Dynamic Procedural Music Generation from NPC Attributes

Megan E. M. Washburn
memwashburn@gmail.com
California Polytechnic State University
San Luis Obispo, California

Foaad Khosmood
foaad@calpoly.edu
California Polytechnic State University
San Luis Obispo, California

ABSTRACT

Procedural Music Generation in Games (PMGG) can enrich the playing experience by providing both entertainment and communication to the player. We present a system that generates unique procedural thematic music for non-player characters (NPC) based on developer-defined attributes and game state. The system responds in real-time to the dynamic relationship between the player and target "boss" NPC. We create a multiplayer 2D adventure game using and evaluate the music generation system by means of user study. Subjects confront four NPC bosses each with their own uniquely generated dynamic track. Results indicate the generated music is generally pleasing and harmonious, and players are able to detect a relationship between themselves and the NPCs as reflected by the music, even if they can not decipher the exact details.

CCS CONCEPTS

• **Applied computing → Sound and music computing**.

KEYWORDS

procedural content generation, procedural music, video games, music generation, non-player characters

ACM Reference Format:
Megan E. M. Washburn and Foaad Khosmood. 2020. Dynamic Procedural Music Generation from NPC Attributes. In *International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3402942.3409785

1 INTRODUCTION AND MOTIVATION

Music in video games is crucial to facilitating immersion in players[13][7]; it is as important for conveying in-game information as it is for player enjoyment. Given the breadth of possible music theory-based and algorithmic means to create procedurally generated video game soundtracks, developing suitable and engaging music is a deeply compelling task. PMGG is directly suited for scoring video games, given their stochastic nature due to player action. Our motivation stems from classic games, and many arcade NPC, particularly boss characters that signal an upcoming confrontation based on visual and auditory clues. A majority of character introductions are signaled by just a simple change in musical track, sometimes different tracks for unique boss characters. But with these musical signals, there is ample opportunity to communicate much more to the player, providing subtle utility and contributing to narrative foreshadowing in the game. We develop a novel PMGG system, integrated into a new multiplayer online adventure game, also developed for this project. Space limitations do not allow for full design discussion of the PMGG and none for the game itself, but we present an abridged version below and we direct the reader to [REDACTED UNPUBLISHED SOURCE] for full details. The PCG system uses the input NPC features to tune and control certain appropriate grammar rules. We evaluate with a 23 person user study verifying both quality, and representation in our system.

FDG '20, September 15–18, 2020, Bugibba, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/3402942.3409785

2 RELATED WORK

The first instance of procedural music in a game consistently traces back to Peter Langston's work for Lucasfilm Games *BallBlazer* games, first released for the Atari 8-bit family in 1984 [11], for which the iMUSE (Interactive Music Streaming Engine) system was developed [19]. This system was used in later games, including both generative and transformative musical soundtracks [10][5].

Procedural music in games widely stayed in the *adaptive* and *transformative* realms as algorithmic music composition was being explored by musicians and composers. By the early 2000s, improvements in vertical and horizontal stem mixing allowed for developers and composers to experiment again with *generative* methods. Notable feats of procedural music generation for video games that adapt to player actions and in-game events can be found in games: *Spore*, *No Man's Sky*, and *Rez Infinite*.

A pioneering scholar in algorithmic music composition is David Cope who successfully demonstrated a rule based musical composition system mimicking the style of classical greats [3][4].

Grammars are one of the earliest representations of music for use in computational generation: by applying semantic analysis concepts, grammars are capable of describing music theory in a structure directly suitable for procedural generation [12], paving the way for systems that codified specific music theory paradigms with enough depth to generate structurally correct music [21][4][20]. Though less adaptive and non-interactive, machine learning systems present the most recent paradigms, showing high-performing, promising music generation results [8].

Birchfield et al. in 2003 introduced a generative model for the creation of musical emotion, meaning, and form using genetic algorithms (GA). The team noted "moments in the generated music where the drive seems to stall," an issue still found often in many generative systems [2].

Multi-modal systems pose perhaps the highest potential for suitable, generative, adaptive music in video games. Prior systems such

as Jewell et al.'s work in 2005 and Scirea el al.'s work on MetaCompose in 2016 gear themselves towards adaptability in video games [9] [18]; their work supports multi-agent systems as the ideal paradigm for improving the current state of PMGG.

MetaCompose presents a promising system design to integrate to game development systems for adaptive, fully generative music in games [18]. It is a significant step in the direction of achieving immersive, affective music while maintaining compositional quality. Emphasis is placed on the distribution of musical tasks to three different subsystems (akin to this system: chord sequence, melody, and accompaniment) in their design to generate the composition, and on the musical constructs based in music theory best create the intended emotional state.

