ISBN: 978-989-20-5488-9

Computational Creativity in Procedural Content Generation: A State of the Art Survey

Nuno Barreto, Amílcar Cardoso and Licínio Roque
CISUC
Department of Informatics Engineering
University of Coimbra
Coimbra, Portugal
{nbarreto, amilcar, lir}@dei.uc.pt

ABSTRACT
Computational Creativity is a research area that encompasses, among other investigation, the study of creativity models applied to the production of artifacts. As such, artistic areas are targeted by Computational Creativity researchers. One such area, that could potentially benefit from the inclusion of artificial creativity models, is Procedural Content Generation. In this paper, we provide a survey of Procedural Content Generators that contain explicit creativity models. We start by presenting an overview of Procedural Content Generation alongside its applications in addition to providing some insights regarding Computational Creativity, in general. We then illustrate the common grounds that Procedural Content Generation shares with Computational Creativity and, finally, we pinpoint future directions for research.

Author Keywords
Computational Creativity; Design Tools; Procedural Content Generation; Videogames.

ACM Classification Keywords
I.2.1 Applications and Expert Systems: Games

INTRODUCTION

Procedural Content Generation (PCG) has been a part of videogames almost since the beginning [23]. For instance, as early as 1984, Elite [4] used PCG as a means to create unique galaxies without having to store their information explicitly. However, while initially it was seen as a method to overcome hardware restrictions, today the increase of videogame development budgets, and subsequent accessibility to development tools have created a need for PCG as either a way to increase a game's replay value without additional cost, or as a means to reduce development time.

As a computational approach to content creation, which is perceived as artistic, PCG could benefit from the inclusion of creativity models studied in Computational Creativity (CC). Although there has been some initial work done in the past, only recently have researchers been starting to incorporate these models to study the resulting enhancement in Procedural Content Generation [34] methods. As a result, the literature regarding this topic is still in its infancy.

In this paper, we provide a state-of-the-art survey of CC in PCG as well as potential future directions this research topic may follow. As such, this paper is structured as follows: Section 2 will present a definition, a contextualization and some state-of-the-art regarding PCG, Section 3 will provide an insight into CC, Section 4 will indicate the links between CC and PCG as well as state-of-the-art applications, and, finally, Section 5 will provide a conclusion to this paper and showcase future work that could be developed.

PROCEDURAL CONTENT GENERATION

To better understand PCG, one must first comprehend its roots: In 1987, Crawford introduced the notion of "process intensity" [15], a notion to express programs that spend more time performing calculations than consulting data, which, according to him, makes the programs more valuable. Additionally, he stated that, as any other program, videogames benefit from "process intensity". Internally, PCG follows this concept.

Togelius et al. [48, 49] and Shaker, Togelius, & Nelson [38] define PCG as "the algorithmical creation of game content with limited or indirect user input". In their definition they state that content can be any game component including sounds, textures, levels, etc. excluding Non-Playable Character AI. Although the authors explain this exclusion by claiming that Computer Intelligence has a more extensive research applied to character behavior than PCG. We, however, find that the generation of behaviors could be included in the definition if we consider emergent, or adaptive, behaviors as it fits the definition presented by Shaker et al.

According to Nareyek [35] and Yannakakis [51], PCG research should be taken into account as future work in AI applied to game development. They claim that the otherwise typical application for game AI, behavior modelling, has recently reached a point where it is practically solved and that AI developers should focus instead on how can PCG be improved, as well as on how it can be used as a design tool. Moreover, Togelius et al. [46] propose that there are several areas of future research that the field of PCG should take into account. For instance, the generation of multi-level multi-content artifacts, a type of artifact that is composed by several layers of content,

dependent of the context of the game and/or game-engine. In addition, the authors pin-point several challenges researchers still need to address, most notoriously, they refer to the fact that most content created may be perceived as generic or bland. Moreover, they identify the need for generators to somehow learn to use an underlying style.

Taxonomies

There are some approaches that attempt to categorize PCG in order to help future developers design potential solutions. For instance, Togelius et al. [49] present a taxonomy that categorizes techniques according to their morphology and application. They introduce five pairs of classes, each with mutually exclusive concepts:

- Content created during the game (*online*) or before its development (*offline*).
- Content required to complete the game (*necessary*) or one that supplements the game (*optional*).
- Generators that receive as input *random seeds* or a *parameter vectors*.
- *Stochastic* or *deterministic* generation.
- Algorithms that include an evaluation function (*generate-and-test*), or not (*constructive*).

