PreGLAM-MMM

Application and evaluation of affective adaptive generative music in video games

Cale Plut
cplut@sfu.ca
Simon Fraser University
Surrey, Canada

Philippe Pasquier
Simon Fraser University
Surrey, Canada
pasquier@sfu.ca

Jeff Ens
jeffe@sfu.ca
Simon Fraser University
Surrey, Canada

Renaud Bougueng
rbouguen@sfu.ca
Simon Fraser University
Surrey, Canada

ABSTRACT

We present and evaluate an application of affective adaptive generative music in a single-player, action-RPG video game. We create a score that serves as an audience to the gameplay, based on the output of PreGLAM, which models the emotional perception of a game audience. We use the Multi-track Music Machine to expand and extend a composed adaptive musical score, and we use industry-standard production techniques to synthesize and perform all of our musical scores. We evaluate our application of generative music in comparison to two composed scores, one adaptive and one linear. Our generative score is rated as nearly equivalent to a composed linear score in perceptions of emotional congruency, immersion, and preference.

CCS CONCEPTS

• **Software and its engineering** → **Interactive games**; • **Information systems** → **Multimedia content creation**; • **Human-centered computing** → **Auditory feedback**; *Empirical studies in HCI*; *Empirical studies in interaction design*; *Systems and tools for interaction design*; • **Computing methodologies** → *Artificial intelligence.*

ACM Reference Format:
Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. 2022. PreGLAM-MMM: Application and evaluation of affective adaptive generative music in video games. In *FDG '22: Proceedings of the 17th International Conference on the Foundations of Digital Games (FDG '22), September 5–8, 2022, Athens, Greece.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3555858.3555947

1 INTRODUCTION

Music is present in some form in almost all video games. Most music in games is composed by one or more humans, and is either performed by human musicians and/or synthesized into audio format. While music is generally linear, and plays without reacting to external input, video games are interactive, and respond to the inputs of one or more players. To create music that matches gameplay, video game composers may use "adaptive music", sometimes called "interactive music", which is music that can be altered based on a control input. Adaptive music is a powerful tool for creating music that matches gameplay, but using adaptive music requires specific techniques that can significantly increase a composers workload. Adaptive music is primarily used when music is serving as an "audience" to the gameplay, commenting on the successes and failures of the player [16].

Generative music is created with some degree of systemic autonomy from its input. Because video games almost universally have some degree of systemic autonomy from their input, it may be argued that all game music is generative. However, we follow Plut and Pasquier's definition of generative music in games as having systemic autonomy from the game logic [21]. For example, if a single piece is cued when the game state changes in an identical fashion each time, we do not consider this generative. Depending on the algorithm, generative music systems are capable of producing large amounts of musical content in minutes, seconds, or even in real-time.

There are two main approaches to applying generative music in video games [21], which we will discuss further in Section 2.2. Academic research generally focuses on the use of novel algorithms for online real-time generation of symbolic music to entirely replace a composed score [10, 17, 28, 31], while approaches from the games industry primarily use stochastic methods to target real-time sequencing of audio stems.

Academic systems most commonly generate and synthesize music in real time with General MIDI sounds. These systems mostly use some form of player experience model, commonly affect-oriented, to control the adaptivity of the generative music. These systems produce novel music that can theoretically match the events of a game, but lack timbral and performative features when compared to contemporary video games.

Systems from the games industry generally use pre-rendered or recorded audio stems, sequenced together with stochastic methods. These systems generally extend adaptive musical methods of *horizontal resequencing* and *vertical remixing* [29]. This approach produces music that has equal performative fidelity to linear music,

FDG '22, September 5–8, 2022, Athens, Greece

ACM ISBN 978-1-4503-9795-7/22/09…$15.00
https://doi.org/10.1145/3555858.3555947

but can often reduce the expressive range of the music, as the music must be composed so that the combined arrangements won't clash with each other [16].

We present a hybrid approach to utilizing generative music in video games, discussed further in Section 3.2. We use Ens and Pasquier's Multi-track Music Machine (MMM) [5] transformer model to generate multi-track symbolic variations of a composed adaptive score, as we discuss in Section 3.3. The composition of our adaptive score is informed by previous research in composing music to express desired affect in a Valence-Arousal-Tension model of emotion [23]. We use the musically-focused audio middleware program *Elias* [4] to control our adaptive scores based on the output of PreGLAM. We also compose a linear score that is based on the adaptive score.

To model the gameplay emotion and inform the musical adaptivity, we use Plut et al.'s *Predictive Gameplay-based Layered Affect Model* (PreGLAM) [22] as discussed in Section 2.3. PreGLAM is an artificial cognitive agent with privileged game information, that models the real-time perceived affect of a biased game spectator. We implement PreGLAM as biased towards the player winning the game, though other biases may be provided.

We use VST instruments to render our scores into audio, to increase the quality of synthesis compared to previous uses of General MIDI. At the time of this writing, real-time synthesis of symbolic music during gameplay is unable to match the quality and fidelity of offline synthesis, such as by VST instruments. We therefore render our symbolic tracks via Ableton Live.

Our approach focuses on providing an application of generative music that builds on previous literature in the area, while increasing the synthesis, production, and performance fidelity of musical scores from previous applications. Our approach also increases the expressive range of the music compared to previous attempts, by utilizing a 3-dimensional VAT model of emotion. We additionally evaluate our generative score in comparison to an adaptive score and a linear score that share identical production methods. Effectively, we target an increase in external validity compared to previous applications, without sacrificing experimental control.

We empirically evaluate our application of generative music in a study with 48 participants, and find that our application of generative music performs consistently with linear music, and outperforms composed adaptive music in participant perception of emotional congruency, player immersion, and preference. Our approach is directly compared to music that is produced using industry-standard techniques.

2 BACKGROUND

2.1 Adaptive music in games

Music can serve multiple functions in games, and occasionally serves multiple functions simultaneously. Winifred Phillips describes one function of music in games as acting as an "audience", which is described as creating a feeling that the music is "essentially watching the gameplay and commenting periodically on the successes or failures of the player" [16].

When composing music to act as an audience, there is an inherent mismatch in the relationship that games and music have with time. Games are interactive, and react to the actions and events of one or more player or non-player agents. Music is most often linear, and generally does not react to external changes. Adaptive music allows for music to be altered based on some control input, such as player health, number of enemies, or game progress [29].

Adaptive music can be a powerful tool for using music as an audience, and Phillips describes adaptive music as "constituting the most complex realization of the music-as-audience approach" [16]. Perhaps the strongest drawback of adaptive music is that it requires a large amount of time investment, and requires early integration into the game design to be effective. Compounding these issues is that music and sound often have fewer resources, lower budgets, and may be added later in the development process than other game features [16, 29].

