AudioInSpace: Exploring the Creative Fusion of Generative Audio, Visuals and Gameplay

Amy K. Hoover, William Cachia, Antonios Liapis, and Georgios N. Yannakakis

Institute of Digital Games
University of Malta
Msida, Malta MSD 2080
amy.hoover@gmail.com, william.cachia.07@um.edu.mt, antonios.liapis@um.edu.mt,
georgios.yannakakis@um.edu.mt

Abstract. Computer games are unique creativity domains in that they elegantly fuse several facets of creative work including visuals, narrative, music, architecture and design. While the exploration of possibilities across facets of creativity offers a more realistic approach to the game design process, most existing autonomous (or semi-autonomous) game content generators focus on the mere generation of single domains (creativity facets) in games. Motivated by the sparse literature on multifaceted game content generation, this paper introduces a multifaceted procedural content generation (PCG) approach that is based on the interactive evolution of multiple artificial neural networks that orchestrate the generation of visuals, audio and gameplay. The approach is evaluated on a spaceship shooter game. The generated artifacts — a fusion of audio-visual and gameplay elements — showcase the capacity of multifaceted PCG and its evident potential for computational game creativity.

1   Introduction

Computer games are a creative outlet for players tasked with solving problems and designers and artists who create a game's multiple facets (e.g. audio, visuals, gameplay, narrative, level architecture and game design) [1]. Often both the creative expression of video game designers and the players' immersion in the game are significantly impacted by the interplay between audio facets (e.g. character themes, sound effects, foreshadowing) and visuals facets (e.g. object and level design) [1]. Historically in popular video games like Super Mario Bros. [2] and Sonic the Hedgehog [3], game composers and developers interweave these facets by assuming that a player will *linearly* progress through known game states. However as players are increasingly drawn toward dynamically constructing their own personalized and replayable game experiences (i.e. *nonlinear* games), game composers and designers are challenged to maintain meaningful interaction between these audio and visual game facets [4].

Procedural content generation (PCG) methods can alleviate some of the demands on game designers, developers, and artists by fully automating the process of creating game facets or within computer-aided tools designed to assist

the human creator [5]. With a few notable exceptions [6, 7], until recently these PCG methods have focused on generating single game facets (e.g. visuals, levels) that are then incorporated into the standard game development pipeline [8].

Instead, this paper introduces a *multifaceted* PCG approach that simultaneously generates audio, visuals, and gameplay facets. Represented by a special type of artificial neural network called a compositional pattern producing network (CPPN) [9], the audio and visual facets are generated with respect to each other (i.e. audio generation considers the currently displayed visuals and vice versa). This paper explores the idea through an extension to a space shooting game introduced by Cachia et al. [10] called AudioInSpace. Players control a spaceship in this test bed game and progress through levels while shooting at enemies and avoiding obstacles. Through *implicit* interactive evolutionary computation (IEC) [11] of the audio and visual CPPNs, over the course of the game players shape their experience by shooting weapons with the most desirable trajectories and colors, and evolving the audio to suit their sound preferences and bullet trajectory needs. The results are personalized weapons and soundtracks created through *mixed-initiative co-creativity*, where the machine mediates the bidirectional relationships between the audio and the visuals while the human player controls the aesthetic outcome via the firing of weapons.

2 Background

While there are many approaches to procedurally generating video game facets (e.g. visuals, audio, narrative, game design, level design and gameplay), the focus is often on a single facet rather than how generated facets combine for a complete game experience. For instance, world maps in *Civilization V* [12], dungeons in *Diablo* [13], racing tracks [14], petal colors and shapes in Petalz [15], and weapons in *Borderlands* [16] are all procedurally generated to provide the player with increased personalization and replayability. Similarly, *Audiosurf* [17] generates game levels from sound files provided by the player. While these generated elements may increase replayability and alleviate authorial burden on developers, these approaches serve as means to an end rather than celebrating the creativity of procedural content generation itself.