Hendrikx et al. [23] provide a different approach to classifying PCG. Instead of grouping PCG techniques, they developed a layered taxonomy for game content in which PCG can be used. In their taxonomy they included:

- *Game bits*: the most elementary units that compose a game including textures, sounds, etc.
- *Game space*: the world where the player navigates.
- *Game systems*: complex systems that increase the game's believability.
- *Game scenarios*: the logical sequence of the games' events.
- *Game design*: the rules and goals of a game.
- *Derived content*: side-products of the game's world including leaderboards and news.

Finally, Khaled, Nelson and Barr [26] developed a set of design metaphors to help understand the relationships between designers and PCG. These metaphors include:

- *Tools*: mechanisms meant to achieve design goals and empower the designer.
- *Materials*: artifacts procedurally generated that can be edited by designers.
- *Designers*: PCG algorithms that aim to solve game-design goals.
- *Experts*: monitors and analysts of gameplay data.

While these taxonomies may seem mutually exclusive, they actually complement each other. Indeed, each set of categories analyses content generation through a different perspective.

State-of-the-Art Applications

One of the most classical applications, and the one which debuted PCG, was to overcome hardware limitations. Initially, computers lacked the hardware power they have today and, as a result, complex and ambitious games could not run on such machines. Games such as Elite [4] and Starflight [1] opposed these restrictions by applying PCG to generate worlds that could otherwise be impossible to fit on 1980s computers' memory. However, nowadays, with the advances in hardware, this type of application is becoming scarcer.

Improving the replay value of games has also been a target of PCG ever since its infancy. While there have been many games in the past, including Rogue [50], Elite and Starflight that have done this, Diablo [2] popularized PCG as a selling-point for "infinite" playthroughs. In addition, there have been game genres, roguelikes and, more recently, infinite runners, that capitalize from PCG as their core mechanics.

Yannakakis [51] and Brown et. al. [6] defend a different application for PCG: by analyzing how a player interacts with the game, developers can use PCG as a means to deliver a more custom experience. More specifically, by having the game build an experience model of its player, a PCG can then use the model to create content. For instance, Galactic Arms Race [20], a space shooter with Role-Playing Game elements, generates ships and weapons to best fit the player's style in a given moment.

Finally, PCG can be part of design tools: by unloading part of the content creation to computers, designers can then focus on improving the content [23, 35, 51] and, consequently reduce game production costs. Moreover, the computer can even act as a co-designer, as stated in Khaled, Nelson and Barr [26]. Craveirinha, Santos and Roque [14] explore the latter by developing an author-centric architecture for PCG which generated and validate levels for a platform game. There are also several commercial design tools that contain some form of PCG including Speed Tree [24], a L-Grammar based tree generator, Euphoria [36], a tool that provides adaptive animations based on the mechanism introduced in [39], to name some.

Search-Based Algorithms in Procedural Content Generation

Since PCG is used on a variety of content, there is a wide range of algorithms that have been used. There are, however, some algorithms that account for classical approaches of certain content. These include Cellular Automata [25], Perlin Noise [27], L-Systems [40], Fractals [40] and Graphs [39], to name a few.

Togelius et al. [49] coined the term Search-Based Algorithms to identify a sub-set of generate-and-test algorithms. The main particularity of this sub-set is that, besides having a generation and evaluation phase, they comprise a genotype (a solution codification that facilitates

ISBN: 978-989-20-5488-9

computations), a phenotype (a representation of the solution that can be understood by humans) and a mapping function that translates genotypes into phenotypes. While these algorithms resemble evolutionary algorithms, the authors stress that search-based algorithm include other kind of algorithms. In addition, the authors also state that search-based PCG is a valuable tool but it should not be used to solve every existing PCG problem.

Liapis, Yannakakis and Togelius [29] present two algorithms for performing novelty searches (a type of genetic algorithms that aim to produce a wide variety of solutions) in large spaces with multiple constraints. In a nutshell, these algorithms use search-based methods yet they use two sets of populations, one with feasible individuals and another with unfeasible ones. Experiments conducted showed that these algorithms seem to generate a large set of feasible maps for strategy games, unlike existing algorithms used in these problems. Liapis, Yannakakis and Togelius also proposed another search-based algorithm, in [28], consisting in an adaptive model for evolving spaceship shapes according to user selection. The fitness function used in this algorithm is also altered to fit criteria underlying the ships visual qualities. Results showed promise when considering one visual criterion yet, when using a weighted sum of multiple criteria, the outcomes seemed arbitrary.

