Thematic Synthesis: Rethinking Generative Music with Compositional Understanding in Game and Software Development

Jasper Moore Tucker
Dartmouth College, jaspermtucker@gmail.com

Follow this and additional works at: https://digitalcommons.dartmouth.edu/masters_theses

Part of the Composition Commons, Computer Engineering Commons, Musicology Commons, and the Music Theory Commons

Recommended Citation

This Thesis (Master's) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Master's Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.
THEMATIC SYNTHESIS: RETHINKING GENERATIVE MUSIC WITH COMPOSITIONAL UNDERSTANDING IN GAME AND SOFTWARE DEVELOPMENT

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of

Master of Science

in

Computer Science

by Jasper Tucker

Guarini School of Graduate and Advanced Studies
Dartmouth College
Hanover, New Hampshire

May 2024

Examinining Committee:

__________________________
Michael Casey, Chair

__________________________
Lorie Loeb

__________________________
William Cheng

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies
Abstract

Generative music, first introduced by composers like Brian Eno and David Cope in the mid-to-late 20th century, has evolved through many stages, with diverse applications across music and technology companies as well as the game industry. However, despite this widespread interest, there remains a notable lack of the foundational understanding of composition and individual expressiveness in current generative systems that was apparent in the work of early composers. This paper advocates for a shift towards prioritizing compositional thought in system design to foster greater diversity and innovation within generative music. To demonstrate this approach, a novel system synthesizing two distinct musical pieces into a single cohesive composition was developed and evaluated. The system’s efficacy was evaluated through two user tests involving 35 participants. In the first test, participants listened to excerpts from various video game composers as well as music generated by combining different pieces. They rated the resulting compositions based on valence and energy, and then discussed their emotional and thematic qualities. In the second test, participants compared music clips generated by different models, including the system developed in this study, rating them on compositional quality. Results exhibit the system’s effectiveness in blending thematic material from its two inputs and producing compositions of comparable quality to existing models. The success of this system underscores the importance of reintegrating musical expression alongside technological advancements in contemporary generative music research and industry practices.
Acknowledgements

Thank you to the defense committee for your support and advice throughout the research process.

Thank you to my friends, family, and girlfriend for their support throughout my Dartmouth career.

Thank you to the DALI Lab for being an integral part of my graduate experience.

Thank you to the Dartmouth computer science faculty for enabling me to grow and expand my skill set.
Contents

Abstract ................................................................. ii
Acknowledgements .................................................. iii
List of Figures ......................................................... vii

1 Introduction 1

2 Related Work 3
  2.1 Overview .......................................................... 3
  2.2 Early Algorithmic Approaches to Composition ............... 4
  2.3 Expressiveness and Compositional Understanding .......... 7
  2.4 Modern Advancements in Generative Music ................. 8
  2.5 Music at the Beginning of the Game Industry .............. 11
  2.6 A Shift to Dynamic Audio .................................. 12
  2.7 Exploratory Academic Methods ................................ 19
  2.8 Bridging the Gap ............................................... 21
  2.9 How Composer’s Think: The Essence of the ‘Idea’ ......... 23

3 System Design and Implementation 26
  3.1 General Overview .............................................. 26
  3.2 Implementation Overview ................................... 29
  3.3 Graphical Representation ..................................... 30
4 Testing Methodology
   4.1 Research Philosophy ........................................... 57
   4.2 Testing Overview .............................................. 58
   4.3 Experiment 1: Combining Material .............................. 59
   4.4 Experiment 2: Achieving Quality Output ...................... 60

5 Results
   5.1 Experiment 1: Combining Material ............................ 63
   5.2 Experiment 2: Achieving Quality Output ...................... 71

6 Discussion
   6.1 ARC’s Music as a Thematic Blend of Two Pieces ............ 78
   6.2 ARC Achieving Comparably Quality Output ................. 86

7 Future Work and Conclusion
   7.1 Extensions and Changes to ARC ............................... 91
   7.2 A New Focus for Research ..................................... 93
   7.3 Application Within Games ..................................... 94
   7.4 Conclusion .................................................... 95

References ........................................................... 96
List of Figures

3.1 Connections Between Notes in a Melodic Graph . . . . . . . . . . . . 36
3.2 Connections Between Notes in a Harmonic Graph . . . . . . . . . . . 36
3.3 A Sample Melody . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
3.4 A Sample Harmony . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
3.5 Melodic Graph for Forest Interlude . . . . . . . . . . . . . . . . . . . 42
3.6 Highlighted Melody Within the Forest Interlude Melodic Graph . . 42
3.7 Extracted Melody from Forest Interlude . . . . . . . . . . . . . . . . 43
3.8 Harmonic Graph for Forest Interlude . . . . . . . . . . . . . . . . . . 43
3.9 Highlighted Harmony Within the Forest Interlude Harmonic Graph . 44
3.10 Extracted Harmony from Forest Interlude . . . . . . . . . . . . . . . 44
3.11 Extracted Melody from Sidon’s Theme . . . . . . . . . . . . . . . . . 52
3.12 Extracted Harmony from Zora’s Domain (Day) . . . . . . . . . . . . 53
3.13 Combined Piece Created by ARC . . . . . . . . . . . . . . . . . . . . 54
5.1 Rated qualities for Animal Crossing: New Leaf music clips . . . . . 64
5.2 Rated qualities for Undertale music clips . . . . . . . . . . . . . . . . 65
5.3 Rated qualities for Minecraft music clips . . . . . . . . . . . . . . . 66
5.4 Rated qualities for The Legend of Zelda: Breath of the Wild music clips 67
5.5 Words used to describe music from Minecraft . . . . . . . . . . . . . 68
5.6 Words used to describe music from Animal Crossing: New Leaf . . 69
5.7 Words used to describe music from *The Legend of Zelda: Breath of the Wild* ......................................................... 70
5.8 Words used to describe music from *Undertale* ................. 71
5.9 Votes for highest ranked clip of generations from *Kamura’s Song of Purification (Hinoa)* .................................................. 72
5.10 Votes for highest ranked clip of generations from *Tifa’s Theme* .... 73
5.11 Votes for highest ranked clip of generations from *Reflection* .... 74
5.12 Quality rankings among clips generated from *Kamura’s Song of Purification (Hinoa)* .................................................. 75
5.13 Quality rankings among clips generated from *Reflection* ........ 76
5.14 Quality rankings among clips generated from *Tifa’s Theme* .... 77
Chapter 1

Introduction

When generative music was first being explored, composers and artists like Brian Eno and David Cope had a fascination with capturing their compositional thought processes within machines. A desire to replicate the creativity and expression of musical understanding within algorithms and generative systems drove the creation of intelligent and interesting programs which still serve as deeply inspirational work to researchers today.

Current academic research, intersecting with both the technology and music industries, has continued to explore algorithmic and deep-learning approaches to music generation. Despite the strides made, current available music generators often lack the unique compositional expression inherent in the work of early, foundational composers.

Concurrently, while the early stages of the gaming industry had music primarily confined to audio loops and simple event-based changes, contemporary developments have brought in a new era of dynamic audio and real-time alterations. While some academic approaches aim for comprehensive real-time music generation, industry initiatives lean towards dynamic track mixing during runtime, prioritizing quality over parametric control. Within the gaming sphere, a disconnect persists between
researchers and industry professionals. Researchers pursue innovative yet often exploratory approaches to music generation, falling short of industry quality standards, while industry professionals opt for more conservative methods to ensure musical excellence. Both, however, have a myopic focus on real-time dynamic changes, which while important, stifles broader avenues of innovation, and often does not engage with music in a compositional and expressive way.

The gaming industry shares a similar issue to the modern academic endeavors within music and technology, where the foundational musical understanding and individual expressiveness inherent in earlier algorithmic work is largely neglected.

I advocate for a paradigm shift towards creating systems rooted in profound musical knowledge. In alignment with this notion, I developed a system that synthesizes two separate compositions into a unified composition, integrating thematic material from both inputs. This novel approach diverges from conventional industry practices, demonstrating that systems imbued with musical understanding can offer new avenues for musical exploration and expression in addition to dynamically changing audio. Moreover, this system employs algorithmic techniques to generate music of comparable quality to state-of-the-art methods, underscoring the indispensability of musical comprehension in system design.

Historically, the gaming industry has disproportionately prioritized visuals and general programming over audio innovation, hindering progress. However, recent trends suggest a growing interest in audio technologies among major gaming companies. Simultaneously, many technology companies are delving into the creation of music generation tools. With these developments, I urge researchers and professionals alike to prioritize musical understanding and compositional expression as the cornerstone of system design, fostering a new wave of innovation in both gaming and technology realms.
Section 2.1

Overview

This section covers a large variety of papers, games, talks, and industry practices which are relevant to generative music. I will first discuss early approaches to generative music, and use that background to set up a definition of ‘compositional expressiveness’ and ‘musical understanding’ which is prolific within early composers’ work on generative systems. Following that, I will discuss more recent research into generative music. Next, I will go into how music was implemented at the start of the game industry, and how generative music is applied in the current industry landscape. Following that, I will discuss more academic approaches to music within games, and how some have looked to unite academic approaches with work applied in the game industry. After discussing the work accomplished in each of these domains, and where each sector struggles, I will cover how various composers have described composition as an introduction into my relevant work.
Section 2.2

The Creativity of Early Algorithmic Approaches to Music Composition

In the 1950s, pioneers like Lejaren Hiller and Leonard Isaacson began exploring generative music, with one of the earliest computer music generation systems emerging in 1956. However, it wasn’t until the 1980s and 1990s that the field witnessed a surge of foundational work from key figures. [37] A great majority of the research and work produced at this time stemmed from an interest in encapsulating the human process within a machine. Researchers wanted to encode their own compositional thought process and philosophy into the systems they built, creating interesting, diverse, and exploratory programs which continue to serve as huge inspirations for those working in generative music and art today.

Brian Eno, who devoted a large part of his career to ambient and generative work, popularized the term ‘generative music,’ and is known for his work on several generative processes. Many of these works illustrate Eno’s interests and inspirations from the avant-garde movement, with him exploring a series of self-regulating and self-generating systems governed by some element of chance. Of particular note, the second half of his album *Discreet Music* has three performers begin at different sections of the *Pachelbel Canon* so that they “overlap each other in ways not suggested by the original score.” In *Fullness of Wind*, each player’s tempo is decreased by the pitch of their instrument. *French Catalogues* groups together sets of notes and melodies with time directions gathered from other parts of the score. In *Brutal Ardour* each player has a sequence of notes related to those of the other players, but the sequences are of different lengths so that the original relationships break down over time. This same thinking would extend to other systems, as Eno would follow inspiration from
Steve Reich’s *It’s Gonna Rain*, which uses different sized tape loops of various recordings playing simultaneously to create ever-changing ambient music. This was most notably used in his album *Ambient 1: Music for Airports*. [24]

John Cage, another pioneer of indeterminacy and non-standard applications in music, also created systems similar to natural processes, such as in *Atlas Eclipticalis*, which was composed by laying score paper on top of astronomical charts and placing notes where the stars appeared. [24] Steve Reich, a musician interested in the minimal aesthetics of art, introduced *It’s Gonna Rain* and *Come Out* in the 1960s, which were made of audio tapes and voice, and shook contemporary classical music and set further development into the minimal aesthetic in both music and art. [21] Similar to Cage and Eno, Reich’s philosophy of music informed his compositional process, thinking of music as a gradual process. In one of the most influential essays Reich has written, he states “I am interested in perceptible processes. I want to be able to hear the process happening throughout the sounding music […] Though I may have the pleasure of discovering musical processes and composing the musical material to run through them, once the process is set up and loaded it runs by itself.” [49] Reich uses his definition of process to define a large majority of his work, such as his compositions with audio tapes, and it is an expressive characteristic that has separated him from contemporary music and influenced other foundational generative musical work.

Another foundational generative composer, David Cope, built *Experiments in Musical Intelligence* (EMI) in the 1980s. EMI is a system focusing on recombinant processes that aim to discover and identify musical structure in various works and create new compositions that are faithful to their style. Cope believed that “recombinancy appears everywhere as a natural evolutionary and creative process” which describes how his philosophy of music is interwoven with the type of system he’s built. [13]
Cope goes into some more detail about parts of EMI in *Recombinant music: using the computer to explore musical style*, where he discusses the types of choices and decisions he’s had to make. In detecting patterns and combining material based on those patterns, Cope’s system finds what may signify a typical genre or style, and then carries out a hierarchical analysis. Cope describes making choices of where harmonic information should lie in saying “the tonic function remains in the same location in the new work, but it can exchange music with other analyzed music of that same function.” He also uses his own melodic intuition in deciding that “rising melodies, for example, can be followed by falling ones for balance. Accompaniments, which otherwise would be a pastiche of various motives, can be made rhythmically consistent so that they flow regularly with the melodic line.” [14] In the creation of EMI, Cope has largely encoded his own understanding of musical processes to create a system which can understand and create music the way he hears it.