Other approaches aim to interweave procedurally generated elements to enhance the game experience. For instance, *Galactic Arms Race* [18] encourages players to interactively evolve weapon visuals and bullet trajectories to their aesthetic and gameplay preferences, while *Game-o-matic* [19] helps players visually construct the topic of their game and then procedurally generates the rules and visuals, challenging players to interpret the designer's intent on topics like religion or politics. *A Rogue Dream* [7] is 2D maze game that gathers and parses internet data to discover relationships between natural language concepts to create visual and gameplay facets. Proteus [20] is a completely procedurally generated pixel-art world with open-ended gameplay encouraging players to spatially and sonically explore a generated island. However, the sonic output does not affect the previously generated landscapes. Even in these integrated envi-

ronments, the output of one procedurally generated facet has little affect on the output of the other.

A notable exception is that of *AudioOverdrive*, a side-scrolling space shooter where players compose music by progressing through the game. It has bidirectional communication between enemies, gameplay, and level visuals [21], but the approach requires heavy human initiative by the developers. The approach in this paper extends ideas from AudioOverdrive by creating a game environment called AudioInSpace with closely coupled audio and visuals that requires minimal hard coding (initiative) from the developers. In this test bed space shooter, the audio, visuals, and gameplay are integrated to generate the appearance of the spaceship's weapons (e.g. color), gameplay effects (e.g. whether it hit an enemy), and the game soundtrack. The soundtrack then influences the firing rate, color and movement patterns of the weapon's particles, thus ensuring a bidirectional communication across three creative facets. While both AudioInSpace and AudioOverdrive explore similar goals, themes, and creative game facets, AudioInSpace specifically uses information on the spaceship's gameplay behavior and on-screen visuals rather than treating gameplay events such as firing a weapon solely on account of its sound effect.

3 Approach

While many PCG approaches focus on creating single game facets to augment the developers' creativity, the approach in this paper is to explore multiple generated game facets that are combined through mixed initiative co-creativity [1], where the machine and the player create and collectively augment the player's gameplay and creative experience. In particular, the machine's creativity is enhanced through a conceptual blending of the audio and visual modules in figure 1 where audio outputs inform the visual module, which in turn is input to the audio module. This model is implemented in a proof-of-concept space shooter game called AudioInSpace (see section 3.1 for details) where players control space ships and fire at enemies. The ways in which the weapons fire (i.e. the bullet trajectories), their shapes, speeds, and colors are affected by the audio, while the audio in turn effects the trajectories. Both audio and visuals are affected by gameplay as players exercise their own creativity through selecting preferred weapons and audio for the soundtrack in AudioInSpace. To co-evolve the individual audio and visual facets, each PCG module bases decisions on information gathered from both domains.

The relationships between the domains and generated outputs are represented by a special type of artificial neural network (ANN) called a compositional pattern producing network (CPPN; shown in figure 2) [9]. Like traditional ANNs, each CPPN is an interconnected network of nodes and connection weights that when provided input, calculate an output value that can theoretically approximate any function[22]. However, unlike traditional ANNs that only compute sigmoid functions at their hidden nodes, CPPNs can compute any function (e.g. Gaussian, sigmoid, sine, ramp, etc.), effectually making them pattern generators
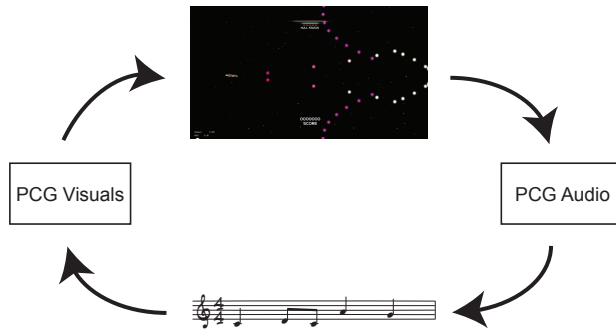
Fig. 1. Mixed-Initiative Co-Creativity with Multiple Facets. In AudioInSpace, weapon visuals and audio are represented by two separate modules that each provide relevant domain information to the other (i.e. the visual module is not only informed by current visual elements in the level, but also by the concurrently generated audio and vice versa). The human initiative in AudioInSpace comes from gameplay and the selection of weapons.
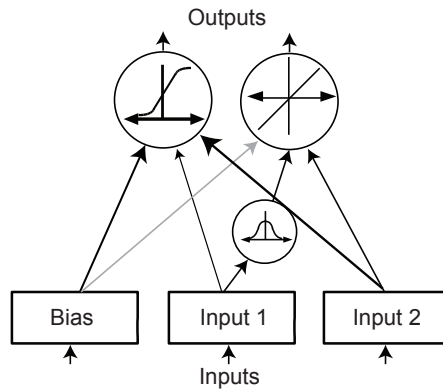
Fig. 2. Compositional Pattern Producing Network (CPPN). This feedforward CPPN is an interconnected network of nodes that take input values (at bottom) from a domain to calculate the output values (at top). Starting from the inputs, at each hidden node (middle) and output node, a value is computed depending on the node's activation function and activation sum, the latter determined by a weighted sum of the values of nodes connected below. Unlike traditional ANNs, a node's activation function can be any function many of which can bias the CPPN toward certain regularities including bilateral symmetry, repetition, and repetition with variation.