Finally, Dahlskog and Togelius [17] introduced an evolutionary system that learns the style [46] present in human-generated levels in order to generate its own. This works by dividing human-generated levels into sequences of micro-patterns and grouping them into meso-patterns to represent the design abstractions present in the levels. By using an evolutionary strategy algorithm [19], alongside three fitness functions that promote the inclusion of patterns, the system is able to create new levels whilst keeping the style it is trying to mimic.

Smith and Mateas [42] introduced a different approach: by defining the design space as a range of answer sets, they can generate content through answer set programming. Both previously mentioned types of algorithms are further explored in Togelius, Justinussen and Hartzen [47]. Here, they create a composition in which a search-based method is used to find the parameters of an answer set program. It is concluded that composition is an ideal framework for PCG as it can be used to circumvent the individual downsides of each type of algorithms and furthermore, it is a framework that supports humans as part of the content creation.

COMPUTATIONAL CREATIVITY

Csikszentmihalyi [16] defines creativity as a process from which results a novel and valuable artifact that contributes to the evolution of culture. He further dissects this notion by stating that creativity is a system composed of three parts: a domain, or set of symbols and rules, a field, or group of people who have the expertise to evaluate given artifacts, and the person from which creativity is originated. As such, creativity can only happen because of the interaction among these three parts.

Boden [3] augments Csikszentmihalyi's definition by introducing new concepts. By dividing the concept of novelty as having a psychological and historical meaning, she introduced the idea of P-creativity, or novel to the person who produced the original artifact, and H-Creativity, a P-Creative artifact that has not appeared in history before. Additionally, Boden presents three types of creativity: combinatorial, transformational and exploratory, each differing in the way the search space is kept static or dynamic throughout the production of artifacts.

Wiggins [53] developed a formalization of Boden's account for creativity. He starts by introducing a mathematical tuple that defines the universe of concepts as well as rules to explore this universe. This sets the foundation for a framework for search algorithms that generalizes upon traditional AI searching methods. Wiggins even states that this framework is capable of more complex searches.

CC attempts to provide the means for computers to express creative behavior. This can be in the form of generating artistic artifacts such as music [13], paintings [7], poems [8], etc., or even solving problems creatively [22]. Computers are, however, argued to lack human factors, such as emotion, and thus cannot be creative [9, 33]. In fact, Colton et al. [9] argue that we should separate human creativity from computer creativity. Nevertheless, computer models of creativity can give insights on how human creativity works [33]. Additionally, there is ongoing research that studies how artificial creativity can be used to stimulate human creativity [9, 34, 53].

All in all, CC tries to model human creativity in an artificial way, either to generate art [41] or to help study creativity at a psychological level. Wiggins' proposed framework encompasses PCG: we may abstract PCG as a search in a conceptual space of designs (artifacts).

COMPUTATIONAL CREATIVITY IN PROCEDURAL CONTENT GENERATION

As mentioned, CC presents automated means to generate creative content. However, while videogames may also be perceived as art [45], there is still little work regarding introducing creativity models in content creation, as stated in Liapis, Yannakakis and Togelius [31]. Because of this, according to Yannakakis, Liapis and Alexopoulos [52], "PCG algorithms are rarely classified as creative." Moreover, this seems to be one of the challenges that PCG research still needs to address [46].

Merrick, et al. [34] argue that creativity models should be used with PCG as "automated game design requires not only the ability to generate content, but also the ability to judge and ensure the novelty, quality and cultural value of generated content." To this end, they developed a system to generate shape grammars, in which a human design is used

ISBN: 978-989-20-5488-9

as a template to generate levels automatically in such way that they "are similar-yet-different to existing human designs" [34]. The notion that creativity should be part of PCG is further explored in Dormans and Leijnen [18]: by assuming that creative artifacts are preferred to those which are not, they provide a PCG technique using a creativity model by introducing an algorithm composed by a module that generates creative and novel, solutions, and another module to evaluate their usefulness.

In the literature, one of the earliest creativity models used alongside PCG is Conceptual Blending. In a nutshell, Conceptual Blending is a notion that explains that creativity can be achieved by mixing preexisting concepts [21]. In Ribeiro, et al. [37], the authors use an authoring tool, named Divago, to generate novel creatures from a preexisting database of decomposable models. Internally, Divago does this by creating conceptual blends and using the database to instantiate those blends. Additionally, created creatures can be fed back to develop even more creations.