Cope also writes “Ultimately, the computer is just a tool with which we extend our minds. The music our algorithms compose are just as much ours as the music created by the greatest of our personal human inspirations.” While said in reference to his own work, I believe Cope’s description of his systems and algorithms extending his mind is a rather powerful summary of how these early composers and musicians made art, and how the music they created was an extension of their own thoughts. Early composers in the influential space of generative music all worked with their own philosophical views of music which has heavily influenced the types of work they’ve each created. All of their work has been vastly influential, and inspired work in various different directions. I firmly believe that these levels of creativity, individuality, and expressiveness were foundational features of each of their work’s designs, and are necessary components of what continues to make their work so inspirational to current researchers.
Defining Expressiveness and Compositional Understanding

In the remainder of this paper, I am going to continue to refer to the presence, or absence, of ‘expressiveness,’ ‘compositional understanding,’ and ‘musical understanding.’ In using these terms, I mean to describe the qualities of systems which use an expressive understanding of musical composition as the guiding factor in their design. Eno, Reich, Cage, and Cope, discussed in the prior section, are prime examples of this. The use of these terms is meant to illustrate the use of one’s philosophical or learned understanding of musical composition as the primary influence in design. In saying that systems are expressive in this way or not, I do not mean to say that they are or are not musically expressive at all. Some may argue that all music is expressive. Others may argue that expressiveness is not inherent in music, and instead in the listener’s perception. I instead mean to emphasize musical expression that is intentionally and purposefully built into how systems function. It is compositional intention which makes this distinction. There are many tools and models which can be used by composers to produce music that one can perceive as expressive, but there is a difference between models that intentionally encode a level of expressive compositional thought in their design and those that do not. Brian Eno’s tape loop systems are a good example of systems which do encode compositional thought, as the design, function, and purpose of the system reflects Eno’s views and understanding of ambience and controlled randomness.
In more recent years, generative music has continued to be of interest to researchers, and new methods, discoveries, and breakthroughs continue to happen as the field evolves rapidly. However, the large majority of recent research efforts appear to lack that expressive and uniquely thoughtful musical underpinning that early composers had.

A number of generative music software have been created over the years by both established and new companies alike, a prominent one being Aiva Technologies, the creator of the Artificial Intelligence Virtual Artist or AIVA. Aiva Technologies describes their work as “an AI capable of composing emotional soundtracks for films, video games, commercials and any type of entertainment content.” AIVA was founded in February 2016 by Pierre Barreau, Denis Shtefan, Arnaud Decker, and Vincent Barreau, and uses a mix of deep neural networks and mathematical rules to make compositional decisions. It has been described by the founder as “able to write beautiful and emotional music, a deed that is considered to be deeply human.” [71] [59] Aiva showcases impressive works, especially with its first album *Genesis*, which the company used as one of the first great examples of what AIVA is capable of. There is unfortunately little literature comparing AIVA to other models or showcasing its practical applications in context. There are, however, publications from the various researchers at Aiva Technologies, including work on constraint-satisfaction problems with music, some notably being *On modelling harmony with constraint programming for algorithmic composition including a model of Schoenberg’s theory of harmony*, and *A model of musical motifs*, both published by Torsten Anders, a current Research and
Development Engineer at Aiva Technologies. [39], [2] Research in these areas involving constraints suggests that AIVA has rules and constraint-based algorithms as part of its implementation, although that is partly speculation, and many details about AIVA’s design are unknown.

Meta has also set out to make music generation software with the creation of MusicGen, a language model conditioned on textual descriptions or melodic features that generates audio. It is similar to AIVA in that MusicGen aims to be a generic composer capable of many styles, but works with text as an input, whereas AIVA uses a set of selected parameters. MusicGen also outputs audio, whereas AIVA outputs MIDI, or Musical Instrument Digital Interface, a standard language that allows musical computer programs to communicate with the same language. MusicGen’s public results are impressive, though it has limited utility due to only being able to generate audio that is 30 seconds long. [15]

OpenAI, a company that has dedicated itself to a large portion of current research on artificial intelligence, created MuseNet in 2019, a deep neural network that generates MIDI from discovered patterns in input MIDI files. The intention of MuseNet is also similar to that of Meta and Aiva Technologies, as it aims to mimic the style of its input, whether that input be a MIDI file, certain parameters, or text prompts. MuseNet is unfortunately no longer available for public use, and only exists as listenable examples from its website. OpenAI’s future plans into generative music are, at the moment, unknown. [44] Google has created a series of music generation systems, one of which is Music Transformer, an attention-based neural network that aims to generate music with long-term coherence. It also functions on MIDI, and is similar to the previous models in that it functions as a general purpose composer, capable of many styles. [28] Lastly, Suno, a company founded within the last couple years, have created a system able to generate both music and lyrics in a variety of styles.
Based off of textual prompts. The quality of music that Suno AI outputs is markedly impressive, and similar to other implementations discussed above such as Meta’s MusicGen. [56]

While all of these companies are doing incredible work, they all seem to be operating with the central goal of making a general music composer. The work they are each carrying out feels measurably separate from the work of early composers like Eno, Reich, Cope, or Cage. While their efforts are certainly pushing research in music generation forward, I believe that push is more in the direction of technical prowess than it is in musical expression. The way in which Steve Reich describes his philosophy in *Music as a Gradual Process* for example, is vastly different to how Google describes Music Transformer. Google writes they “demonstrated that the Transformer equipped with relative attention is very well-suited for generative modeling of symbolic music. The compelling long-term structure in the samples from our model leaves us enthusiastic about this direction of research. Moreover, the ability to expand upon a primer, in particular, suggests potential applications as creative tool.” [28] Reich forefronts everything he creates in alignment with his creative vision and intent. His philosophy towards musical understanding, perception, and creation are the focal point of each work he has created. Google’s work, as described in the creation of their transformer model, aims at improving the efficiency of models, and interprets music generation as a technical task. Creative intention feels almost as if it is a second thought. This is a common thread in the descriptions of many modern music generators. Modern research has pushed away from generative music’s roots in musical curiosity and a desire to extend our own thought-processes into that of machines, and has instead moved towards a desire to solve the large technical problem of ‘can we make a machine make music?’ Due to this, I believe we have reached stagnation. We are no longer seeing the creation of systems which think uniquely
2.5 Music at the Beginning of the Game Industry

It was the diversity of thought and approaches to composition that made the work of Eno, Reich, and others so foundational.

I do not mean to argue that the work being done by companies like Aiva Technologies, Meta, OpenAI, Google, or Suno in the generative music space is not helpful, quite the opposite in fact. These companies are making strides in the development and refinement of new technologies, and keeping a large interest in generative music. Where I believe they miss the mark however, is that compositional thought, and a desire to create an expressive system, ought to be the essential underlying characteristic of generative music systems. The diversity of philosophies towards musical composition inherent in our human understanding of music is necessary for the diversity and expressiveness of generative music systems.

Section 2.5
Music at the Beginning of the Game Industry

Having now gone over both early and recent research efforts in generative music, I’m going to introduce audio in the gaming space. First, I will introduce how music began in the industry, before then moving to describe similar problems with generative music being faced in the current industry landscape.

In the 1970s and 1980s with the release of the first video game consoles, music and audio in games were limited by hardware constraints. The Atari VCS, released in 1978, had just two channels to play audio, where music and sound effects could only be heard from two voices mixed into a single mono output. The Nintendo Entertainment System (NES), released in 1985, had a sound chip with a five-channel programmable sound generator with two pulse waves, a triangle wave, a noise channel, and a sample channel. With these limitations comes constraints on what composers and sound designers can do with sound quality, music length, and differing tracks or
voices. Continuous music was introduced in 1978 with Space Invaders and Asteroids, with simple looping non-diegetic music that only consisted of a few tones. [11]

Looping music was a prevalent practice, largely due to the limited memory available. Many of Nintendo’s earliest games, such as Donkey Kong, Donkey Kong Jr, Popeye, and Devil World, all used one or two bar loops. Even now, with lesser constraints on memory and hardware, looping music tracks continues to be a popular practice, although with much longer tracks. Karen Collins, a pivotal researcher and author in interactive audio for games, notes that “rather than being the consequence of the limited memory available on the systems, loops were, at least in part, an aesthetic that grew as the games became more popular and more complex.” [11]

Section 2.6

A Current Shift in the Game Industry to Dynamic Audio Without Largely Adopted Musical Consideration

As games have since grown to be more complex than their predecessors, music and audio have seen vast improvements in quality, length, and varied practices. A particular area which has gotten lots of attention is that of dynamic and adaptive music and audio, or audio that is able to change and react in real-time to either the player or the environment. However, while incredible developments have pushed what is possible with audio in games in these categories, there is a lack of development of systems or tools within the industry which think uniquely expressively with regard to music and composition. By this, I continue to refer to systems which use an expressive understanding of musical composition as the guiding factor in their design. Many tools and examples used in the current industry landscape work within the space of
dynamic mixing, real-time alterations to pitch, volume, or tempo, or spatialization.
Yet, these tools are often aimed at creating a more realistic or more pleasant sounding landscape, making sounds clearer, or making dynamic musical changes to match on-screen content. These are interesting and worthy areas to explore, but there is a missed opportunity in not building systems with expressive intent. By making musical expression and compositional thought central to the design of systems, developers can make more uniquely engaging and varied systems that are more emotionally resonant for players. In the remainder of this section, I will cover a series of talks, games, software, and papers which describe the current work and research going into game audio, and how some may miss an expressive or evocative musical underpinning.

At this past Game Developers Conference (GDC) held in San Francisco in March of 2024, where over 30,000 industry professionals gathered to attend talks, roundtables, and networking events, a series of sessions were held on the audio of recently released games. The talk Mixing, Music, and Mods: ‘Cyberpunk 2077’ Audio Technical Pipelines, led by both Colin Walder and Giuseppe Marano of CD Projekt Red, focused on dynamic transitions between gameplay segments and developing tools and pipelines for audio mixing and spatialization. [67] The talk Dialogue Innovation in ‘Marvel’s Spider-Man 2’: Challenges and Opportunities of Ushering in the Breathing System by Matthew Strasser from Insomniac Games focused on building a large-scale dialogue system, and discussed the technical and organizational challenges that came with such a task. [55] The session In Your Head: Audio-Driven Everything in ‘Alan Wake 2’ and its Mind Place led by Joshua Bell of Remedy Entertainment explored Alan Wake 2’s approach to cross-collaboration across departments to create more compelling and cohesive audio-visual experiences. [5] The talk Mixing ‘Marvel’s Spider-Man 2’ by Blake Johnson of Insomniac Games covers how the audio team balanced the sounds from the game’s city, score, and frequent lines of dialogue. [31]
A Shift to Dynamic Audio

While all of these talks explore important and intriguing aspects of game and audio development, they all focus on a similar set of categories: dynamic mixing for clarity, improving the connection between audio and other game elements, and technical challenges faced during development. These themes are present throughout many other talks, and are illustrative of the current landscape of game audio. However, none of these talks discuss audio systems that engage with a deeper musical purpose. The focus is on creating a clear mix, or a better workflow, or better collaboration, and not on how to create a new compositional system, or a uniquely evocative musical experience.

Looking at wider industry practice, middleware tools like Wwise and FMOD used for interactive audio in games shape how developers create and engage with audio. A 2023 survey carried out by Brian Schmidt covering the game audio industry found that around 70% of AAA game studios are using Wwise, along with 40% of mid-sized studios, and 30% of smaller indie developers. The survey covered 645 respondents. [51] Wwise emphasizes its uses in dynamic mixing, spatial audio, rule-based musical track changes, and creating sound design variety via synthesizers. While Wwise is an incredible tool, and a necessity for many studios, these features largely support and encourage real-time mixing, and don’t engage with musical meaning on a deep level. Certain capabilities like rule-based musical changes can be expressive, but that hasn’t been the main industry use. Wwise gives tools for creatives to use, and it is up to sound designers and composers to place their expressive work into the tools. But the tools Wwise develops depend on shifting industry trends and use-cases, so the focus has been on real-time alterations for mixing and transition purposes. The tools don’t encourage compositional expressiveness in design themselves. Via their current set-up, they encourage dynamic mixing and limited approaches. [30]

By looking at some popular recent releases that used Wwise significantly in their
development, you can see how the tools within Wwise are being applied. A great illustration is Bethesda Game Studios’ *Starfield*, released in September of 2023 and acquiring over 12 million players by December of 2023. [70] In an interview between Jennifer Walden and Starfield’s sound team, Dave Schreiber, a senior sound designer on the team, described how a large part of creating the ambience in various parts of the game’s atmosphere relied on developing procedural ways of placing various sound effects throughout certain biomes or cities. Another senior sound designer, Casey Coffman, explained how voiceovers could be manipulated using a series of switches, states, and effects in conjunction with game data in order to alter audio to better match a current scene according to location and distance. [66] Another great example of a recent release which used Wwise is *Avatar: Frontiers of Pandora*, which was developed by Massive Entertainment at Ubisoft and released in December of 2023. In a PlayStation Blog detailing how the game makes use of the PS5 console’s unique features, attention is brought to 3D-positionality of every sound in space, and adaptability to the player and environment. In particular, the blog provides an example of how the sounds of wildlife will react to the presence or lack of gunshots or machinery, or how plants rustle in the presence of the player. [42] Both Starfield and Avatar: Frontiers of Pandora exhibit the usual uses of dynamic and adaptive audio in games that are in-line with how Wwise promotes and describes their tools. While these go a long way to aid the developers in their goals of creating a sense of immersion, they are illustrative of the point that the industry is currently limited in its approaches to generative or dynamic systems with audio.

FMOD, an alternative middleware solution to Wwise for adaptive audio in games, emphasizes similar capabilities, such as the use of randomization or modulation to create variety in sounds, or various methods for dynamically mixing audio in responses to changes within a game. [60] In *Gameplay as Discrete Form: Leveraging Procedural*
Audio for Greater Adaptability in Video Game Music, Christiaan Clark highlights the problem with audio middleware not advocating for musical expressiveness in design. He comments that “These software are not without problems - as previously stated, all musical expression and variance is completely tied to its individual audio file.” Clark emphasizes that the design of the tools available in both Wwise and FMOD do not lend themselves to creative expression, and that expressive intention is instead completely up to the composer or sound designer. There is a large difference between creating tools that creatives can use, and creating tools that further enable or extend creatives.

