporary keyboard music. *STL-QPSR*, 28(4):79–85, 1987.

Supergiant Games. Hades, 2020. Video game.

Supermassive Games. Until Dawn, 2015. Video game.

Michael Sweet. *Writing Interactive Music for Video Games*. Addison-Wesley, Upper Saddle River, NJ, 2015.

Hao Hao Tan, Yin-Jyun Luo, and Dorien Herremans. Generative Modelling for Controllable Audio Synthesis of Expressive Piano Performance. `https://arxiv.org/abs/2006.09833`, 2020. Accessed: 2021-05-10.

Abbas Tashakkori and Charles Teedlie. *Handbook of mixed methods in social & behavioral research*. Sage Publications, 2010.

James Sebastian Tate. *Creating a coherent score: the music of single-player fantasy Computer Role-Playing Games*. PhD thesis, Durham University, 2021.

Oscar Taylor-Kent. Final fantasy vii remake: Fantasy finally becomes reality. *PlayStation Official Magazine - UK*, page 11, April 2020a.

Oscar Taylor-Kent. Exclusive collector's guide to final fantasy vii remake. *PlayStation Official Magazine - UK*, page 11, April 2020b.

Renaud Bougueng Tchemeube, Jeffrey Ens, Cale Plut, Philippe Pasquier, Maryam Safi, Yvan Grabit, and Jean-Baptiste Rolland. Evaluating Human-AI Interaction via Usability, User Experience and Acceptance Measures for MMM-C: A Creative AI System for Music Composition. volume 6, pages 5769–5778, August 2023. doi: 10.24963/ijcai.2023/640. URL `https://www.ijcai.org/proceedings/2023/640`. ISSN: 1045-0823.

Team Ninja. Strangers of Paradise: Final Fantasy Origins, 2022. Video game.

Holly Tessler. The new MTV? Electronic Arts and'plying'music. In *From Pac-Man to Pop Music*, pages 13–25. Routledge, 2008. URL `https://www.taylorfrancis.com/chapters/edit/10.4324/9781351217743-2/new-mtv-electronic-arts-plying-music-holly-tessler`.

Chance Thomas. *Composing Music for Games: The Art, Technology and Business of Video Game Scoring.* Routledge, 1 edition, 2016.

Ryan Thompson, Dana Plank, Jullianne Grasso, and Karen Cook. The music of midgar: Comparing ff7 remake's soundtrack to ff7. VGMTogether, video (MP4), 2020. URL `https://www.twitch.tv/videos/947097640`. Duration: 1:05:00.

Julian Togelius, Georgios N Yannakakis, Kenneth O Stanley, and Cameron Browne. Search-based procedural content generation. In *Applications of Evolutionary Computation: EvoApplicatons 2010: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Istanbul, Turkey, April 7-9, 2010, Proceedings, Part I*, pages 141–150. Springer, 2010.

Wiebke Trost and Patrik Vuilleumier. Rhythmic entrainment as a mechanism for emotion induction by music: a neurophysiological perspective. In *The Emotional Power of Music: Multidisciplinary Perspectives on Musical Arousal, Expression, and Social Control.*, pages 213–224. Oxford, 2013.

Augoustinos Tsiros and Alessandro Palladini. Towards a Human-Centric Design Framework for AI Assisted Music Production. In *Proceedings of the international conference on New Interfaces for Music Expression*, 2020.

Alan Mathison Turing. Mind. *Mind*, 59(236):433–460, 1950.

Ubisoft Montreal. Far Cry 5, 2018. Video game.

Nobuo Uematsu. Let the Battle Begin! Track on: Final Fantasy VII Original Soundtrack [CD], 1997a. Tokyo: Square Enix Music.

Nobuo Uematsu. Opening - Bombing Mission. Track on: Final Fantasy VII Original Soundtrack [CD], 1997b. Tokyo: Square Enix Music.

Nobuo Uematsu. Mako Reactor. Track on: Final Fantasy VII Original Soundtrack [CD], 1997c. Tokyo: Square Enix Music.

Nobuo Uematsu. Fight On! / Those Who Fight Further. Track on: Final Fantasy VII Original Soundtrack [CD], 1997d. Tokyo: Square Enix Music.

Nobuo Uematsu. Let the Battles Begin! Track on: Final Fantasy VII Original Soundtrack [CD], 1997e. Tokyo: Square.

Nobuo Uematsu. Fanfare. Track on: Final Fantasy VII Original Soundtrack [CD], 1997f. Tokyo: Square.

Nobuo Uematsu. Shinra, Inc. Track on: Final Fantasy VII Original Soundtrack [CD], 1997g. Tokyo: Square.

