Case Study: A Small Step Toward Combining Procedural Content Approaches

Anonymous

Anonymous

Abstract. While generating content for video games is a growing field, many approaches focus on isolating single areas in games and inserting the new content into established game and game creation paradigms. This paper, however, introduces two new methods: one for creating music and the other for weapon visuals,

1 Introduction

For most video games, the interplay between game music (e.g. character themes, sound effects, foreshadowing) and visuals (e.g. level design,) is crucial for the creative expression of the game designers and immersion of the players [**?**]. However, as *nonlinear games* (i.e. games with many different that give the player) are surging in popularity (TODO stats), composers and designers are increasingly challenged to create meaningful TODOmoments/scenes for players who are now dynamically constructing their own game experiences.

As traditional composers and designers are adapting to this new medium [**?**], simultaneously approaches in procedurally generating assets for video games (called procedural content generation (PCG)) are emerging to enhance TODO-gameplay/replayability and alleviate some of the additional demands on designers, musicians, and artists. However, while these approaches in PCG are automating aspects game development, like asset creation, character behavior, and even developer tools, their focus ihas until recently been on inserting these assets and tools into the standard game development pipeline. (TODO- However, games based on PCG are interesting in their own right [**?**], emergence of playable PCG.) Instead, this paper not only explores two minimal PCG methods for generating music and weapon visuals and gameplay, but as a proof-of-concept also addresses how they can potentially influence each other to create a unique experience for each player.

(expressivity of the game, search in more than one facet).

(TODO-note incongruence between nonlinear games and the linear game we currently have - fix)

Add paragraph about multifaceted game creativity, see paper of liapis, distinguishes us from audiooverdrive, human for fitness, that it (limited human initivate) - explain different types

Mention domain of space shooter interactive evolution

As the field of procedurally generated content grows, so do the aims and scopes of what can be accomplished.

While procedurally generating content for video games may have begun as a way to relieve developers, the field has grown

The hope is that by exploring the interrelationships between two pcg systems within the same game, that a step can be made toward actualizing games with fully autonomous systems.

Michael Cook (Liapis paper) - look specific to music, ICCC 2014 ICCC Traenor (Liapis paper) proof of concept

2 Background

While there are many approaches to procedurally generating content or facets for video games (e.g. visuals, audio, narrative, game design, level design and gameplay), the focus is often on the single facet rather than how they can combine to create a complete game experience. For instance, world maps in *Civilization V* (Firaxis 2010), dungeons in *Diablo* (Blizzard 1996), racing tracks [8], and quest locations [2] are all procedurally generated to provide the player with increased personalization and replayabity. Simiarly, Audiosurf [] generates aspects of levels based on a given musical background. TODO-moremusic generated stuff. While these elements are awesome, games themselves are a complex set of interwoven assets.

and weapons in *Borderlands* (Gearbox 2009) are all procedurally generated.

Academic interest in procedural content generation for games is more recent, but has seen a plethora of novel and sophisticated algorithms applied for the purposes of generating a broad range of content from racing tracks [8] to quest locations [2] and from puzzles [7] to collectible flowers [6]. Search-based Procedural Content Generation (SBPCG) [9] often uses evolutionary computation to generate game content according to fitness functions (based on game literature or on estimated gameplay experience) or via interactive evolution.

Procedural content generation usually focuses on one specific type of game element (e.g. levels), and often with a specific (commercial) game in mind — e.g. maps for *Starcraft* (Blizzard 1999).

However, games have many different facets which interweave to create the final player experience: [5] identify visuals, audio, narrative, game design, level design and gameplay as creative facets of games.