biased toward the particular regularities indicated by the hidden node activation functions. Through this representation, the aim is that both weapon visuals and audio are represented as patterns of each other.

Each PCG module is represented by a separate CPPN that inputs domain information and outputs instructions for generating either audio or visual pat-

terns that not only affect player immersion but also gameplay. To personalize the relationship between audio, visuals, and gameplay, these CPPNs can be evolved through a process similar to animal breeding called interactive evolutionary computation (IEC), wherein the human rather than an explicit fitness function determines the fitness of candidate individuals. CPPNs are evolved through the NeuroEvolution of Augmenting Topologies (NEAT; [23]) algorithm that was originally developed to solve control and decision tasks but generates audio and visuals in this paper. Through (NEAT), each CPPN can start minimally and complexify as necessary over evolutionary time. By adding hidden nodes and connections, and changing activation functions and weight values, each new individual can expand the relationship between inputs and hidden nodes. Thus, player selection is an important aspect of this approach and toward facilitating mixed-initiative co-creativity.

3.1 AudioInSpace: The Game

The multifaceted PCG approach is evaluated in a space shooter, called AudioInSpace, which extends approaches by Hastings et al. [18] for weapon particle generation through the integration of an audio component. As the game progresses, the player moves through different space-based game levels encountering obstacles and hostile enemies (as shown in figure 3).

The player encounters large rocks to avoid, and small rocks to avoid and shoot, while implicitly evolving weapon trajectories and aesthetics, and audio based on the amount of time the user has the weapon and audio track equipped. With each bullet fired, a weapon is assigned a preference point while the audio is rated for each new note. The longer each weapon and audio CPPN are in play, the higher that CPPN is rated. It is assumed that the longer weapons and visuals are in play, the more preferable they are to a particular player. Therefore through IEC players evolve the visual and audio networks toward their aesthetic preferences. By directing the two CPPNs to create patterns from information gathered from both domains, this approach incorporates mixed-initiative co-creativity while combining two different PCG modules.

Upon launch, the colors and trajectory of the first bullet are calculated when a random note from the C Major pentatonic scale is played and input to the visual CPPN in figure 4a. Together with the pitch information from the initially random MIDI note, this CPPN also inputs the $(\delta x, \delta y)$ position between where the bullet was fired and where it is currently located with respect to where it was fired, and the time t since firing. Note that before the bullet is fired, the initial $(\delta x, \delta y) = (0, 0)$ and $t = 0$. Outputs of the visual CPPN determine the red, green, and blue (RGB) color values for each bullet, and the (x', y') outputs dictate the bullet's new position (i.e. its trajectory). The color and trajectory of each bullet is calculated every time a new note sounds.