Nobuo Uematsu. Battle 1. Track on: Final Fantasy IX Original Soundtrack [CD], 2000a. Tokyo: Square.

Nobuo Uematsu. Victory Fanfare. Track on: Final Fantasy IX Original Soundtrack [CD], 2000b. Tokyo: Square.

Nobuo Uematsu. Danger in the Forest. Track on: Final Fantasy IX Original Soundtrack [CD], 2000c. Tokyo: Square.

Nobuo Uematsu and Daiki Ishikawa. Victory Fanfare. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020. Arranged by Ishikawa Daiki. Tokyo: Square Enix Music.

Nobuo Uematsu and Tadayoshi Makino. Noises in the Night. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020. Arranged by Tadayoshi Makino. Tokyo: Square Enix Music.

Nobuo Uematsu and Yoshinori Nakamura. Anxiety. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020. Arranged by Yoshinori Nakamura. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Let the Battles Begin! - Break Through. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020a. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Let the Battles Begin! - Ex-SOLDIER. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020b. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Let the Battles Begin! - The Hideout. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020c. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Let the Battles Begin! - A Merc's Job. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020d. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Bombing Mission. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020e. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Mako Reactor 1. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020f. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Mako Reactor 1 - Battle Edit. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020g. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Shotaro Shima. Scorpion Sentinel. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020h. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu and Yoshitaka Suzuki. Abzu. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020a. Arranged by Yoshitaka Suzuki. Tokyo: Square Enix Music.

Nobuo Uematsu and Yoshitaka Suzuki. Crab Warden. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020b. Arranged by Yoshitaka Suzuki. Tokyo: Square Enix Music.

Nobuo Uematsu and Yoshitaka Suzuki. Getaway. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020c. Arranged by Yoshitaka Suzuki. Tokyo: Square Enix Music.

Nobuo Uematsu and Yoshitaka Suzuki. Specimen H0512. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020d. Arranged by Yoshitaka Suzuki. Tokyo: Square Enix Music.

Nobuo Uematsu, , and Shotaro Shima. Shinra's Theme. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020a. Arranged by Shotaro Shima. Tokyo: Square Enix Music.

Nobuo Uematsu, Shotaro Shima, and Tsutomu Narita. Midgar, City of Mako. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020b. Arranged by Shotaro Shima and Tsutomu Narita. Tokyo: Square Enix Music.

Nobuo Naoyuki Honzawa Uematsu. Let the Battles Begin! - REMAKE. Track on: Final Fantasy VII Remake Original Soundtrack [CD], 2020. Arranged by Naoyuki Honzawa. Tokyo: Square Enix Music.

UKIE. 2023 Video Game Industry Valuation UK, April 2024. URL `https://ukie.org.uk/news/2024/04/2023VideoGameIndustryValuation`.

Jean-Frederic Vachon. Avoiding Tedium - Fighting Repetition in Game Audio. In *Proceedings of the 35th International AES Conference: Audio for Games*, 2009.

Johnny van Doorn, Alexander Ly, Maarten Marsman, and E-J Wagenmakers. Bayesian Rank-Based Hypothesis Testing for the Rank Sum Test, the Signed Rank Test, and Spearman's ρ. *Journal of Applied Statistics*, 47 (16):2984–3006, 2020.

Soumya Sai Vanka, Maryam Safi, Jean-Baptiste Rolland, and George Fazekas. Adoption of AI Technology in the Music Mixing Workflow: An Investigation, April 2023. URL `http://arxiv.org/abs/2304.03407`. arXiv:2304.03407 [cs, eess].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1–11, 2017.

Valerio Velardo. Transformers Explained Easily: Part 1 - Generative Music AI, 2024a. URL `https://www.youtube.com/watch?v=FtXT-AFzSvg`.

Valerio Velardo. Transformers Explained Easily: Part 2 - Generative Music AI, 2024b. URL `https://www.youtube.com/watch?v=ctbvMnbylsA`.

Viswanath Venkatesh. Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model. *Information Systems Research*, December 2000. doi: 10.1287/isre.11.4.342.11872. URL `https://pubsonline.informs.org/doi/abs/10.1287/isre.11.4.342.11872`. Publisher: INFORMS.

Viswanath Venkatesh and Fred D. Davis. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46(2):186–204, February 2000. ISSN 0025-1909. doi: 10.1287/mnsc.46.2.186.11926. URL https://pubsonline.informs.org/doi/10.1287/mnsc.46.2.186.11926. Publisher: INFORMS.

