Evolving in-game mood-expressive music with MetaCompose

Marco Scirea
University of Southern Denmark

Peter Eklund
Deakin University, Australia

Julian Togelius
New York University, USA

Sebastian Risi
IT University of Copenhagen, Denmark

ABSTRACT

MetaCompose is a music generator based on a hybrid evolutionary technique that combines FI-2POP and multi-objective optimization. In this paper we employ the MetaCompose music generator to create music in real-time that expresses different mood-states in a game-playing environment (Checkers). In particular, this paper focuses on determining if differences in player experience can be observed when: (i) using affective-dynamic music compared to static music, and (ii) the music supports the game's internal narrative/state. Participants were tasked to play two games of Checkers while listening to two (out of three) different set-ups of game-related generated music. The possible set-ups were: static expression, consistent affective expression, and random affective expression. During game-play players wore a E4 Wristband, allowing various physiological measures to be recorded such as blood volume pulse (BVP) and electromyographic activity (EDA). The data collected confirms a hypothesis based on three out of four criteria (engagement, music quality, coherency with game excitement, and coherency with performance) that players prefer dynamic affective music when asked to reflect on the current game-state. In the future this system could allow designers/composers to easily create affective and dynamic soundtracks for interactive applications.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**;
• **Applied computing → sound and music computing**;

KEYWORDS

Music generation, Affective expression, evolutionary algorithms

ACM Reference Format:
Marco Scirea, Peter Eklund, Julian Togelius, and Sebastian Risi. 2018. Evolving in-game mood-expressive music with MetaCompose . In *Audio Mostly 2018: Sound in Immersion and Emotion (AM'18), September 12–14, 2018, Wrexham, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3243274.3243292

1 INTRODUCTION

This paper investigates the use of the MetaCompose [32, 33] music generator in a game-playing context. Scirea *et al.* previously showed that the MetaCompose system is able to reliably express moods in music [30], however it has to this point not been demonstrated that such an affective music system can improve player experience in the domain of games. In this paper we aim at providing a first look at how much (if any) effect the use of this kind of music system would have on the perception of a game. The game chosen for this experiment is American-checkers (see Section 4 for description of this well-known game). Checkers is chosen for three reasons: it is well-known, the game has simple rules that are easy to grasp for even those players unfamiliar with it, and it has a minimal amount of intrinsic narrative. While the two former come from practical considerations, we wanted a game that satisfied the latter requirement to remove as many variables as possible that could influence perceptions of the game.

The research questions this study addresses are:

(1) can any difference in player experience (emotionally) be observed when presented with affective-dynamic music compared to static music?
(2) can differences be observed when the music is supporting the game's internal narrative/state?

These research questions correspond to two hypotheses:

(1) players prefer background music with dynamic affective expression;
(2) players prefer background music where the affective expression is consistent with the game state.

In short, the first question/hypothesis pair allows us to explore the effect of *dynamic* music, while the second tests

AM'18, September 12–14, 2018, Wrexham, United Kingdom

ACM ISBN 978-1-4503-6609-0/18/09…$15.00
https://doi.org/10.1145/3243274.3243292

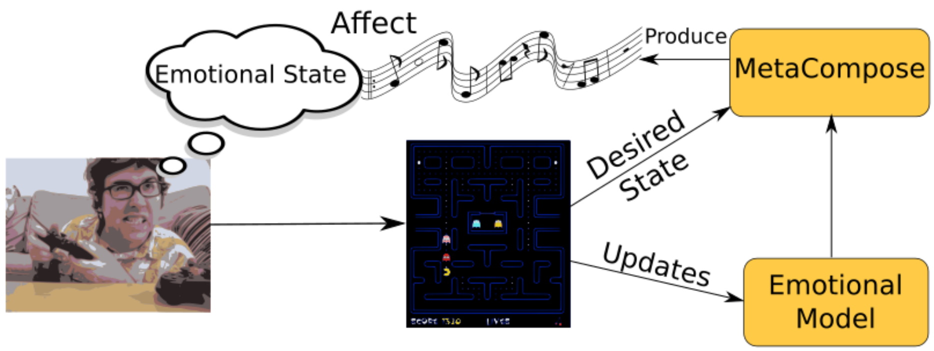


Figure 1: A player interacts with a game and through this interaction updates an emotional model and an affective music generator (MetaCompose). The game, potentially through designer annotation, tells the music generator in what emotional state it would like the player to be in. At the same time an emotional model of the player estimates the current emotional state of the player. The music composer, by consulting the emotional model, decides what affective expression to match to the music. Finally by listening to the music, the player's emotional state is influenced.

the effect of *adaptive* music. To this end we present and discuss the results of a participant-based evaluation in which test-subjects are tasked to play two Checkers games with musical accompaniment. Both self-reported data is collected – through a questionnaire – as well as physiological data via sensor input.

