2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)
Banff Center, Banff, Canada, October 5-8, 2017

Facilitating Player Progression by Implementing Procedural Music in Videogames

Jayden Chan, Justin John Daza, William Kwan and Anup Basu
Department of Computing Science, University of Alberta, Edmonton, Canada, T6G 2S4
Email: {jayden1, justinjo, wkwan, basu} <AT> ualberta.ca

Abstract – **While the multi-million-dollar videogame industry sees constant improvements in visuals and processing power with every new processor, gaming console, or graphics card release, is there any avenue to pursue new innovations in this discipline? One answer lies in the introduction of procedural music in videogames as an innovative means of overcoming the repetitive nature of traditionally composed game music and to enhance the immersive response of game music to player actions. While most studies are preoccupied with the descriptive evaluation of how entertaining procedural music is in comparison to traditionally composed music, we pursue a novel study on the utility of implementing procedural music in videogames as a tool for facilitating player progression. To do so we employ objective, quantitative measures to gather results that can quantify the utility of implementing procedural videogame music, unlike other studies. We demonstrate that users playing a game with a procedural music model that actively instructs and assists players can complete game levels in a significantly more time efficient manner and are more likely to rate the procedural music model as having significantly contributed to the game's entertainment and engagement.**

I. INTRODUCTION

Contemporary videogames are the intersection of all multimedia content forms involving diverse combinations of graphics, animations, visual texts, music, audio effects, and player (user) interactions. While constant improvements in computer processing power with every new processor, gaming console, or graphics card release may allow for continuously evolving visuals and player interactivity, researchers continue to demonstrate interest in studying the implementation of procedural videogame music as a means to overcome the repetitive linearity of human composed videogame soundtracks. Such music models may also assist in reducing the development time and cost of having to compose many individual tracks for a single game [1], [2].

Within the perspective of our study we consequently define procedural videogame music as the implementation of non-linear, often synthetic sounds that are generated in real-time based on a combination of player input and a game's algorithmic rules and procedures [3] to produce spontaneously dynamic soundtracks that immerse players in the game through emotional induction [2] that ultimately results in the players being emotionally involved. As an example, upon reading the algorithmic parameters for how much remaining life a player character has remaining or how many points (s)he may have accrued thus far, a procedural soundtrack may spontaneously compose or generate audio that perhaps reflects the peril of a player near death and/or the triumph of almost having collected enough points to complete a level.

While nearly all contemporary retail videogames feature music, it usually serves the principal task of emphasizing the overall atmosphere of the games [1]. Furthermore, the majority of such game music is usually linear human composed soundtracks much like those of theatrical film scores [1], [4]. That is, just as every scene in a film typically has a specific music track composed for it with the purpose of eliciting the emotion the movie director wishes to associate with the scene, every level, environment, or cut scene cinematic of a game has a certain linear music track composed for it with the purpose of eliciting the emotion that a game developer wishes to have the player feel. Although players can appreciate the professional film-like scale and quality of this kind of videogame music, it generally suffers from repetitiveness and player fatigue [5] – especially if players spend a long time in particularly difficult or extended sections of a game [2]. Accordingly, it appears as if the primary outcome of interest for many researchers at this time is to evaluate whether prospective players would actively appreciate and approve of their particular experimental model of procedural videogame music. However, because research regarding procedural videogame music is new [1], [4], the breadth of academic articles on the matter is limited [4]. Moreover, most articles that are available tend to be descriptive and/or predominantly qualitative in the reporting of their results [4] – a trend that is demonstrated by many of the studies [5], [6], [7] discussed in this article. By contrast, in this paper we introduce an approach to implementing and studying procedural videogame music that focuses on players' appreciation of the music's ability to actively facilitate their progression through a game as opposed to their subjective evaluation of the music's emotional engagement or ability to entertain as compared to traditional human composed music. Furthermore, we put into practise an experimental design that endeavours to obtain repeatable, quantitative results that can demonstrate the objective utility of implementing procedural videogame music designed to facilitate player progression, in addition to gathering qualitative player evaluations of the experimental procedural music model used.