The use of language around dynamic and adaptive audio in games also reveals the lack of attention that has been given to deep music engagement. In The Effect of Dynamic Music in Video Games, An Overview of Current Research, Léo Smith gives a distinction between interactive and adaptive audio, explaining that interactive audio is audio that is controlled in real-time by the player, and adaptive audio is sound that reacts to and supports the current state of the game. However, the array of examples provided for both of these definitions are either for mixing purposes or for altering music according to simple variables without expressive musical intention. Smith cites the accomplished video game composer Michael Sweet in saying that “the second technique is called vertical re-orchestration which ‘uses different musical layers (or tracks) to alter the overall musical intensity.’” [53] [57] In describing one of the main techniques used in dynamic audio, Sweet explains that it is used to alter ‘musical intensity.’ Yet, this is a rather limited definition, and doesn’t recognize that vertical re-orchestration could be used for countless other creative possibilities. Smith continues forwards, citing Karen Collins who discusses that many generative methods used in games have yet to have an emotional context, writing that “as of yet, procedural music struggles to effectively accompany the narrative functions of music in games.
which include ‘anticipating action, drawing attention, serving as leitmotif, creating emotion, representing a sense of time and place, signaling reward, and so on.’” [53], [12] Generative music in games has yet to demonstrate a deeply expressive or evocative purpose, and this is likely in part due to the limited scope of the language and tools surrounding what is possible for audio in games.

As further proof of Collins’ remark, there are many critically acclaimed games which use striking examples of interactive or adaptive audio, yet they remain confined to the purposes discussed above. *Killer Instinct*, a fighting game series originally created by Rare, syncs punctual, melodic sounds in time with the animation sequence of a player’s finishing move, quantizing them to the appropriate musical beat. *Ape Out*, a beat ‘em up developed by Gabe Cuzzillo and published by Devolver Digital utilizes a frantic, percussion-driven jazz soundtrack, where the underlying beat responds to each action taken by the player. [40] These are both vastly intriguing and interesting demonstrations of what game audio can do, but they do not focus on leitmotif, or a deep change in emotional perception that Collins refers to.

Lastly, Cale Plut and Philippe Pasquier give an extensive overview of adaptive and generative music in games in *Generative music in video games: State of the art, challenges, and prospects*. They cover 34 different games that implement various systems spread out within industry and in academia. As one of their main concluding points, they state that some of the main benefits of generative music are “a greater variety and adaptivity to music than is possible with composed music, and with less required labour. Generative systems can also create endless amounts of music, which is well suited to longer-duration games. Generative systems can also provide large amounts of variety, which is particularly useful in run-based games.” These are all valid uses of generative music, and largely illustrative of the current ways in which generative music is viewed and used, but this fails to mention that generative systems are also
2.6 A Shift to Dynamic Audio

Related Work

capable of creating deeper expression and emotional engagement, which I believe is overlooked. Of the systems covered, the only one which appears to accomplish this is within *Spore*, a game developed by Maxis in 2008. [47] Interestingly enough, this observation makes sense, given the development of *Spore’s* audio occurred alongside Brian Eno. Eno’s previous work on generative composition, and especially ambient composition, as outlined earlier in this paper, matches closely with the work on *Spore*, with both consisting of themes related to evolution. The Rolling Stones described the music within *Spore* in writing that the “music a player hears will develop and mutate along with their style of play.” [54] However, since *Spore’s* release in 2008, there has been very little exploration of similar systems within the game industry.

The only other large similar example of this is Epic Games’ current efforts with Unreal Engine Audio and MetaSounds. Epic Games, creator of Unreal Engine, one of the most widely used game creation tools, has released a number of features for their audio engine with the release of Unreal Engine 5. These include features related to audio modulation, analysis, and integration with game logic. [22] MetaSounds, a digital signal processing rendering graph, is also a current initiative of Epic Games to increase the amount of possibilities for sound designers in creating and interfacing with audio. [23] There have even been some systems already taking advantage of these tools, such as Sonic Lifeforms, created by Mitch Dörfler. Sonic Lifeforms is a procedural music generator that focuses on ambient music, and is even similar to the work of Brian Eno and Steve Reich. Dörfler mentions that the system “layers loops of different lengths on top of each other,” which is reminiscent of the tape loop systems of Eno and Reich. [17]

Throughout the game industry’s evolution, game audio has gone through several levels of innovation and advancement, starting from simple loops of short note melodies, and now achieving dynamic, adaptive, and interactive orchestral scores.
Yet despite these advancements made, a large amount of energy and resources continue to hone in on the tried-and-tested uses of dynamic audio: generating variation, creating better sounding mixes, altering intensity, and matching player action or on-screen content. With a few exceptions, the industry has been conservative in its recent approaches to audio, and recent titles have largely not considered the musical imagination as the key aspect of their design.

In addition to the work being carried out in industry, there exists a large body of research related specifically to games. This work tends to be more exploratory in nature, and while a much larger variety of topics and potential avenues within generative music are explored, there are two main issues: For one, many academic papers do not produce work that can match industry quality standards. Secondly, and similarly to the work done in industry, they engage with music on a high level that overlooks deeply expressive musical engagement.

In *Automatic Real-Time Music Generation for Games*, Steve Engels, Fabian Chan, and Tiffany Tong of University of Toronto create a system capable of producing generated music that imitates a certain style in real-time. They implement the system as a markov-model, a stochastic method that makes decisions from previous states. Their main use case that motivated their research is the ability to more quickly create original music for a game, and explain that recombinant practices are used to rearrange existing portions of a musical piece in new ways. [18] Though like other systems used in industry, this system functions as a variation generator, and does not encode a musical understanding of its own, or explore new ways for a system to
create unique musical material. Other researchers follow a similar direction, such as Cameron Bossalini of University of Texas Dallas as well as William Raffe and Jaime Garcia of University of Technology Sydney who generate audio with parameters for adjusting pitch and tempo according to player input. [6] Their focus is on increasing variety and adaptivity to the player, but do not engage with deep compositional processes. Another similar example is the work done by Marek Kopel, Dawid Antczak and Marcin Waczyński of Wroclaw University of Science and Technology, who created a recurrent neural network to generate music in real-time for a game, motivated by the efficiency gained in generated music being a more cost-effective alternative to man-made composition. Their generative compositions proved to be more favorable to players than linearly composed soundtracks, but they’re focus is similarly on creating greater adaptability and variation. [34]

A common short-coming of several academic approaches is that they struggle to reach industry levels of quality in their produced sound, limiting them from having realistic applications within industry. *A Perceptual and Affective Evaluation of an Affectively-Driven Engine for Video Game Soundtracking* uses algorithmic composition in order to more greatly match an emotional congruence with gameplay. While the authors found that perceived levels of valence of arousal were consistently rated by users to be similar between the music and the gameplay, their music failed to score highly on ‘immersion.’ They note that this is likely due to the relatively worse audio quality to that of linearly composed music, and is a drawback of real-time composition. [68] Other authors have explored various other methods and directions with real-time composition, and found similar results. *Experience-Driven Procedural Music Generation for Games* proposes that researchers and developers experiment more with experience-driven parameters in generative music, as they can produce better variety and a more relevant emotional connection to the current game state.
The generative system described changes according to perceived levels of frustration, challenge, and fun that a player is experiencing, but due to the real-time composition practice, quality suffers. [46]

Lastly, *Algorithmic interactive music generation in videogames* discusses many of the challenges faced by generative music in games, and acknowledges that much of generative music has been for the task of creating more variety, and that note-by-note generation struggles to match human-like quality and a distinctive narrative style. [16] While the investigative work done by researchers probes into interesting territories not fully embraced by the game industry, many approaches fall short of creating quality sounding music that is needed for the industry, or fail to interact with music on a deeply emotional or compositional level.

Section 2.8  
**Bridging the Gap Between Academia and the Game Industry, and Where it Falls Short**

Several authors have looked at the separation between academia and the game industry, and have attempted to fill the hole that exists. Nonetheless, the focus usually remains distinct from the early research of generative composers, and the conversations lack an engagement with expressive musical knowledge within generative systems.

Cale Plut and Philippe Pasquier, mentioned earlier for their extensive overview of current research in generative audio, highlighted the shortcomings of both academia and the industry in their work *Deep Dive: A framework for generative music in video games*. They propose moving towards a future where both academia and the industry can learn from the other. They explain that a better collaboration between academic and industrial approaches to generative music is necessary to moving towards greater innovations in both. They write that “if we see a gap in knowledge in current aca-
demic research in this area, it is mainly in the practical understanding of interactive music design and game design, which the games industry is full of. If we see a missed opportunity in industrial approaches in this area, it is in its overly conservative approach to music, missing the potential transformational power to game music design by favoring the most basic workable approach. In bringing together the two camps, we believe we can exploit the advantages of both.” In support of this assertion, they have created a model that composes an adaptive score along valence, arousal, and tension variables in order to have variability and adaptivity while also expressing particular emotions. They name it the 'Multi-Track Music Machine' (MMM), and demonstrate it is usable in the current context of audio implementation in games, by using compatible formats and realistic approaches to working with existing software.

Plut and Pasquier’s work is important since it brings to light more experimental and creative work usually done in a purely academic setting, but demonstrates its viability in an industry context. I find their paper highly influential, and strongly resonate with their message of encouraging more radical and different approaches to building systems in industry, while keeping an eye on the needs of the industry landscape. But at the same time, the tool they develop, MMM, still gives all the creative vision to the composer, and largely functions as a variation-creation machine. MM itself does not make compositional choices, nor does it encode any level of its own musical understanding. It is not expressive in its own right, and therefore still suffers from the same shortcomings of software like Wwise or FMOD: It is a tool for creatives to use, but not a tool that can further extend and enable a creative’s mind.

It’s also worth mentioning Meta’s efforts into combining research with the game industry. Meta has devoted large amounts of their companies resources to grow the presence of virtual and augmented reality, with a large focus and application within
games. Meta Reality Labs, a large research division of Meta, has been prototyping and producing production ready tools and applications for use within Meta’s products and platforms. Reality Labs includes teams focused on researching and creating tools for audio, with one of their current efforts supporting Meta’s XR Audio SDK Features, a set of tools for developers to use within the popular Unity game engine. This exposes some of Meta’s current research into sound for virtual and augmented reality spaces for developers to readily create within their products. However, the supported features consist largely of functions related to spatialization, acoustics, attenuation, and reflection, and largely don’t engage with specific musical material at all. [38]

Section 2.9

How Composer’s Think: The Essence of the ‘Idea’

A fundamental quality that I have described as being missed by many researchers and professionals who are developing generative systems is that of compositional expression. Whether that be in the research produced by large established technology companies, universities, exploratory academic papers, or game industry professionals, current research has moved away from what made generative music so intriguing and evocative in the first place: uniquely expressing emotion and compositional thought through generative processes. Many researchers are caught up in the technical challenges of generative music, where I believe more attention should be given to the compositional challenges of generative music. An understanding of musical processes ought to be the first aspect of design. In alignment with this notion, I have developed a system which encodes my understanding of musical composition to create uniquely expressive music. But before I describe my system, I am going to start by looking at
how various composers describe their compositional process, and how that informed my system design.

Hans Zimmer, in describing writing for films on the ’Mixing with the Masters’ program, states that “the first thing I need to do is figure out an idea [...] There has to be a concept.” In finding what that concept is, he continues to say that “with a score at least you know the title of the film and there will be something in the images, or in the dialogue, or in the story. There will be something that points you in a direction. All you have to do is figure out what the subtext is.” [36] Zimmer defines that a driving factor for his compositional process is finding the ‘idea’ which a film is trying to convey, and understanding how he can best illustrate that idea. Similarly, in an interview with Gramophone, John Williams commented on trying to find the concept of a film, asking “What is the texture, what is the tone of this particular film?” [25] While his music is demonstrably very different from Zimmer’s, Williams still looks to capture an essential tone.

Looking at some influential composers of the 19th and 20th centuries, this theme is consistent. Kenneth Brown, in his dissertation on Igor Stravinsky, mentions a quote from him in his interview with NBC: “Fingers are not to be despised: they are the great inspirers, and, in contact with a musical instrument, often give birth to subconscious ideas which might otherwise never come to life.” [8] While in a much different context, Stravinsky recounts the creation of musical ideas as originating in a very tactile way with his piano. In Ten Schoenberg’s Philosophy of Composition: Thoughts on the “Musical Idea and Its Presentation”, Patricia Carpenter and Severine Neff quote Schoenberg in saying “A creator has a vision of something which has not existed before this vision. And a creator has the power to bring his vision to life, the power to realize it.” [7] Schoenberg explains that the creative process begins with a new unique vision that the creator can then bring to life.
Moving to more game-specific composers, in an interview with Native Instruments, Christopher Larkin, the composer for indie hit Hollow Knight, said that “Regardless of the synth, sampler or plug-in, the most essential aspect of music- and sound-making is the idea.” He continues to say that “In terms of the personal side of things, everything I write is in some way a reflection of myself. But I’m also very conscious of crafting the emotional experience the player is ultimately going to have.” Larkin too emphasizes the ‘idea’ as having principal importance, and follows it up by saying that his music is a reflection of himself. This is reminiscent of the way Brian Eno, Steve Reich, John Cage, and David Cope have described their own compositional processes in developing musical systems. In a 2001 interview, Koji Kondo, longtime Nintendo composer, comments that “Regardless of how the hardware changes, what remains most important is that we are making music for video games, and remembering that the music should fit the game. I’m talking about musical ideas which enhance the interactivity of the game, ideas which could only be thought of in a video game.” In the same way that both Zimmer and Williams describe writing music for film, Kondo illustrates that musical ideas are essential for enhancing and fitting the scene of a game.