This paper presents results that support the two hypothesis, especially in showing preference towards dynamic affective music. The results showcase the potential of the MetaCompose system – and by extension other affective music generator systems – and, more importantly, they suggest that the affective/dynamic music generation paradigm MetaCompose is based on (see Figure 1) can lead to an improved player experience.

2 MUSIC GENERATION AND GAMES

Procedural generation of music is a field that has received much attention in the last decade [18], which is briefly summarized in this section.

Wooller *et al.* [36] identifies two categories of procedural music generation, namely *transformational* and *generative* algorithms. MetaCompose [32], falls in the latter category. *Transformational* algorithms act upon an already prepared structure (audio clips, MIDI files, etc.), for example by having music recorded in layers that can be added or removed at a specific time to change the feel of the music. Note that this is only one example and there are a great number of transformational approaches [1, 2], but a complete study of these these is beyond the scope of this paper. *Generative* algorithms instead create the musical structure themselves, which leads to a higher degree of complexity in keeping the produced music of consistent quality and coherence, especially when wanting to connect the music to game events. Such an approach requires more computing power, as the musical content has to be created dynamically and on the fly. An example of this approach can be found in the game *Spore*: the music generators were created by Brian Eno with the *Pure Data* programming language [21], in the form of many small samples that assemble to create the soundtrack in real-time.

MetaCompose adopts the latter approach, in particular focusing on generative procedural music generation in games for emotional expression. While the topics of affect [3], semiotics [10] and mood-tagging [14] are also interesting and significant, the focus of this system is *real-time generation of background music able to express moods during game play.*

Many projects focus on expressing one (or more) affective states; an example is described by Robertson [23], where a music generator is developed to express fear. There are parallels between Robertson's work and MetaCompose, for example musical data is represented via an abstraction (in Robertson's case via the CHARM representation [34, 35]), yet Scirea *et al.* [32] claim their system has a higher affective expressiveness since it is designed to express multiple moods in music. A more extensive example of a generative music system targeted at expressing particular emotions is described by Monteith *et al.* [20] using Markov models, *n*-grams and statistical distributions from a training corpus of music. Chan and Ventura's work [6] focuses on expressing moods; yet their approach relies on changing the harmonization of a predefined melody, while MetaCompose generates the complete musical piece.

There are many examples of evolutionary algorithmic approaches to generating music, two notable examples are the methods to evolve piano pieces by Loughran *et al.* [15] and Dahlstedt [7], although many more can be found in the

Evolutionary Computer Music book [19]. Other examples of real-time music generation can be found in patents. Two examples are a system that allows the user to play a solo over some generative music [22], and another that creates complete concerts in real-time [17]. An interesting parallel between the second system [17] and MetaCompose is the incorporation of a measure of "distance" between music clips in order to reduce repetition. Still, neither of the patented systems present explicit affective expression techniques.

As the final objective, MetaCompose is designed to be employed to create computer game music. It is therefore important to mention the work by Livingstone [14], which defines a dynamic music environment in which music tracks adjust in real-time to the emotions of the game character (or game state). While this work is interesting, it is limited by the use of predefined music tracks for affective expression. Finally, another notable project in affective expressive music in games is *Mezzo* [4], a system that composes neo-Romantic game soundtracks in real-time and creates music that adapts to emotional states of the character, mainly through the manipulation of *leitmotifs*.

3 METACOMPOSE

Scirea *et al.*'s MetaCompose consists of three main components: (i) *composition generator*, (ii) *real-time affective music composer*. This section presents a summary of the music generation method employed by MetaCompose, a more complete description can be found in [33].

The *composition generator* (i) creates the basic abstraction of a score that will be used by the *real-time affective music composer* in order to (ii) create the final score according to a specific mood or affective state. In other words, the *composition generator* (i) serves as a composer that only writes the basic outline of a piece, while the *real-time affective music composer* (ii) acts as an ensemble, free to interpret the piece in different ways. The system also has an *archive* which maintains a database of all the previous compositions connected to the respective levels/scenes of the game-state while also allowing a rank to be computed that measures the novelty of future compositions compared to those previously generated. MetaCompose is designed to react to game events depending on the effect desired. Examples of responses to such events include: a simple change in the affective state, a variation of the current composition, or an entirely new composition.