There has been limited interest in attempting to generate game content which falls into more than one facet: notable exceptions include the *Galactic Arms Race* [3] where particles (visuals) have certain gameplay properties as weapon missiles (gameplay), *Game-o-matic* [10] which merges human-authored rhetoric (narrative) with generated rules (game design) and visuals, and *A Rogue Dream* [1] which assigns game effects (game design) to associations between words, provides visuals for them and places them in procedural generated levels (level design). Of particular interest is *AudioOverdrive* which uses a soundtrack to generate ludic elements (e.g. spawning enemies when a clap sample plays), while the sound

effects from player actions influence the enemy behavior in the same way as the soundtrack. Beyond games, **(author?)** [4] provide images as inspirational input for generating musical pieces. However, existing work rarely investigates the interaction between the multiple facets being generated; the generation of each facet is decoupled from the other in the sense that e.g. the art style of the visuals in *A Rogue Dream* does not inform the look and feel of the level architecture. CURRENTPROJECTTITLE attempts to orchestrate the generation of sound, visuals and gameplay: in the testbed space shooter game, the appearance of the spaceship's weapons (e.g. color) and their gameplay effects (e.g. whether it hit an enemy) affect the game's soundtrack; the soundtrack in turn affects the firing rate, color and movement patterns of the weapon's particles, thus ensuring a bidirectional communication across creative facets. While similar to AudioOverdrive in terms of goals, theme and creative game facets tackled, CURRENTPROJECTTITLE 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.

IEC: used in games, artwork

PCG games

While there are many approaches to procedurally generating particular aspects of video games (e.g. creatures, music, gameplay, weapons), the focus is often on the single asset rather than how they can combine to create a complete game experience. This section first discusses existing approaches for generating music and visuals for games followed by relevant representations and evolutionary mechanisms.

Audiosurf

NeuroEvolving Robotic Operatives (NERO) is a game where a team of collaborators is spawned and evolved in real time. Individuals who are rated poorly and replaced in real time with the children of successful team members. Similarly. However, how they can influence todo is todo. While they have the potential for combination, they are currently singular aspects of games.

Similarly,

Because video game music can be heavily dependent on environment, recently there have been several approaches to having gameplay and terrain influence music (and vice versa). Audioverdrive -bidirectional, heavily scripted, while maintains nice cohesion (important note amy, this program works much better than the shooter music program, less human initiative Proteus

these facets can combine to create a complete game experience is often they often focus on "plugging" the procedural content into an established game. The question of how these aspects can influence each other is For instance

they typically address issues solely related to that one particular aspect. While these approaches answer interesting questions related to these subgenres, the nontrivial question of how these ideas can interact

to do something cool growing issue in the field is generating these game facets is interesting in itself, an important question in the field of procedurally generated For instance, Nero is a shooter

This paper not only examines generating music

2.1 Evolving CPPNs with NEAT

TODO how CPPNs are important for this project. This transformation occurs through a variant of an ANN called a compositional pattern producing network (figure 1a).

The CPPN is a network of interconnected nodes similar to ANNs. However, unlike traditional ANNs, each node in a CPPN can compute a different type of function (e.g. Gaussian, sigmoid, linear, sine, multiplicative, etc.), thereby biasing the search space toward results with particular regularities. For instance, Gaussians tend to generate bilateral symmetry while sine hidden nodes suggest repeating patterns. These nodes are arranged within an arbitrary topology and, as in an ANN, the nodes send their outputs over outgoing connections by multiplying their outputs by the connection weights. In effect the CPPN is like an ANN with multiple possible activation functions that can be expressed within the same network.

CPPNs are chosen because they are suited to producing patterns with regularities. They are in effect generic pattern generators capable of producing patterns in space (such as images) just as they can produce patterns in time (such as music). While CPPNs generate music and weapon particles, their pattern-generating capabilities were originally demonstrated through generating images (i.e. spatial patterns). To give an intuitive idea of how a CPPN can generate such a pattern, figure 1 shows how CPPNs transform pixel positions into a collection of shadings that paint images. The (x,y) position of each pixel on the canvas in figure 1b is input to the CPPN in figure 1a to produce a shading. Figure 1b shows how these shadings are collected over the canvas to create complete pictures. As shown in figure 1b, the resultant patterns exhibit regularities, as should musical sequences as well.

