

# Procedural Generation of Music-Guided Weapons

William Cachia, Luke Aquilina, Héctor P. Martínez and Georgios N. Yannakakis

Institute of Digital Games

University of Malta

Msida, Malta MSD 2080

{william.cachia.07, luke.aquilina.10, hector.p.martinez, georgios.yannakakis}@um.edu.mt

**Abstract**—**Beyond the standard use of music as a passive and, sometimes, optional component of player experience the impact of music as a guide for the procedural generation of game content has not been explored yet.** Being a core elicitor of player experience music can be used to drive the generation of personalized game content for a particular musical theme, song or sound effect being played during the game. In this paper we introduce a proof-of-concept game demonstrator exploring the relationship between music and visual game content across different playing behaviors and styles. For that purpose, we created a side-scroller shooter game where players can affect the relationship between projectiles' trajectories and the background music through interactive evolution. By coupling neuroevolution of augmented topologies with interactive evolution we are able to create an initial arsenal of innovative weapons. Those weapons are both interesting to play with and also create novel fusions of visual and musical aesthetics.

## I. INTRODUCTION

Music is generally considered a fundamental component for a satisfactory gameplay experience and one of the core facets of *computational game creativity* [4]. Following practices in film-making, level designers have used music in various ways to elicit or amplify a mood in certain parts of the game. As the interaction in games becomes more complex, studies have attempted to augment the role of music in games by creating personalized music for different players (e.g. [5]). Moreover, using music not only as a complement to player experience but as a core gameplay mechanic has led to the creation of highly engaging experiences — e.g. in the *Guitar Hero* (Harmonix, 2005) and *Pulse* (Cipher Prime, 2011) games. Within this genre, some games have already explored the automatic creation of levels given a song or music library such as *Audiosurf* (Dylan Fitterer, 2008), *Audioverdrive* [2], *Beattthebeat* [3] and *Symphony* (Empty Clip Studios, 2012)). In these games, the designer has fixed a mapping between the characteristics of a music segment and the behavior of particular game objects. However, as the experience of music is subjective, we believe that allowing players to participate in the definition of such a mapping would generate a more personalized experience and, thereby, more engaging games.

In this paper we present a procedural game content generator that is built on interactive evolution [1] allowing players to influence the relation between the music and the behavior of game objects. In particular, we present a demonstrator that implements this system in a side-scroller shooting game in which the representation and trajectories of different weapons are orchestrated by the background music. Preliminary evaluations suggest that not only players create different representations for the same music but, most importantly, that the evolution

process itself produces an engaging gameplay mechanic.

## II. THE INFINITE SIDE-SCROLLER SHOOTING GAME

In this demonstrator we use an infinite side-scroller shooting game (see Fig. 1). The player controls the movement of a spaceship displayed from the side. As the game progresses, the player has to avoid or shoot down incoming obstacles and enemies to survive. On its basic variant, the spaceship shoots individual white projectiles that follow a straight path until they disappear off the screen or collide with an obstacle or enemy. When music is added to the game, the color and trajectory of each individual projectile is directed by the music played currently. During the creation of the weapons, players are offered an arsenal of different weapons that display various behaviors dependent on the background music.

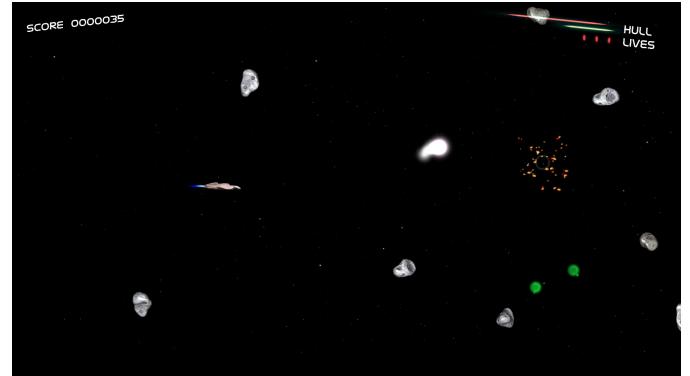


Fig. 1. A screenshot of the infinite side-scroller shooting game.

## III. MUSIC-DRIVEN WEAPON GENERATION

In order to allow a wide variety of mappings between music and the behavior of game objects we use artificial neural networks (ANNs) as they can represent any continuous function. Inspired by [1] we use the following ANN inputs (normalized to [0,1]):

- The average intensity of six different music frequency bands extracted using a Fourier transformation with a Blackman-Harris window: 20Hz - 80Hz (Low Bass); 80Hz - 320Hz (Hi Bass); 320Hz - 1280Hz (Midrange); 1280Hz - 5120Hz (High Midrange); 5120Hz - 20840Hz (High Treble).
- The distance between the projectile and the spaceship.
- The time since the projectile was fired.

- The x and y coordinates of the projectile (relative to the spaceship).

The ANN outputs the frame-to-frame movement ( $\delta x$  and  $\delta y$ ) and the color of the projectile (represented as RGB values). In every frame, the position and color of every projectile on the screen is updated according to the ANN output.