Composition in the context of MetaCompose refers to an abstraction of a music piece composed by a *chord sequence*, a *melody* and an *accompaniment*. It is worth noting that the term *accompaniment* denotes another abstraction (a simple rhythm and an *arpeggio*), not the complete score of a possible accompaniment. The main reason for the deconstruction of compositions is to produce a general structure (an abstraction) that we believe makes music recognizable and provides identity. Generating abstractions, which themselves lack some information that one would include in a classically composed piece of music (e.g. tempo, dynamics, etc) allows MetaCompose to modify the music played in real-time depending on the affective state the interactive media wishes to convey through the mood expression theory. The generation of compositions is a process with multiple steps: (i) creating a chord sequence, (ii) evolving a melody fitting this chord sequence, and (iii) producing an accompaniment for the melody/chord sequence combination (see Figure 2).

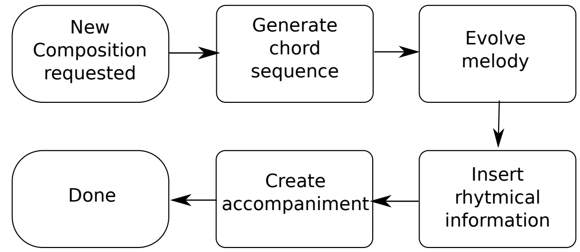


Figure 2: Steps for generating a *composition*.

Scirea *et al.* [28, 31] define a number of features to include (objectives) and to avoid (constraints) in melodies, these are based on classical music composition guidelines and musical practice. The constraints define that a melody should: i) *not have leaps between notes bigger than a fifth*, ii) *contain at least a minimum amount of leaps of a second* (50% in the current implementation) and iii) *each note pitch should be different than the preceding one*. Three objectives are used to compose the fitness functions: a melody should i) *approach and follow big leaps (larger than a second) in a counter step-wise motion (explained below)*, ii) where the melody presents big leaps the *leap notes should belong to the underlying chord* and finally iii) *the first note played on a chord should be part of the underlying chord*.

When dealing with constrained optimization problems, the approach is usually to introduce penalty functions to act as constraints. Such an approach strongly favors feasible solutions over the infeasible ones, potentially removing infeasible individuals that might lead to an better solutions. There have been many examples of constrained multi-objective optimization algorithms [5, 9, 11, 12]. MetaCompose's approach to melody generation uses a combination of the Feasible/Infeasible two-population method (FI-2POP [13]) and NSGA-II [8] dubbed Non-dominated Sorting Feasible-Infeasible 2 Populations (NSFI-2POP [32]). This approach combines the benefits of maintaining an infeasible population, which is free to explore the solution space without being dominated by the objective fitness function(s), and finding the Pareto optimal solution in the presence of multiple objectives. The algorithm takes the structure of FI-2POP, but the objective function of the feasible function is substituted with the NSGA-II algorithm.

4 CHECKERS

Checkers (*Draughts*, in British English) is a family of strategy board games for two players. The invariant characteristics of any Checkers-game variant are: pieces are uniform (they all display the same game-play rules): movement is strictly diagonal; captures are achieved by jumping over opponent pieces; and captures are mandatory. There are many Checkers-variants, which usually change the board-size (such as the Polish-draughts, which plays on a 10×10 board) or variants based on capture constraints, but variations can also include changes to the core rules of the game (in Italian-draughts *pedine* (*men*) pieces cannot capture *kings*). The specific version we use in this study is the *American-checkers* (or *Straight-checkers*). This version is played on an 8x8 checkered board with 12 pieces per side. The objective of the game is to remove all enemy pieces from the game. There are two types of pieces:

Men: initially all the pieces on the board are "men" (uncrowned pieces). These pieces can move one step forward diagonally or, in the event there is an empty space behind an enemy piece, they can perform a capture by "jumping" the opposition.It is also possible for the man to chain multiple enemy-captures if capture constraints are satisfied.

Kings: Whenever a *man* ((uncrowned piece) reaches the enemy end of the board (*the King's row*), it becomes a *king*. In physical games this is marked by the addition of an further *man* token on top of the promoted one, in software a crown sprite can added to the piece, as in our experiment. *King* pieces obtain the ability of moving and capturing backwards, they also have the ability to chain captures.