New notes are fired when the bullet hits an enemy or otherwise at the end of the current note's duration. As seen in figure 4b, the music CPPN determines the pitch and the note's duration through the respective Pitch and DeltaTime outputs. These values are based on the (x, y) position of where the last bullet

Fig. 3. AudioInSpace. In AudioInSpace, the user controls the spaceship on the left side of the screen while moving through each space level. The health of the ship is shown at the top of the screen as "hull strength" while the weapon score is displayed at the bottom. Two mirrored visual beams project from the player's spaceship shooting patterns directed by the current visual CPPN. The rate of fire and the pitches heard result from the audio CPPN's outputs.

was fired, time since firing t, whether the bullet struck an enemy h, and its RGB values.

4 Experiments

Experiments in this section illustrate the potential of this multifaceted PCG proof-of-concept by exploring the impact of the audio and visual CPPNs on the game and on each other. In AudioInSpace, each CPPN module is evolved separately by the player. That is, the audio network is held constant while the weapon network is evolved and vice versa. The first set of experiments explores each network, one at a time. To start, three weapon CPPNs are evolved with a constant, simple audio input (i.e. a repeating eighth note at a pitch of MIDI value 72). After exploring the weapon effects with this repeated input, the audio CPPN is evolved for each of the weapon CPPNs. The results of the new and varied MIDI inputs are then explored through the weapon visuals. To further explore these effects, ascending and descending MIDI notes between MIDI values 20 and 90 are also input to the visual CPPN.

The second set of experiments explores the complete interaction between the two networks when evolved by players. In these experiments, a note in the
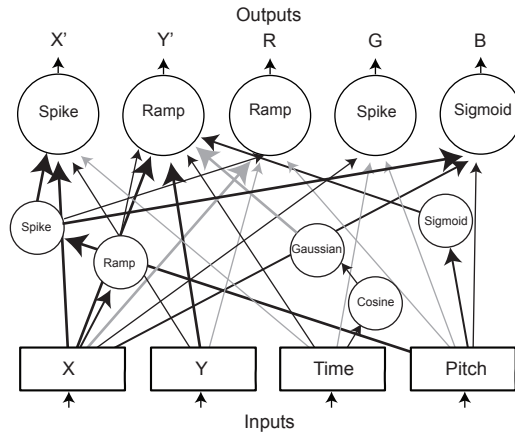
(a) Visuals and Gameplay (Weapon) CPPN
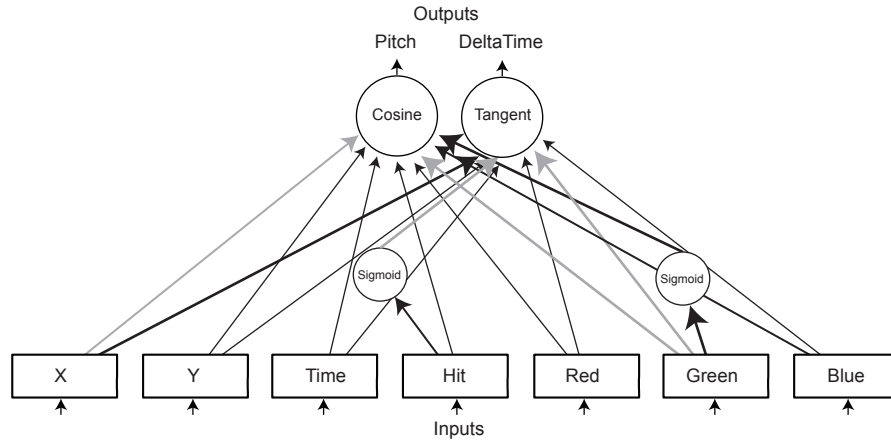
(b) Audio CPPN

Fig. 4. CPPNs for AudioInSpace. Each PCG module is enabled by its own CPPN. Darker connections represent a positive weight while lighter connections are a negative weight. The magnitude of each weight is represented by the line width where thicker lines indicate a higher absolute value. In (a), the inputs to the visual CPPN are the (x, y) coordinates from where the bullet is initially fired, the time since firing, and the current pitch being input from the CPPN in (b); it outputs a new (x', y') coordinate for the next frame and RGB values. On the other hand, (b) inputs the outputs from (a) as well as whether a bullet has just struck an enemy.

pentatonic C major scale is sent to the visual network. The visuals are then generated in loop with the MIDI notes.