The theory on affective state present in MetaCompose and previous work by Marco Scirea references Jacek Grekow's work on defining emotion in music by mapping their descriptors to a space composed of the axes *valence* and *arousal*, the former corresponding to a scale of positivity and negativity, the latter to high to low energy [17][6]. Earlier works also used this scale to structure metrics with which to generate music [1][16].

3 SYSTEM DEVELOPMENT

We present a multi-agent expert system, which generates components of music by multiple means. Each algorithm is developed to handle a specific musical composition aspect, all of which are synthesized to a single MIDI file that contains all the movements, sections, rhythms, notes, and instrumentation necessary to playback a full composition in-browser, in real-time for the game.

This system draws inspiration from the interactive music generation concepts proposed in Scirea et al.'s work on MetaCompose [18], but includes various improvements and alterations that allow the music generation system to adapt to NPCs in a more responsive and expressive manner.

3.1 Composition Algorithm

As input, the system receives four values that are resolved from developer input: arousal, valence, time period, and seed (randomly generated by default). These values together resolve to various aspects of the final composition.

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

3.1.2 Chord Sequence Generation. With the resolved mode, we can apply diatonic music theory rules to a parameterized, random generation of chord progressions. Given the rule structure of chord progression theory, designing an algorithmic grammar was most appropriate for this section.

Each scale degree's chord in a given mode has an assignable quality. The classification mostly relies on whether or not the chord contains *fundamental* or *characteristic* tones: the former tends to sound mode-agnostic, while the latter tends to establish to the ear that we are definitely in the given mode.

For the purposes of this user study, **A B A B** was the only song form used. With these classifications, we can construct an overarching chordal structure based on the tendencies of the notes within the given mode's chords. Within chord progressions, passages can be further classified by *closed* = c and *open* = o. This is where grammars lend well; each of these consist of a specific ordered collection of the above abstract chord quality classifications. Using string replaceall, we can string together a variety of these progressions that are both varied and harmonically correct. Each **A B** section consisted of an *origin*, which was either: c c, c o, c o c, or c o o c.

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

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

$$\mathbf{A} \Rightarrow \text{c o c} \Rightarrow \text{[T S D T2] [D T S D] [T D2 D S T2] :‖}$$

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

3.1.4 Pitch Contour Generation. The randomly generated input binary string serves as the generative seed for the base pitch number string. This base pitch number string is then adjusted by passing through the following rules, each weighted against one another as distributions. The included subset of composition rules for melodic line are as follows: (1) Remove third+ repeating note; (2) Force a 2^{nd} interval jump if a FARJUMP has occurred; (3) Added dissonance (DISS.) for interest; (4) Biased range: keep notes within a certain range; (5) Close Jumps interspersed among Far Jumps.

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

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

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

3.1.6 Mapping of Attribute to Musical Feature. Figure 2 represents the most appropriate set of static and dynamic NPC attributes for the game used in this user study, and illustrates the flow of how and where these attributes affect the generated music.

4 USER STUDY AND RESULTS

We conduct a 23 person user study based on an adaptation of the game PhaserQuest (which was adapted from the browser-MMO BrowserQuest) injected with this PMGG system. 15 participants are

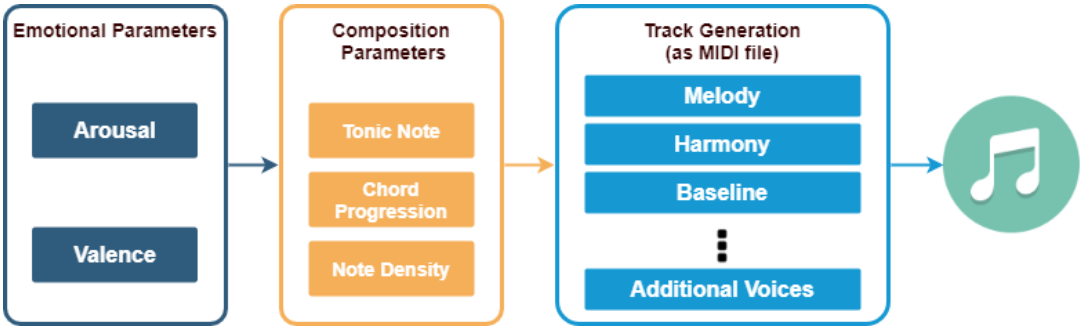

Figure 1: Simple music composition feature dependencies

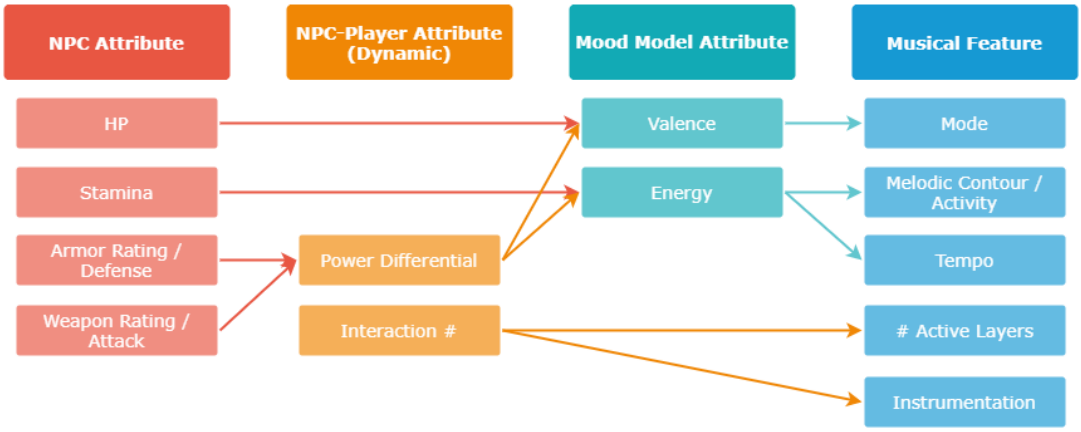

Figure 2: Illustrates dynamic relationships of game features to musical features within the generation system

Table 1: Music quality questions (n=23)

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

Table 2: User study free form responses: percent identifying a representation and top answers (n=23)

aspect	% id'ed	top answers	true representation
tempo	91.3	"environment", "risk", "game play"	boss speed
melody	83.0	"environment", "risk", "NPC"	boss difficulty
harmony/accom.	69.6	"environment", "not sure"	boss strength / hp
musical mode	47.8	"progression"	power diff.
instrumentation	78.3	"environment", "not sure"	number of encounters

male, and 7 are female. All are students recruited from a university computing program. Two objectives of the user study are: first determine if the generated music is good, and second to measure to what extend can users decipher the communication that occurs through the procedural music.

For the first objective, we use the criteria from Scirea et al.'s evaluation work for their own generative music system, *MetaCompose*

[15]. The average scores were higher than expected for musical quality as seen in Table 1. The *overall score* for musical quality incorporating all attributes is **7.16 / 10** with a *standard deviation* = $\sqrt{\sigma^2}$ of **4.40**. Most responses are very positive.

Users also feel certain attributes in-game are reflected in the generated compositions. This assessments is done through a set of free form responses to the questions of the form "did [aspect] represent anything?", where [aspect] is the specific musical variable in our PMGG algorithm. We also ask what the representation itself is. The results in Table 2 indicate users clearly identifying some correlation but did not have enough exposure or recollection to isolate the actual representation.

Power differential was the primary factor that affected the **valence** parameter of a new instantiation of a generated music composition. Recall that the **valence** measure of a piece of music generally accounts for the perceived happiness or sadness of that piece. Conversely, the **energy** level accounts for the perceived arousal or sleepiness of a piece of music. Together, this dyad creates a two-dimensional emotional space that can account for a broad range of emotions [14], all of which can be broadly associated with musical mode, and dialed in with musical details such as tempo, rhythmic choices, instrumentation choices, dynamics, and more (such as was done in this experiment). A majority of my participants were able to identify this attribute as one of the main factors that affected the music generation.

The question that addressed the core of relating a composition to a given NPC's attribute set asked participants if the music helped to indicate Boss HP, *Boss Stamina, Boss Weapon Rating / Attack, Boss Armor Rating / Defense, Power Differential, or Interaction Number*. The characteristics aside from power differential I listed in this multiple choice question were also in-game attributes that affected the generated composition, but to lesser degrees than the power differential did. For example, a higher *Boss HP* raised the value of **valence**, but to a lesser degree than the *Power Differential*. *Boss Stamina*, or boss speed in this case, was the principal attribute that affected the **energy** value in the generative algorithm; higher speed = higher energy. *Boss Weapon Rating / Attack* and *Boss Armor Rating / Defense* also minimally weighed in on the valence value due to the fact both attributes were being compared to the player's own attack rating (as that is how *power differential* is calculated), but were primarily responsible for **instrumentation** choice. Changes in the music due to *Interaction Number* may have been more difficult to pick up on, due to the fact that this would only be predominant if the player had visited the boss with low attack or low defense (bad weapons and bad armor) first, then returned to visit the boss with better gear (higher attack or higher defense). Recall from figure 2, this attribute accounted for instrumentation choices and number of musical layers present. This concludes that the system did in fact generate relevant compositions given specific NPC attributes to a certain extent.