AI for Checkers

Checkers was one of the first games to which AI was applied, the earliest program was developed in 1951 by Christopher Strachey. Since the 1990s Schaeffer's *Chinook* program has been the strongest Checker-AI, performing at the highest human-levels. In 1996, it won the U.S. National Tournament by an incredible margin [27]. Compared to other games such as Go, Checkers is relatively simple, with its $\approx 10^{20}$ possible positions and game-tree complexity of about $\approx 10^{40}$. It is therefore not surprising that in 2007 Schaeffer *et al.* wrote that "Checkers is solved" [26], declaring the game to have been (weakly) solved from a computational point of view, namely given both players never make mistakes, choosing the best possible moves available to them, the game will always end in a draw.

Another notable research work on Checkers was done by Arthur Samuel, which in his 1959 seminal paper provided arguably the world's first reinforcement learning agent [25]. Moreover to battle the memory restrictions of the time he designed what is now referred to as *alpha-beta pruning*.

5 EXPERIMENT DESIGN

The objective of this study is to explore if any differences in the enjoyment and perception of Checkers game-play can be observed through different configurations of affective music produced by MetaCompose.

An experiment was devised where participants would play two games of checkers, while listening to two (out of three) different setups of generated accompanying music. During game-play players were asked to wear an E4 Wristband, this allows various physiological measures such as: Blood Volume Pulse (BVP), galvanic skin resistance (GSR), and peripheral skin temperature to be recorded The three possible experimental setups were:

- Static expression: MetaCompose rendered the accompanying music piece without change in affective expression throughout the game-play. This acts as a control group to find if differences arise between a static and dynamic music accompaniment.
- Consistent affective expression: at the start of the player's turn, an evaluation of the game-state is made and representative values of valence and arousal are passed to MetaCompose which then returns a suitable music track as accompaniment. In this way the music reflects the state of the game-play.
- Random affective expression: at the start of the player's turn, random values for valence and arousal are passed to MetaCompose. This acts as a control group (like group 1), but allows us to differentiate between differences due to random and guided dynamics.

Afterwards participants were tasked with answering four comparative questions regarding the two games. Martinez and Yannakakis [16] suggest that ranking produces more consistent and reliable data when annotating affect information, participants are therefore asked to compare two pieces of music and rank them. The questions are:

- **Which game did you find more engaging?**
  "The first one"/"The second one"/"Neither"/ "Both Equally"
- **In which game was the music best?**
  "The first one"/"The second one"/"Neither"/ "Both Equally"
- **In which game did the music better match how exciting the game was?**
  "The first one"/"The second one"/"Neither"/ "Both Equally"
- **In which game did the music better match how well you were playing?**
  "The first one"/"The second one"/"Neither"/ "Both Equally"

We also include the more neutral answers "Neither" and "Both Equally" to avoid randomness in the data from participants who cannot decide which clip satisfies the evaluation criterion better or worse.

A survey was prepared with HTML and PHP, using a MySQL database to hold the data collected. The Checkers-framework used is an open-source AI framework called *raven-checkers*[1] which includes provision of a computer game-playing agent. The PHP code externally invokes the Checkers-framework through the *exec()* function, which effectively stops the execution of the PHP code until the game terminates. The experiment was designed for the participants to play two games of Checkers with 2 randomly chosen set-ups (repetitions of the same set-up were not allowed). As each game can take between 5 and 10 minutes, the experiment was designed to last between 10 and 20 minutes for each participant.

6 EVALUATION OF THE GAME STATE

MetaCompose generates affective expression music based on two parameters, valence and arousal [32]. To pass MetaCompose the valence/arousal coordinates, the state of play for the human player needs to be evaluate so that the affective music generated reflects his/her game-state. We decided to derive the valence value from a 'utility-value': meant to describe how good the current board configuration is for the human-player, while arousal is based on the possible moves that the human-player can take (reflecting the sentiment "how much is at stake for the next move?"). In this section we will describe in detail how these calculations are made.

The utility-value used to describe valence is identical to the metric used by *raven-checkers* to evaluate the game state. It is derived from various features of the current game-state: *Piece count, Cramp*[2]*, Back-rank Guard, Double-Corner, Centre-positions, Edge-positions, and Tempo*. For a more detailed explanation of these features we reference the reader to the *raven-checkers* github page.