3 Approach

Because it is important for the individual music and visual/gameplay domains to interact with each other at a fundamental level, each PCG module bases decisions on information gathered from both domains. These relationships between the domains and generated outputs are represented by a special type of artificial neural network called a compositional pattern producing network (CPPN; shown in figure 1) [**?**], which is in effect a patten generator biased toward certain regularities. By representing both the generated musical and visual/gameplay aspects through patterns of each, the aim is that TODO-somekindofcoolrelationship

Each PCG module is represented by a separate CPPN that inputs domain information and outputs instructions for generating either musical or visual patterns (TODO- note somewhere that visual patterns affect gameplay).

Because each CPPN

Domain patterns are sent to the CP

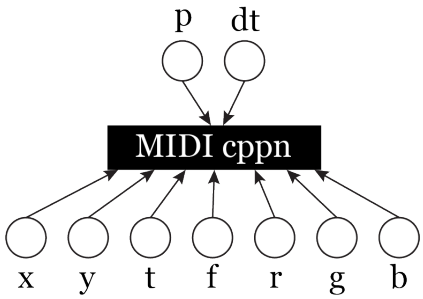


Fig. 1. Compositional pattern producing networks (CPPNs; **?**) are a special type of artificial neural network where activat. Like traditional ANNs, CPPNs are an interconnected network of nodes with bias outputs toward certain types of regularities. TODO:brief description of CPPNs

Explain how they interact, both take each other as input: Game loop
IEC

Each CPPN Output patterns are selected by the player through a process called interactive evolutionary computation, wherein the user rather than a predetermined function determines fitness of the individuals in the population.

These patterns are evolved through interactive evolutionary computation (IEC),

Because CPPNs produce patterns of the inputs, it is important to choose meaninful inputs Because patterns are important, generating desirable melodic and visual patterns is dependent on the types of inputs sent through the CPPN.

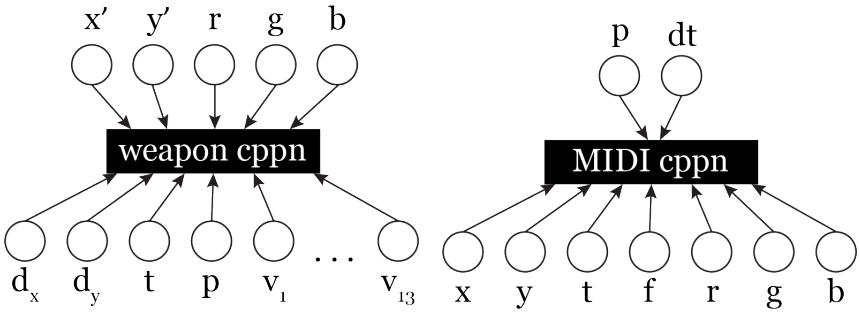
Relationships in TODO music-game are represented as compositional pattern producing networks evolved with neat. Each game asset has a different set of inputs and outputs.

3.1 GamePlay Testbed

This mixed initiave design is presented through a TODO-typeofgame(shoot-them-up) space shooter, called TODO-name [**?**]. The player moves through space levels attempting to avoid and shoot enemies while simultaneously evolving weapon trajectories and visuals and the accompanying music. Each bullet fired depends on the note played once fired. TODO-motivationsentence

Upon launch, a random note from the C Major pentatonic scale is sounded, and sent to the visual CPPN in figure 2a. Along with the pitch information from the MIDI note, this CPPN also inputs the (x, y) position where the bullet was fired, and the time t since firing. Information from the

The game's main research points are to observe the patterns of bullet movement through the use of Neural Networks with MIDI sound as the main focus and the generation of MIDI sound based on the visual of the bullet and in-game events. The two AI requirements needed the use of two networks to work in conjunction.



(a) Weapon CPPN (b) Music CPPN

Fig. 2.

The methodology of this research is to have a simple loop construct between the two networks where one network tells the in-game bullets how to travel whereas the other tells the game what sound to produce.