Each of these composers describe the compositional process as originating from nothing but the ‘idea.’ It functions as the most basic and essential part, and inspiration for the story that is present throughout everything in the composition. For some this is more abstract, perhaps for Stravinsky or Schoenberg, and for others there is an explicit subcontext, as mentioned by Kondo, Zimmer, and Williams. While it is impossible to exactly define what the ‘idea’ is, it is clear that each composer is working with their own individual understanding. Referring back to the comments from Schoenberg and Larkin, this idea is often a reflection of our own thought processes.
Chapter 3

System Design and Implementation

Section 3.1  
General Overview and Motivation

Along with the goal of capturing musical understanding within system design, I see the essence of the 'idea' as being a foundational aspect of music creation and listening. Connecting to the game and movie industries, expressing a central 'idea' within music is integral to how composers convey a specific emotional tone, as described by Zimmer, Williams, Larkin, and Kondo. In looking specifically at the game industry, certain games have hundreds to thousands of music tracks with musical themes representing certain characters, locations, events, or objects. These associations between music and in-game events are what largely comprise the emotional landscape of a game. Almost all people at least recognize if not associate pleasant memories with the iconic Super Mario Bros. Theme or Halo Theme. [33] [43]

In creating my own work within game development and music composition, I’ve wanted to more greatly integrate the importance of musical themes with generative
systems beyond the current conservative approaches of the industry. As such, I have
developed a system which aims to capture the essence of the 'idea.' At a high level,
I have created a system which synthesizes two separate pieces of music into a new
singular composition. This system largely works with the understanding that musical
pieces convey a central 'idea,' and that those 'ideas' can be synthesized together in
a musical form. Implemented with algorithmic and rule-based methods, this system
works in two major parts: extracting important thematic material from each of the
two inputs, and then combining those themes together. This systems operates with
my own compositional understanding of what constitutes musical themes, and utilizes
my own musical intuitions in combining them together. I will refer to this system
as the Algorithmic Recombinant Composer, or ARC. This system presents a unique
use-case for game music, where different themes with different in-game associations
can be blended together, bringing about a new meaning. This allows for creative
musical expression to manifest both within a composer’s work and the system’s out-
put. As for further personal motivation, designing a system in this way allows for my
own musical expression to arise from both musical composition, as well as algorithmic
system design and implementation.

To motivate this further, I’m going to use a musical moment from *The Legend of
Zelda: Breath of the Wild* to illustrate how systems designed in this way can be mean-
ingful and engaging for players. Within the game, there is a location named Tarrey
Town. This area starts off very baron and its music track reflects this, with a sparse
instrumentation and melody. The player’s goal with this area is to bring residents of
other regions to this town, creating a new community. As more residents are brought
to Tarrey Town, elements of the music from their regions are added to Tarrey Town’s
theme. Once a resident from each region is brought to Tarrey Town, the final musical
track is a transformed version of its prior self, now with various new instruments,
harmonies, and melodies from other parts of the game. [41] This a hugely impactful moment for players emotionally, and demonstrates how game music can affect people in a way unique from other media. This moment, which synthesizes and combines different musical ideas, served as a large inspiration for ARC. Practitioners in the industry can aim to have generative systems do more than alter musical intensity in real-time. Creating systems that are different and exploratory can further enhance the player experience.

The design of ARC is expressive in similar ways to the systems made by Brian Eno, David Cope, and others. In deciding how to operate on the input music, given as two distinct MIDI files, I made explicit compositional decisions on how to represent and manipulate music. Later on when combining the thematic ideas of each piece into a new composition, I used my own compositional intuition and experimentation to drive the implementation of ARC. As such, ARC executes with my own musical understanding of what a musical ‘idea’ is, and is therefore able to create new compositions that express a unique compositional language. At the same time, it presents a new and unique example of how generative music can be applied in the context of the game industry.

ARC takes two separate MIDI files as input and then gives a single MIDI file as output, representing the new composition. This choice is partly inspired by Plut and Pasquier’s MMM model mentioned earlier, and how they describe their system fitting into an industry context. MMM similarly operates on MIDI, so that the files given by the machine can be produced within a digital audio workstation by a composer or sound designer, allowing it to achieve high quality. The produced music track or track stems can then be used in-engine however a game studio sees fit. [48] ARC works the same way, giving a MIDI file as output, thus allowing for better achievable quality than real-time composition methods and integrating well with existing indus-
3.2 Implementation Overview

System Design and Implementation

try pipelines. ARC is built within C++, and makes use of the MidiFile library from Craig Sapp for reading and writing to files. [50] The choice of programming language and libraries is not vital to how ARC functions given that it is a standalone program, but I opted to use C++ due to the useful available libraries such as MidiFile and since it is a largely used language within the game industry.

ARC is an exploratory work, diverging from what is usually done within generative music, yet it is feasible within the context of the game industry, due its ability to produce outputs that can achieve higher quality than real-time composition and fit into existing industry pipelines. But more importantly, it demonstrates a uniquely expressive and creative use of generative music, demonstrating that having musical understanding as the underpinning of generative systems can spark further creative innovation in the space.

Section 3.2

Implementation Overview

Before getting into finer details about ARC’s function, I am going to establish one of the main data structures used throughout the program and describe the system’s high-level architecture.

There are two major parts within ARC: extraction and combination. The extraction portion aims to extract important thematic material from the two input pieces, while the combination portion focuses on combining that material to form a new composition. For extraction, it is important to represent the musical information within each input MIDI file in a way that is decipherable by both the program and the user. As such, each MIDI file is read into a respective note information map, represented in C++ as the data structure `map<int, vector<int>>`. This map holds information related to each note within a file. The key of the map is a unique ID value for a note,
starting with an ID of 0 for the first note within the file, and incrementing by one
for each new note. The value is a vector of integers holding values that represent the
note value, start time in the original piece, end time in the original piece, duration,
and velocity. The note value corresponds to the actual pitch of the note. All time
related values are in ticks, the smallest unit of time represented in MIDI files. The
velocity refers to how strong a note is played (like the strength of pushing down a
piano key, or amount of force used in strumming a guitar). Having this information
for each input file gives a convenient way of analyzing and making decisions from the
data.

In order to extract important thematic material from the input pieces, it is necessary
to define what ‘important’ means, and how importance can be represented within a
program. This is where my understanding of musical composition is required,
as defining what ‘important’ material is within a piece of music is an inherently
subjective decision. The musical intuition I will use to prescribe importance is that
repetition is often an indicator of significance. By observing the repetition of certain
properties within music, one can identify frequently repeated phrases or themes. It is
highly likely that repeated themes within a piece of music will sound familiar and are
representative of the piece of music. This is an example of how my understanding of
music is coupled with the decisions made in creating ARC; other musical intuitions
will arise later when introducing certain functions of the program.

To find repeated phrases or themes within a given input file, the note information
needs to be organized in a way such that ARC can search through it to find a theme.
To accomplish this, the note information of an input piece is represented as a graph,
wherein notes are vertices and edges denote the connections between them. Edges, or the connections between notes, are determined based on proximity in both time and pitch interval. For example, if a note begins right after another note begins, and their pitches are only four steps apart, then there will likely be an edge between them. However, if a note begins three full minutes after a different note, there will not be an edge between them. In creating this graph, ARC has a representation of the piece of music in linear time, where it can traverse from note to note in order to search for repeated musical phrases.

ARC looks for both important melodic and harmonic information within each piece of music. By ‘melodic,’ I am referring to a sequence of pitches that are horizontal in time. That is, each note occurs after the previous note to form a melody. By ‘harmonic,’ I am referring to a sequence of pitches that is both vertical and horizontal in time. That is, each note occurs either after or at the same time as the previous note to form a harmony. The musical intuition here is that melodies are often a sequence of notes that occur one after the other, while harmonies could be represented as chords with many notes playing on top of each other at once, or as arpeggiations where notes occur after each other in succession, or some combination of both. To encapsulate this difference, ARC constructs two slightly different graphs to represent each piece of music: one for melodic content, and the other for harmonic content. The only difference comes in how the graph considers potential edges: the melodic graph considers an edge to be possible between notes if one note occurs after another, while the harmonic graph considers an edge between notes if one note occurs at the same time or after another.

In order to capture important repeated segments, a number of maps are used to score certain qualities. One is an interval score map, which scores how many times certain intervals appear within a given piece. An ‘interval’ in this context is the pitch
difference between two notes. For example, middle C, or C3, is the MIDI note value 60, while E3 is note value 64, so from the note C3 to E3 is an interval of 4. Another map is a successive pitches map, which scores how many times a pair of successive pitches are seen. This is similar to the interval map, but instead tracks the number of times two exact pitches happen in order. For example, if the note E3 occurred right after the note C3, then the successive pitch sequence ‘C3, E3’ would increase.

A couple more important rules are added to this scoring system in order to improve the ranking of notes. If an interval between two notes is 0, that is, the two notes are the same, then they are given a score of 1 in both maps, regardless of how many times they appear. This is to prevent ARC from scoring sequences of the same note playing repeatedly highly. These sequences are often not very melodically or harmonically interesting, so ARC largely ignores them. A second rule is that scores in each map are increased by the inverse of the interval between the two notes. For example, since there is an interval of 4 from the note C3 to E3, then each time that sequence of notes is seen its score is increased by \( \frac{1}{4} \). The intuition here is that notes that are closer together in pitch are more likely to be of the same sequence than notes that are very far away in pitch. This scoring system prioritizes notes that are close by to one another, without completely discounting the possibility that notes within the same sequence may be distant in pitch. The decision to score intervals in this way is as much a musical decision as it is a technical one, partly taking inspiration from how scoring appears in weighted K-nearest neighbors within machine learning. Within weighted K-nearest neighbors, the strength between data points is proportional to the inverse of the distance between them, similar to how the strength between notes in ARC is proportional to the inverse of the interval between them. [29]

Lastly, in order to construct a graph of each musical piece, ARC iterates through all vertices, the notes of the piece, and then decides where to place edges. ARC first
searches for ‘candidate followers’ of each note, before then deciding on the ‘followers’
of each note. Each note can be considered to be a ‘source’ which has a number of
note ‘followers.’ These terms are meant to describe the relationship between the left
hand and right hand side of an edge. On the left side of an edge, there is a source
note, while the right side of an edge has a note which follows this source note, or a
follower.

A ‘candidate follower’ of a note is any note which occurs after another note,
but within a certain allotted time frame. These two notes could be considered as a
potential edge. ‘Followers’ of a given note are then any potential candidate followers
which are the closest in both time and interval. ARC iterates through each note in
the piece, developing a representation of which notes follow which other notes, or in
other words, placing edges between nearby notes in the graph.

Having now described the reasoning behind creating a graphical representation of
each of the input pieces and some of the musical intuitions used, below is pseudocode
which depicts how the graph is created in larger detail:

```java
void findCandidateFollowers (currentNote):
    for each note after currentNote:
        if looking for a melody:
            if note startTime > currentNote startTime and
            note startTime < currentNote endTime + window size:
                candidateFollowers[currentNote].Add(note)
        else if looking for a harmony:
            if note startTime >= currentNote startTime and
            note startTime < currentNote endTime + window size:
                candidateFollowers[currentNote].Add(note)
        else:
            break
```
void createGraph():
    for note in notes:
        candidateFollowers[note] = {}
        followers[note] = {}

    // for each note, first find its candidate followers
    for each currentNote in notes:
        potentialSources = {currentNote}
        sharedCandidates = {}
        findCandidateFollowers(currentNote)

    // then for each note after, find their candidate followers
    for each note after currentNote:
        if candidateFollowers[note] is empty:
            findCandidateFollowers(note)

        // keep track of candidates which are shared
        // between the current note and any following notes
        for each candidate in candidateFollowers[note] and
            candidateFollowers[currentNote]:
            sharedCandidates.Add(candidate)

        // if this note and currentNote share candidate followers,
        // then it is a potential source for the candidates
        if candidateFollowers[note] and sharedCandidates
            share any notes:
            potentialSources.Add(note)

    // turn candidate followers into followers if
    // they are not shared by any other notes
for candidate in candidateFollowers[currentNote] and not in sharedCandidates:
    followers[currentNote] = candidate
    updateScoreOfEdge(currentNote, candidate)
    sharedCandidates.Remove(candidate)

// for any candidate followers that are shared between notes,
// have them follow the potential source which is closest
// in both interval and start time
for candidate in shareCandidates:
    closestSources = source from potentialSources...
    closest in start time to candidate
    nearestSource = source from closestSources...
    closest in interval to candidate

    followers[nearestSource] = candidate
    updateScoreOfEdge(nearestSource, candidate)

    for otherSource in potentialSources that
    are not nearestSource:
        candidateFollowers[otherSource].remove(candidate)

To make the distinction between a melodic and a harmonic graphical representation more clear, the below images display sections of a small MIDI file with notes connected by arrows. The notes are the colored rectangles, where the vertical position represents pitch and the horizontal position represents time. The arrows denote which notes are followers of which other notes, or the edges of the graph. The melodic graph for a simple MIDI file is shown in figure 3.1.
Figure 3.1: Connections Between Notes in a Melodic Graph

Notice how the edges are largely horizontal, with each arrow moving from left to right. A harmonic graph for the same sample is shown below in figure 3.2.

Figure 3.2: Connections Between Notes in a Harmonic Graph

Within the harmonic graph, there are more edges that are vertical, as well as edges that are horizontal. This is meant to capture the structure of various harmonies within
3.4 Melody and Harmony Search

With a graph constructed representing each input piece, ARC can then search through the melodic graph to find a melody and the harmonic graph to find a harmony for each piece. Since each edge of the graph has a score corresponding to the repeated use of specific intervals and pitches, ARC can search for the highest scoring sequence of a specified length, returning a sequence of notes where each interval between notes occurred frequently.

There are a few more important design choices made in extracting melodies and harmonies from the graph that rely on musical knowledge. While the scoring of the graph described in the previous section discourages sequences of singular repeated notes, it is possible that ARC may have scored a sequence that repeats a simple pattern many times very highly. This is common for several pieces of music, where an ostinato, short arpeggiation, or simple background harmony may repeat over and
over, thus acquiring a high score. However, to find the most interesting material within a piece, an additional step is added to ARC to prevent these types of musical themes from being the extracted melody or harmony. ARC will carry out the following operations for a number of iterations: first, ARC searches for the highest scoring sequence of notes of a specified length via dynamic programming. Then, it decreases the score of each edge in the extracted sequence to 1 in the graph. Then, it will search for identical subsets of a certain length within the extracted sequence. If it finds one, then it deletes each note in the sequence from the identical subset forward. Next, if the extracted sequence of notes has a higher cumulative score than any previously found, it is set as the best sequence. After all iterations are complete, ARC will have extracted a sequence of notes with few repeating subsections.