To calculate arousal we have decided to use, as a measure of "tension", how many moves the player can take and how much these can change the course of the game. For example, if the player can make five moves but all are of little consequence to the game-play, the arousal expressed in the music will be low, while if there are moves that can improve/worsen the game-play situation, the music will reflect this by becoming more stressful. This is implemented by:

(1) Calculate all possible moves m from the current game-state s.
(2) For each resulting game-state $s_i = T(s, m)$ (where T is the transition function that returns a new board state given an initial state and a move), calculate the best move a that the adversary can take, and save the utility value of state $s_i i = T(s_i, a)$.
(3) Once we have calculated the worst situation achieved with every possible move, we calculate the standard deviation of such values. This gives us a measure of how much the game could change from the current game board, which we then use as input to MetaCompose's arousal parameter.

[1]https://github.com/bcorfman/raven-checkers

[2]Checkers a "cramp" is a restriction of mobility in a region of the board

7 MUSIC GENERATION

The music in this experiment is not generated beforehand, instead the music is generated in real-time by MetaCompose. For each participant, MetaCompose creates a single composition and uses that as the basis for the music generated in both of the player's play-through of both Checkers set-ups. In this way we ensure no difference in the player's response due to a potential quality difference between two compositions, the baseline accompaniment is identical.

8 RESULTS AND ANALYSIS

The data collected corresponds to 29 self-reported comparisons and 34 recordings of physiological measurements, respectively from 29 and 17 participants. We have only gathered physiological measurements from 17 out of the 29 participants due to an issue in the recording software. The analysis of the physiological data is not included in this paper as it didn't produce any conclusive results.Recall that each participant was presented with a randomized selection of 2 distinct experimental set-ups (from the three described in Section 5).

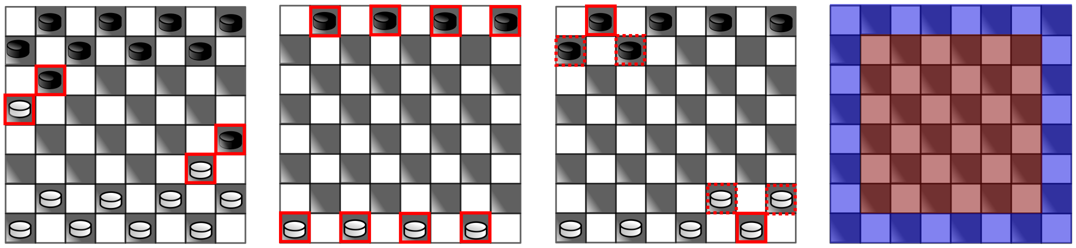
Demographics

Out of the 29 experiment participants, 19 are males, 9 females, and 1 participant did not express gender. The participants' age has an average of ≈ 28.9 years ($stdev \approx 5.9$). In regards to the other demographic features, expressed in 5-point Likert scale (0–4), most people self-reported little experience with the game of Checkers ($avg = 0.89$, $stdev 0.87$, $mode = 1$), and a considerable experience with computer video-games ($avg = 2.68$, $stdev \approx 1.05$, $mode = 2$). No matter how we divide the population, the results are not significantly different, possibly because of the limited number of participants and their relative homogeneity.

Self-report

This section will discuss the results obtained through the survey portion of the experiment. As shorthand we will refer as the criteria the participants used to evaluate the games they played using the labels: *engage*, *best*, *exciting*, *well*. Refer to Section 5 for the complete text of the questions.

For now only consider definitive answers are considered (i.e. the participant chooses one of the music clips presented);



(a) (b) (c) (d)

Figure 3: Visualisations of some of the metrics used to judge the state of the board: (a) examples of cramp, (b) the back-rank guard, (c) the double-corner configuration, and (d) the two positioning areas (centre and edge).

Table 1: Participants' answers to our criteria when comparing the consistent and random set-ups. Also shown are the p-values, calculated using a two-tailed binomial test, and the Binomial Effect Size Display.

	Engage	Best	Exciting	Well
Preferred consistent	6	7	7	5
zP Preferred random	1	0	2	3
No preference	3	3	1	2
Total succ+fail	7	7	9	8
Binomial test	5.47E-02	**7.81E-03**	7.03E-02	2.19E-01
BESD	71.40%	100%	55.60%	25%

Table 2: Participants' answers to our criteria when comparing the consistent and static set-ups. Also shown are the p-values, calculated using a two-tailed binomial test, and the Binomial Effect Size Display.