The remainder of this paper is organized as follows. In Section II we present related research in implementing procedural videogame music. Section III provides an overall exposition of our procedural videogame project. Then, in Section IV we discuss our conceptual procedural videogame music design model, which is followed by a description of our algorithmic implementation of the design model in Section V. In Section VI and VII the experimental design and results are reviewed respectively. Finally, we end with concluding remarks in Section VIII.

II. RELATED WORK

Though few in number, procedural videogame music has been the main focus of various contemporary multimedia research studies. In this section, we compare our approach with some of those studies. Hoover et al. [6] explores the intersection between player actions and machine learning in the spontaneous and interconnected procedural generation of various videogame aspects like audio, visuals, and interactivity all at once in the their AudioInSpace game project. Lopes et al. [7] combine procedurally generated music and game level environments in their ongoing Sonancia project. In particular, the procedural music generated in the Sonancia project is specifically designed to represent the horror genre of videogames. Despite the Sonancia project's sound effort to accentuate player immersion, the narrow and nearly entirely descriptive focus on generating player emotions of foreboding fear and suspense also makes it difficult to generalize any findings to other game projects. Moreover, there is very little discussion of experimental results in the Sonancia papers [7] as Lopes et al. are also ultimately concerned with ongoing evaluations of how "[their procedural sound] should be distributed in order for the experience to be more enjoyable to players." Plans and Morelli [5] encoded procedural music based on a player experience driven procedural content generation framework (EDPCG) to the MarioAI Championship engine [8]. Numerous player parameters like time spent running, monsters defeated, times fell into endless gaps, times ducked, time spent standing still, etc. were recorded and used as variables to compute and determine an overall player excitement score. Nevertheless, the experimental results were inconclusive as Plans and Morelli acknowledged a potentially "poor test design" [5] in their study since it only consisted of a subjective questionnaire offering "only three choices: 'I enjoyed this [game] round', 'I almost enjoyed this [game] round', and 'I didn't enjoy this [game] round'" in reference to whether test subjects found playing rounds of their MarioAI procedural music project enjoyable. Additionally, Plans and Morelli's selection of broad, non-specific player parameters and focus on MarioAI's 'platformer' genre of games in which player characters jump and dodge enemies and obstacles between various platforms is also a challenge to generalization.

All of the aforementioned studies focused predominantly on the descriptive evaluation of whether or not players would find the implementation of procedural music in videogames entertaining, immersive, or otherwise emotionally engaging compared to traditionally composed videogame music. Consequently, none of the studies provided any objective, repeatable, or quantitative experimental results. The novelty in our paper lies in studying the utility of implementing procedural music in videogames as a tool for facilitating player progression as opposed to focusing on the descriptive evaluation of player enjoyment. Moreover, our study also employs objective, repeatable, and quantitative measures to gather results that can quantify the utility of implementing procedural videogame music; something other studies have not.

III. OVERALL GAME DESIGN

The core of this project revolves around a fully functional Unity 3D game environment that demonstrates our procedural music model's ability to facilitate a player's progression in completing a given level. Stylistically, the player is cast as a suave master thief in disguise who is given the basic controls of full 360 degree running movement and the abilities to jump or crouch while moving. The level itself is designed to look like an art museum with various rooms housing a myriad of priceless artifacts. Although the player must ostensibly find and 'steal' or collect eight specific objects out of over 60 object assets placed in the environment, the level and its visuals are designed convincingly such that it is not immediately obvious or easy for first time players to know or find the necessary objects without any hints. Furthermore, the level is populated with dozens of non-player-characters (NPCs) – most of whom are visiting pedestrians who simply wander randomly around the environment and do not affect the player's character avatar in any way. Nevertheless, there also exist a handful of plain-clothes undercover police officers – enemy NPCs who blend easily into the background by looking similar to many other wandering NPCs. Upon contact with any one of these enemies, the player loses one of their five health points and a game over state is rendered when all five health points are lost. Finally, when the player has collected all eight necessary objects (s)he must find an escape vent that is cleverly hidden in the level to 'escape' and ultimately complete the game project level successfully.