Adding these functions that discourage repeated subsections and favor longer, unique sequences is a creative choice I made in implementation. I believe that ARC performs better and extracts more interesting material with this design. One could argue that these short, frequently repeated sequences that I am discouraging in ARC’s design are actually more representative of certain pieces or styles of music. However, I believe that speaks to the strength of making musical decisions within system design. Another person creating a system similar to ARC may make entirely different decisions than I, potentially creating a system that produces music that sounds completely different than ARC. This is why underscoring each function within musical systems with a musical understanding is vital for creating more varied and intriguing systems.

Below is some pseudocode more clearly defining how ARC extracts a melodic or harmonic sequence from a graph representation of the music.
void searchGraph(notes, followers, scores, iterations, maxSequenceLength, maxSubsetLength):

    bestSequence = []

    for iterations:
        edgesInCurrentSequence = []
        // find the current highest scoring sequence
        currentSequence = findHighestScoringSequence(notes, followers, scores, maxSequenceLength)

        // keep track of the edges within the extracted sequence
        for each edge in currentSequence:
            edgesInCurrentSequence.Add(edge)

        // search for identical subsets of a certain length and
        // delete notes if one is found
        deleteIdenticalSubsets(currentSequence, maxSubsetLength)

        // calculate the total score of the sequence
        // after making deletions
        Score = 0
        for edge in currentSequence:
            score += scores[edge]

        // keep track of highest scoring sequence
        if score > all previous scores:
            bestSequence = currentSequence

        // lower the scores of the edges
        // of the most recently extracted sequence
        for edge in edgesInCurrentSequence:
            
scores[\text{edge}] = 1

\text{return bestSequence}

To better visualize the result of this process, the below images display extracted melodies and harmonies from various pieces. Referring back to the small sample MIDI file used in the prior section in figures 3.1 and 3.2, an extracted melody for this piece may look like that within figure 3.3.

![Figure 3.3: A Sample Melody](image)

Each note is followed another note in time horizontally. A harmony for the same piece may look like the one displayed below in figure 3.4.
This extracted phrase has more connections vertically in time, with notes beginning at the same time. To give a more illustrative and complex example, below is a plausible melodic graph for a portion of the piece *Forest Interlude* from *Donkey Kong Country 2* by composer David Wise [69], shown in figure 3.5.
Melody and Harmony Search System Design and Implementation

Figure 3.5: Melodic Graph for *Forest Interlude*

The same graph is shown below, but the red edges display the melody that ARC chose when executed. This is displayed in figure 3.6.

Figure 3.6: Highlighted Melody Within the *Forest Interlude* Melodic Graph

That melody extracted from the piece then appears as below in figure 3.7.
Now to demonstrate the difference between the extracted melody and harmony, figure 3.8 below shows a plausible harmonic graph for the same piece. Note how, similar to the sample MIDI file shown earlier, there are more vertical edges between notes.

The red edges below in figure 3.9 then display what ARC selects as the harmony for the piece.
3.4 Melody and Harmony Search System Design and Implementation

Figure 3.9: Highlighted Harmony Within the *Forest Interlude* Harmonic Graph

That harmony, isolated from the rest of the piece, then appears as below in figure 3.10.

Figure 3.10: Extracted Harmony from *Forest Interlude*

By constructing the graphs in different ways with a different formation of edges, ARC can extract different information for a melody and a harmony. The melody tends to be more vertical in time, with each note following the last, while the harmony allows for notes to begin at the same time, representing a more vertical as well as horizontal
There are a few important hyperparameters that have a significant impact on what melodies or harmonies that ARC extracts. These are iterations, or the number of times the graph will be searched for a sequence, maxSequenceLength, the maximum number of notes allowed within an extracted sequence, and maxSubsetLength, the number of notes in a sequence that need to be identical for ARC to consider them as identical subsets. The number of iterations has the least impact on ARC’s function, as the best sequence will likely be found within the first few iterations, and adding more and more iterations is unlikely to produce a higher scoring sequence than the ones already found. The max sequence length will determine how many notes a melody or harmony can have. This shouldn’t be too short, since ARC may then not have a chance to find interesting sequences, and it shouldn’t be too long, because then ARC may return as much of the original piece as it can. This parameter is a balance between limiting the sequence, and allowing it to be long enough to represent interesting material. Lastly, the max subset length arguably has the highest impact of what melody or harmony is found. This parameter decides how many consecutive notes need to be the same for the program to consider them identical. To illustrate this, if maxSubsetLength is set to three, and the extracted sequence contains the notes C3, E3, G2 at the beginning of the sequence, and then C3, E3, G2 again later in the sequence, then ARC will delete this second appearance of these notes as well as every note that follows them. Having this parameter too low will result in ARC making deletions too frequently, and having this parameter too high will result in ARC making almost no deletions at all. Some pieces of music may benefit from a lower threshold, while others would require a higher one. Similarly to the max sequence length, the max subset size is a delicate balance between eliminating too much and too little information.
The last two sections, consisting of constructing graphical representations of the two input pieces and searching them for melodies and harmonies, were part of the first component of ARC: extraction. This section will go into details about the second large component of ARC: combination. This portion of the program is focused on taking the information given by the extraction processes, and assembling them into a new composition.

In capturing the information of the input pieces, each piece was read into a note information map implemented as `map<int, vector<int>>`. When extracting the graphs for melodies and harmonies, the sequences of notes returned were implemented as `vector<tuple<int, int>>`. This data structure represents a list of notes, where each note is an ID and a note pitch. This was helpful in the extraction part of the program since many of the operations on extracted sequences considered the note values, or pitches. For the combination part of the program, the new composition will be represented as a number of new data structures. There are two structures of `vector<tuple<int, int, int>>` representing two different lines with a note pitch, duration, and velocity. One line will represent the new composition’s melody, and the other the harmony. They each have an accompanying `vector<int>` structure representing the times at which the notes at the corresponding index start.

The general structure of the combination portion of ARC follows a number of heuristics and algorithmic processes to merge together the extracted melodic and harmonic information of the two input pieces. As before, a number of my own musical choices will need to be made in the design of the system. There are an infinite number of ways in which two different compositions could be combined into one, and
the design of ARC follows just one of those possibilities. The structure of the new
composition produced by ARC follows an ABAB structure, where A and B are two
contrasting sections. The A section will use an extracted melody from one piece, and
the extracted harmony from the other, while the B section will use the melody and
harmony not featured in the A section. The ABAB structure is inspired by the use of
looping music that is still dominant in much of the game industry, as the composition
made by ARC could very feasibly be played on a loop with this structure. [11]

To start the combination process, ARC first chooses what the first melody will
be. Whichever extracted melody is longer in terms of the number of notes will be the
melody of the A section. Thus, the A section will use the harmony of the opposite
piece. The program then decides what length of time the A section will play the
first melody for. In order to make the A and B sections a similar length of time, the
greatest common denominator of the melody lengths is taken, and is then added iter-
atively to the section duration until the duration is divisible by both melody lengths.
This ensures that the A and B sections will be balanced since each melody will play
for a similar amount of time, and each section will end around the end of their re-
spective melodies. Once decided on length, some of the melodies and harmonies are
transposed so that all music is within the same key. The key of the A section melody
is determined by analyzing the pitch content of the note information map created
earlier during extraction. The same is then done for the A section harmony and B
section melody, which both come from the opposite piece. The A section harmony
and B section melody are then transposed to the key of the A section melody by mov-
ing each scale degree to the corresponding pitch in the new key’s scale. This allows
all the music to be compatible within the same key while retaining the intervallic
relationships between all notes.

With the melodies and harmonies chosen, structure decided, and section lengths
and pitch content established, ARC can then arrange the music together. Two 
\texttt{vector\langle tuple\langle int, int, int\rangle\rangle} structures are created, one for the new melodic 
line and the other for the new harmonic line. An additional two \texttt{vector\langle int\rangle} struc-
tures are created to represent the note start times of each respective line. Notes 
from the A section melody are added into the new melodic line and corresponding 
note start times data structures until the decided section duration is reached. Sim-
ilarly, notes from the A section harmony are added into the new harmonic line and 
corresponding note start times data structures until the same duration is reached. 
Simple motivic transformations are applied to the notes in the melody and harmony 
as they are added to the new composition. These transformations include inversion, 
retrograde, as well as a small amount of random noise. The choice to create motivic 
transformations of the input material is a common practice, discussed in depth by 
famous composer Schoenberg, as it increases the variety of the new composition while 
retaining the identity of the original material. [45] After inputting the notes into the 
new composition, the start times of the notes in the harmonic line are then quantized 
to be at the same time as the notes in the melodic line if they are too close together. 
This ‘fits’ the notes of the harmonic line to the melodic line, letting the melodic line 
largely dictate the rhythm of the composition, and give the composition a feeling of 
a more solid structure.

Next, notes from the B section melody are added into the melodic line and corre-
sponding note start times. The same is done for the B section harmony and harmonic 
line. A few additional steps are added here to ensure there is a smooth transition from 
the A to the B section. The B section melody may be far away in pitch from the A 
section melody, so it is transposed up or down a number of octaves until it is as close 
as possible in pitch. The same is independently applied to the B section harmony so 
that it is close in pitch to the A section harmony. Then, the pitch of the notes at the
end of the previous A section melody are moved progressively up or down to be closer to the starting pitch of the following B section melody. Similarly, the velocities of the notes at the end of the A section melody progressively interpolate to the velocity of the first note of the B section melody. These same processes are applied to the harmony. This allows for a smoother transition in both pitch content and velocity between the A and B sections, and contributes to making the new composition feel more cohesive. The above procedures are then applied again, adding in material for the next A section, then interpolating the previous B section to transition into it, then adding in material for the final B section, and interpolating the last A section to transition into it. This completes adding all notes into the full ABAB structure of the new composition. As a final step, ARC conducts an individual analysis of both the melodic and harmonic lines to measure the standard deviation of the note distribution. If the notes from the melodic and harmonic lines significantly overlap, and there is a range of notes not utilized by either line, then the melody may be transposed up an octave, harmony transposed down an octave, or both to increase separation between them. This promotes a clearer distinction between parts of the composition, and prevents pitches from clashing too much.

The pseudocode below showcases the above described process:

```cpp
void getMaterial(piece1Melody, piece1Harmony, piece2Melody, piece2Harmony):
    // gets all extracted information ready for combination

    // assign longer melody as leading melody
    if piece1Melody is longer than piece2Melody:
        melodyA = piece1Melody
        harmonyA = piece2Harmony
        melodyB = piece2Melody
```

49
3.5 Algorithmic Recombination System Design and Implementation

harmonyB = piece1Harmony

else:
    melodyA = piece2Melody
    harmonyA = piece1Harmony
    melodyB = piece1Melody
    harmonyB = piece2Harmony

// decide on duration that each section will last
gcd = greatestCommonDenominator(melodyA, melodyB)
duration = length of melodyA
while duration is not divisible by both melodyA and melodyB:
    duration += gcd

// transpose material so all music is in same key
scale = getScale(melodyA)
transposePiece(melodyB, melodyA, scale)
transposePiece(harmonyA, melodyA, scale)

// return required information for combination
return melodyA, harmonyA, melodyB, harmonyB, duration

void combineMaterial(melodyA, harmonyA, melodyB, harmonyB, duration):
    // combines prepared material into new composition

melodicLine = [], harmonicLine = []
melodicStartTimes = [], harmonicStartTimes = []

// add in notes for first A section,
// quantizing harmony to melody
addNotesToMelodicLine(melodyA, melodicLine, melodicStartTimes, duration)
addNotesTo HarmonicLine(harmonyA, harmonicLine,
harmonicStartTimes, duration)
quantizeNotes(harmonicStartTimes, melodicStartTimes)

// add in notes for first B section, quantizing harmony, then
// interpolating values for a smoother transition
addNotesToMelodicLine(melodyB, melodicLine,
  melodicStartTimes, duration)
addNotesToHarmonicLine(harmonyB, harmonicLine,
  harmonicStartTimes, duration)
quantizeNotes(harmonicStartTimes, melodicStartTimes)
interpolateNoteValues(melodicLine, harmonicLine)
interpolateVelocities(melodicLine, harmonicLine)

// add in notes for second A section, quantizing harmony and
// interpolating values
addNotesToMelodicLine(melodyA, melodicLine,
  melodicStartTimes, duration)
addNotesToHarmonicLine(harmonyA, harmonicLine,
  harmonicStartTimes, duration)
quantizeNotes(harmonicStartTimes, melodicStartTimes)
interpolateNoteValues(melodicLine, harmonicLine)
interpolateVelocities(melodicLine, harmonicLine)

// add in notes for second B section, quantizing harmony and
// interpolating values
addNotesToMelodicLine(melodyB, melodicLine,
  melodicStartTimes, duration)
addNotesToHarmonicLine(harmonyB, harmonicLine,
  harmonicStartTimes, duration)
quantizeNotes(harmonicStartTimes, melodicStartTimes)
interpolateNoteValues(melodicLine, harmonicLine)
interpolateVelocities(melodicLine, harmonicLine)
3.5 Algorithmic Recombination System Design and Implementation

// separate the pitches of melody and harmony if possible
separateLines(melodicLine, harmonicLine)

// return information that will be written to new MIDI file
return melodicLine, melodicStartTimes,
    harmonicLine, harmonicStartTimes

To illustrate how ARC combines material together, below is an example of some material that ARC extracted, as well as how it merged them together. Figure 3.11 shows the extracted melody from Sidon’s Theme from The Legend of Zelda: Breath of the Wild, composed by Hajime Wakai, Manaka Kataoka, and Yasuaki Iwata. [64]

Figure 3.11: Extracted Melody from Sidon’s Theme

Figure 3.12 below then shows the extracted harmony from Zora’s Domain (Day) from the same game and composers. [65]
3.5 Algorithmic Recombination System Design and Implementation

Figure 3.12: Extracted Harmony from Zora’s Domain (Day)

Figure 3.13 then displays the piece created by ARC which combines the above mentioned pieces. The start of the composition uses the melody from Sidon’s Theme and harmony from Zora’s Domain (Day). For a clearer view of the separate components, the melody from Sidon’s Theme is shown in green, while the harmony from Zora’s Domain (Day) is shown in shades of blue and purple.
Within the figure above, some of the musical choices made in the algorithmic design of ARC are visible, such as how some notes between the melody and harmony are quantized to start at the same time, or how the harmony is inverted in a series of parts, with arpeggiations moving downwards and upwards, while the extracted harmony only moved upwards in pitch.

While I’ve already discussed several decisions made for musical purposes such as choosing the section duration length or interpolating note values, there are a number of hyperparameters that further impact how the new composition is created. A significant one is how close two notes between the harmony and melody should be in order to quantize them. While ARC is combining material from separate compositions with their own rhythms, the new composition should still have rhythmic consistency. Therefore, notes within the harmony need to be quantized to notes in the melody if they start at similar points in time. If this threshold is too low, then not enough notes will be quantized and the composition will lack rhythmic consistency, but if
this threshold is too high, then too many notes in the harmony will quantize to the melody, making it more difficult to hear the harmony as unique. Another important hyperparameter is how many notes at the end of each section should interpolate to the first note of the next section. This interpolation between pitches and velocities is necessary for a smooth transition between sections. However, if too many notes are used for interpolation, then the material may fail to sound recognizable, as ARC would be changing the pitches and velocities too much. Yet, if there are too few notes used for interpolation, then the transition between sections will feel sudden and jarring. There are many more hyperparameters within ARC and musical decisions behind each of them, but these examples are illustrative of how a unique musical understanding of composition is inherent in ARC’s design.

Section 3.6

System Output and Summary

When ARC has finished running, it outputs a singular MIDI file that represents the thematic material of both input files while making certain compositional choices that express a unique style of composition. There are hundreds of design choices in building ARC that are intertwined with my understanding of music, such as the choice to represent the new composition as a melodic and harmonic line, how I’ve defined what ‘melodic’ and ‘harmonic’ mean in this context, and what constitutes important thematic material. There is a musical choice made for every data structure used and algorithm written, and another person’s implementation of ARC may likely produce music completely different than mine. The beauty of designing systems with musical understanding as the cornerstone of development is that it creates systems which further extend how their creators express musical thought. ARC showcases how musical systems can be built expressively with regard to composition in new
exploratory ways, while still being practical in an industry context.
Chapter 4

Testing Methodology

Section 4.1 Research Philosophy

In order to test the effectiveness of ARC, I will refer back to the introduction of this paper, where I ask for a paradigm shift towards making systems rooted in musical knowledge in order to make further advancements in musical expression and exploration within generative music. ARC is an example of a new creative use for generative music that displays both my own compositional understanding, while also incorporating other thematic material. Therefore, testing will focus on two questions: 1. Does ARC express qualities of a thematic blend of the two input pieces, thereby accomplishing the purpose it was built to do? In other words, does ARC convincingly synthesize the thematic material of two inputs into a new composition? 2. Can ARC produce outputs that are of comparable quality to other available generative systems?

Depending on the answers to these questions, testing will reveal if exploratory generative systems made with compositional thought as the cornerstone of design can produce new interesting material that is on par with current research and appli-
cations. These questions are aimed at discovering if the creative purpose with which ARC was built is fully realized, and if it demonstrates that quality can still be reached via exploratory, expressive approaches.

Section 4.2

Testing Overview

Answering questions about how people perceive the thematic essence of music involves a largely subjective evaluation of how music emotionally impacts someone. As such, evaluation will largely consist of qualitative measures and open-ended discussion regarding the emotional qualities of certain pieces of music. Some quantitative measures will also be included with a series of likert scales to evaluate other metrics related to musical outputs.

Two experiments will be run to answer each of the research questions. The first experiment will focus on the musical output that ARC gives, and if it represents a combination, or thematic blending, of its two input pieces. The second experiment will focus on the musical output of ARC in comparison to the outputs of other available generative models.

Participants will be from a variety of musical backgrounds to represent a breadth of different perspectives related to how music is perceived. The experiments are now discussed in more detail below.
Section 4.3

Experiment 1: Representing and Combining Thematic Material

This experiment focuses on the question: Does ARC express qualities of a thematic blend of the two input pieces, thereby accomplishing the purpose it was built to do? For this experiment, participants will listen to four groups of three musical pieces. They will listen to one group of three music clips at a time. In each group of music clips, two clips will be from a video game composer, and one will be generated by ARC as a combination of the other two. While listening to the three clips, participants will rate each music clip on two five-point likert scales: valence and energy. This is inspired by a number of studies which use valence, arousal, tension, and energy as emotional metrics, most notably within Plut and Pasquier’s evaluation of their MMM system described earlier. [48] To maintain simplicity, this evaluation method follows the method used in *Feel the Moosic: Emotion-based Music Selection and Recommendation*, which uses the two qualities valence and energy in building an intuitive emotion-based music recommendation application. [26] Participants will also write a few words describing the music clips they listen to. After listening to each group, each participant will then be asked to verbally compare the music clips with regard to their compositional material and emotional tone. All music will be represented as MIDI files, and produced on the same virtual instrument so that sound quality is consistent. This will keep the focus on compositional and perceived emotional differences rather than differences in sound quality. The virtual software instrument used is the Autograph Grand by Spitfire Audio used in the digital audio workstation Logic Pro. [4] [3]

All music clips will be roughly 30 seconds long both to ensure consistency and
so that participants do not spend too long on any one music clip. They will also be presented in a random order. The music clips in each group are from the games: *Minecraft*, *Animal Crossing: New Leaf*, *Undertale*, and *The Legend of Zelda: Breath of the Wild*. The specific pieces in each section that are not generated are *Sweden* and *Living Mice* by C418 from *Minecraft*, [10] [9] 1 A.M. and 5 P.M. by Kazumi Totaka, Manaka Kataoka, and Atsuku Asahi from *Animal Crossing: New Leaf* (as well as other games in the series), [61] [62] *It’s Raining Somewhere Else* and *Premonition* by Toby Fox from *Undertale*, [19] [20] *Sidon’s Theme* and *Zora’s Domain (Day)* by Hajime Wakai, Manaka Kataoka, and Yasuaki Iwata from *The Legend of Zelda: Breath of the Wild*. [64] [65] These pieces are used to introduce a variety of different compositional styles as well as energy and valence levels. All music is also chosen from varying game series to demonstrate ARC’s utility with game soundtracks.

In summary, participants will listen to a series of groups of three music clips. In each group, two clips will be from a video game composer, and one generated by ARC. Participants will rate each music clip on valence and energy, write a few words describing them, and then verbally compare them with regard to compositional and emotional similarities and differences.

**Section 4.4

Experiment 2: Achieving Quality Output in Comparison to Other Generative Models**

This experiment focuses on the question: Can ARC produce outputs that are of comparable quality to other available generative systems? The design of this experiment is similar to the last. Participants will listen to three groups of three music clips, listening to one group at a time. In each group, one music clip will be generated...
4.4 Experiment 2: Achieving Quality Output Testing Methodology

by ARC, another will be generated by Aiva Technologies’ AIVA, and another generated by Google’s Magenta Studio. While listening to the music clips within a group, the participants will rate the perceived compositional quality as well as their level of interest in each one on five-point likert scales. They will also note which of the three music clips they believe is best. After listening to a group of three clips, the participant will also be asked to verbally compare each composition with regard to compositional quality.

Similarly to the last experiment, all music will be from MIDI on the Autograph Grand virtual software instrument within Logic Pro to maintain consistent sound quality, and all music will be roughly 30 seconds long. For accurate comparisons, the first 30 seconds of music will be used from each generated composition. They will also be presented in a random order.

AIVA and Magenta Studio were chosen because each system is able to take a MIDI file as input, and produces MIDI as output, thereby being as similar to ARC as possible compared to other generative systems. Both are also recent and ongoing research projects from Aiva Technologies and Google, demonstrating an accurate representation of the current work in generative music. [58] [1]

All three models, ARC, AIVA, and Magenta Studio, will be given the same MIDI file as input. Since ARC requires two pieces in order to combine them, which is different from other generative systems which can only take one input, it will be given the same MIDI file twice. Therefore, ARC will be creating a composition having the same input that the other models will receive. The compositions used as input for the models are also from a series of games. Those compositions are Reflection by Christopher Larkin written for the game Hollow Knight, [35] Kamura’s Song of Purification (Hinoa) by Satoshi Hori for Monster Hunter Rise, [27] and Tifa’s Theme by Nobuo Uematsu for Final Fantasy VII. [63]
4.4 Experiment 2: Achieving Quality Output  Testing Methodology

Both AIVA and Magenta Studio have a series of parameters that can be altered to affect the composition. To restrict the amount of human intervention and leaving more up to the choices of the systems, I only altered parameters related to the length of the music to ensure it would be at least 30 seconds. All other parameters related to instrumentation or style were left to the individual systems to decide. After the music was generated by all systems, they were then produced on the same virtual software instrument mentioned above.

In summary, participants will listen to a series of groups of three music clips. In each group, one clip will be generated by ARC, one by AIVA, and another by Magenta Studio. Participants will rate each one on compositional quality and how interesting they find the clip, decide on the best of the three, and then verbally compare them with regard to compositional quality.
Participants were sourced around a college campus, with a total of 35 participants completing both experiments. Participants were between ages 19 and 32, with 16 being male and 19 female. Participants had a variety of musical backgrounds, with 12 identifying as having a large background in formal musical study or performance, 14 identifying as having some amount of formal musical experience, and 9 identifying as having little to no formal musical experience.

Section 5.1

Experiment 1: Representing and Combining Thematic Material

This experiment had participants rate the valence and energy levels of various music clips, as well as describe their emotional and compositional characteristics. In each group of music clips, two were from a video game composer and one was a combination of the two created by ARC. On average, the music clip produced by ARC had valence and energy levels that were between those of the two others. This is most clearly apparent with the samples from the games *Animal Crossing: New Leaf* and *Undertale*, present in figures 5.1 and 5.2. The standard error is also shown on each plot. Across
5.1 Experiment 1: Combining Material

the board, the standard error for each rating is rather small.

![Valence and Energy Levels:](image)

Figure 5.1: Rated qualities for *Animal Crossing: New Leaf* music clips
5.1 Experiment 1: Combining Material Results

Figure 5.2: Rated qualities for *Undertale* music clips

Similar results appeared for the other music samples, but there was slightly more variance with regard to valence. In the music clips from *Minecraft* and *The Legend of Zelda: Breath of the Wild*, the music clips produced by ARC had energy levels between the two others, but a lower valence than both. In both cases, the valence was below, but close to, the lower of the two other compositions. The summary of these ratings is present within figures 5.3 and 5.4.
5.1 Experiment 1: Combining Material

Valence and Energy Levels:

Minecraft

Figure 5.3: Rated qualities for Minecraft music clips
5.1 Experiment 1: Combining Material

With regard to the words participants used to describe each music clip, as well as their verbal descriptions comparing the music clips on compositional and emotional qualities, all of the examples had evidence displaying that ARC’s musical output had qualities that were similar to or in-between the other two music clips in each group. Word clouds were created to represent the frequency of word usage in participants’ written descriptions of each piece of music. The more a word was used by participants, the larger it is in the graphic. Words demonstrating that the music produced by ARC combined thematic qualities can be observed most clearly in the descriptions for the pieces from *Minecraft*, *Animal Crossing: New Leaf*, and *The Legend of the Zelda: Breath of the Wild*, shown in figures 5.5, 5.6, and 5.7.
5.1 Experiment 1: Combining Material Results

Figure 5.5: Words used to describe music from Minecraft

(a) Sweden

(b) Living Mice

(c) Combination
5.1 Experiment 1: Combining Material

Figure 5.6: Words used to describe music from *Animal Crossing: New Leaf*
5.1 Experiment 1: Combining Material Results

The word usage for the music from *Undertale* may indicate that each music clip had a more unique identity perceived by participants, with each piece receiving a large number of unique words in their descriptions. However, many verbal descriptions that participants used to compare the pieces reveal a similar trend of ARC’s output feeling as a mixture or in-between state of the other two pieces. This will be discussed in further detail in the following Discussion section. The word cloud for the music clips from *Undertale* is shown in figure 5.8.
This experiment had participants rate the compositional quality and level of interest in various music clips. In each group of music clips, one was generated by ARC, another by AIVA, and another by Magenta Studio. In addition to rating each one on
quality and their level of interest in the music, participants also identified the best composition of each group, and verbally compared them with regard to compositional quality. Results tended to show a general preference towards ARC’s output compared to either AIVA or Magenta Studio. This is most prominently displayed by the music clips participants chose as the best composition for the examples generated from the piece Kamura’s Song of Purification (Hinoa), and Tifa’s Theme, shown in figures 5.9 and 5.10.

![Highest Ranked:](image)

**Figure 5.9:** Votes for highest ranked clip of generations from Kamura’s Song of Purification (Hinoa)
5.2 Experiment 2: Achieving Quality Output

Figure 5.10: Votes for highest ranked clip of generations from *Tifa’s Theme*

For the other group, where music was generated by *Hollow Knight’s Reflection*, most participants still preferred ARC’s output, but with a larger number of people preferring AIVA’s music. Varied levels of preference were also described in participant’s verbal responses, which will be discussed further into the Discussion section. This is displayed in figure 5.11.
In ranking the compositional quality and level of interest of each piece of music, participants rated ARC as having both the highest quality and interest for the samples generated from *Kamura’s Song of Purification (Hinoa)*, although the differences between each piece are minor. This is shown in figure 5.12.
5.2 Experiment 2: Achieving Quality Output

The groups of music clips generated from Reflection and Tifa’s Theme show similar results, although rankings are even closer. ARC outperforms in quality by a small amount in each group. For level of interest, ARC is tied for the highest rating in one group and a close second in the other. These similarities in ratings are also common topics that participants mentioned in their verbal responses, which are discussed further below. This is displayed within figures 5.13 and 5.14.

Figure 5.12: Quality rankings among clips generated from Kamura’s Song of Purification (Hinoa)
Figure 5.13: Quality rankings among clips generated from Reflection
5.2 Experiment 2: Achieving Quality Output

**Quality Metrics:**

*Pieces Generated from ‘Tifa’s Theme’*

![Quality Metrics Chart]

Figure 5.14: Quality rankings among clips generated from *Tifa’s Theme*
Chapter 6

Discussion

The data collected from participants showcases that ARC was largely successful in its two main goals: creating a composition that evokes a thematic blend of two pieces, and achieving similar quality output to existing generative models. In discussing each portion of the results, I will first go over data related to ARC’s output being perceived as a thematic blend, and then go over results related to comparing models based on compositional quality. In talking about each of the participants, I will refer to them as P1 through P35, or participant 1 through participant 35. There are many quotes where participants refer to “clip 1” or “music clip 2,” in reference to one of the compositions. While the participants did not know which music clip pertained to which composer or model, for readability I will replace these instances with the composition they are referring to in square brackets. For example, instead of “I enjoyed clip 1 the most” I will write “I enjoyed [Living Mice] the most.”

Section 6.1

ARC’s Music as a Thematic Blend of Two Pieces

Regarding how participants rated the valence and energy levels of each music clip, the combination piece created by ARC was largely in-between the two other clips in
each group. Energy was consistently rated in the middle for every example. This can be observed in figures 5.2, 5.3, and 5.4. In some cases, valence was lower or closer to one of the inputs, especially in figure 5.4. There are likely two main causes for this. For one, there were many cases where participants would describe ARC’s music as more closely aligning with one piece’s thematic material than the other.

In listening to music from *Minecraft*, a number of participants compared ARC’s output to *Living Mice*. P4, P8, P10, P16, P21, P22, P30, and P35 all said that ARC’s output and *Living Mice* both had similar qualities. P4, P21, and P22 all described them both as having the same energy levels and similar moving patterns. P35 noted that they sounded strikingly similar, adding that “[ARC’s output] compliments [Living Mice].” P10 said that they both had “similar levels of complexity” and were “more upbeat than [Sweden].” On the other hand, a few participants had thought that ARC’s output and *Sweden* were more similar to each other instead. P32 said that they shared similar elements, and P2 said that “[ARC’s output] was depressing like [Sweden]; more energetic, but still not joyful.”

There was a similar occurrence with the music from *Animal Crossing: New Leaf*. Many participants noted similarities to ARC’s output and *5 P.M.*, with P16, P23, and P29 all referring to them both as swingy, jumpy, and jazzy. P20 thought that “both [ARC’s output] and *5 P.M.* felt like they’d be heard in a coffee shop.” Meanwhile, P32 compared ARC’s output to *1 A.M.*, saying that “[ARC’s output] is a faster version of *1 A.M.*”

For the music from *Undertale*, P7, P14, P27, and P32 likened ARC’s output to *It’s Raining Somewhere Else*, with P7 calling *It’s Raining Somewhere Else* “a progression of [ARC’s output]” and P27 similarly referring to *It’s Raining Somewhere Else* as “a heavier version of [ARC’s output].” P14 and P32 thought they had very similar energy levels. Yet P17, P18, P20, P21, P24, and P30 all thought ARC’s output
6.1 ARC’s Music as a Thematic Blend of Two Pieces

and *Premonition* were more similar. P17 and P30 noted that both compositions had similar pauses between notes, P20 referred to them as both “calming,” and P21 said that “the first two are more similar; [*Premonition*] is a more stretched out version of [ARC’s output].” P24 even said they were “much more similar to each other than [*It’s Raining Somewhere Else*].”

For the music from *The Legend of Zelda: Breath of the Wild*, P1, P19, P21, P24, and P25 said that *Sidon’s Theme* and ARC’s output both shared qualities related to tempo, being funky, perky, short, choppy, or jumpy. P12 said that [ARC’s output] was “a remix of [*Sidon’s Theme*].” On the other side, P4 and P18 said that ARC’s output and *Zora’s Domain (Day)* conveyed similar feelings of peacefulness, pleasantness, and exploration. P31 noted that they both used the same theme, and P6 thought that “[ARC’s output] was a more extreme version of [*Zora’s Domain (Day)*].”

Another possible reason for ARC producing music with a valence that is lower than the inputs is that some found ARC’s output to be less pleasant than the other compositions due to dissonances and some “odd” or “weird” notes.

After listening to the clips from *Minecraft*, P6, P7, P14, and P15 found ARC’s output to be less organized than the others, as well as more chaotic and sporadic. P19 noted that it was “sometimes dissonant” which made them rate it lower on valence. After listening to music from *Animal Crossing: New Leaf*, P3, P15, P30, and P31 commented on how they thought ARC’s output was more disorganized or random. While listening to music from *The Legend of Zelda: Breath of the Wild*, P5, P6, P15, and P23 thought that ARC’s output was less pleasant and cohesive.

While ARC’s valence may be lower in some cases for these reasons, it is strong evidence that ARC can produce music that convincingly shares similar thematic and emotional qualities of its two inputs if some participants compare it heavily to one composition while some participants compare it heavily to the other. As even stronger
6.1 ARC’s Music as a Thematic Blend of Two Pieces

According to the evidence, many participants more explicitly described ARC’s output as a mix between the other two.

In listening to the music from *Minecraft*, P2 said ARC’s output felt like a “collage” of the others. P18 described it as “a balance between the two, like dawn if the others were night and day.” P26 stated that *Sweden* felt “dreamy,” *Living Mice* “had more tension,” and ARC’s output was “also dreamy, but there’s a lot of movement.” P27 said that “[*Living Mice*] is a more hyper version of [ARC’s output],” and then continued to say that the pieces went in the order of *Living Mice*, ARC’s output, then *Sweden* in terms of timbre, placing ARC in the middle. P28 said that *Sweden* was “definitely saddest,” *Living Mice* had “a hint of hope,” and ARC’s output was “not as sad as [Sweden],” but felt “defeated.”

For the music clips from *Animal Crossing: New Leaf*, P1, P4, P5, P22, and P28 all noted that the pieces became faster and “groovier” throughout, where ARC’s music clip was in the middle between a slower and faster piece. P1 asked afterwards: “is [ARC’s output] a blend?” Both P9 and P13 said ARC’s output felt “in-between,” and P17 said the pieces became gradually more upbeat, saying that “[*1 A.M.*] was like the beginning of spring, and [ARC’s output] was more tentative like ‘we’re getting there,’ and [*5 P.M.*] was more cheerful.” P19 had a similar sentiment saying that ARC’s output was “more playful” than *1 A.M.*, but *5 P.M.* was the “most playful.” In discussing the music from *Undertale*, P2, P3, and P13 all said that each piece of music had a similar mood. P5 stated that “[ARC’s output] was a mix of both, having the suspense of [*Premonition*], but faster.”

In talking about the music clips from *The Legend of Zelda: Breath of the Wild*, P2 believed that “[ARC’s output] combines elegant qualities of [*Zora’s Domain (Day)*] and passionate qualities of [*Sidon’s Theme*].” P17, P27, and P28 all described ARC’s output as feeling neutral between the other pieces, where *Sidon’s Theme* was more
6.1 ARC’s Music as a Thematic Blend of Two Pieces

upbeat and adventurous, and Zora’s Domain (Day) was smoother and lighthearted.” P32 stated that Sidon’s Theme was “very energetic” and Zora’s Domain (Day) felt “cold”. ARC’s output was then “less energetic than [Sidon’s Theme]. Maybe joyful in the moment, but not for long,” demonstrating a mix of the perceived properties of the other two music clips. These above examples demonstrate that ARC is capable of combining the thematic and emotional elements of two distinct compositions.

Additionally, while some people preferred ARC’s composition to the others, several people noted it didn’t have as clear a theme compared to the other compositions within each group. This sentiment was most commonly brought up in reference to music from The Legend of Zelda: Breath of the Wild. A couple participants mentioned it with the music from Minecraft and Undertale as well.

For the music from Minecraft, P10, P11, P12, P20, and P34 all shared that they thought ARC’s output was better than the other compositions. P34 believed it was “far better” than the others. For the music from Undertale, P22 said that ARC’s output sounded “like [it] would be an actual song.” However, other participants found it to be less engaging, with P12 calling it “unremarkable,” and P28 thinking it “wasn’t super memorable.” This thought was the most frequent while listening to the music from The Legend of Zelda: Breath of the Wild. P8 said that Sidon’s Theme felt like a “character’s theme,” and Zora’s Domain (Day) was like a “classical concert piece,” yet in response to ARC’s output they said “I haven’t heard something like that.” P9 similarly thought that ARC’s output “evoked less emotion,” and P13 “couldn’t put much of a feeling to it.” P26 thought there was “not an emotionally clear message.”

Meanwhile, the emotions for the other pieces were clear, with P7 saying that Zora’s Domain (Day) was like “observing something of grandeur,” and P9 likened it to “an expensive hotel lobby.” P20 believed it felt like “visiting a new place.” Sidon’s Theme being referred to as a character’s theme is obviously fitting, given it quite literally is
a character’s theme, and *Zora’s Domain (Day)* having themes of grandeur and new locations is also appropriate given the music is for a grand location in the game’s world.

A potential downside of mixing thematic material discovered in these comments is that a blend may not have as strong a message in either direction of the two inputs it is blending, not holding as clear an identity. In the case of *The Legend of Zelda: Breath of the Wild*, Sidon’s Theme was an extremely adventurous, upbeat and fun theme meant to convey a confident, whimsical character. On the opposite side, *Zora’s Domain (Day)* was much more calm, grand, and bliss, meant to evoke a vast and serene location. Some people found the combination of the two to give interesting results, and they enjoyed the composition, while other people found its emotional and thematic message to be vague and unclear compared to the others.

It is worthy to note the some participants did describe ARC’s output from this section as having its own identity, with P33 referring to it as “exploring the unknown” and both P14 and P23 using the analogy of raindrops, which is oddly suitable considering Sidon is a fish and Zora’s Domain is a kingdom in the water.

Some people recognized the music from a few games, mostly with *Minecraft*, and a few with *Animal Crossing: New Leaf* and *The Legend of Zelda: Breath of the Wild*. It’s interesting to note how this knowledge impacted how they emotionally received certain music. Participants who recognized a piece usually gave them a higher valence in the ratings. Interestingly enough, some participants incorrectly identified ARC’s output as being from the video game series.

In listening to music from *Minecraft*, P2, P3, P23, and P29 all recognized *Sweden*, while P8, P14, P16, and P22 thought it sounded like *Minecraft*, but did not recognize that it in fact was. For *Animal Crossing: New Leaf*, P14 thought that one of the tracks reminded them of Nintendo, and P19 said they felt “like background music
6.1 ARC’s Music as a Thematic Blend of Two Pieces  

Discussion

to Animal Crossing or something.” P35 knew it was from a game, but did not remember which. P2 was the only participant to recognize music from *Undertale*, but said “this is Undertale isn’t it?” while listening to ARC’s output. For *The Legend of Zelda: Breath of the Wild*, P3 and P7 recognized *Zora’s Domain (Day)*, and P7 also recognized *Sidon’s Theme*.

While it is interesting data to see whether participants recognize a piece of music or not, or believe that they recognize it, it did not have a significant impact on the results. Recognizing a music clip only affected how participants perceived a piece of music emotionally. In this case, it is still important to see if ARC’s music can produce a thematic blend whether the input music is recognized or not. In a practical setting where ARC would be used within a game, it is highly plausible that the two compositions that ARC would blend together would be recognizable pieces.

Now looking at the word usage from the written descriptions given by participants, there is also evidence supporting that ARC’s output was a thematic blend of the inputs, particularly with *Minecraft*. Looking at the word clouds in figure 5.5, *Sweden* was often described as slow, somber, melancholic, tranquil, and calm, while *Living Mice* was more moving, hopeful, inspirational and dramatic. ARC’s output was then in the middle as bittersweet, sad, calming, and melancholic, sharing words with the other pieces, and also having some word usage that lied between the slow and hopeful themes, such as “bittersweet.” The same appeared for the other examples, where *Animal Crossing: New Leaf* in figure 5.6 had many participants referring to both ARC’s output and *5 P.M.* as jazzy, energetic, and rhythmic, but noting that *5 P.M.* was the highest energy. *The Legend of Zelda: Breath of the Wild* in figure 5.7 had a similar occurrence, with both ARC’s theme and *Sidon’s Theme* being energetic and fast and then ARC’s theme and *Zora’s Domain (Day)* seeing themes of fantasy and exploring a new location. *Undertale* also saw shared themes in figure 5.8, with
ARC’s theme and *Premonition* being slow and calming. There were some differences people perceived, with *Premonition* being mysterious, suspenseful, and anticipatory while *It’s Raining Somewhere Else* felt full, relaxing, moving, beautiful, and dynamic. Meanwhile, ARC’s theme was melancholic, serene, calming, slow, yet uncertain and somber. Each piece in this section seemed to hold a more unique identity, but the qualities people described seemed to fit ARC’s output between the others.

Lastly, while some participants more explicitly described ARC’s output as a thematic blend, and others thought it was more similar to one piece of music than the other, there was also a consistent theme that each group of music clips displayed a cohesive idea, demonstrating ARC’s ability to stay within the context of its inputs.

For the music from *Minecraft*, P9, P11, P17, and P29 said that all pieces had a sad connotation, and P23 thought they were all melancholic. P24 said they all had very similar valence and energy. For the music from *Undertale*, P10 called all pieces “graceful,” and P23 and P31 called them similar in tone. For the music from *The Legend of Zelda: Breath of the Wild*, P8 and P9 both thought that each of the pieces were all relatively high energy.

The analysis of ARC’s output as a thematic blend of two pieces reveals intriguing findings across all the various musical contexts. Participants consistently placed ARC’s compositions between the thematic and emotional qualities of its inputs. In many instances, participants described ARC’s music as a “mix” or “blend.” Even when participants did not explicitly state that ARC felt like a blend, they frequently gave comparisons between ARC’s music and other pieces in the same group, showing that ARC consistently produced outputs of similar thematic qualities. However, some participants noted that while ARC’s output displayed a mix of thematic characteristics, it sometimes lacked a clear identity compared to the original compositions. Despite differing perceptions, participants’ ratings, written descriptions, and verbal
comparisons all display ARC’s ability to evoke thematic blends. This proves that ARC accomplished what it was designed to do, and illustrates that generative systems can be built with new, interesting, and diverse goals that are perceptible to listeners.

Section 6.2

ARC Achieving Comparably Quality Output

Now looking at the data collected comparing ARC to other generative models, the results demonstrate that ARC produces compositions of a similar or higher perceived compositional quality to other generative models. ARC was ranked highest in every category among the other models, with the greatest difference displayed in the group of music generated from Kamura’s Song of Purification (Hinoa) in figure 5.9, and a smaller difference in the sections with music generated from Reflection and Tifa’s Theme in figures 5.10 and 5.11. There was an interesting level of variance in what participants considered to be good compositional quality. Many expressed that they enjoyed ARC’s compositions since they found them to be more interesting and they didn’t feel rigid or predictable, while others did not like the dissonant choices ARC occasionally made, and thought the compositions were less intuitively structured.

For example, in the section with music generated from Reflection, P1 thought ARC had the most interesting melody, and P3, P4, P5, and P30 thought it felt the most “full” and “complete.” P3 “liked [ARC] a lot more,” and P14 believed it had a “greater base” compared to the other pieces. P17 said it had “the greatest capacity to be a standalone piece.” For music generated from Kamura’s Song of Purification (Hinoa), P1, P29, and P30 thought that ARC’s composition had the most depth. Similarly, P35 said it had “a little more oomph” compared to the other pieces. P2 said it wasn’t too chaotic, but also “a little unpredictable,” yet it felt “wondrous and
beautiful.” P18 thought it had the “greatest elaboration,” and P19 thought it “made
the most sense lyrically.” For music generated from Tifa’s Theme, P3, P5, P7, P11,
P13, and P15 thought ARC’s output was the most interesting. Some noted that there
was a stark difference between pieces, with P7 saying it was their favorite “by far,”
and P14 saying “easy, [ARC]” when picking the best composition. P35 said that the
composition was “something I would try to play.” But on the other hand, a large
number of participants also found the same piece to be unpleasant. P2 said that the
harmony at times was “cacophonous.” P4, P16, and P19 thought it had too much
dissonance, and P23 thought it was “too muddy.” P6 referred to it as “chaotic.” Dif-
ferent participants viewed many of the same qualities in different lights, showing the
discrepancy in how people perceive music, and what some may view as high quality.

On a similar note, many preferred AIVA’s compositions since it produced music
that was much more consonant on average and followed a predictable pattern. Yet
many also said they found it to be too boring and stale.

For the music generated from Reflection, P7 and P8 found AIVA’s music more
interesting since it was “less random.” P19 thought the structure made the most
sense, and P29 thought it “felt cohesive.” Yet P1 said it was “boring,” and P5 called
it “very basic.” P8 and P32 referred to it as bland, and P21 said it was “way too
simple.”

For the music generated from Kamura’s Song of Purification (Hinoa), P9 and P28
thought AIVA’s output was “pleasant” and “consistent.” P16 noted that it had a nice
structure and the chords “weren’t as dissonant compared to [ARC].” P24 said that
they enjoyed the repetitive pattern and P26 thought it was “elegant” and “easy to
understand.” However, P4, P15, P16, P19, P21, P30, and P32 all expressed that it
was too predictable and very repetitive. P7 thought that the pattern was nice, but
“[ARC] had a larger progression.”
For music generated from *Tifa’s Theme*, P2, P4, and P6 found AIVA’s music to be interesting, and P16 and P19 specifically mentioned that it had “no dissonance” or was “nice and consistent.” P26 found that it had “clear changes.” But P5, P14, P28, and P30 thought it had worse quality, with P30 saying that it had “nice chords, but wasn’t overly interesting.”

These comments, with some participants preferring ARC’s bolder choices with dissonance over AIVA’s consistent melodies, and others finding ARC to be messy while AIVA was pleasant and structured, align with the close ratings between ARC and AIVA in figures 5.10 and 5.11. While ARC was usually picked as the best composition in each group, AIVA was a preference for a large number of participants.

Continuing on this same theme of participants enjoying different aspects of music, some participants found Magenta Studio’s melodies to be intriguing and follow interesting contours, while the large majority found them confusing and off-putting.

For music generated from *Reflection*, P2 thought that the unpredictable nature of the melody made it “more interesting.” But P1, P8, P26, and P32 expressed that they found it to be too random or “weird.” P13 and P22 used the analogy of someone who “just learned piano,” and P28 found the randomness “frustrating.”

For music generated from *Kamura’s Song of Purification (Hinoa)*, P8 enjoyed Magenta Studio’s output since it had the least dissonance, and P21 thought it was “nice and regular with some surprises hidden within rests.” P31 thought it was “mysterious,” and that “the silence added to the second part.” However, P1, P14, P26, P28, and P29 all thought that the same silence was too long. P26 said that the large rest “threw me off and didn’t catch me again.” P12 thought it was “monotonous,” and P5 thought it felt like it was being played by someone who “just started learning.”

Similar sentiments were shared for the music generated from *Tifa’s Theme*, with P2, P4, P21, P28, P29, P30, and P32 calling Magenta Studio’s output too random or
boring. P6 said it “felt kind of flat,” and both P14 and P16 said that they couldn’t understand the melody. P14 commented that “[Magenta Studio] sounded too scared, as if AI were to make music.”

In looking at how participants rated the relative compositional quality and level of interest for the pieces, the results aligned with the participants’ ratings for the best compositions. However, in all categories there was not much variance. Both quality and level of interest were ranked from 1 to 5, and the highest average quality achieved was a 3.89 by ARC in the section generated from _Kamura’s Song of Purification (Hinoa)_ in figure 5.12 and the highest average level of interest achieved was a 3.74 shared by both ARC and AIVA for music generated from _Tifa’s Theme_ in figure 5.14. Many comments also noted that the difference between the clips was not drastic, while some others felt that there was a greater distinction between some.

For music generated from _Reflection_, there was a common sentiment that none of the compositions were stellar. P4 said that “I didn’t like any of them,” and P8 thought “none were great.” P9, P12, P19, P23, P30, and P35 all concluded that they were of very similar quality. However, P3 said that they “liked [ARC] a lot more.” For music generated from _Kamura’s Song of Purification (Hinoa)_, P7 thought all pieces were similar in quality, but P11 said “definitely [ARC]” when choosing the best composition. A similar moment happened for music generated from _Tifa’s Theme_, with P14 saying “easy, [ARC]” when picking the best music clip, and P7 saying it was their favorite “by far.”

The overall attitude that these music clips were not fantastic may emphasize the overall weakness of current generative music’s difficulty in creating quality compositions on par with human composition. While this seemed to be common, with some people rating ARC’s valence lower in the first experiment due to it feeling like lower quality, and the quality ratings in the second section never reaching an average above
4 out of 5, some people were surprised to hear that the first section had generated music, with others incorrectly identifying ARC’s output as human-made, certainly displaying the potential and ability for generative systems to make quality and interesting music.

The findings discussed in this section showcase that ARC produces compositions of similar or higher perceived compositional quality compared to other current models. Although there was variance in individual preferences, ARC consistently ranked highest in every category among the other models. This underscores ARC’s capability to produce compositions that are both intriguing and of commendable quality, even with differing perceptions and preferences among listeners.
Chapter 7

Future Work and Conclusion

While ARC is a successful demonstration of how compositional thought can continue to be central to the design and implementation of generative music systems, there are certainly improvements that can be made to how it operates. In addition, the success of ARC is also demonstrative that there is value in this approach to generative music, that is, creating systems with compositional understanding as the first essential pillar of design. This also extends to applicable use cases within the game industry, which has a history of applying generative music systems in games ranging from research projects to AAA releases.

Section 7.1

Extensions and Changes to ARC

While ARC has certainly shown that it can produce music that can blend the thematic material of two other compositions as well as achieve a level of quality that surpasses other existing models, it certainly struggles to achieve these consistently. For some inputs, ARC does a great job in both understanding what some melodic and harmonic themes are, and merges them in interesting ways. In other cases, ARC can struggle to find a melody or harmony, sometimes returning a series of notes that
don’t make much sense to a human listener. In other cases, ARC may extract good melodies and harmonies, but make poor choices in combining them together. Sometime, two pieces of music may be so different from each other that combining them together makes it nearly impossible for ARC to make the result sound good. Other times, ARC may fail to account for certain possibilities. After all, attempting to have a series of algorithms understand all of Western music is a near impossibility. There is so much information in music that is inherent in how we perceive it, that encoding my own level of musical composition into ARC naturally limits it to perform best with the types of music I have experience with. With more time, I would like to make ARC more robust and consistent with its outputs, to make it more viable as a tool. In its current state, ARC is an excellent prototype that showcases that having compositional understanding in design can produce new, meaningful, and quality results. However, more needs to be done to bring it from a prototype stage to a finished product.

Additionally, there are a number of musical intuitions and functions that I had attempted to use in ARC’s design, but they had either failed to produce desired results, or created unexpected problems. For example, in one iteration, the harmonic line that ARC produced was the result of a statistical harmonic analysis. Details such as the standard deviation of the range of notes, direction of the melody, and median note duration were used to make compositional choices. However, this created music that was more often than not confusing, awkward, and inconsistent. In a different iteration, the tempo at the ends of sections would be increased or decreased to transition between material of different pieces. This was a nice and interesting effect, but later failed to make sense when notes from both pieces would play simultaneously. Trying different approaches and experimenting with different musical ideas is necessary to create a generative system with a compositional focus, but I believe there
are further choices that could be made to give ARC a clearer and more identifiable style. To refer back to the Discussion section, when testing ARC some participants thought the emotional message of ARC’s music was not as clear as other human-made compositions. With more time, I would like to explore further options to give ARC a more robust and identifiable style that better communicates my understanding of music.

Lastly, with more time ARC could benefit from additional testing with generated music integrated into a game environment. The questions asked within this paper were targeted at users’ perceptions of ARC’s generated music, so the testing done to evaluate ARC on a musical basis was done within an isolated context. However, different questions related to ARC’s effectiveness specifically within gaming could certainly benefit from additional research and testing.

Section 7.2

**A New Focus for Research**

There are certainly changes that could be made for ARC to be more robust, but for the purposes of this paper, it serves as a strong example of how compositional expression can be integrated within generative music systems. Combining thematic material from different compositions is a unique focus for a musical system, and requires that the system understand what ‘thematic material’ is. Through my own musical understanding, I was able to create a system which accomplished this. Not only that, but it can produce music of better quality than other models, and has the ability to fit within the context of the game industry. This should show researchers that it is possible for exploratory methods such as this one to be intriguing, unique, and ingrain an individually creative vision that differentiates itself from other uses, and still be viable and plausible in industry settings. I urge other researchers to be
more creative and expressive with how they conceive generative systems, and give a level of musical expression to their systems that allows them to be more individual and unique. We ought to remember what made generative music so fascinating to composers such as Brian Eno and Steve Reich, and use that fascination to continue to develop innovative and expressive systems. It is the individuality and variety in our perceptions and interpretations of music that drive the uniqueness and innovation in generative music.

Section 7.3

Application Within Games

I have highlighted the game industry as a context in which generative music is frequently used. I have also pointed out that a large majority of efforts from both researchers and industry professionals have gone towards real-time alterations. While this is certainly an interesting capability of generative systems, and a worthy feature given the adaptive and interactive nature of games, I would like to further emphasize that creating musical systems with a compositional underpinning should also be a focus.

Following the footsteps of Plat and Pasquier discussed earlier in both the Related Work and System Design and Implementation sections, it is feasible to create these types of systems and still reach a level of quality necessary for industry needs. [48] ARC is created within a programming language that is frequently used in the game industry, and since it operates as a standalone system, can create MIDI files that can then be produced in a digital audio workstation to raise their sound quality before then placing them in-engine. This establishes the practicality of incorporating compositional exploration into game development processes.

The growing attention to audio in the game industry, exemplified by Epic Games’
advancements in Unreal Engine Audio and MetaSounds, underscores the potential for innovation in compositional expression. While developments such as Sonic Lifeforms showcase promising strides in procedural music generation, it is essential for the industry to continue innovating in the realm of unique compositional expression alongside technical advancements in audio technology.

Section 7.4  

Conclusion

Pioneers of generative music such as Brian Eno, Steve Reich, and David Cope embedded their compositional understanding into the systems they created. However, in contemporary research and industry applications, there is a tendency to prioritize technical challenges, established industry practices, and the desire to create a generalized generative composer. In this pursuit, the essence of extending a composer’s creativity through musical systems has been overlooked.

To foster greater diversity, interest, and innovation in generative systems, I advocate for a shift in how researchers approach generative music. The expressive and compositional insights that characterized the work of early algorithmic composers were instrumental in shaping the field’s initial allure and are indispensable for driving further innovation.
Bibliography


[29] Lakshmi Illuri, *Understanding weighted k-nearest neighbors (k-nn) algorithm*, 2023, Medium.


[59] ______, *Composing the music of the future*, 2016, Medium.


[63] Nobuo Uematsu, Tifa’s theme, 1997, Final Fantasy VII.


[67] Colin Walder and Giuseppe Marano, Mixing, music, and mods: ’cyberpunk 2077’ audio technical pipelines, 2024, Game Developer’s Conference.


[70] Wesley Yin-Poole, Starfield has now had over 12 million players, phil spencer says, 2023, IGN.