There are two main techniques for creating adaptive music: Horizontal resequencing, and vertical remixing [29]. Music is often read left-to-right through time, and horizontal resequencing refers to the adaptive alteration of music through time. In horizontal resequencing, the music generally adapt to game state — musical cues will loop until certain conditions are met or the game state changes, a transition is played, and a new musical cue begins looping, matching the new game state.

Instruments in sheet music are vertically aligned, and vertical remixing refers to the adaptive addition or subtraction of audio stems, depending on the input. When using vertical remixing, the music generally responds to some variable such as "intensity", and adds or subtracts tracks based on a provided mapping. *Mass Effect 2* presents a common use of vertical remixing: the music in *Mass Effect 2* adapts based on a measure of combat intensity, adding additional layers as combat becomes more intense [2].

2.2 Generative music in games

Generative music, also known as procedurally generated music or algorithmic music, is music that is partially or wholly created by some form of systemic autonomy [15]. Depending on the specifics of a particular system, generative music algorithms are capable of generating music quickly, potentially in real-time, based on a set of input parameters. Because generative music can produce music quickly and produce large amounts of music based on the provided input, it may be used to address the drawbacks to using adaptive music.

Plut and Pasquier survey uses of generative music in video games in both the games industry and academic research, and identify several trends [21]. Primarily, generative music in the games industry is used to extend composed scores in the audio domain, mostly through stochastically re-arranging musical cues and stems based on input from the game. In contrast, generative music in academic research mostly targets the replacement of a composed score with adaptive music generated and synthesized in real time.

2.2.1 Academic applications. Academic approaches primarily focus on applying novel generative algorithms to create a general system capable of real-time, adaptive symbolic music generation. These systems commonly use generative music instead of composed music, with the musical adaptivity most often based on an affective model of player experience. These affective models generally map a set of game variables to one or more affective dimensions. Academic systems are often empirically evaluated, and the evaluation

is often focused on whether the generated music is perceived as expressing similar affect to the game.

Plans and Morelli create a system that generates music for the MarioAI Championship engine, a game used in procedural level generation research [17]. Plans and Morelli describe an "excitement" metric based on aggregate counts of game events and variables, and map several musical features to the excitement metric. A harmonic sequence are generated by a genetic algorithm design, using notes from the C major scale or a subset of notes from the C major scale. Additionally, a melody is created, first by creating set of phrases are generated by applying minor transformations to a smaller set of composed phrases, and then combining a sequence of these phrases into a melody. The music is synthesized by the "SawLPFInstRT2" instrument, from the Jmusic library [3].

Plans and Morelli evaluate their system by comparing results from the generative system to a precomposed linear MIDI track. Plans and Morelli collect the output of their affect model from playthroughs, and ask player-participants to rate a level of enjoyment after playing. While the gameplay-derived frustration value was on average lower when utilizing generative music, other measures, including self-reported enjoyment, are consistent between the two conditions.

Prechtl presents a system that uses weighted Markov models to generate real-time chord progressions, which can be played by a single loaded VST instrument. The chords are played both as a block chord and an arpeggio, and the chord contents are selected base on an input "tension" value. Prechtl also presents a horror-genre game *Escape Point*, created to implement and evaluate the generative system. Prechtl maps a tension value to the distance between the player and the nearest mobile object (mob) while navigating a maze. Mobs follow a pre-determined path, and if the player comes into contact with a mob, they lose the game.

Prechtl evaluates his system, and finds that the adaptive generative score invoked more tension and excitement based on skin conductance. After playing *Escape Point*, participants report perceiving more tension and excitement with the adaptive generative score than with linear generative music or no music [24]. Participants who like the horror genre prefer the generative score to the linear score or no music, and find the game more fun to play with the generative score. However, all three conditions are evaluated as roughly equal in preference and fun ratings among participants who do not like the horror genre.

Scirea presents *Metacompose*, which uses hybrid evolutionary techniques to generate music [27]. Metacompose generates a chord progression, and evolves a melody based on that chord progression. Metacompose then realizes the chord progression into an accompaniment in the form of a block or arpeggiated chord. The music generation responds to input values for the dimensions of valence and arousal.

Scirea implements and evaluates Metacompose in the game of checkers, synthesized via a solo piano. A valence value is determined by evaluating "how good the current board configuration is for the human-player", and an arousal value is determined by evaluating the range of evaluations for possible moves, described as reflecting the sentiment "How much is at stake for the next move?". Metacompose outperformed random music and non-adaptive music in an empirical user preference study.

Williams et al. present an "affectively-driven algorithmic composition" system that primarily uses Markov generation with post-hoc transformations for affective expression [31]. While this system is capable of real-time generation, generated sequences were rendered into audio files, played on a solo piano, for the evaluation.

To evaluate their generative system, Williams et al. select a specific in-game section of the MMO *World of Warcraft*. Situations that occur within the section are manually tagged with affective targets, and the music system selects generated clips to match the affective target. Williams et al.'s system outperforms both the composed score and silence in user ratings of "emotional congruence", in gameplay, and outperforms silence in user ratings of immersion. However, the generated score shows a "marked decrease" in user ratings of immersion compared to the composed score.

2.2.2 Industry applications. Industry applications of generative music most commonly sequence composed and pre-rendered or recorded audio stems together in new ways. Mick Gordon describes an example of using generative music to extend horizontal resequencing in *DOOM (2016)* [25]. Gordon assigns fully arranged clips into "buckets", that generally follow a structure such as "verse", "chorus", and "bridge". During gameplay, while certain conditions are met, the system continuously randomly selects clips from within a bucket. When conditions change, the system plays a transition as in typical horizontal resequencing, and then being playing randomly selected clips from the new bucket.

Red Dead Redemption makes aggressive use of generative music addressing the arrangement task [26] — in *RDR*, all music is written at 130 beats per minute, in the key of a minor. The music in *RDR* is divided into orchestral function e.g. "melody" or "bass", and associated game states e.g. "riding horse" or "combat". When the game state changes, the generative system in *RDR* selects a set of instruments/functions, and randomly selects a loop for each selected instrument. Additionally, the system adds or removes layers based on game variables within some situations, using elements of both horizontal resequencing and vertical remixing.

2.3 PreGLAM

As mentioned in Section 2.2.1, the most common application of generative music in games uses an affect-based model of player experience to influence the adaptivity of the generative score. The adaptivity of our score is influenced by the Predictive Gameplay-based Layered Affect Model, or *PreGLAM* [22]. *PreGLAM* is a cognitive agent that models a spectator with a provided bias. In our implementation, we use PreGLAM to model an audience who is biased in favour of the player to create music that affectively comments on the successes and failures of the player.