The ANNs are trained using neuroevolution of augmented topologies (NEAT) [7] with an additional mutation operator introducing activation functions commonly used in compositional pattern producing networks [6]. The fitness function is directly linked to the usage of each weapon and it is obtained *implicitly* through interactive evolution: i.e. every time a weapon of the arsenal is used, the fitness function of the associated ANN is incremented by 1 while the fitness of the remaining ANNs are lowered by 0.5.

#### IV. PRELIMINARY EVALUATION

Preliminary tests with this demonstrator showed that players would tend to evolve (after a large number of generations) weapons with a low dependency with respect to music; these weapons fire projectiles that follow simple trajectories easy to predict. Figure 2 shows two weapons evolved during this evaluation. The *opposite shots* weapon fires projectiles that follow a south-west trajectory that is reversed when a particular beat/drum is heard. The *wave shots* weapon creates a west-east trajectory that oscillates when a lot of bass is heard.

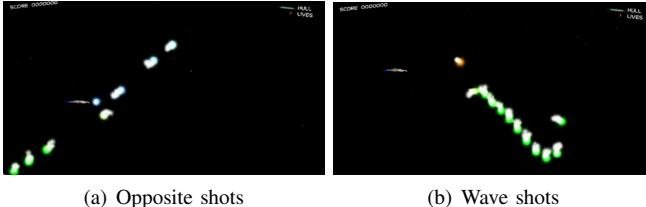


Fig. 2. Weapon patterns with low music dependency

In order to force stronger connections between music and projectile trajectory, especially during early generations of weapon evolution, we normalized the music ANN inputs so that to induce a higher effect of the frequency bands on the trajectory over the other inputs of the network. This experiment resulted in more exaggerated weapons but interesting nonetheless. For example, the weapons depicted in Fig. 3 fire projectiles that jump to their next position whenever bass is heard or else they shake from one side to another with the sound of an acoustic string.

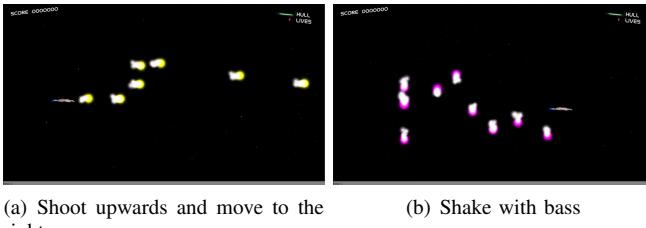


Fig. 3. Weapon patterns with high music dependency

As the screenshots depicted in Fig. 2 and Fig. 3 can only show snapshots of the generated weapons we refer the reader to the corresponding videos<sup>1</sup>.

#### V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a proof-of-concept of an interactive evolution approach with the aim to create personalized game content directly linked to music. The paper introduced a possible orchestration between two facets of computational game creativity [4] and their framing within a game: music and visuals. For that purpose we evolved NEAT mappings between projectile trajectories and certain basic characteristics of the background music, and allowed players to influence those mappings implicitly by using them. Future steps in this study will formally evaluate the differences in weapons evolved by independent users in order to study the relationship between subjective perception of music and its preferred visual representation. Furthermore, we will evaluate whether the preference for one weapon over others holds across songs and music genres.

As we represented the mapping between music and projectile trajectories through a neural network, any continuous function can potentially be evolved yielding an infinite number of weapons. Nevertheless, the current mapping is limited by the inputs selected; thus, in future work we will explore a wider space of possible music representations that can allow more perceptive users to define finer-grained, and potentially more interesting, connections between music and visuals.

#### ACKNOWLEDGMENTS

Thanks to Ryan Abela for his input on designing the sound extraction methods. The research was supported, in part, by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665).

#### REFERENCES

- [1] Erin J. Hastings, Ratan K. Guha, and Kenneth O. Stanley. Evolving content in the galactic arms race video game. In *Proceedings of the 5th International Conference on Computational Intelligence and Games*, CIG'09, pages 241–248, Piscataway, NJ, USA, 2009. IEEE Press.
- [2] Nils Iver Holtar, Mark J. Nelson, and Julian Togelius. Audiooverdrive: Exploring bidirectional communication between music and gameplay. In *Proceedings of the 2013 International Computer Music Conference*, 2013.
- [3] Annika Jordan, Dimitri Scheftelowitsch, Jan Lahni, Jannic Hartwecker, Matthias Kuchem, Mirko Walter-Huber, Nils Vortmeier, T Delbrugger, U Guler, Igor Vatolkin, et al. Beatthebeat music-based procedural content generation in a mobile game. In *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*, pages 320–327. IEEE, 2012.
- [4] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Computational game creativity. In *Proceedings of the Fifth International Conference on Computational Creativity*, 2014.
- [5] David Plans and Davide Morelli. Experience-driven procedural music generation for games. *IEEE Trans. Comput. Intellig. and AI in Games*, 4(3):192–198, 2012.
- [6] Kenneth O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.
- [7] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99–127, June 2002.

<sup>1</sup><http://goo.gl/T2kkby>