Viswanath Venkatesh, Michael G. Morris, Gordon B. Davis, and Fred D. Davis. User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3):425–478, 2003. ISSN 0276-7783. doi: 10.2307/30036540. URL https://www.jstor.org/stable/30036540. Publisher: Management Information Systems Research Center, University of Minnesota.

Veera Vimpari, Annakaisa Kultima, Perttu Hämäläinen, and Christian Guckelsberger. " An Adapt-or-Die Type of Situation": Perception, Adoption, and Use of Text-To-Image-Generation AI by Game Industry Professionals. *arXiv preprint arXiv:2302.12601*, 2023.

Paul C Vitz. Affect as a function of stimulus variation. *Journal of experimental psychology*, 71(1):74, 1966.

Eric-Jan Wagenmakers, Tom Lodewyckx, Hichem Kuriyal, and Raoul Grasman. Bayesian Hypothesis Testing for Psychologists: A Tutorial on the Savage-Dickey Method. *Cognitive Psychology*, 60(3):158–189, 2010. doi: 10.1016/j.cogpsych.2009.12.001.

Jennifer Walden. The fantastic sound of final fantasy 7 remake, August 2020. URL https://www.asoundeffect.com/final-fantasy-7-remake-sound/. Accessed: 2020-12-20.

Joshua Walker. Gears: A look inside the final fantasy vii game engine, February 2013. URL https://q-gears.sourceforge.net/gears.pdf. Accessed: 2025-06-20.

Sebastian Walter, Guillaume Mougeot, Yan Sun, Lihong Jiang, Kuo-Ming Chao, and Hongming Cai. Midipgan: A progressive gan approach to midi generation. In *2021 IEEE 24th international conference on computer supported cooperative work in design (CSCWD)*, pages 1166–1171. IEEE, 2021.

Megan Washburn and Foaad Khosmood. Dynamic Procedural Music Generation from NPC Attributes. In *International Conference on the Foundations of Digital Games*, pages 1–4, Bugibba Malta, September 2020. ACM. ISBN 978-1-4503-8807-8. doi: 10.1145/3402942.3409785. URL https://dl.acm.org/doi/10.1145/3402942.3409785.

Ryan Watkins. *Procedural Content Generation for Unity Game Development*. Packt Publishing Ltd, January 2016. ISBN 978-1-78528-967-5. Google-Books-ID: 8GwdDAAAQBAJ.

Paul Weir. The Sound of 'No Man's Sky'. Game Developer's Conference, 2015.

Paul Weir. The Sound of 'No Man's Sky', 2017. URL `https://www.gdcvault.com/play/1024067/The-Sound-of-No-Man`.

Paul Weir, Timea Farkas, Kyle Worrall, and Alessandro Coronas. Procedural audio in narrative game design panel, nov 2021. URL `https://www.youtube.com/watch?v=AllYuKKxks8`.

Christopher William White, Kavi Kapoor, Nicole Cosme-Clifford, James Symons, and Leonardo von Mutius. Humans perceive ai-generated music as less expressive than comparable human-made content. *Available at SSRN 5087035*, 2025.

Gerhard Widmer. Machine Discoveries: A Few Simple, Robust Local Expression Principles. *journal of New Music Research*, pages 37–50, 2002.

Gerhard Widmer and Werner Goebl. Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216, 2004. Publisher: Taylor & Francis.

Geraint A Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-based systems*, 19(7):449–458, 2006.

Tom Wijman. Newzoo's year in review: the 2023 global games market in numbers, December 2023. URL `https://newzoo.com/resources/blog/video-games-in-2023-the-year-in-numbers`.

Martin Wilde. *Audio Programming for Interactive Games*. Taylor Francis, New York, NY, US, first edition, 2014. URL `https://www.taylorfrancis.com/books/mono/10.4324/9780080472065/audio-programming-interactive-games-martin-wilde`.

A. Williamon, S. Thompson, T. Lisboa, and C. Wiffen. Creativity, originality and value in music performance. In Irène Deliège and Geraint A. Wiggins, editors, *Musical creativity: multidisciplinary research in theory and practice*, pages 161–180. Psychology Press, 2006.

Chris Wilson. Designing 'Path of Exile' to Be Played Forever, 2019. URL https://www.gdcvault.com/play/1025784/Designing-Path-of-Exile-to.

Ludwig Wittgenstein. *Philosophical investigations*. Basil Blackwell Ltd, 3rd edition, 1986. ISBN 0-631-11900-0. URL https://ld.circuitdebater.org/w/archive_files/Philosophical%20Investigations.pdf/507953616/Philosophical%20Investigations.pdf.