	Engage	Best	Exciting	Well
Preferred consistent	8	7	6	3
Preferred static	1	1	1	1
No preference	1	2	3	6
Total succ+fail	9	8	7	4
Binomial test	**1.76E-02**	**3.13E-02**	5.47E-02	2.50E-01
BESD	77.80%	75%	71.40%	50%

we will look at the impact of the neutral answers at the end of this section. Under the definite choice constraint, the data becomes Boolean: the answers are either *"user preferred the first set-up"* or *"user preferred the second set-up"*. To analyse this data a two-tailed binomial test is used, with as null hypothesis that both categories are equally likely to occur and, as we have only two possible outcomes, that probability is 0.5. The Binomial Effect Size Display (BESD) [24] is another way of looking on the effects of treatments by considering the increase of success through interventions. This is an interesting measure, as it elucidates how much of an effect is created, in our case, by the different set-ups.

Consistent versus Random expression changes

From our test subjects, ten were shown the comparison between the consistent and static set-ups. As can be seen in Table 1, a strong statistical significance can only be observed for the *best* criterion. If we had adopted less stringent criteria and used p-value cut-offs of .05 or .1, we would have seen significant effects also for the *engage* ($\approx$ .05) and *exciting* ($\approx$ .07) criteria. Thus the null hypothesis can be refuted (at least for the *best* criterion) and a difference in distribution can be inferred between preferring the set-up with consistent affective expression compared to the random expression set-up. This shows how the *consistent* set-up is perceived as possessing a better overall music quality, and hints at leading to better engagement and better expression of excitement in the game. The final criterion (*well*) gives us more inconsistent results. In the next sections we will see how that seems to be a consistent behaviour and we will discuss at the end of the section why this could be such a difficult criterion to evaluate. The BESD values reflect what can be inferred from the p-values, especially highlighting how, while we do not have strongly significant p-values, we see a high increase in successes in the *engage* and *best* criteria.

Consistent versus Static expression

Ten participants were shown the comparison between the consistent and static set-ups. As can be seen in Table 2, we can observe statistically significant results for the *engage* and *best* criteria. Moreover we can observe some relatively low p-values for the *exciting*($\approx$ .05) criterion. As such, it can be observed that, compared with the static set-up, the *consistent* set-up seems to be perceived as: providing better engagement, having higher quality, and hints at better expressing in-game excitement. These results seem to reflect those reported in the previous section, showing how the consistent experimental set-up seems to be better perceived than the other two game set-ups. We refer the reader to Section 9 for more discussion on the *well* criterion. The BESD values highlight that there is a significant increase in preference for the *engage*, *best*, and *exciting* criteria when using the

Table 3: Participants' answers to our criteria when comparing the random and static set-ups. Also shown are the *p*-values, calculated using a two-tailed binomial test, and the Binomial Effect Size Display.

	Engage	Best	Exciting	Well
Preferred random	2	2	6	1
Preferred static	5	4	2	1
No preference	2	3	1	7
Total succ+fail	7	6	8	2
Binomial test	1.64E-01	2.34E-01	1.09E-01	5.00E-01
BESD	-42.90%	-33%	50.00%	0.00%

consistent set-up, although the calculated p-values for the *exciting* criterion is not strictly significant.

Random versus Static expression

From our participants, nine were shown the comparison between the consistent and static set-ups. Table 3 shows no *p*-values with strong statistical significance for any criteria. While the quantity of data collected at the time of this writing is not sufficient to draw reliable conclusions, they can help us formulate hypotheses about how the games are differently perceived in these two set-ups. It seems that static music seems to be considered more engaging and – to a lesser degree – have overall better quality than music with random changes in affective expression. Conversely, it appears that random changes in affective expression are still perceived as more supportive to the excitement of the game. Intuitively we can imagine how participants could have found the static music less disruptive than the accompaniment presenting random expressive changes (hence better scores for *engage* and *best*). At the same time, it is likely that at times the random changes in expression might have matched (at least in some sense) the current state of the game or the excitement perceived by the player, thus leading to a better score for the random set-up in the *exciting* criterion. While there is less evidence than the results discussed in the previous sections, these hypotheses are corroborated by the BESD values. Nonetheless, with the current data at our disposal we have to consider these two set-ups as being relatively equivalent.

9 DISCUSSION

The main questions explored in this study are: can any difference in player experience be observed when presented with affective-dynamic music compared to static music, and can any difference be observed when the music is supporting the game's internal narrative/state. An experiment was designed where participants were tasked with playing two games of Checkers out of three different experimental set-ups.