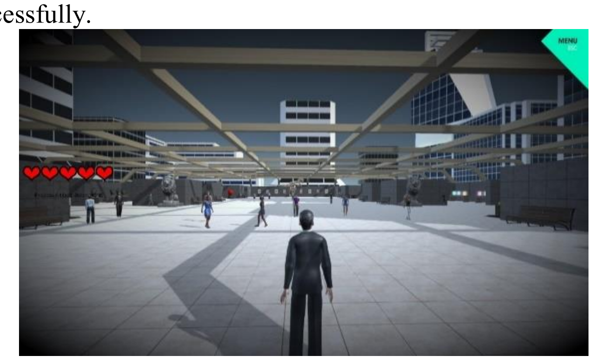

Fig. 1. Players must avoid unidentified enemies and find the correct items

The procedural music model that we implemented in the game project level facilitates a player's progression in completing the level. A number of pre-composed audio tracks comprised of short and simple sets of beats or instrument rhythms are stored in the game's algorithm. On one hand, short tracks of beats per minute – not unlike that of one's heart rate – play faster and more intensely with more beats added when enemies get close to the player, when many enemies are following the player at once, or when the player's health worsens. Being able to maintain good health or escaping far from pursuers' causes those beat tracks to subside until danger arises again. On the other hand, when players move within a certain proximity of an item that needs to be collected, randomized uplifting tracks of instrumental (piano, for example) rhythms play louder. When a player has finally collected all the eight necessary items, similar instrumental rhythms of success play the closer (s)he moves towards the hidden escape vent. Incidentally, the vent is located under a texture that can be rendered to look like a water fountain's pool. Because the level does not prevent the player from exploring or jumping into the pool, the level design and the procedural music collaboratively assist the player in finding the exit. It is simple to observe how the project's procedural music model facilitates player progression by instruction and assistance. Still, depending on how different players choose to idiosyncratically move about the level while spontaneously searching for objects and avoiding enemies, the procedural music is capable of sounding quite different from one moment to another. Additionally, we decided to implement a 3D environment so as to take advantage of the multi-directionality of stereophonic sound. As a result, not only do players know they are close to an object, enemy, or escape route but depending on whether they 'hear' the procedural music emanating from in front, behind, to the side, above, or below, they also receive clues as to where those things may be located.

Fig. 2. Enemies are not immediately distinguishable from harmless NPCs.

The overall design of our game can be generalized to many contemporary retail titles. In that sense, our decision to select and record specific game and player parameters like objects to collect, number of objects collected, goal area location, enemy NPCs, player proximity to these instances, and player health also match closely with variables like collectable items, number of items collected, goal area waypoints, opponent NPCs, and player character health – all of which are very universal and generalizable aspects found across almost any genre of 2D or 3D videogame. Given how these same parameters are used to elicit the procedural music in our game, one can see how these kinds of parameters can assist in facilitating player progression.

IV. PROCEDURAL MUSIC MODEL DESIGN

In the process of designing how to represent player instruction and assistance with procedural music, we had to first decide what instructions we wanted to present to the player. As our overall game design consisted of having a 3D player avatar explore a large open 3D environment comprised of hostile enemies and collectible objects. The predominant instruction we decided to convey to the player was that of 'successfully collecting all the items while avoiding enemies.' Nevertheless, given most procedural music studies' preoccupation with how pleasing or appropriate their procedural music models sounded, we sought to find a simple and straightforward method to be able to generate procedural music that provided us with enough control over how the actual music should sound.

