

The Future of Procedural Content Generation in Games

Gillian Smith

Northeastern University, Playable Innovative Technologies Group
360 Huntington Ave, 100 ME, Boston MA 02115
gillian@ccs.neu.edu

Abstract

The future of procedural content generation (PCG) lies beyond the dominant motivations of “replayability” and creating large environments for players to explore. This paper explores both the past and potential future for PCG, identifying five major lenses through which we can view PCG and its role in a game: data vs. process intensiveness, the interactive extent of the content, who has control over the generator, how many players interact with it, and the aesthetic purpose for PCG being used in the game. Using these lenses, the paper proposes several new research directions for PCG that require both deep technical research and innovative game design.

Introduction

There is a strong community of both researchers and practitioners around procedural content generation (PCG), largely devoted to furthering technical research in how to create content generators that are controllable and expressive (Togelius et al. 2011; Hendrikx et al. 2011). PCG makes many promises to designers—the replayability that can come from many different potential experiences for a player, the shifting of design burden from designer and artist to programmer, the ability to adapt content to meet player’s desires, and environments larger than feasible to author by hand for the player to explore. Though not all of these promises are fulfilled, these appeals have driven technical research in PCG, resulting in research that typically focuses on personalization (Yannakakis and Togelius 2011) and PCG interaction with designers (Liapis, Yannakakis, and Togelius 2013; G. Smith, Whitehead, and Mateas 2011). This research tends to concern itself with trying to improve that which is already present in games, asking questions such as can PCG make Mario more fun for players (Shaker et al. 2011)? How can designers create levels more efficiently or

more creatively (Liapis, Yannakakis, and Togelius 2013)? How can quests be adapted to best suit individual players (Thue, Bulitko, and Spetch 2008)?

Even the most innovative games typically use PCG in similar ways and for similar aesthetic goals. The aesthetics of discovery and challenge are common to most games that use PCG, and even games that use PCG as a mechanic (Hastings, Guha, and Stanley 2009; Risi et al. 2012; G. Smith et al. 2012). Are infinitely explorable environments and adaptive gameplay the best we have to hope for from PCG in game design? These are noble goals, to be sure, but surely not the only ones that should be motivating a community of researchers and designers.

This paper envisions new futures for PCG, and its potential to drive radical innovation in the kinds of games we can make. It contributes a theory for how PCG has been used in the past up until today: how has it changed in the last several decades, and what common themes can we draw from these changes? Then, we will use these themes to project forward to imagine potential futures. Identifying these futures provides more clarity to the current, somewhat ill-defined motivations of using PCG to imbue games with replayability and adaptiveness, and also offer alternatives, as well as offering entirely new motivations that can drive new research.

PCG’s Past

In order to reason about the future, one must first understand the past. PCG has been used in digital games since the early 1980s, when *Rogue*’s (Toy et al. 1980) procedurally generated dungeons and *Elite*’s (Braben and Bell 1984) procedurally generated galaxies provided players the ability to explore unexpected and vast spaces. It has since been used in many other games, from generating territory to explore in *Civilization* (Firaxis Games 2005) and *Minecraft* (Persson 2011) to platforming challenges in *Robot Unicorn Attack* ([adult swim games] 2010) and

Canabalt (Saltsman 2009). The technical approaches to PCG have been well-summarized in two major survey articles (Togelius et al. 2011; Hendrikx et al. 2011), and I have previously published a play-centric examination of the role of PCG in games (G. Smith 2014), upon which the following analysis of PCG’s history is based.

Over the course of PCG’s history in digital games, there are five main themes that we can draw out in how it is used and interacted with, each of which will be explored in more detail in this section. Each of these themes can be considered spectra along which individual PCG systems can be positioned. These spectra are being identified in this paper by their extremes.

Data Intensive to Process Intensive Systems

Data-intensive PCG uses large building blocks that have been made by a human designer. It requires heavy authoring from a human, and often has lightweight algorithms to piece these building blocks together, as in endless runner games such as *Robot Unicorn Attack* ([adult swim games] 2010). Process-intensive PCG, on the other hand, places much of the design responsibility on an algorithm or process. The computer is responsible for making creative decisions about how to piece together small building blocks or reason about abstract constraints on the generative space.

Note that there is a distinction between a data *intensive* PCG system and a data-*driven* PCG system (though a system may indeed be both of these). A data-driven system uses external data to inform the generation of content; for example, a system to create monopoly boards based on real-world demographic data (Friberger and Togelius 2012). This content could then be generated using a data-intensive approach, with data filling in large templates created by a human, or a process-intensive approach, with the data being one of many components that informs the algorithm’s decisions.