The self-reporting task of the experiment showed how music with affect expression consistent with the game-state appears to be better perceived than the other two set-ups. In particular, it was observed that it is perceived as having better overall quality, leading to a more engaging experience, and – to a lesser degree – better matching the perceived excitement in the game. The static and the random expression set-up appear to be more equivalent, although we can observe some non-significant differences between the two: the static set-up seems to be generally better perceived (more engaging and overall better quality), while the random set-up seems to better match the perceived excitement of the game. We hypothesize that the random set-up has too many disruptive changes in expression to be particularly liked by the listener, while the static one by definition does not present any changes that might match game-play, leading to lower ratings in regards to excitement criterion. When looking at the answers for the last criterion ("in which game did the music better match how well you were playing?") we find inconsistent results between each of the groups. This may be caused by the complexity of the question, which requires the participant to also evaluate her own game-play performance in the two games.

A limitation we need to address is that, while it seems reasonable, there is no assurance that the estimation of valence/arousal (see Section 6) is necessarily correct in expressing the game-state. A preliminary study could have been designed to validate this state-evaluation, for example by asking expert checkers players to give us an evaluation of specific states and comparing it to our measurements.

The reader might have been wondering about the choice of the game of Checkers in this study: why choosing this classical board-game compared to other games? We thought Checkers would provide a good first use-case of MetaCompose thanks to its extremely minimal narrative and aesthetics; in fact, when comparing it to similar games like Chess, Checkers only has two types of pieces and even these pieces have weaker narrative-intrinsic names: men and kings (in some languages "ladies"). A natural next step in exploring the effect of the music generated by MetaCompose on player experience would be to apply it to an opposite case: a so-called adventure game (the genre is also referred to as 'point-and-click', from the interaction mode used by classical examples in this genre). We plan to conduct an analogue experiment using the adventure game described in the paper *Evaluating musical foreshadowing of video-game narrative experiences* [29]. This experiment would focus on observing how differently people would perceive the game based on three experimental settings: 1) the background music reflects and supports the in-game narrative, 2) the changes in mood expression happen randomly while during game-play, and 3) the music never changes expression. An additional set-up that would also be interesting to observe is one presenting the user with dynamic and adaptive music, but contrasting the narrative of the game. This study would not only give us invaluable insight in the effect of music on narrative perception, but coupled with the Checkers experiment would create the basis for analyzing the effect of affective expressive music in more complex games.

To conclude, via this experiment we found that participants self-reportedly preferred the dynamic affective music provided by METACOMPOSE when trying to reflect the current game-state in three out of four criteria. This result is especially significant for the perceived quality of the music.

REFERENCES

[1] Steven Abrams, Daniel V Oppenheim, Don Pazel, James Wright, et al. 1999. Higher-level composition control in music sketcher: Modifiers and smart harmony. In *Proceedings of the ICMC*. Citeseer.

[2] John Biles. 1994. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, 131–131.

[3] 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*. 99–104.

[4] Daniel Brown. 2012. Mezzo: An Adaptive, Real-Time Composition Program for Game Soundtracks. In *Proceedings of the AIIDE 2012 Workshop on Musical Metacreation*. 68–72.

[5] Deepti Chafekar, Jiang Xuan, and Khaled Rasheed. 2003. Constrained multi-objective optimization using steady state genetic algorithms. In *Genetic and Evolutionary ComputationâĂŤGECCO 2003*. Springer, 813–824.

[6] Heather Chan and Dan A Ventura. 2008. Automatic composition of themed mood pieces. (2008).

[7] Palle Dahlstedt. 2007. Autonomous evolution of complete piano pieces and performances. In *Proceedings of Music AL Workshop*. Citeseer.

[8] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.

[9] Kalyanmoy Deb, Amrit Pratap, and T Meyarivan. 2001. Constrained test problems for multi-objective evolutionary optimization. In *Evolutionary Multi-Criterion Optimization*. Springer, 284–298.

[10] Mirjam Eladhari, Rik Nieuwdorp, and Mikael Fridenfalk. 2006. The soundtrack of your mind: mind music-adaptive audio for game characters. In *Proceedings of Advances in Computer Entertainment Technology*.

[11] Amitay Isaacs, Tapabrata Ray, and Warren Smith. 2008. Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. In *IEEE Congress on Evolutionary Computation*. IEEE, 2780–2787.