In particular, Plans and Morelli's [5] experimentation with EDPCG in which various parameters and elements of a player's actions and experience in a game would determine how the game music that was generated provided strong inspiration. With the EDPCG framework in mind, we actively selected specific parameters of our game that would work well in representing player instruction and assistance. We consequently mapped parameters like number of items collected and enemy proximity to variables that would influence the music generated. Considering the instructions that we wished to convey to players, the variables that we decided to integrate these specific parameters into were labelled *intensity* and *progress*. *Intensity* acts as an indicator of danger and urgency for the player to take caution. *Progress* indicates whether or not the actions a player takes is progressing the player towards desired goals.

Table 1: Factors contributing to our procedural music model.

Multiplier	Intensity/Tension	Progress
Enemy proximity	-1	0
Collectible/Goal proximity	0	-1
Number of collectibles obtained	-1	1
Remaining health (5 to 0)	1	0

The models we used to calculate *intensity* and *progression* are as follows:

$$Intensity\ (or\ Progression) = a_1 \times Enemy\ Proximity \times Number\ of\ Enemies$$
$$+ a_2 \times Item/Goal\ Proximity$$
$$+ a_3 \times Items\ collected$$
$$+ a_4 \times Remaining\ health$$

$$Intensity\ (or\ Progression) = \sum_i a_i c_i \text{, where } a_i = \text{ weight for } ith \text{ metric, } c_i.$$

Since we use two variables – *intensity* and *progression* – we need to have a procedural music generation model that reflects this two-dimensional spectrum. For the purpose of substituting written or visual instruction, we could not overlook the fact that the quality or type of sound used would play a significant role. These considerations led to the idea of using a *riffology* inspired design in our model. *Riffology* makes use of piecing together small, pre-composed tracks of music into a variety of fuller, unique compositions [2]. By employing this technique, we are able to incorporate, with some degree of control, various instrumental sounds that are not normally heard in conventional procedural music generators. Although the collection of smaller 'riffs' can be combined in various ways to create various permutations of unique tracks, one caveat is that more permutations would ultimately require a composer to compose more riffs, despite them being smaller and more simplistic than full pieces of traditionally composed music.

To simplify the composition process we separated *intensity* and *progress* into two exclusive sections of the game level's full composition. In an effort to reinforce the match between *intensity* and caution, *intensity* was coupled with percussive riffs, in our case being electronic beat sounds. On the other side of the composition spectrum, *progression* handles instrumentation that does not classify under percussion. In terms of *progression*, it influences the 'uplifting' quality of the music tracks. Higher or lower progression would correlate to the number of layers of musical tracks making the overall background composition sound more or less grandiose or of the sense that the player is progressing more, or perhaps less, towards success.

To further simplify our procedural music generation model, we quantized *intensity* and *progression* into four levels ranging from 0 to 3. The purpose of this was that it complemented our compositional design choice of layering musical riffs on top of each other, in addition to being relatively easy to organize and manage. The four states were defined as follows. State I: Player has sufficient distance from threats and desired goal. State II: Player is in close proximity to threats but far from the goal. State III: Player is in close proximity to both the goal and threats. State IV: Player is sufficient distance away from threats but is close to the goal

V. IMPLEMENTATION OF THE PROCEDURAL MUSIC MODEL ALGORITHM

Our model was implemented using three main components: *ProceduralMusicInitalize.cs*, *ProceduralMusic.cs*, and *musicObjectInit.cs*. To begin, one would first attach the *ProceduralMusicInitialize.cs* script onto an object inside the level environment and then employ the *ProceduralMusic.cs* script.

ProceduralMusicInitalize.cs

This component initializes a number of key classes that are used in the implementation of the model. The first class is *musicObject*. This class is created to help organize objects that the algorithm would need to keep track of in order to apply the model. The variable *self* is the identifier for the object which uses instanceIDs to index and *type* is a string that is later used to identify the kind of object referenced. The labels can be determined by the particular developer of the project. In our project we specifically used the labels enemy, item, and object.