Non-Interactive to Interact-able Content

Three major threads of research have formed during PCG’s history, each related to the nature of the content being created. Early content generation systems, such as L-systems and particle systems, were used to create content that players do not typically interact with heavily in the game, such as textures, trees, or special effects like smoke and fire (Ebert 2003). In the 1980s, we saw the beginnings of PCG being more concerned with interact-able content, such as quests, levels, and weapons (Hendrikx et al. 2011). In the late 1990s and 2000s, there has been a rise in research into systems that can create the rules of a game itself, and even other content generators (A. M. Smith and Mateas 2010; Orwant 2000).

Developer Control to Player Control

Most early PCG systems were created by practitioners who took on the role of both developer and designer. Much of the control over content created in those systems lay entirely in the algorithm, with little room for a non-technical user to interact with it. *Elite* and other similar systems, such as those used in demoscenes (.theprodukt 2004), used PCG to address technical problems such as low memory. More commonly, systems attempt to act as a designer and most make some informed decisions about content at runtime, rather than simply decompressing content that has been expressed in code rather than stored in memory.

Recently, however, there has been a push towards supporting a player interacting with a content generator, even to the extent of supporting the player taking on some of the authoring burden for the game. *Spore*’s creature creator (Maxis 2008) is an example of such a system, with procedural animation and texturing support helping the player act as a designer; however, this is happening mostly at the level of non-interactable content.

Single-Player to Multi-Player

PCG began in digital games as a way to create content for a single player, and the vast majority of PCG systems are still considering an individual player’s experience in a single-player game. The few multiplayer games that use PCG, such as *Civilization IV*, are typically creating content that is in a single instance—maps that many players will interact but are the same for each player. There are very few games that have multiplayer, multi-instance PCG, where multiple players are interacting with unique, realtime generated content that is based on the behavior of multiple players—*Galactic Arms Race* comes close, with weapons tailored for individuals that are born from a pool that is available to all players (Hastings, Guha, and Stanley 2009)—or that can influence what other players will see in their own version of the game.

Common Game Aesthetics

Challenge, *Discovery*, and *Fellowship* are the three most common aesthetic properties that arise from the use of PCG in games (G. Smith 2014). Challenge arises through the use of PCG to provide tailored and surprising environments; for example, *Rogue* (Toy et al. 1980) and roguelikes typically use PCG to give players new and unknown environments to test their skills in. PCG supports *Discovery* through providing new environments for the player to explore, or new systems for them to learn about over time. And finally, *Fellowship* is an aesthetic that has come about more recently, through the creation of an emergent system that encourages player communication outside of the game, such as in forums for the game

Civilization IV, where long discussions about generated maps (and even how the generator works) are the norm.

PCG's Present

The current state of PCG, both in industry-created games and in academia, still sits in one of the categories described above. Many generators are still quite data-intensive, due to the desire to place authorial control in the hands of designers. Very little PCG research looks at how to create deeply interactable content, let alone entirely new games. There is research in how to create PCG that can be controlled by human designers, but little that could reasonably be used by *players*—this requires additional work in understanding how it hooks into a game, can be made enjoyable, and can be used by truly novice designers. The majority of PCG research exists to create content for single players or as single instances. Experimental games such as *Proteus* (Key and Kanaga 2013) aim to use content generation to provoke other aesthetics such as *Sensation*, though these are currently in the periphery of the PCG community.

PCG's Future

PCG research is currently making forward progress within the motivations that are laid out above. Innovation can also happen in the “low” end of the scale in these PCG trajectories; for example, procedurally generating non-interact-able content is still a highly active research area. Though this research can be worthwhile, it is all towards doing better in an established area and towards an established set of motivations, rather than pushing the envelope of PCG's capabilities and what it can do to drive innovative game design.

What are the new and exciting problems in PCG? What kinds of games could we make with *different* kinds of PCG technology? Let's turn PCG “up to eleven”—what kinds of games and PCG systems could exist if we moved far beyond what has already been done?

Deeply Process-Intensive Generation

A deeply process-intensive generator should be able to create content with minimal input from its creator. Part of process-intensiveness is tied to the kind of knowledge representation used, removing design knowledge from the building blocks used and placing it into the algorithm being followed. A deeply process-intensive generator is one that meaningfully follows a design process, and can rationalize the decisions it makes. It could create or modify its own concept of the quality of what it generates based on its prior experiences, and it could draw design inspiration from an external source.

Of course, the creation of a truly process-intensive generator is an AI-complete problem. However, even small steps in this direction have the potential to create exciting new games and design tools. Imagine a design tool incorporating a content generator that is aware of its own process, and can explain to a player not only *what* is has created but *why* it was created. Or a journaling game where the player can provide photographs and a description of an event, and the system can create a video game about that event.

Creating New Mechanics and Genres

Some of the most delightful moments in games come where the game delivers *surprising* content that follows the theme of what the player has seen previously, such that the player must learn how it works and build new game strategies. For example, new level elements such as platforms where the player controls their movement in *New Super Mario Bros* (Nintendo EAD 2006). While there has been research in procedurally generating level progressions (Butler et al. 2013), this work uses pre-defined progressions and game elements. How can we create systems that can generate their own progressions? This requires a kind of player modeling that operates at a greater level of detail than numerical scores for enjoyment and frustration, combined with a representation for the mechanics of the game and how game components use them.

Another avenue for research in interactive content generation is the generation of game *genres*: what is the generator that can create sets of games that are meaningfully different from each other, yet share a common theme?

Power to the Player

User-created content is a popular aspect of many games; however, it is difficult for players (who typically lack design experience) to create engaging content for themselves and others. *Spore* is a game where user-created content was highly successful, allowing players to create professional-quality content using simple tools, imbued with enough intelligence to empower the player without overriding their intent. What are the “magic crayons” (Gingold 2003) of interactable content? Work in creating PCG-based design tools has largely focused on providing brainstorming support to designers, creating many variants that can then be tweaked, but how can we create tools that help a player realize a vision that they already have, but do not know how to express?

Multiplayer and Multi-Instance PCG

What does it mean for multiple players to really engage with generated content, and for a generator to be designed

with multiple players in mind? A multiplayer game with multi-instance PCG could have everyone seeing different content while inhabiting the same space, with support for mechanics that allow players to influence each other's environments. For example, imagine a collaborative multiplayer platforming game where each player's actions causes new content to be created in another's, and players must find ways to communicate about how to achieve some common goal.

In the context of design tools, there is the question of how a tool could learn from previous users' experiences. Can it incorporate previously created designs into its own design suggestions? Or learn common patterns of user interaction?

Beyond Current Aesthetics

In their paper suggesting the adoption of the MDA framework, Hunicke et al. identify eight "kinds" of aesthetics for games (Hunicke, LeBlanc, and Zubek 2004). Discovery and Challenge are two kinds that have already been well-explored with PCG and form the major motivations for using PCG in games, while Fellowship is an aesthetic that emerges from the use of PCG, though is not typically a motivating aesthetic. The remaining six aesthetics of Sensation, Fantasy, Narrative, Expression, and Submission are all fertile ground for new PCG systems and PCG-based games.

For example, what is the game that is built around the aesthetic of Sensation? This would require imbuing the generator with an understanding of this aesthetic: what makes for visually or aurally beautiful games as experiences, and how can a generator amplify this? This is an area that is explored more in generative art (Boden and Edmonds 2009) research, though its implications for game design are less clear.

The Path to the Future

The future of PCG-based game design requires a combination of deep research into new algorithms and systems, as well as innovative game design. We can pay more attention to both the designer and the design *process* when creating systems, rather than relying on design knowledge to be embedded in building blocks or applied as an evaluation function on an otherwise design-blind process. We should consider methods for generating new games that do not rely on templates of game mechanics. We need to consider the desires of the player and the game motivations for user-created content when making PCG-based design tools. We have the opportunity to create exciting new PCG systems that can interact with and learn from multiple users. And, perhaps most importantly, we should be considering game aesthetic motivations for PCG

beyond Discovery and Challenge. There are also many lessons to learn from other communities that touch the PCG community, such as generative art and design; indeed, others have also argued for the expansion of the community to encompass these other methods (Compton, Osborn, and Mateas 2013).

We're barely scratching the surface of PCG's potential. The existing common motivations of "replayability" and "adaptability" do not fully capture the power of PCG for games. There are games that are waiting to be made, if only we had the AI systems necessary to make them, and the ability for designers to use those systems.

Acknowledgments

My thanks to Noah Wardrip-Fruin, for fruitful conversations that led to identifying several of these PCG trajectories. And to the growing PCG community and to the reviewers of this paper, whose current activity has made thinking about the community's future all the more exciting.