In the initial set up of the game, a random MIDI NoteOn event is generated from a predefined pentatonic scale. As soon as the game starts, the first NoteOn event triggers a fire function that launches a bullet from the ship. The trajectory of the bullet depends on the pitch produced from the triggering NoteOn event and the volume data occurring per frame. The structure of the CPPN is very similar to a previous iteration of the project that worked with music files injected into the game [1].

The CPPN's inputs are: Distance X relative to where the shot was fired Distance Y relative to where the shot was fired Time since bullet was fired NoteOn MIDI Pitch Volume Frequencies o 20hz - 39hz o 40hz - 59hz o 60hz - 99hz o 100hz - 199hz o 200hz - 399hz o 400hz - 599hz o 600hz - 999hz o 1000hz - 1999hz o 2000hz - 3999hz o 4000hz - 5999hz o 6000hz - 9999hz o 10000hz - 15999hz o 16000hz - 20000hz The CPPN's outputs are: X' Translation Y' Translation R Color G Color B Colorv

The second CPPN that is responsible for the generation of MIDI NoteOn events is called when a bullet is shot and when an enemy is hit. The inputs for the second CPPN are:

X position of bullet where it was fired or where it hit Y position of bullet where it was fired or where it hit How long the bullet was out Was it fired or did it hit (0 or 1) R Color value G Color value B Color value From those values, the output values generated for a new NoteOn event are: Note pitch MIDI Deltatime As the game goes on, the networks keep working together in order to produce the sound and visuals of the game.

The networks both start off as Single-Layer Perceptrons, allowing for an observation from a simple network to a more complex one as more Hidden Nodes are added to the network. The fitness function determined for each network is different between the bullet network and the MIDI network. The bullet CPPN is evaluated based on how many times a shot was fired using that network.

The MIDI CPPN is evaluated based on how long the player remained with that network.

References: [1] Cachia, William, et al. "Procedural generation of music-guided weapons."Computational Intelligence and Games (CIG), 2014 IEEE Conference on. IEEE, 2014.

[2] Hastings, Erin J., Ratan K. Guha, and Kenneth O. Stanley. "Evolving content in the galactic arms race video game." Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on. IEEE, 2009.

Step 1: Random note from pentatonic scale Step 2: triggered by Step 1, creates a bullet with the note value from step 1 Step 3: triggered by Step 2, activate trajectory and color network

Note IEC left open

3.2 general

weird abstract picture

3.3 game

The

3.4 particular approach

detailed CPPNs for weapons and music

4 Experiments

5 Results

time results

initial feed for music, constant on either side, see which generate what

6 Discussion and Future Work

Monitors on the music to trigger events in game - appearance of enemies, frequencies, boss battles

Evaluation: studies with designers, humans, player study, listener's study, control - potential of the fusion on creativity/human, GAR fed with number of weapons, see which weapon music combination that they use, what they like/preference comes directly from the use, what they like about the weapons why do they like the combo

Narrative?Level design

7 Conclusion

Summary of the paper
- Limitations of approach
- Extensibility to other games, other disciplines, out of the box

8 Acknowledgements

Bibliography

[1] Cook, M., Colton, S.:

[2] Hartsook, K., Zook, A., Das, S., Riedl, M.O.: Toward supporting stories with procedurally generated game worlds. In: Proceedings of IEEE Conference on Computational Intelligence and Games. pp. 297–304 (2009)

[3] 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), 245–263 (2009)

[4] Johnson, D., Ventura, D.:

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

[6] 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)

[7] Smith, A.M., Butler, E., Popović, Z.: Quantifying over play: Constraining undesirable solutions in puzzle design. In: Proceedings of ACM Conference on Foundations of Digital Games (2013)

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

[9] Togelius, J., Yannakakis, G., Stanley, K., Browne, C.: Search-based procedural content generation: A taxonomy and survey. IEEE Transactions on Computational Intelligence and AI in Games (99) (2011)

[10] Treanor, M., Blackford, B., Mateas, M., Bogost, I.: