PCG Shotgun: 6 Techniques for Leveraging AI in Content Generation by Tyler Coleman, Zach Aikman, Tanya Short, Tarn Adams, Mitu Khandaker-Kokoris, Luiz Kruel

Author: Thomas Nuschy

May 5, 2017

Tags: PCG, procedural story generation, procedural personality generation. procedural level generations
Track: GDC 2017 - AI Summit
Url: http://www.gdcvault.com/play/1024146/PCG-Shotgun-6-Techniques-for
Speaker: Mitu Khandaker-Kokoris, Spirit AI
Speaker: Tanya Short, Kitfox Games
Speaker: Zach Aikman, 17-BIT
Speaker: Tarn Adams, Bay 12 Games
Speaker: Luiz Kruel, SideFX Software
Speaker: Tyler Coleman, Retora Games

Abstract

This work gives an short overview of different procedural content generation (PCG) tools in games. PCGs are used in various aspects of game development. This summary touches the topics of procedural story generation, personality generation and level generations. It shows that designers sometimes have to think outside of the box to get the result they want.

1 Summary of Talk

In the GDC talks *PCG Shotgun: 6 Techniques for Leveraging AI in Content Generation* Tyler Coleman, Zach Aikman, Tanya Short, Tarn Adams, Mitu Khandaker-Kokoris and Luiz Kruel present different applications of procedural content generation in gamedevelopment. The talk gives six short introductions of six different applications.

1.1 Procedural Character-Based Narratives

Mitu Khandaker-Kokoris from Spirit AI talks about techniques used for procedural narrative generation. Using such procedural narratives allows game designer to tell various different stories not just one. The first method presented is *Agent-driven Stories*. This method is for simulating the interaction of agents that tell the player something about the word. The story is told by a mechanism called *noticing*. A game in which this mechanic is used is Little Invasion Town . Little Invasion Town utilising *noticing* for telling the story of the game by showing text bubbles and the behaviour of the agents in the game. The second method is *Story-driven Agents*. In this case one specific agent knows a lot about a specific plot. The player learns about this specific plot by interacting with this agent. This interaction can be a dialog or something else.

1.2 Maximizing the Impact of Generated Personalities

Tanya Short from Kitfox Games about how to generate personalities using procedural methods. The generate a personally we need two main parts. First a *Reasoning*, meaning why does the agent to something. Secondly a *Behaviour*, meaning what is the agent doing. In human generated personalities agents show a behaviour and its up to the player to interpret their reasoning. This can be called *Behaviour First*. This is more natural because its how it works in real life. In contrast to that stands *Reasoning First*. This method is used by most procedural games. The agent tells the player his reasoning and then acts. Its up to the us as players to notice what the agent does with its reasoning.

A problem with this is that in can result in a lot of information for the player but as players get more accustomed to procedural personalities game designer don't have to expose all traits of agent. It can become part of the gameplay to find this hidden traits. An example for such a game is The Shrouded Ilse.

There a several ways in which the impact of procedural personalities can be maximized. First, in general personalities can be very subtle, even extremes. Therefore it is import to support these personalities with some sort of behaviour. Secondly, procedural generated people do not act like real people, which leads to some level of comedy. Third, its much more economically to generate reactions in agents from one action, allowing the player to learn more about the agents in the game. And last, change in personality is normal.

1.3 In-Game Content Generation by AI-Agents

Tarn Adams from Bay 12 Games talks about AI algorithms to create game objects. AI algorithms can get quit complicated with a lot of parameters to configure. This parameters can be used to restrict the output of the algorithm. For the parameters everything can be used events, locations, peoples, relationships or personality. With the created objects its possible to create an emerging narrative by using the output of an algorithm as input. The example that is given in the talk is that the player is elected mayor. This triggers that an AI agent creates a statue of the player. This statue holds some data about the player or an vet. One thing you can do with this data is let AI agents react to it or you can simply use it to distributed information about an event.

By using generated game objects as input for other procedural algorithms we can create some sort of event tree. This tree can contain objects from the past, present and future. This allows the game to refer back to past events and let new content emerge from it.

1.4 Procedural Systems and AI in Galak-Z

Zach Aikman from 17-BIT talks about procedural level generation and how to make NPCs more alive in their game Galak-Z. Galak-Z is a 2d space shooter game that uses procedural generated levels. They create the used a two step method.

The first step was fo generate cavers using *Cellular Automata*. *Cellular Automata* work on a cellular grid with a finite number of states for each cell. States transitions are described by a constraints. With the right constraints it is possible to create room like structures. In 1.4 you can see an example of a *Cellular Automata*. 17-Bit used this technique to create chunks of a level that are later stitched together. The problem is that generating this chunks during runtime is too inefficient.

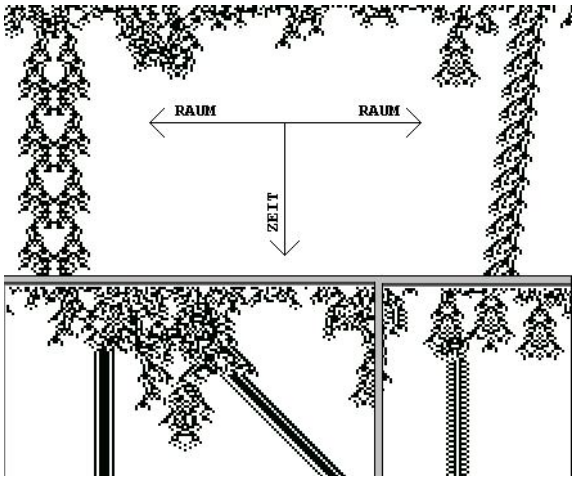

Figure 1: Example of a Cellular Automata[2]

The second step was to stitch the stitch the level chunks together. This was done by utilising Hilbert Curves. In 1.4 you can see the first to third order hilbert curve. In Galak-Z they used this curves to stitch the level chunks created by the cellular automata together to create the levels. To avoid that every level has the same base structure they took higher level Hilbert curves and took move some sort of sliding window around to get more complex levels.

In Galak-Z they used *Contextual Barks*[4] to make their NPCs look more alive. *Contextual Barks* are simple described a set of rules that if met lead to a dialog. A problem that can arise

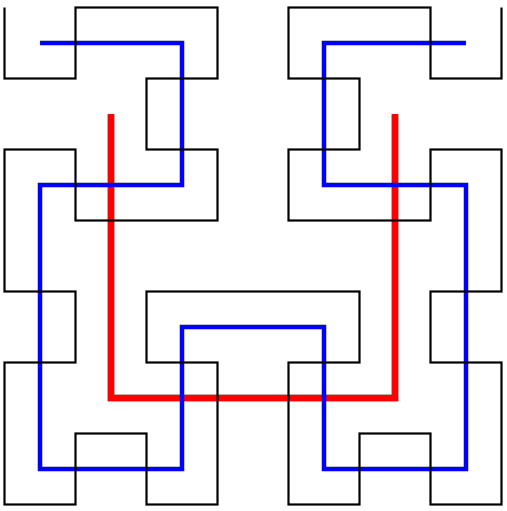

Figure 2: First to third order Hilbert Curves[2]

with *Contextual Barks* is that the number of rules can be very high and lead to a very complex systems.

1.5 Procedural Generated FPS

Luiz Kruel from SideFX Software about how to create levels for a FPS using procedural methods in AAA quality. He used a tool called Sentient Sketchbook[5] to create a basic room layout. For generating the level they use a backwards approach the designer creates a basic level layout with connected rooms. The level layout is then in combination with the SEntient Sketchbook to create the content of the rooms.

1.6 Pseudo is your Friend

Tyler Coleman from Retora Games talks about how pseudo randomnumber generators can be used in games and how you can take advantage of the pseudo nature of these generators. Usually randomnumber generators need some sort of seed. In games the seed might be responsible what kind of experience happens in the course of the game.

Their are different options on how to get a seed. The first one is to let the user input the seed. This allows players to share their experiences with others as the same seed triggers the same events. Another option would be to user the users input as seed. An example would be something like the Assassins Creed loading screens where the player can run around and do stuff or something simple like entering the username. Another seed possibility is to take the X, Y and Z coordinates from an object. This seed is unique to that item. Also UserID can be used as seed. This allows a unique experience for a specific player. Another type of seed is *Lifetime Seed*. The idea behind this type of seed is that the gamer experience stays consistent through multiple gameplays although it is different between players.

2 Overview and Relevance

The talk shows that procedural methods can be used in very different way in every aspect of game design form level to personality creation and also at some level for story telling. Some methods present in the GDC talk are more or less standard methods but as the use of *Hilbert Curves* for level generation shows game designers have to sometimes think outside of the box when conventional methods do not work for the problem at hand.

I think that the relevance of procedural generated content will get higher but more in the aspect of story telling. Procedural generated stories make a game much more replayable than dungeons that are to some point random, like dungeons in *Diablo*.

3 References and Further Sources

[1] Little Invasion Tales, http://littleinvasiontales.com/

[2] Cellular Automaton, https://en.wikipedia.org/wiki/Cellular_automaton

[3] Hilbert Curves, https://en.wikipedia.org/wiki/Hilbert_curve

[4] Dynamic Dialog, Ruskin Elan, http://www.valvesoftware.com/publications/2012/GDC2012_Ruskin_Elan_DynamicDialog.pdf

[5] Sentient Setchbook, http://www.sentientsketchbook.com/