[12] Fernando Jimenez, Antonio F Gómez-Skarmeta, Gracia Sánchez, and Kalyanmoy Deb. 2002. An evolutionary algorithm for constrained multi-objective optimization. In *Proceedings of the Congress on Evolutionary Computation*. IEEE, 1133–1138.

[13] Steven Orla Kimbrough, Gary J Koehler, Ming Lu, and David Harlan Wood. 2008. On a Feasible–Infeasible Two-Population (FI-2Pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *Eur. J. Operational Research* 190, 2 (2008), 310–327.

[14] Steven R Livingstone and Andrew R Brown. 2005. Dynamic response: Real-time adaptation for music emotion. In *Proceedings of the 2nd Australasian Conference on Interactive Entertainment*. 105–111.

[15] Roisin Loughran, James McDermott, and Michael O'Neill. 2015. Tonality driven piano compositions with grammatical evolution. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2168–2175.

[16] Hector P Martinez, Georgios N Yannakakis, and John Hallam. 2014. DonâĂŹt classify ratings of affect; rank them! *Affective Computing, IEEE Transactions on* 5, 3 (2014), 314–326.

[17] Sidney K Meier and Jeffrey L Briggs. 1996. System for real-time music composition and synthesis. US Patent 5,496,962.

[18] Eduardo Reck Miranda. 2013. *Readings in music and artificial intelligence*. Vol. 20. Routledge.

[19] Eduardo Reck Miranda and Al Biles. 2007. *Evolutionary computer music*. Springer.

[20] Kristine Monteith, Tony Martinez, and Dan Ventura. 2010. Automatic generation of music for inducing emotive response. In *Proceedings of the International Conference on Computational Creativity*. Citeseer, 140–149.

[21] Miller Puckette et al. 1996. Pure Data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts* (1996), 37–41.

[22] Alexander P Rigopulos and Eran B Egozy. 1997. Real-time music creation system. US Patent 5,627,335.

[23] Judy Robertson, Andrew de Quincey, Tom Stapleford, and Geraint Wiggins. 1998. Real-time music generation for a virtual environment. In *Proceedings of ECAI-98 Workshop on AI/Alife and Entertainment*. Citeseer.

[24] Robert Rosenthal and Donald B Rubin. 1982. A simple, general purpose display of magnitude of experimental effect. *Journal of educational psychology* 74, 2 (1982), 166.

[25] Arthur L Samuel. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development* 3, 3 (1959), 210–229.

[26] Jonathan Schaeffer, Neil Burch, Yngvi Björnsson, Akihiro Kishimoto, Martin Müller, Robert Lake, Paul Lu, and Steve Sutphen. 2007. Checkers is solved. *science* 317, 5844 (2007), 1518–1522.

[27] Jonathan Schaeffer, Robert Lake, Paul Lu, and Martin Bryant. 1996. CHINOOK the world man-machine checkers champion. *AI Magazine* 17, 1 (1996), 21.

[28] Marco Scirea. 2013. Mood Dependent Music Generator. In *Proceedings of Advances in Computer Entertainment*. 626–629.

[29] Marco Scirea, Yun-Gyung Cheong, Byung Chull Bae, and Mark Nelson. 2014. Evaluating musical foreshadowing of videogame narrative experiences. In *Proceedings of Audio Mostly 2014*.

[30] 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*. ACM, 211–218.

[31] Marco Scirea, Mark J Nelson, and Julian Togelius. 2015. Moody Music Generator: Characterising Control Parameters Using Crowdsourcing. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer, 200–211.

[32] Marco Scirea, Julian Togelius, Peter Eklund, and Sebastian Risi. 2016. MetaCompose: A Compositional Evolutionary Music Composer. In *International Conference on Evolutionary and Biologically Inspired Music and Art*. Springer, 202–217.

[33] 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.

[34] Alan Smaill, Geraint Wiggins, and Mitch Harris. 1993. Hierarchical music representation for composition and analysis. *Computers and the Humanities* 27, 1 (1993), 7–17.

[35] Geraint Wiggins, Mitch Harris, and Alan Smaill. 1990. *Representing music for analysis and composition*. University of Edinburgh, Department of Artificial Intelligence.

[36] Rene Wooller, Andrew R Brown, Eduardo Miranda, Joachim Diederich, and Rodney Berry. 2005. A framework for comparison of process in algorithmic music systems. In *Generative Arts Practice 2005 — A Creativity & Cognition Symposium*.