In each of the experiments, tournament selection occurs amongst a population of fifty individuals, ten of which are rated by the player through IEC. Players can switch the current weapon or audio representation by pressing a

button to activate a different CPPN chromosome whose effects are immediately heard and seen in AudioInSpace. The evolutionary parameters are set through preliminary testing, and the probability of adding a new node or connection is 30% and 15%, respectively. Similarly, the activation functions at any node can be replaced by one of the following with an even chance: Sigmoid, Hyperbolic Tangent, Sine, Cosine, Bipolar Sigmoid, Gaussian, Ramp, Step, Spike [18]. However, the activation function of any particular node is only changed with a 20% chance. Weights are real values between $[-3, 3]$ that are mutated with a uniform 90% chance. Occasionally (1% chance), connections are disabled.

5 Results

Due to space constraints we will only present a number of indicative audio-visual patterns generated through the experiments conducted. For the interested reader several videos of the results presented in this section (and beyond) are available at http://music-ai.org/musicgame/.

The first set of results explores the combination of visual and audio CPPNs by first examining the weapons with a predetermined MIDI pattern. MIDI note 67, or G, is input to the visual network in eighth note quantization. As shown in figure 5 and accompanying video, visuals generated from this single repeated note create a constant visual stream with a predictable trajectory dictated by the repeated eighth note inputs. Each visual CPPN has four inputs, five outputs, and in this result, which was evolved in fifty generations, also has six hidden nodes as shown previously in figure 4a. As these weapons are patterns created by human evolved CPPNs, many different colors and trajectory shapes are possible. As already noted, several additional examples are available at the website for the reader of this paper.

Once the audio CPPN is evolved by the player for the previously evolved weapon CPPN, the bullets still maintain much of the blue color present in figure 5a, but as shown in figure 5b also start shooting red colored bullets over time. The trajectory from the CPPN in 5(a) results in a predictable repetitive pattern. However, once the MIDI network is evolved in 5(b) the varied audio inputs impact the original path in 5(a). In fact, with this MIDI network, the bullets are fired with a slightly backwards trajectory that at times allows users to build posterior bullet defenses against potential upcoming enemies. In this way, players can simultaneously evolve networks suited for their individual game experiences.

Figure 6(a) shows a weapon CPPN that is evolved with a repeated MIDI input like figure 5a. However, 6(b) shows the same weapon from 6(a) but with MIDI notes that are input in ascending and descending order between 20 and 90. Over time, the overall trajectory shape of 6(a) is maintained, but the length of time between bullets increases and decreases according to the MIDI input. Videos of the complete system interaction are available at http://music-ai.org/musicgame/ where the player cycles through several different weapons and MIDI CPPNs.
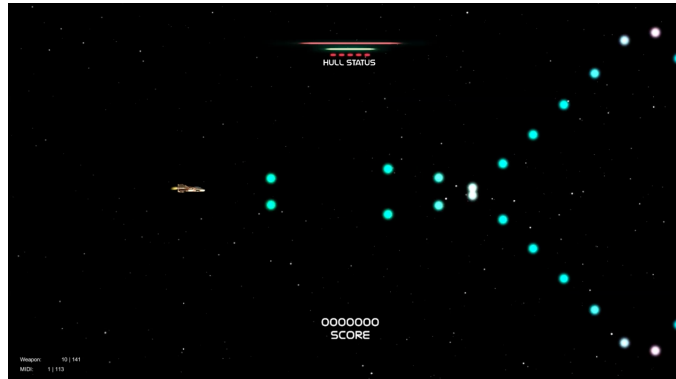
(a) Weapon with Repeated Single Eighth Note
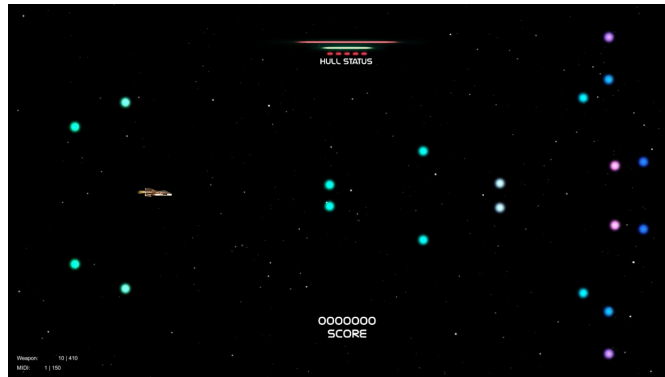
(b) Weapon with Repeated Variable Eighth Note