Thomas Wolf. A cognitive model of musical sight-reading. *Journal of Psycholinguistic Research*, 5(2):143–171, April 1976. ISSN 0090-6905, 1573-6555. doi: 10.1007/BF01067255. URL https://link.springer.com/10.1007/BF01067255.

Rene Wooller, Andrew Brown, Eduardo Miranda, and Joachim Diederich. A framework for comparison of process in algorithmic music systems. *Generative Arts Practice:*, pages 109–124, 2005. Publisher: Creativity & Cognition Studio Press.

René William Wooller. *Techniques for automated and interactive note sequence morphing of mainstream electronic music*. PhD thesis, Queensland University of Technology, 2007.

Kyle Worrall. Remaking Music for Modern Sensibilities: A case study in the evolution of music design for a more critical audience., 2021.

Kyle Worrall. Final Fantasy VII Remake: music redesign for evolved expectations across console generations. *JSMG*, 5(4):1–20, 2024.

Kyle Worrall and Tom Collins. Considerations and Concerns of Professional Game Composers Regarding Artificially Intelligent Music Technology. *IEEE Transactions on Games*, pages 1–13, 2023. ISSN 2475-1510. doi: 10.1109/TG.2023.3319085. URL https://ieeexplore.ieee.org/document/10262334. Conference Name: IEEE Transactions on Games.

Kyle Worrall, Zongyu Yin, and Tom Collins. Comparative evaluations in the wild: Systems for the expressive rendering of music. *IEEE Trans. AI*, pages 1–15, 2024.

Jian Wu, Changran Hu, Yulong Wang, Xiaolin Hu, and Jun Zhu. A hierarchical recurrent neural network for symbolic melody generation. *IEEE transactions on cybernetics*, 50(6):2749–2757, 2019.

Jian Wu, Xiaoguang Liu, Xiaolin Hu, and Jun Zhu. Popmnet: Generating structured pop music melodies using neural networks. *Artificial Intelligence*, 286:103303, 2020.

Sarah H Wu and Kevin J Holmes. Is there a "mind" behind the music? attributing music to ai can suppress narrative meaning-making. *Cognitive Research: Principles and Implications*, 11(1):19, 2026.

Shih-Lun Wu and Yi-Hsuan Yang. Compose & Embellish: Well-Structured Piano Performance Generation via A Two-Stage Approach. `http://arxiv.org/abs/2209.08212`, September 2022. Accessed: 2022-09-26.

Xinda Wu, Zhijie Huang, Kejun Zhang, Jiaxing Yu, Xu Tan, Tieyao Zhang, Zihao Wang, and Lingyun Sun. Melodyglm: multi-task pre-training for symbolic melody generation. *arXiv preprint arXiv:2309.10738*, 2023.

Zhiyang Xiang and Yibo Guo. Controlling Melody Structures in Automatic Game Soundtrack Compositions With Adversarial Learning Guided Gaussian Mixture Models. *IEEE Transactions on Games*, 13(2):193–204, June 2021. ISSN 2475-1510. doi: 10.1109/TG.2020.3035593. Conference Name: IEEE Transactions on Games.

S. Yıldırım and M. Asgari-Chenaghlu. *Mastering Transformers*. Packt Publishing Ltd, Birmingham, 2021. ISBN 978-1-80107-765-1. URL `https://www.packtpub.com/product/mastering-transformers/9781801077651`.

Zongyu Yin, Federico Reuben, Susan Stepney, and Tom Collins. "A Good Algorithm Does Not Steal – It Imitates": The Originality Report as a Means of Measuring When a Music Generation Algorithm Copies Too Much. In Juan Romero, Tiago Martins, and Nereida Rodríguez-Fernández, editors, *Artificial Intelligence in Music, Sound, Art and Design*, pages 360–375. Springer International Publishing, 2021.

Zongyu Yin, Federico Reuben, Susan Stepney, and Tom Collins. Measuring when a music generation algorithm copies too much: The originality report, cardinality score, and symbolic fingerprinting by geometric hashing. *SN Computer Science*, 3(5):1–18, 2022.

Zongyu Yin, Federico Reuben, Susan Stepney, and Tom Collins. Deep learning's shallow gains: A comparative evaluation of algorithms for automatic music generation. *Machine Learning*, 112(3):1785–1822, 2023.

Yi Yu, Abhishek Srivastava, and Simon Canales. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1):1–20, 2021.

P. Zanon and G. De Poli. Estimation of Parameters in Rule Systems for Expressive Rendering of Musical Performance. *Computer Music Journal*, 27(1):29–46, 2003. URL `http://muse.jhu.edu/article/41105`.

ZeniMax Online Studios. The Elder Scrolls Online, 2014. Video game.