A system to keep track of the riffs, or audio clips, that we are using would also be needed. Another study that this design took inspiration from is Lopes et al.'s [7] Sonancia project. Whereas that project used a system to reduce the repetition of music clips by only playing each clip once until all of them had been played at least once before allowing repeats to occur, we created a similar system with the appropriately named *riffList* class. When the active dictionary results in an empty attribute level indicating that all clips of that attribute level had already been used or played, audio clips are moved back from the off dictionary to the active dictionary and shuffled to further minimize repetition. Similarly, a global dictionary is used to store all the music object assets.

ProceduralMusic.cs

The principle algorithmic implementation of procedural music model is inside this module. *ProceduralMusic.cs* is a relatively large class with many individual functions. The first few functions discussed are *calculateTension*() and *calculateProgression*() as both are vital to the model. As the name suggests, *calculateTension*() is used to calculate the relative *intensity* occurring in the game. What the algorithm does is iteratively look at all music objects and checks their distance from the player. As the player collects more items, the game becomes less intense as the player gets closer and closer to completing the level. As a result, each item collected serves to reduce the collective *intensity*. We also check the player's health. As it gets lower the *intensity* also increases. At the end of the function we check for boundaries based on the previously defined *intensityLevels*.

Conversely, *calculateProgression*() calculates the relative *progression* level in the game. The function's implementation looks at the distance between the player and collectible items and, like *calculateTension*(), adds up these distance values. Next, the algorithm looks at how many items the player has collected with a higher *progression* level as more items are collected. It is the *countMusicObject*() function that actually counts the amount of *musicObjects* of a specific label that exist in a Unity scene. Finally, in order to assist players find the hidden escape exit in the level, the algorithm heavily weights the *goal* area object – but only after all the items have been collected.

The method *prox*() is a proximity method used for non-directional audio. This is useful for changing the music without giving away the spatial stereophonic position of the *musicObject* it is being influenced by. Alternatively, the proxDirec() method is more specialized than prox(). It uses the audiosources of the musicObjects. This means it can allow for directional stereophonic audio which allows players to get a sense of where the music is coming from.

musicObjectInit.cs

The last of the three main scripts is musicObjectInit.cs. The sole duty of this script is to be attached to objects that need to be kept track of. Researchers would have to create one of these scripts for each object so that they can change the label to the relevant one.

VI. EXPERIMENTAL DESIGN

To evaluate our game project with sample players, we created two variations of our game level. One variation had all our procedural music replaced with a single track of traditionally composed background music. The single track of music was chosen to reflect and accentuate the stylistic theme of tension and intrigue associated with theft in a museum. This variation was considered the control level. Alternatively, the second variation of our level included all of our procedural music and played as intended based on our overall game design and procedural music model implementation. This second variation was used as the test level. Twelve players were recruited to participate in the experiment, with six players each for the control and test groups. Although the variation of the game level played depended on the group a player belonged to, the conditions before, during, and after play did not vary between the players. At most one researcher accompanied each player during a play session. The sessions took place on a laptop with headphones at a desk on a quiet library floor. Just before playing a researcher made a player wear a wristwatch style heart rate monitor and measured the player's initial heart rate. The researcher then gave each player a short, identical explanation of the game's controls and the basic premise that a player had to "collect the correct items and then find the hidden escape exit" to complete the level. During active play time, the researcher would record the time it took a player to complete the level but no further questions about the game or experiment were asked. Once a player completed his or her variation of the game level the researcher would again record the player's heart rate and then administer a player experience questionnaire. After the player completed his or her questionnaire the experiment was deemed complete. All experimental data, including each player's heart rate before and after playing the level variation, each player's completion time, and each player's questionnaire response scores were then compiled and evaluated.

VII. RESULTS AND DISCUSSION

As the total sample size was small we must be careful in interpreting the results. Nevertheless, the results do offer a number of general insights. One-tailed two-means t-tests were performed on each of the heart rate, completion time, and questionnaire response scores to facilitate the discussions.

Table 2. Heart rates of six control group sample players (SP) and six test group sample players before and after playing their level.