Fig. 5. Evolved weapon with repeated and variable notes. While the weapon in (a) maintains a steady trajectory due to the repeated eighth note input, the same weapon in (b) deviates in bullet timing, color, and trajectory once the MIDI network is evolved, thereby illustrating the influence of the MIDI CPPN.

6   Discussion and Future Work

While this paper is a proof-of-concept of multifaceted game creativity as influenced by the player (mixed-initiative), it is important to investigate whether players actually enjoy this new mixed-initiative experience. Future work aims to explore and evaluate players' appreciation of the visuals, audio, and gameplay, and whether the mixed-initiative interaction enhances the overall experience. The question is whether players prefer the sound of the additional audio and the control that it provides over the weapon visuals. AudioInSpace can be play tested — with the work by Cachia et al. [10] as a control — examining average game time, perceived enjoyment, generations per audio and visual CPPNs, and complexity of the relationships between inputs and outputs of each CPPN.
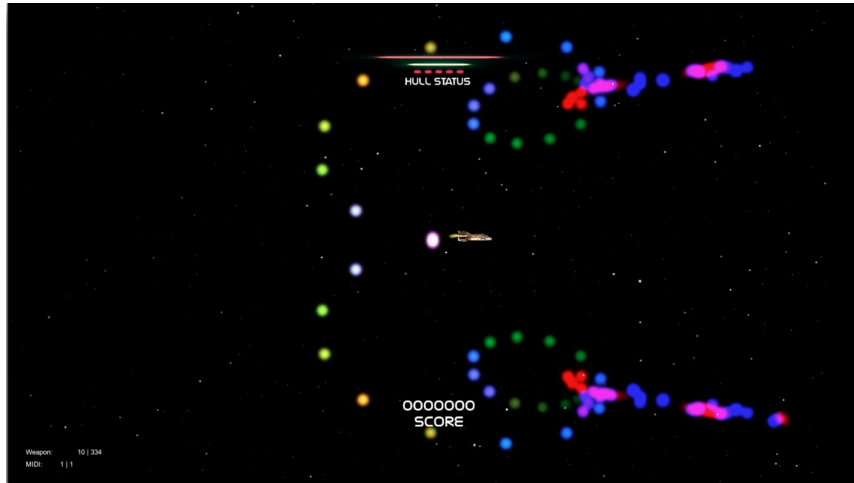
Fig. 6. Ascending and descending pitches When ascending and descending notes between MIDI values 20-90 are sent to this weapon network, the smooth transitions between colors and trajectories are evident. The corresponding video is available at http://music-ai.org/musicgame/.

While it is currently assumed that the number of times a gun is fired and the length of time an audio network is heard reflect player preference, another interesting question is why players choose certain weapons and audio. For example, do players focus more on the look of the weapon or the rate of fire determined by the audio? Answers to these questions could help identify player styles and potential areas for more development.

There are also many different avenues for increasing the quality of the visuals and audio, and the cooperation between the two. Like Galactic Arms Race, for instance, the initial generation of weapon and audio CPPNs could be preselected for quality and gameplay. Furthermore, more constraints on the audio output (e.g. restriction to key) could enhance the sound of the output. Another interesting area of research is in adding "listeners" to generated audio and spawning monsters depending on the pitches and durations of note patterns. For instance, when low-pitched long notes play, a larger more important and difficult boss could appear, whereas higher pitched notes could signal intermittently approaching comets.