REFERENCES

[1] Timothey Adam, Michael Haungs, and Foaad Khosmood. 2014. Procedurally generated, adaptive music for rapid game development. In *FDG 2014 Global Game Jam Workshop, Foundation of Digital Games.*

[2] David Birchfield. 2003. Generative Model for the Creation of Musical Emotion, Meaning, and Form. In *Proceedings of the 2003 ACM SIGMM Workshop on Experiential Telepresence* (Berkeley, California) *(ETP '03)*. ACM, New York, NY, USA, 99–104. https://doi.org/10.1145/982484.982504

[3] David Cope. 1991. Recombinant Music: Using the Computer to Explore Musical Style. *Computer* 24, 7 (July 1991), 22–28. https://doi.org/10.1109/2.84830

[4] David Cope. 2013. The Well-Programmed Clavier: Style in Computer Music Composition. *XRDS* 19, 4 (June 2013), 16–20. https://doi.org/10.1145/2460436.2460444

[5] Chris Crawford. 1984. *The Art of Computer Game Design.* McGraw-Hill, Inc., New York, NY, USA.

[6] Jacek Grekow. 2016. Music Emotion Maps in Arousal-Valence Space. In *Computer Information Systems and Industrial Management*, Khalid Saeed and Władysław Homenda (Eds.). Springer International Publishing, Cham, 697–706.

[7] Mark Grimshaw, Craig Lindley, and Lennart Nacke. 2008. *Sound and Immersion in the First-Person Shooter: Mixed Measurement of the Player's Sonic Experience.* www.audiomostly.com. https://doi.org/images/stories/proceeding08/proceedings_am08_low.pdf

[8] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. 2017. Counterpoint by Convolution. In *Proceedings of ISMIR 2017.* https://ismir2017.smcnus.org/wp-content/uploads/2017/10/187_Paper.pdf

[9] Michael O. Jewell, Lee Middleton, Mark S. Nixon, Adam Prügel-Bennett, and Sylvia C. Wong. 2005. A Distributed Approach to Musical Composition. In *Knowledge-Based Intelligent Information and Engineering Systems*, Rajiv Khosla, Robert J. Howlett, and Lakhmi C. Jain (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 642–648.

[10] Michael Land and Peter McConnell. 2000. Sound Engine: iMuse. https://www.mobygames.com/game-group/sound-engine-imuse

[11] Peter S. Langston. 1986. 201 644-2332 or Eedie & Eddie on the Wire: An Experiment in Music Generation. *Peter Langston* (1986), 13–27.

[12] Author(s) C. Roads, Paul Wieneke, and C. Roads. 1979. Grammars as representations for music. *Computer Music Journal* (1979), 48–55.

[13] Timothy Sanders and Paul Cairns. 2010. Time Perception, Immersion and Music in Videogames. In *Proceedings of the 24th BCS Interaction Specialist Group Conf.* (Dundee, United Kingdom) *(BCS '10)*. British Computer Society, Swinton, UK, UK, 160–167. http://dl.acm.org/citation.cfm?id=2146303.2146327

[14] Emery Schubert. 1999. Measuring Emotion Continuously: Validity and Reliability of the Two-Dimensional Emotion-Space. *Australian Journal of Psychology* 51, 3 (1999), 154–165. https://doi.org/10.1080/00049539908255353 arXiv:https://aps.onlinelibrary.wiley.com/doi/pdf/10.1080/00049539908255353

[15] Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. 2017. Can You Feel It?: Evaluation of Affective Expression in Music Generated by MetaCompose. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Berlin, Germany) *(GECCO '17)*. ACM, New York, NY, USA, 211–218. https://doi.org/10.1145/3071178.3071314

[16] Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. 2018. Evolving In-game Mood-expressive Music with MetaCompose. In *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion* (Wrexham, United Kingdom) *(AM'18)*. ACM, New York, NY, USA, Article 8, 8 pages. https://doi.org/10.1145/3243274.3243292

[17] Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. 2018. Towards an Experiment on Perception of Affective Music Generation Using MetaCompose. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Kyoto, Japan) *(GECCO '18)*. ACM, New York, NY, USA, 131–132. https://doi.org/10.1145/3205651.3205745

[18] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. 2016. MetaCompose: A Compositional Evolutionary Music Composer. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, Colin Johnson, Vic Ciesielski, João Correia, and Penousal Machado (Eds.). Springer International Publishing, Cham, 202–217.

[19] Willem Strank and Peter Moormann. 2013. *The Legacy of iMuse: Interactive Video Game Music in the 1990s.* Springer Fachmedien Wiesbaden, Wiesbaden, 81–91. https://doi.org/10.1007/978-3-531-18913-0_4

[20] Geraint Wiggins. 2007. Computer Models of Musical Creativity: A Review of Computer Models of Musical Creativity by David Cope. *Literary and Linguistic Computing* 23 (12 2007). https://doi.org/10.1093/llc/fqm025

[21] Rene Wooller, Andrew R Brown, Eduardo Miranda, Joachim Diederich, and Rodney Berry. 2005. A framework for comparison of process in algorithmic music systems. *Creative Industries Faculty* (2005). https://eprints.qut.edu.au/6544/