Heart Rate	SP 1	SP 2	SP 3	SP 4	SP 5	SP 6
(Control) Before	72	69	78	81	66	73
(Control) After	80	75	86	92	73	82
(Test) Before	83	70	65	79	72	71
(Test) After	93	77	72	90	78	79

Despite the small sample size, there was no statistical significance in the heart rate differences between the two experimental groups, and the sample players of both groups experienced similar degrees of increased heart rate after playing the game level variation. Since the players of the control group experienced increases in heart rate, this suggests that the basic game level that we created – even without any kind of reactive, procedural music – was stimulating or otherwise emotionally engaging enough to elicit noticeable increases in heart rate in the control players. At the same time, because the control variation of the game level essentially represented gameplay with traditionally composed videogame music and the test group players experienced degrees of heart rate increase similar to the control group players, this may also suggest that the use or implementation of procedural videogame music can elicit a degree of game playing stimulation or engagement that is at least as high as when playing with traditionally composed game music.

Conversely, various studies have suggested that while heart rate is a legitimate index of emotional arousal [9], it is not necessarily a particularly reliable metric of measure because it has been shown to be influenced by both the parasympathetic and sympathetic nervous systems [10]. Furthermore, a test subject's heart rate may be subject to medical conditions that may or may not be known to either the researcher or the subject. As a consequence, more reliable measures, perhaps like those obtained from an individual's electrodermal activity (EDA), need to be tested. However, the equipment necessary to gather EDA measurements is complex. Previous studies performed by Nacke et al. [9] on subjects who played with traditionally composed videogame music and subjects who played without game music have shown that EDA measurements may not capture any statistically significant results either. Consequently, the challenge is to find an objective, repeatable, and quantifiable physiological measure that is effective in evaluating the differences between procedural videogame music and traditionally composed videogame music in engaging players.

Table 3. Times to complete a level variation for each sample player (SP).

Time	SP 1	SP 2	SP 3	SP 4	SP 5	SP 6
Control	13m21s	15m03s	15m33s	16m23s	18m51s	17m47s
Test	06m09s	08m17s	16m46s	07m31s	07m58s	08m57s

In general, the test group players required a noticeably shorter amount of time to complete the game level and the overall times recorded do result in a statistically significant difference in the completion times between the control and test groups. This result was expected considering that incredible care and effort was expended in specifically designing the procedural music. We must nonetheless be wary that such time results may vary significantly between different types of individuals recruited. For instance, we experienced at least one outlier sample player in the test group who, despite agreeing that the procedural music was significantly helpful in facilitating their progress, took a particularly long time to complete the test level variation simply because (s)he wanted to explore the entire level environment out of curiosity. Nevertheless, we observe that the longer it took control group players to complete the level, the more frustrated they became in having to continuously identify and dodge enemies when they ended up being too close or not being able to find all the items or escape exit without the aid of the procedural music model.

Table 4. Cumulative questionnaire response scores for each sample player (SP).

Survey	SP 1	SP 2	SP 3	SP 4	SP 5	SP 6
Control	12	9	9	6	13	10
Test	28	26	23	35	24	31

The player experience questionnaire administered was as below. Each item was scored as (Not at all = 0), (Slightly = 1), (Moderately = 2), (Fairly = 3), (Extremely = 4).

1. I thought it was easy to find and collect the items.
2. I thought it was easy to identify and avoid the enemies.
3. I thought it was easy to find and reach the level escape area.
4. I felt the game was engaging and immersive.
5. I though the game was entertaining and felt content to have had the chance to experience it.
6. I thought the music and sound effects helped me to find and collect the items.
7. I thought the music and sound effects helped me to identify and avoid the enemies.
8. I thought the music and sound effects helped me to find and reach the level escape area.
9. I felt the music and sound effects contributed significantly to making the game entertaining.
10. I felt the music and sound effects contributed significantly to making the game engaging and immersive.