Future work includes incorporating more interwoven facets into this existing multifaceted PCG approach to explore the boundaries of the idea across different game types. While an obvious future direction is investigating level architecture facets, the complexity of extending the approach across a variety of games becomes greater with the generation of additional game facets. However, combining additional facets into a meaningful game that provides a positive player experience and potentially enhancing his or her creativity depends on more factors requiring more extensive experimentation.

7 Conclusion

This paper introduced an approach for procedurally generating multiple game facets simultaneously that depends on initiatives of both computational and human creators (mixed initiative co-creativity). Through a space shooter game called AudioInSpace, players evolve their own weapons and audio to suit their aesthetic tastes and tactical demands of the current level. Results from this proof-of-concept game illustrate that a significant impact is made on the visuals when combined with simultaneously generated audio patterns. While obtained results are already encouraging, there is room to improve the overall interaction between the player and the game and the generated audio in future version of the game. However, these initial results already suggest the potential of a multifaceted PCG approach toward realizing computational game creativity.

8 Acknowledgments

References

[1] Liapis, A., Yannakakis, G.N., Togelius, J.: Computational game creativity. In: Proceedings of the Fifth International Conference on Computational Creativity. (2014)

[2] Nintendo: Super mario bros. (1985)

[3] Sega: Sonic the hedgehog (1991)

[4] Collins, K.: An introduction to procedural music in video games. Contemporary Music Review **28**(1) (2009) 5–15

[5] Yannakakis, G.N.: Game ai revisited. In: Proceedings of the 9th conference on Computing Frontiers. (2012) 285–292

[6] Scirea, M.: Mood dependent music generator. In Reidsma, D., Katayose, H., Nijholt, A., eds.: Advances in Computer Entertainment. Volume 8253 of Lecture Notes in Computer Science. Springer International Publishing (2013) 626–629

[7] Cook, M., Colton, S.: A rogue dream: Automatically generating meaningful content for games. In: Proceedings of the AIIDE Workshop on Experimental AI and Games. (2014)

[8] Togelius, J., Shaker, N., Nelson, M.J.: Introduction. In Shaker, N., Togelius, J., Nelson, M.J., eds.: Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer (2014)

[9] Stanley, K.O.: Compositional pattern producing networks: A novel abstraction of development. Genetic Programming and Evolvable Machines Special Issue on Developmental Systems **8**(2) (2007) 131–162

[10] Cachia, W., Aquilina, L., Martinez, H.P., Yannakakis, G.N.: Procedural generation of music-guided weapons. In: Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG). (2014)

[11] Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games **3**(3) (2011) 172–186

[12] Firaxis Games: Civilization v (2010)

[13] Blizzard: Diablo (1996)

[14] Togelius, J., Nardi, R.D., Lucas, S.M.: Towards automatic personalised content creation for racing games. In: Proceedings of IEEE Symposium on Computational Intelligence and Games, IEEE (2007) 252–259

[15] Risi, S., Lehman, J., D'Ambrosio, D., Hall, R., Stanley, K.O.: Combining search-based procedural content generation and social gaming in the petalz video game. In: Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference. (2012)

[16] Gearbox Software: Borderlands (2009)

[17] Audio Surf LLC: Audiosurf (2011)

[18] Hastings, E.J., Guha, R.K., Stanley, K.O.: Automatic content generation in the galactic arms race video game. IEEE Transactions on Computational Intelligence and AI in Games **1**(4) (2009) 245–263

[19] Treanor, M., Blackford, B., Mateas, M., Bogost, I.: Game-o-matic: Generating videogames that represent ideas. In: Proceedings of the FDG Workshop on Procedural Content Generation. (2012)

[20] Key, E., Kanaga, D.: Proteus. http://www.visitproteus.com/ (2011)

[21] Holtar, N.I., Nelson, M.J., Togelius, J.: Audioverdrive: Exploring bidirectional communication between music and gameplay. In: Proceedings of the 2013 International Computer Music Conference. (2013)

[22] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems **2**(4) (1989) 303–314

[23] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation **10** (2002) 99–127