PreGLAM is a layered, gameplay-based affect model, based on NPC models of affect [1, 6]. A base mood value is provided to PreGLAM that represents a general, environmental affective feeling. PreGLAM models emotions as the responses to emotionally evocative game events (EEGEs). EEGEs have a provided base emotion value, a set of intensity modifier variables, and a time scalar. PreGLAM calculates an output affect value for each dimension based on the provided mood value, as well as the summed emotional responses to EEGEs, modified by their intensity modifier variables and time scalar.

Figure 1: A possible flank in XCOM

PreGLAM models EEGEs that occur in the game, and also models emotional responses to prospective EEGEs. Prospective events are events that PreGLAM expects to happen, given the current state of the game. One example of how we model prospective events can be seen in Figure 1, which shows a scenario from *XCOM: Enemy Within*. In Figure 1, the player's selected unit is able to flank an opposing unit. The opposing unit is otherwise in cover, which gives it a tactical advantage that can be removed when flanked. Because the gameplay in *XCOM* primarily involves manipulating tactical positioning in relation to cover, we can predict that the player will move their unit into a flanking position and attack the opposing unit. Importantly, PreGLAM models that a spectator emotionally perceives the possibility of a flank even if the player does not take the predicted action — the prospect of the flank is not affected by whether or not it is realized.

PreGLAM is integrated into the game *Galactic Defense* (GalDef), which will be further described in Section 2.5. PreGLAM's application in GalDef is based on informal experiential playtesting, with all EEGEs, mood values, and conditions for predicting EEGEs created during the design process. We assign threshold values for 5 levels of each dimension, which influences the adaptivity of our musical score.

2.4 IsoVAT Composition guide

Plut et al.'s IsoVAT composition guide presents a set of Western musical features, and the perceived changes in emotional expression that changes in these musical features are associated with [23]. This guide is aggregated from a broad overview of research in music and emotion. The IsoVAT guide represents emotion using a Valence-Arousal-Tension model, and is intended to be used across Western pop, jazz, and classical genres. For example, increases in the melodic range, contour, and direction are strongly associated with increases in arousal, while decreases in harmonic consonance are strongly associated with increases in tension.

The IsoVAT composition guide is empirically evaluated by producing a corpus of clips that express varying levels of valence, arousal, or tension. These clips are organized by the affective dimension that they manipulate, and further divided into sets of 3. Each set shares an instrumentation and genre, and is composed to express a low, medium, and high level of the assigned emotional dimension.

2.5 Galactic Defense

Plut et al. integrate PreGLAM into a custom video game, *Galactic Defense* (*GalDef*) [1], designed for use in game emotion and perception research [22]. This follows a common approach in research and emotion [9, 10, 13, 24], and is further described in the PreGLAM paper [22]. We implement and evaluate our use of generative music in games using GalDef. Figure 2 provides an annotated screenshot of gameplay. We integrate our adaptive scores into *GalDef*, using PreGLAM's output to control the adaptivity. *GalDef* also serves as an environment for evaluating our application of generative music.

GalDef is an action-RPG game, where the player uses a set of abilities to defeat a series of opposing units in real-time. The abilities that the player has access to have situational strengths and weaknesses, with the intent of encouraging moments of gameplay where the player is appraising the current game state, and using that appraisal to make choices about their next move. The player must manage a small, recharging limited resource pool for both themselves and the opponent, and must take care not to use certain abilities while under threat of attack.

The player controls a spaceship in *Galactic Defense*, and must defeat several opposing AI-controlled spaceships to win the game. The player has four moves, which are shown in Figure 2. In terms of resources, the player has a weak shield that constantly recharges, and a pool of health points. When the player uses any ability, the shield is temporarily deactivated, and therefore any incoming attack will directly drain health points. All opponents have the moves of *attack pattern*, *heavy laser*, and *repair*.

Both the heavy laser and repair abilities are interruptible when used by the player. If the player receives any damage while using these abilities, the damage will be multiplied and the ability will be cancelled. Most of the gameplay in *GalDef* is in tactical decisions of when to use each of the four moves. The basic attack pattern does small but consistent damage, the heavy laser deals large damage in some situations, but is vulnerable to counterplay. The "absorbitive reactor" parry ability is extremely powerful, but requires precise timing and is purely situational. Self-repair is often necessary, but as with the heavy laser, the player is vulnerable while using it. This design provides fluid gameplay and highlights the contextual nature of game emotions.

2.6 PreGLAM implementation

Figure 3 shows how PreGLAM appraises game data to select music, based on a perceived valence, arousal, and tension, acting as an audience.

Mood values are provided to PreGLAM based on the designed difficulty levels of each gameplay segment. Each gameplay segment involves 2-3 combat encounters, which rise in difficulty as the game progresses. We model PreGLAM with a desire of the player winning, and derive a set of EEGEs, shown in Table 1. In Table 1, we abbreviate "Player" to "P", and "Opponent" to "O". These EEGEs are created through an iterative process of playtesting with a focus on informal evaluation of experienced emotions.

Table 1 gives the base assigned value for the associated emotional perception of each event. These values are based on an initial unit of 1, and values represent the intensity of the emotional response

[1] GalDef is available at GitHub (https://tinyurl.com/75mfvw92)