There are also design tools that attempt to bring forth creativity through a cooperation between human designers and tools. These tools follow a principal in which the computer plays the role of a colleague [32], a creative peer that works alongside human designers and, according to Yannakakis, Liapis and Alexopoulos [52], they can be classified as mixed-initiative. One such tool is Tanagra [43, 44]. Tanagra is a 2D platformer level design authoring tool. Here, both designers and computers play an active role: designers can define the constraints of the level both spatially and temporally. In addition, they may define gameplay patterns, known as beats, and change the level's layout. The tool, on the other hand, can generate layouts from the given conditions, suggest changes for the level design and assure that the level is playable. Another example is Sentient Sketchbook [30], a tool that generates levels through designer sketches, while the tool evaluates the originated map for gameplay constraints and suggests map layouts.

As a final point, there is some work regarding procedural game design. In Browne and Colton [5], the authors study the extents of creativity expression in exploratory creativity. Browne and Colton defined a closed rule-set for game designers to develop a game, and attempted to determine whether creativity could arise in such a closed system. They concluded that, in fact, creativity could be achieved although the system was constrained. This helped defined future research in implementing the work's findings as a computation design collaborator. Another related work is that of ANGELINA [10, 11, 12]. ANGELINA is an automated game design generator, or, in other words, a computational game designer. In its early versions [10], it generated puzzle games using an evolutionary multi-faceted algorithm. In recent versions, however, ANGELINA 3 [11, 12] is able to create platform games using contextual information from the "The Guardian".

CONCLUSION

To sum up, we presented a paper that surveyed the state-of-the-art regarding CC applied to PCG. While both CC and PCG are study fields with extensive research done and well-defined applications, introducing CC to PCG is still in its infancy, even though both areas share common grounds.

There is still further work that could be done. For instance, while Merrick, et al. [34] argue in favor of an artificial evaluator, there are no further studies regarding the implementation of one in the context of videogames. Artificial evaluation of game content should take into account the context of the game itself, i.e. the genre, setting, and other factors, in order to produce better results as what works in a particular game context may not work in another.

Additional studies should be made to investigate to what extent creativity models enhance PCG. Dormans and Leijnen's work [18] is a step towards this direction yet, further tests should be made to compare the overall results of using traditional PCG with PCG with creativity models.

Most of the literature focuses on applying creativity to game design, in general. PCG is a vast area encompassing several types of content, it would be interesting to assess the results of applying creativity models to content other than game design.

Another interesting aspect that could be further developed is to introduce the enhanced PCG to improve the player's experience. By introducing creativity to procedurally generate content that stimulates, for instance, surprise, the games may improve their overall experience.

Finally, most of the literature points to the direction of fully automated PCG, however, PCG can also be used to empower content designers, as mixed-initiative tools. It would be interesting to explore how a mixed-initiative paradigm can stimulate designers' creativity. In fact, by giving the tools the ability to engage in creative behaviors through underlying creativity models, research can explore to what extents this can stimulate and promote the production of creative artifacts.

ACKNOWLEDGMENTS

This work was supported by the CISUC ICIS project CENTRO-07-0224-FEDER-002003.

REFERENCES

1. Binary Systems. Starflight, Eletronic Arts(1986).
2. Blizzard North. Diablo, Ubisoft (1996).
3. Boden, M. A. Computer Models of Creativity. *AI Magazine 30*, 3(2009), 23-35.
4. Braben, D. and Bell, I. Elite, Acornsoft (1984).
5. Browne C. and Colton, S. Computational Creativity in a Closed Game System. In *Proc. CIG 2012*, IEEE(2012), 296-303.