The difference in player experience questionnaire response scores between the control group and test group was also statistically significant despite the small sample size. In particular, the test group players provided significantly higher response scores. However, given the fact that the test group players were able to play with the procedural music and subsequently complete the game level significantly faster than the control group players, it is plausible that the test group players would have enjoyed the experience more and given higher response scores for the questionnaire items.

In addition, although the primary focus of this study was not on evaluating whether or not players found our procedural videogame music model to be at least as effective as traditionally composed videogame music, the fact that most of the test group players noted that our procedural music seemed to contribute significantly to making the game level entertaining, engaging, and immersive suggests that they may have felt engaged and immersed by a procedural music system that constantly and actively aided them during gameplay, almost like a constant companion that was playing collaboratively with them. However, the lack of any apparent correlation between the test group's higher response scores and the degrees of increase in their heart rates in comparison to the control group may further corroborate Nacke et al.'s [9] intimation that objective, physiological data and evidence may not necessarily correlate with subjective data and evidence in similarly themed studies on immersion and videogames. The control group players' consistently low questionnaire scores do however highlight that a repetitive, looping soundtrack does not contribute any more to the entertainment and engagement of a game as it can initially, when players hear it for the first few times.

VIII. CONCLUSION

In this study we demonstrated the utility of implementing procedural videogame music as a tool for facilitating player progression through instruction and assistance. Despite an admittedly small sample size of players, our game design and procedural music model demonstrated that users playing with actively instructive and assistive procedural music could progress, advance, and complete a game level significantly faster than without. Furthermore, users who played with our facilitative model of procedural music also tended to concur that the music contributed significantly to making our game entertaining, engaging, and immersive.

In addition to the above, the ability for our player assistance modeled procedural music to generate engaging experiences for players – for example by cuing players into the location of a hidden escape exit only after all necessary items are collected – demonstrates the creative and interactive capability for procedural music to spontaneously create new experiences for players dynamically whereas linear and continuously looping traditionally composed scores can ultimately serve only to passively and statically reflect what is already present in an environment. Nonetheless, the inconclusive results stemming from the heart rate measurement aspect of our experimental design serves only to perpetuate the challenge of finding an objective, reliable, repeatable, and quantitative measure for effectively evaluating the differences between procedural videogame music and traditionally composed videogame music to engage and immerse players. The relative success and generalizability of our design and model, however, allows us to consider continued work into making our code and algorithms open source, generalizable, and ultimately usable across any project using our game development toolset.

REFERENCES

[1] Julian Togelius et al. "Procedural content generation: Goals, challenges and actionable steps." Dagstuhl Follow-Ups. Vol. 6. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[2] Karen Collins. "An introduction to procedural music in video games." Contemporary Music Review 28.1 (2009): 5-15.

[3] Andy Farnell. "An introduction to procedural audio and its application in computer games." Audio mostly conference. 2007.

[4] Rod Munday. "Music in video games." Music, sound and multimedia: From the live to the virtual (2007): 51-67.

[5] David Plans and Davide Morelli. "Experience-driven procedural music generation for games." IEEE Transactions on Computational Intelligence and AI in Games 4.3 (2012): 192-198.

[6] Amy Hoover et al. "Audioinspace: Exploring the creative fusion of generative audio, visuals and gameplay." International Conference on Evolutionary and Biologically Inspired Music and Art. Springer International Publishing, 2015.

[7] Phil Lopes, Antonios Liapis, and Georgios N. Yannakakis. "Sonancia: Sonification of procedurally generated game levels." Proceedings of the 1st computational creativity and games workshop. 2015.

[8] J. Togelius, "2011 Mario AI Championship," [Online]. Available: http://www.marioai.org/

[9] Lennart Nacke. Affective ludology: Scientific measurement of user experience in interactive entertainment. Diss. Blekinge Institute of Technology, 2009.

[10] J.F. Papillo and D. Shapiro. The cardiovascular system. In J. T. Cacioppo & L. G. Tassinary (Eds.), Principles of psychophysiology: Physical, social, and inferential elements (pp. 456–512). New York: Cambridge University Press, 1990.