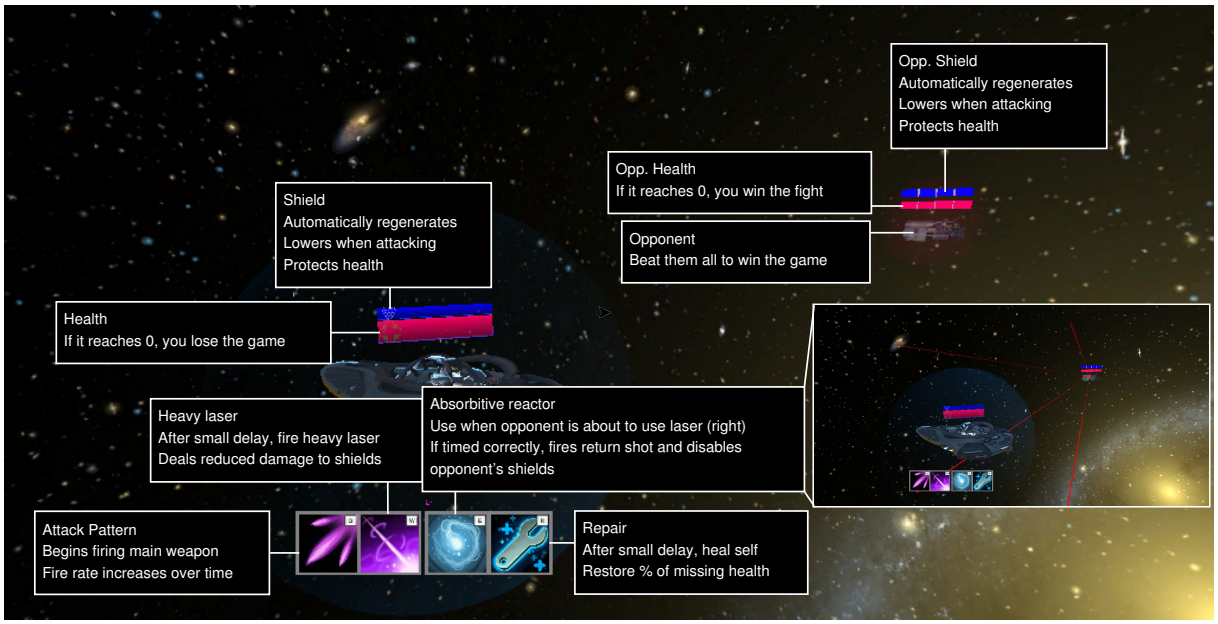


Figure 2: Visual tutorial for Galactic Defense

to the EEGE. We represent all intensity modifiers as percentages, which scale the emotional values between 100 and 200%. Tension values are only computed for prospective events, as tension arises from the prospect of events [14]. As an example, the "Player shield down" EEGE has a base value of -2 valence, 1 arousal, and 2 tension. These values are modified based on how much health the player has remaining — losing the shield is more of a problem if the player's health is also low. If the player has, e.g. 50% of their maximum health and is expected to lose their shield, output values will scale to 150% of their base value, and the output values at the moment that the shields are expected to go down are 3 valence, 1.5 arousal, and 3 tension. We note that during actual gameplay, these values are additionally scaled through time. As mentioned, PreGLAM is further described in a separate paper [22].

Table 1: Emotionally evocative events in *GalDef*

Event	Valence	Arousal	Tension	Modifiers
P. complete atk combo	1	1	1	Missing O. shield
P. heavy atk	1	1	1	Missing O. health
O. atk combo	1	1	1	Missing P. shield
O. heavy atk	-2	1	2	Missing P. health, Parry active
P. shields down	-2	1	2	Missing P. health
O. shields down	2	1	2	Missing O. health
P. exploit O. disable	3	1	2	Missing O. health
P. death	-3	1	3	P. shield recharge time
O. death	3	1	3	O. shield recharge time
P. heal	2	1	2	Missing P. health
O. heal	-2	1	2	Missing O. health

3 MUSICAL SCORES

3.1 Linear score

We compose a linear score that attempts to create moments of "serendipitous sync" [29], where a linear score that is written with changes in emotion over time occasionally synchronize with the changing emotions of gameplay. This score is musically based on the adaptive score, and mostly consists of manually re-arranged tracks and sections of tracks from the adaptive score. We arrange the musical ideas from the adaptive score into a linear score that has varying rises and falls in valence, arousal, and tension. The approximate levels of each dimension through the linear score's 128 bars is shown in Figure 4. As we expect the gameplay of GalDef to also demonstrate moments of rising and falling valence, arousal and tension, we expect that there may be moments where the linear music aligns with the GalDef's perceived emotion. The linear score is available to listen on SoundCloud [18].

3.2 Adaptive score

We compose our affectively adaptive score following the IsoVAT composition guide [23], as described in Section 2.4. The IsoVAT guide provides an ordinal description of how musical features affect emotional perception, and we use the guide to create clips that express three levels of each dimension: low, medium, and high. While the IsoVAT corpus adjusts music along individual dimensions, we use the guide to compose a 3-dimensional adaptive score, that can express any combination of 3 levels of 3 affective dimensions. Therefore, we compose 3^3, or 27 clips.

Each clip is at a tempo of 130 beats per minute. Each clip has 5 tracks, and is composed for the same instrumentation, divided into "melody" and "rhythm/harmony" sections, as shown in Table 2. Table 2 also provides the VST instrument used for each instrument. We note that the guitar part alternates between using a distorted electric effect and using an acoustic guitar, and the two guitar parts share a track. We also note that while piano is often considered a rhythm section instrument, we use piano as a melody instrument

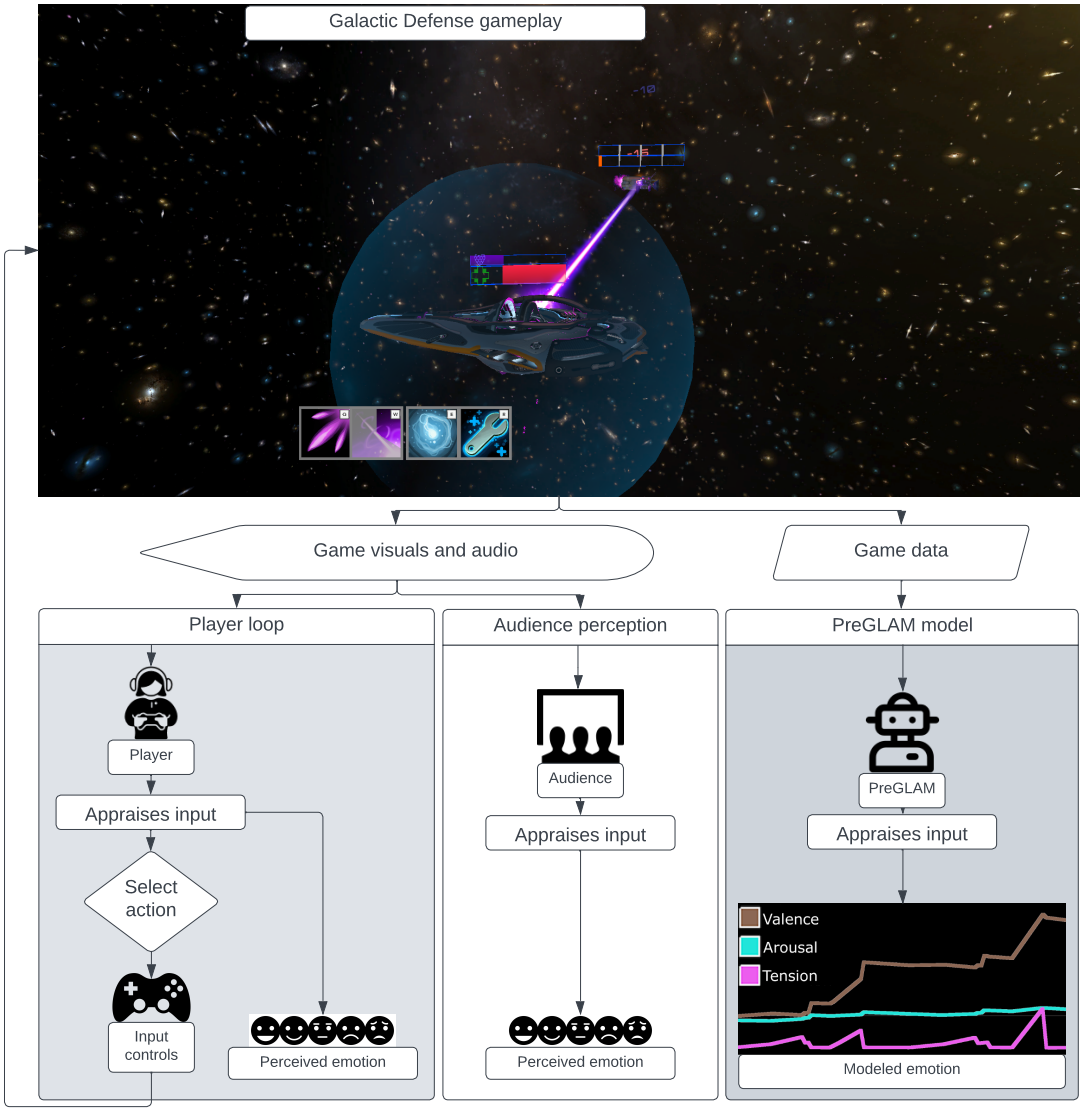


Figure 3: Diagram showing how PreGLAM-MMM fits into game loop

in our adaptive score. The use of VST instruments will be further discussed in Section 3.4.

We expand our 3 levels of adaptivity into 5, adding medium-low and medium-high levels via adaptive re-sequencing. These levels are differentiated by instrument section — only the melody section adapts to medium-low and medium-high levels. The rhythm section, in contrast, only adapts to levels of low, medium, and high. Section levels are independently set, so when transitioning from a high or low level to a medium-high/low level, the rhythm section continues to play the high/low clips until the corresponding dimension reaches a medium level. This further expands our adaptivity from 5 to 7 possible output levels for each dimension: low, low→medium, medium→low, medium, medium→high, high→medium, and high, creating a total of $7^3 = 343$ unique arrangements. Due to the interactive nature of our adaptive and generative scores, we implement a "Music explorer" in *GalDef*, where users can freely navigate the emotion space of the score outside of the gameplay.

In terms of harmonies, the keys/modes used in the clips are:

(1) b minor/aeolian, primarily used for low valence
(2) D Major/Ionian, primarily used for high valence
(3) G Lydian, primarily used for high valence with high tension

These keys and modes share a key signature, and therefore the adaptive score can theoretically navigate the harmonic space without jarring transitions.

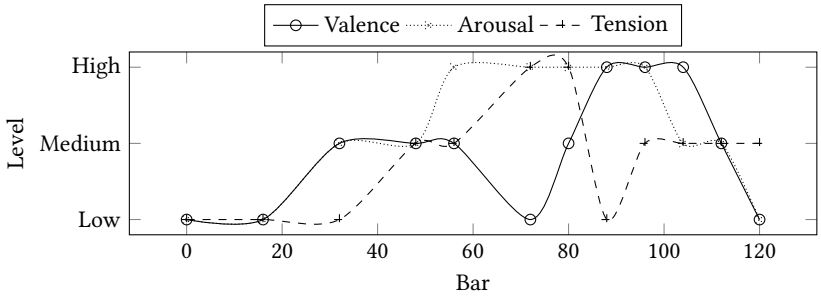


Figure 4: Affective levels by bar in the linear score

Table 2: Instrumentation of *Galactic Defense* score

Instrument	Section	VST bank	VST instrument	VST source
Bass	Rhythm	Analog Essentials	80ties Dance	Applied Acoustic Systems
Drums	Rhythm	LABS	Drums	Spitfire Audio
Strings	Rhythm	BBC Symphony Orchestra	Violas	Spitfire Audio
E. Piano	Melody	Lounge Lizard Session	Bite	Applied Acoustic Systems
Guitar (Electric)*	Melody	Strum Session	Ballistic Squeeze	Applied Acoustic Systems
Guitar (Acoustic)*	Melody	Strum Session	Dreadnought Smooth	Applied Acoustic Systems

Table 3: MMM Generation parameters

Parameter	Tracks per step	Bars per step	Shuffle	Percentage	Temperature	Model size
Value	1	4	True	90%	1.0	8-bar

3.3 Generative score

We utilize Ens and Pasquier's *MMM* 8-bar transformer model, which generates symbolic multi-track music [5], using the parameter settings in Table 3. While MMM has a host of features, we primarily use *Bar inpainting*. Bar inpainting involves resampling a subset of the bars present in one or more tracks, or altering a subset of musical material conditioned on the remaining unaltered musical material.

As mentioned in Section 3.2, our score is composed as a set of 8-bar clips, and the instrumentation is separated into sections. Because the melody and rhythm sections adapt as groups, we condition the generation of new melody bars on existing rhythm bars, and the generation of new rhythm bars on existing melody bars. We create 3 additional variations per section, for each of the 27 clips in our adaptive score. When the music adapts, we randomly select from the 4 possible variations (1 composed and 3 generated) independently for each instrument. This creates a total of $343^4 = 13,841,287,201$ unique arrangements.

The MMM model is currently too heavy and slow to generate music in real-time. However, we believe that the amount of generative musical content is indistinguishable from real-time generated music during gameplay in terms of musical variety. By utilizing offline generation, we are able to increase the audio quality over previous real-time uses of symbolic generative music. As technology improves, we believe that our approach could implement real-time generation.

3.4 Synthesis and Arrangement

Video game composers commonly use libraries of virtual instruments to provide some or all of the synthesis of their music [16]. These virtual instruments are generally controlled via MIDI, and input can be recorded on MIDI controllers and/or manually adjusted. As our score is in MIDI format, we use VST instruments to synthesize both our composed and generative score.

For our composed scores, we record data from a MIDI keyboard directly into *Ableton Live*, a common digital audio workstation (DAW). We primarily use VST sources from Spitfire audio's LABS libraries [8] and libraries from Applied Acoustic Systems [30]. We record the performance at 1/2 speed, played on a MIDI keyboard - this ensures precision in following the composed score while allowing for human articulation and velocity data.

Each instrument part has 27 unique levels, encompassing the VAT space that the adaptive composition expresses in total. As mentioned in Section 2.3, we label thresholds for PreGLAM's output to trigger a corresponding categorical level of each emotional dimension. We use "smart transitions" in *Elias*, which attempts to transition individual parts only during silence based on an analysis of the audio file. This creates transitions that somewhat more resemble transitions using symbolic notation instead of rendered audios, as there is some musical consideration for the timing of the transitions.

4 EMPIRICAL EVALUATION

4.1 Empirical Methodology

To evaluate our application of generative music in video games, we collect real-time user annotations from 48 video spectators. Our annotation software is available on GitHub [19], and is similar to *RankTrace* [11] and *PAGAN* [12]. Our annotation interface is shown

Figure 5: Screenshot of participant annotation interface

Table 4: Empirical study conditions

Condition	Music Source	Adaptivity	Relevant Section
No music	None	N/A	N/A
Linear score	Composed	Linear	3.1
Adaptive score	Composed	Adaptive	3.2
Generative score	Generative	Adaptive	3.3

in Figure 5. While watching a video of gameplay, user can press the up or down arrows to indicate an emotional change. As with RankTrace and PAGAN, unbounded input is collected every 250 ms, and the user is provided a visual graph of their annotation so far.

We create 20 videos of *Galactic Defense* gameplay. Each video is $\approx 3-4$ minutes in length, and we select clips that have clear changes to their emotional expression, particularly within a single affective dimension, based both on PreGLAMs output during the video and our informal evaluation. Each video has an accompanying output file generated by PreGLAM. We divide these videos into 4 sets of 10, based on the source and adaptivity of the musical accompaniment, as shown in Table 4.

Prior to annotating video, participants familiarize themselves with the gameplay of *GalDef*. Figure 2 shows an image tutorial, and a video tutorial is available for them to watch [20]. Participants are given 25 minutes to familiarize themselves with *GalDef*. During this 25 minutes, after downloading and completing the tutorials for the game, players freely play *GalDef*. After the 25 minutes, participants begin the annotation tasks.

Each participant completes one annotation curve per condition per video, annotating a single affective dimension, for a total of four annotation curves per participant. After completing their annotation, participants are presented with the four videos that they provided annotations for. They are then asked to select one video for each of the following questions:

(1) In which video do you feel the music most closely matches the events and actions of the gameplay? (gameplay match)

Table 5: Results by musical condition and dimension

Measure	Model	Result	None	Linear	Adaptive	Generative	Valence	Arousal	Tension
DTW	PreGLAM	Distance	16.30	19.48	17.84	19.33	22.52	13.52	17.39
		SEM	1.05	1.48	1.20	1.44	1.19	1.20	0.71
	Random walk	Distance	24.20	25.63	25.44	25.64	27.53	24.61	25.71
		SEM	1.60	1.95	1.80	1.96	1.46	2.00	1.30
RMSE	PreGLAM	RMSE	1.06	1.04	0.99	1.07	1.23	0.73	1.08
		SEM	0.06	0.06	0.05	0.06	0.04	0.05	0.04
	Random walk	RMSE	1.34	1.38	1.35	1.38	1.36	1.28	1.41
		SEM	0.05	0.06	0.06	0.06	0.04	0.06	0.04

(2) In which video do you feel that the music most closely matches the emotion that you perceive from the gameplay? (emotion match)
(3) In which video did you feel most immersed in the gameplay? (immersion)
(4) Which video's music did you enjoy the most? (preference)

4.2 Results

48 participants take part in our study. Of these, 23 use he/him pronouns, and 25 use she/her. 55% of participants report playing between 0-4 hours of games per week, and the average age of participants is 23.60 years old. 39 participants are recruited from undergraduate students at (institution withheld for anonymous review), 4 participants are recruited via email and message boards, and 5 participants are recruited using Amazon's Mechanical Turk platform. For all participants, the study is identical.

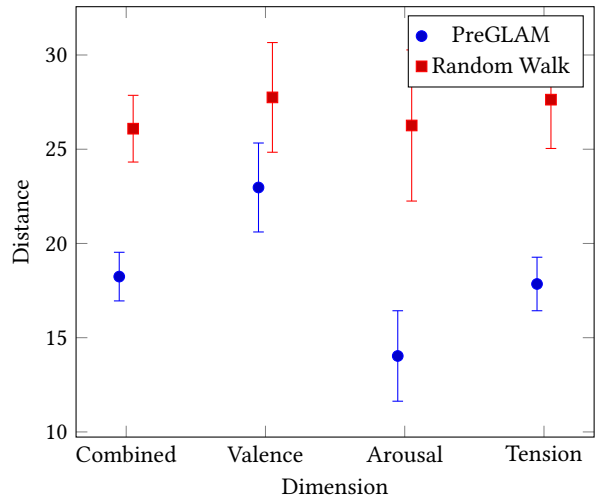
We analyze our results using Dynamic Time Warping (DTW), with the dtw-python library [7], and calculate the Root Mean Squared Error (RMSE) based on z-score scaling. Table 5 shows these values, and Figure 6 shows the DTW Distance and 95% confidence interval. DTW is a measurement of similarity between two time series that may vary in speed. RMSE is a commonly used measure of the similarity between predicted and actual values. These measures provide both a measure of contour similarity with DTW, and overall similarity with RMSE.

In Table 5, the responses for musical condition are aggregated across affective dimension, and the dimension responses are aggregated across conditions. In other words, the DTW Distance between PreGLAM and the ground-truth annotations for the "linear" condition represents the combined average distance of valence, arousal, and tension annotations when the linear score is played.

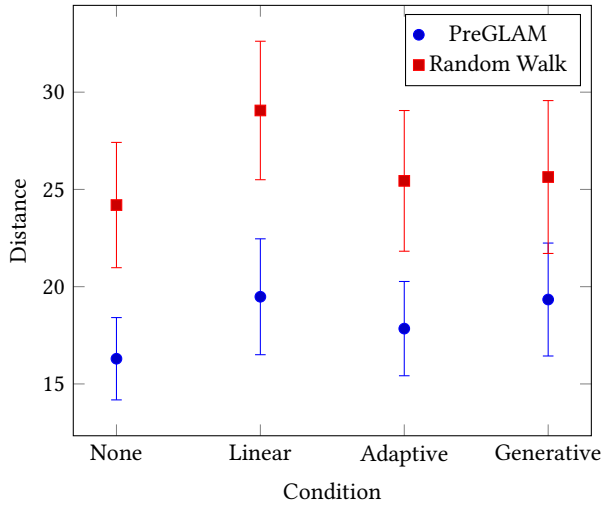
Each participant's annotation curve is compared directly to PreGLAM's output annotation. Additionally, we provide a more absolute measure by comparing each participant's annotation curve to a random walk time series. These results therefore demonstrate the distance measures between PreGLAM and ground-truth annotations, in comparison to the distance measures between the random walk and the ground-truth annotations.

We test the assumption of normality, and find that the data is normally distributed in all four measures. We perform a t-test to compare results and find significant difference between PreGLAM and random walk compared to user annotations, $p < 0.01$ for both metrics. We perform post-hoc two-way t-tests separated by condition and dimension. Results of these t-tests are shown in Table 6

PreGLAM significantly outperforms the random walk in both DTW-Distance and RMSE across all conditions. We perform an



(a) Affective dimension



(b) Musical condition

Figure 6: DTW-Distance between PreGLAM and annotations, compared with Distance between random walk and annotations

Table 6: T-test results by musical condition and dimension

Measure	None	Linear	Adaptive	Generative	Valence	Arousal	Tension
Dtw-Distance	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p = 0.08$	$p < 0.01$	$p < 0.01$
RMSE	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p < 0.01$	$p = 0.09$	$p < 0.01$	$p < 0.01$

ANOVA across all conditions, and find no significant effects from changes in musical condition. Separated by dimension, PreGLAM significantly outperforms the random walk for arousal and tension, but does not significantly outperform the random walk for valence measures. We perform an ANOVA across all dimensions, and find that the three dimensions are significantly differentiated from another. Post-hoc Tukey tests show that all pairwise comparisons of dimensions are also significantly different — modeled arousal is significantly more accurate than modeled tension, which is significantly more accurate than modeled valence.

Figure 7 shows the distribution of questionnaire responses. In these responses, the composed linear score is rated as the highest in all questions. In terms of emotional congruency, immersion, and preference, the generative score is rated as a close second, with the composed adaptive score in a more distance third. In terms of matching the events and actions of the gameplay, the adaptive score slightly outperforms the generative score.

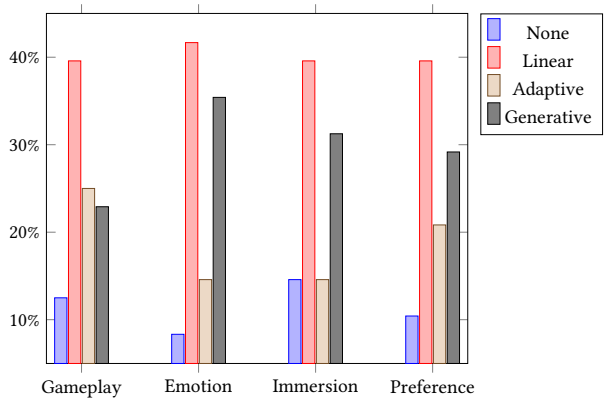


Figure 7: Distribution of questionnaire responses

4.3 Discussion

Overall, PreGLAM presents a viable emotion model for controlling adaptive music, outperforming a random walk in matching ground-truth annotations. There is a marginal increase in the distance between PreGLAM's ouput and user annotations when any music is introduced. Within the musical conditions, the distance is lowest with composed adaptive music, and highest with the composed linear score. We did not find significant differences between the distances between PreGLAM and ground-truth annotations when separated by the musical condition. In other words, the musical conditions are not significantly differentiated from each other according to real-time perceived ground-truth annotations.

The post-hoc questionnaire questions indicate support for the generative approach. As mentioned in Section 2, Williams et al. find that participants report a decreased immersion when playing with a single-instrument MIDI track, compared to an original orchestral score. We address this by using identical production processes across our three musical conditions, therefore isolating the compositional element of the musical generation. Participants judge the generative score slightly lower than the linear score in all questions, but generally much higher than the composed adaptive score. The composed adaptive score outperforms the generative score in terms of matching the actions and events of gameplay, but the generative adaptive score presents an increase in participant ranking for perceived emotional congruency, immersion, and preference.

The linear score is the only score that has composed transitions. While the linear score does not adapt its emotional expression based on gameplay, it does have rising and falling emotional arcs through time, and may produce serendipitous sync [16]. In Williams'

previous research, the generative affective score is compared with a linear score that has a mostly consistent emotional expression. While Williams' generative system outperforms their compared linear score in emotional congruency, the linear score outperformed the generative score in immersion [31].

While our generative score is close in ranking to our linear score, our linear score outperforms our generative score in all questionnaire responses. This may seem to show a step backwards from the work presented by Williams et al. [31]. We draw attention to several differences that may explain some of this discrepancy. Our emotional model adapts in response to the actions and events of gameplay in real-time, rather than associating each emotion with a single game state. Our linear musical score changes in emotion over time, rather than expressing a mostly static affect. Additionally, our linear, generative, and adaptive scores are synthesized using identical production techniques, bringing musical features such as instrumentation, timbre, tempo, genre, synthesis, and production quality to parity with the generative music. We believe that this provides a more isolated understanding of the compositional aspects of the generative score.

A linear score may be preferred by listeners due to the smoothness of transitions, and the pre-determined intentionality of its emotional expression. Contrastingly, our composed adaptive score has a limited amount of musical content for each adaptive level compared to the generative score, which may lead to the musical transition point between adaptive levels in the composed score being jarring and/or repetitive. While the application of generative music does not bring an adaptive score to full parity with a composed linear score in post-hoc participant responses, the generative score improves upon our adaptive score and upon previous applications of generative music in games.

Overall, these results indicate that while the real-time perceived effects of musical accompaniment to gameplay shown in Table 5 and Figure 6 are small, our approach to generative music is mostly comparable to linear music in terms of the emotional congruency, immersion, and preference in post-hoc responses from participants, and improves upon these features compared to purely human-composed adaptive music. This demonstrates the strength of MMM in assisting a composer to create and extend a highly adaptive score with generative music.

5 CONCLUSION

We identified several differences between academic approaches to using generative music in games, and approaches taken from the games industry. Academic systems tend to use MIDI synthesis of symbolic generative music, often with a single piano instrument. Academic systems generally use an emotion model that directly relates the absolute values of game variables to emotion values for one or two dimensions. Systems from the game industry generally use audio recordings of instruments and/or VST instruments to synthesize and produce the music offline. Industry systems rarely use an abstracted model of emotion, instead directly relating a set of game variables to musical adaptivity.

We present a hybrid approach to using generative music in video games that uses generative composition to extend and expand a composed adaptive score. This approach attempts to utilize the advantages of using advanced generative music algorithms within a score that is aesthetically similar to scores from commercial video games. We believe that this represents an evaluation of generative music in games that more closely measures how generative music may be used in real-world games than previous approaches.

This approach presents a somewhat idealized version of generative music used in video games, given current technological constraints. While our generative score technically produces unique music that matches gameplay, it does not compose music in real-time to match gameplay as the MMM algorithm is not currently capable of real-time generation. Our generative score is generated using symbolic notation, but tracks are rendered into audio files, as real-time synthesis cannot currently match the fidelity or computing performance of offline synthesis.

Our results are consistent with previous approaches to using generative music in games. While the differences are marginal, real-time annotations of perceived emotions match our predicted perceived emotion more with generative and adaptive music than with the composed linear score. Participants rank our generative score as on par with our linear score in terms of emotional congruency, immersion, and preference, and far above our composed adaptive score.

6 FUTURE WORK

In focusing on the aesthetic fidelity of our application of generative music in games, we do not necessarily exploit the full strength of generative music. While PreGLAM outputs unbounded floating point values for Valence, Arousal, and Tension, we use 5 categorical levels of emotion — We control the adaptivity of the score separately from the composition in order to use adaptive music techniques from the industry.

Additionally, we manually design the adaptivity of our score, and compose a score that has the same 3-dimensional adaptivity as the generated score. While the use of generative music allows us to easily and quickly expand the composed adaptive score, the original composition, and therefore the generated music that is based on the composition, is still somewhat restricted in expressive range to allow for relatively smooth musical transitions.

Generative music that is composed and synthesized in real-time could exhibit more musical flexibility than our composed score, and could provide more continuous adaptivity. Additionally, generative music that is composed and synthesized in real-time could have smoother transitions, as the transitions could be directly generated.

In addition to future work in the technological implementations of generative music in games, we note that we evaluate generative music acting as an audience for a single-player action-RPG game. There are many ways to use music in games, and this application of generative music may not be suitable for all of them — for example, Phillips describes the use of music as "branding", which uniquely generated music may be very poorly suited to. While we believe that our application represents a scenario for which generative music is most well suited, there are many other possible applications of generative music in video games.

REFERENCES

[1] Shakir Belle, Curtis Gittens, and TC Nicholas Graham. 2019. Programming with Affect: How Behaviour Trees and a Lightweight Cognitive Architecture Enable

the Development of Non-Player Characters with Emotions. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–8.
[2] BioWare. 2007. Mass Effect.
[3] Andrew R Brown and Andrew C Sorensen. 2000. Introducing jmusic. In *Australasian computer music conference*. 68–76.
[4] Kristofer Eng and Philip Bennefall. 2021. https://eliassoftware.com/
[5] Jeff Ens and Philippe Pasquier. 2020. MMM : Exploring Conditional Multi-Track Music Generation with the Transformer. arXiv:2008.06048 [cs.SD]
[6] Patrick Gebhard. 2005. ALMA: a layered model of affect. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. 29–36.
[7] Toni Giorgino. 2009. Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package. *Journal of Statistical Software, Articles* 31, 7 (2009), 1–24. https://doi.org/10.18637/jss.v031.i07
[8] Christian Henson and Paul Thomson. 2007. Labs. https://labs.spitfireaudio.com/
[9] Phil Lopes and Ronan Boulic. 2020. Towards Designing Games for Experimental Protocols Investigating Human-Based Phenomena. In *International Conference on the Foundations of Digital Games*. 1–11.
[10] Phil Lopes, Antonios Liapis, and Georgios N Yannakakis. 2015. Sonancia: Sonification of Procedurally Generated Game Levels. *Proceedings of the 1st Computational Creativity and Games Workshop*. (2015).
[11] Phil Lopes, Georgios N Yannakakis, and Antonios Liapis. 2017. Ranktrace: Relative and unbounded affect annotation. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 158–163.
[12] David Melhart, Antonios Liapis, and Georgios N Yannakakis. 2019. PAGAN: Video Affect Annotation Made Easy. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 130–136.
[13] David Melhart, Antonios Liapis, and Georgios N Yannakakis. 2021. The Affect Game AnnotatIoN (AGAIN) Dataset. *arXiv preprint arXiv:2104.02643* (2021).
[14] Andrew Ortony, Gerald L Clore, and Allan Collins. 1990. *The cognitive structure of emotions*. Cambridge university press.
[15] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. 2017. An Introduction to Musical Metacreation. *Computers in Entertainment* 14, 2 (2017), 1–14. https://doi.org/10.1145/2930672
[16] Winifred Phillips. 2014. *A Composer's Guide to Game Music*. The MIT Press, Cambridge, MA.
[17] D. Plans and D. Morelli. 2012. Experience-Driven Procedural Music Generation for Games. *Computational Intelligence and AI in Games, IEEE Transactions on* 4, 3 (2012), 192–198.
[18] Cale Plut. 2021. Galactic Defense linear score. https://soundcloud.com/cale-plut/galactic-defense-linear-score
[19] Cale Plut. 2021. GalDef Annotation software. https://github.com/CalePlut/GalDef_Annotation
[20] Cale Plut. 2021. How to play Galactic Defense - Video Tutorial. https://www.youtube.com/watch?v=YQtF9s5fVyc
[21] Cale Plut and Philippe Pasquier. 202. Generative music in video games: State of the art, challenges, and prospects. *Entertainment Computing* 33 (202), 100337.
[22] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. 2022. "(Preprint) PreGLAM: A predictive gameplay-based layered affect model". *IEEE Transactions on Games* (2022).
[23] Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng. 2022. "(Preprint) The IsoVAT Corpus: Parameterization of musical features for affective composition". *Transactions of the International Society for Music Information Retrieval* (2022).
[24] Anthony Prechtl. 2016. *Adaptive music generation for computer games*. Ph.D. Dissertation.
[25] Lucas Reycevick. 2016. The Brilliance of DOOM's Soundtrack. https://www.youtube.com/watch?v=7X3LbZAxRPE
[26] Rockstar Games. 2010. *Red Dead Redemption*. Game.
[27] 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*. 1–8.
[28] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. 2017. Affective evolutionary music composition with MetaCompose. *Genetic Programming and Evolvable Machines* 18, 4 (2017), 433–465.
[29] Michael Sweet. 2015. *Writing interactive music for video games : a composer's guide*. Addison-Wesley, Upper Saddle River, NJ.
[30] Marc-Pierre Verge. 1998. Applied Acoustic Systems. https://www.applied-acoustics.com/
[31] Duncan Williams, Jamie Mears, Alexis Kirke, Eduardo Miranda, Ian Daly, Asad Malik, James Weaver, Faustina Hwang, and Slawomir Nasuto. 2017. A perceptual and affective evaluation of an affectively driven engine for video game soundtracking. *ACM Computers in Entertainment* 14, 3 (2017).