References

- .theprodukt. 2004. *.kkrieger (PC Game)*. <http://www.theprodukt.com/kkrieger>.
- [adult swim games]. 2010. *Robot Unicorn Attack (PC Game)*.
- Boden, M.A., and Edmonds, E.A. 2009. "What Is Generative Art?" *Digital Creativity* 20 (1-2): 21–46.
- Braben, D., and Bell, I. 1984. *Elite (BBC Micro)*. Acornsoft.
- Butler, E., Smith, A.M., Liu, Y. and Popovic, Z. 2013. "A Mixed-Initiative Tool for Designing Level Progressions in Games." In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, 377–86.
- Compton, K., Osborn, J.C., Mateas, M. 2013. "Generative Methods". In *Proceedings of the Fourth Workshop on Procedural Content Generation in Games*, co-located with *Foundations of Digital Games*. Chania, Greece, May 15, 2013.
- Ebert, D.S. 2003. *Texturing & Modeling: A Procedural Approach*. Morgan Kaufmann.
- Firaxis Games. 2005. *Civilization IV (PC Game)*. 2K Games.
- Friberger, M.G., and Togelius, J. 2012. "Generating Interesting Monopoly Boards from Open Data." In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 288–95. doi:10.1109/CIG.2012.6374168.
- Gingold, C. 2003. "Miniature Gardens and Magic Crayons: Games, Spaces, & Worlds". Master of Science in Information, Design & Technology, Georgia Institute of Technology. <http://levitylab.com/cog/writing/thesis/>.
- Hastings, E. J., Guha, R.K., and Stanley, K.O.. 2009. "Automatic Content Generation in the Galactic Arms Race Video Game." *IEEE Transactions on Computational Intelligence and AI in Games* 1 (4): 245–63.
- Hendriks, M., Meijer, S., Van der Velden, J., and Iosup, A. 2011. "Procedural Content Generation for Games: A Survey." *ACM*

Transactions on Multimedia Computing, Communications and Applications.

Hunicke, R., LeBlanc, M., and Zubek, R. 2004. "MDA: A Formal Approach to Game Design and Game Research." In *Proceedings of the 2004 AAAI Workshop on Challenges in Game Artificial Intelligence*. San Jose, California: AAAI Press.

Key, E., and Kanaga, D. 2013. *Proteus (PC/Playstation 3)*. Curve Studios.

Liapis, A., Yannakakis, G.N., and Togelius, J. 2013. "Sentient Sketchbook: Computer-Aided Game Level Authoring." In *Proceedings of ACM Conference on Foundations of Digital Games*. Chania, Greece, June 2013.

Maxis. 2008. *Spore (PC Game)*. Electronic Arts.

Nintendo EAD. 2006. *New Super Mario Bros. (Nintendo DS)*. Nintendo.

Orwant, J. 2000. "EGGG: Automated Programming for Game Generation." *IBM Systems Journal* 39 (3.4): 782–94.

Persson, M. 2011. *Minecraft (PC Game)*. <http://www.mojang.com/notch/mario/>.

Risi, S., Lehman, J., D'Ambrosio, D.B., Hall, R., and Stanley, K.O. 2012. "Combining Search-Based Procedural Content Generation and Social Gaming in the Petalz Video Game." In *Proceedings of the 2012 Conference on Artificial Intelligence and Interactive Digital Entertainment*. Palo Alto, CA.

Saltsman, A. 2009. *Canabalt (PC Game)*. Adam Atomic. <http://www.adamatonic.com/canabalt/>.

Shaker, N., Togelius, J., Yannakakis, G.N., Weber, B., Shimizu, T., Hashiyama, T., Sorenson, N., et al. 2011. "The 2010 Mario AI Championship: Level Generation Track." *IEEE Transactions on Computational Intelligence and AI in Games* 3 (4): 332–47.

Smith, A. M., and Mateas, M. 2010. "Variations Forever: Flexibly Generating Rulesets from a Sculptable Design Space of Mini-Games." In *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, Copenhagen, Denmark, August 18-21, 2010.

Smith, G. 2014. "Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games." In *Proceedings of the 2014 ACM Conference on Computer-Human Interaction*. Toronto, Canada.

Smith, G., Othenin-Girard, A., Whitehead, J., and Wardrip-Fruin, N. 2012. "PCG-Based Game Design: Creating Endless Web." In *Proceedings of the International Conference on the Foundations of Digital Games*, 188–95. FDG '12. Raleigh, NC.

Smith, G., Whitehead, J., and Mateas, M. 2011. "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design." *IEEE Transactions on Computational Intelligence and AI in Games (TCAIG), Special Issue on Procedural Content Generation* 3 (3).

Thue, D., Bulitko, V., and Spetch, M. 2008. "PaSSAGE: A Demonstration of Player Modelling in Interactive Storytelling." In *Proceedings of the Fourth Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE '08)*, 227–28. Palo Alto, CA: AAAI Press.

Togelius, J., Yannakakis, G.N., Stanley, G.N., and Browne, C. 2011. "Search-Based Procedural Content Generation: A Taxonomy and Survey." *Computational Intelligence and AI in Games, IEEE Transactions on* 3 (3): 172–186.

Toy, M., Wichman, G., Arnold, K., and Lane, J. 1980. *Rogue (PC Game)*.

Yannakakis, G.N., and Togelius, J. 2011. "Experience-Driven Procedural Content Generation." *IEEE Transactions on Affective Computing* 2 (3): 147–61.