ISBN: 978-989-20-5488-9

6. Browne, C., Yannakakis G. and Colton, S. Guest Editorial: Special Issue on Computational Aesthetics in Games *IEEE Transactions on Computational Intelligence and AI in Games 4*,3(2012), 149-151.
7. Colton, S. The Painting Fool Teaching Interface. In *Proc. ICCC2010*, 2010.
8. Colton, S., Goodwin J. and Veale, T. Full-FACE Poetry Generation. In *Proc. ICCC2012*, 2012.
9. Colton, S., López de Mántaras R. and Stock, O. Computational Creativity: Coming of Age. *AI Magazine 30*, 3(2013), 11.
10. Cook M. and Colton. S. Multi-Faceted Evolution Of Simple Arcade Games. In *Proc. CIG 2011*, IEEE(2011), 289 - 296.
11. Cook M. and Colton, S. ANGELINA – Coevolution in Automated Game Design. In *Proc. ICCC2012*, 2012, 228.
12. Cook, M., Colton S. and Pease, A. Aesthetic Considerations for Automated Platformer Design. In *Proc. AAAIDE'12*, AAAI(2012), 124-128.
13. Cope D., Computer Models of Musical Creativity, Cambridge: MIT Press(2006).
14. Craveirinha, R., Lucas S. and Licínio R. An Author-Centric Approach to Procedural Content Generation. In *Proc. ACE 2013*, Springer(2013), 14-28.
15. Crawford, C., Process Intensity. *The Journal of Computer Game Design 1*, 5(1987).
16. Csikszentmihalyi, M. *Creativity: the psychology of discovery and invention*, Harper Perennial, New York City, NY, USA, 1997.
17. Dahlskog S. and Togelius, J. Procedural Content Generation Using Patterns as Objectives. In *Proc. EvoGames*, 2014, 1-12.
18. Dormans J. and Leijnen, S. Combinatorial and Exploratory Creativity in Procedural Content Generation. In *Proc. PCGames '13*, Society for the Advancement of the Science of Digital Games(2013), 1-4.
19. Eiben A. E. and Smith J. E. *Introduction to Evolutionary Computing*, Spring, Berlin, Germany, 2007.
20. Evolutionary Games. Galatic Arms Race, Evolutionary Games(2010).
21. Fauconnier G. and Turner, M. The Way We Think: Conceptual Blending And The Mind's Hidden Complexities, Basic Books, New York City, NY, USA, 2002.
22. Hélie S. and Sun, R. Incubation, insight, and creative problem solving: A unified theory and a connectionist model. *Psychological Review 117*, 3(2010), 994-1024.
23. Hendrikx, M., Meijer, S., Van Der Velden, J. and Iosup, A. Procedural Content Generation for Games: A Survey. In *Proc. TOMM*, ACM Press(2013).
24. Interactive Data Visualization, Inc. SpeedTree, 2013.
25. Johnson, L., Yannakakis G. N. and Togelius, J. Cellular automata for real-time generation. In *Proc. PCGames '10*, ACMPress(2010).
26. Khaled, R., Nelson, M. J. and Barr, P. Design metaphors for procedural content generation in games. In *Proc. CHI '13* , ACMPress(2013), 1509-1518.
27. Lagae, A., Lefebvre, S., Cook, R., DeRose, T., Drettakis, G., Ebert, D. S., Lewis, J. P., Perlin K. and Zwicker, M. A Survey of Procedural Noise Functions. *Computer Graphics Forum 29*, 8(2010), 2579-2600.
28. Liapis, A., Yannakakis G. N. and Togelius J. Adapting Models of Visual Aesthetics for Personalized Content Creation. *IEEE Transactions on Computational Intelligence and AI in Games 4*,3(2012), 213-228.
29. Liapis, A., Yannakakis G. N. and Togelius J. Enhancements to Constrained Novelty Search: two-population novelty search for generating game content. In *Proc. GECCO '13*, ACMPress(2013), 343-350.
30. Liapis, A., Yannakakis G. N. and Togelius, J. Sentient Sketchbook: Computer-Aided Game Level Authoring. In *Proc. FDG '13*, ACMPress(2013), 213-220.
31. Liapis, A., Yannakakis G. N. and Togelius, J. Computational Game Creativity. In *Proc. ICCC2014*, 2014, 54-62.
32. Lubart, T. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies - Special issue: Computer support for creativity 63*, 4-5(2005), 365-369.
33. McCormack J. and d'Inverno, M. Computers and Creativity: The Road Ahead. In *Computers and Creativity*, Springer(2012), Berlin, Germany, 421-424.
34. Merrick, K. E., Isaacs, A. and Barlow, M., A Shape Grammar Approach to Computational Creativity and Procedural Content Generation in Massively Multiplayer Online Role-Playing Games. *Entertainment Computing 4*, 2(2013), 115-130.
35. Nareyek, A. Game AI Is Dead. Long Live Game AI! *IEEE Intelligent Systems 22*, 1(2007), 9-11.
36. NaturalMotion. Euphoria, 2008.

ISBN: 978-989-20-5488-9

37. Ribeiro, P., Pereira, F., Marques, B., Leitão B. and Cardoso, A. A Model For Creativity In Creature Generation. In *Proc. GAME-ON'03*, 2003, 175-179.

38. Shaker, N., Togelius, J. and Nelson, M. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Springer, 2014.

39. Sims, K. Evolving 3D Morphology and Behavior by Competition. *Artificial Life 1*,4(1994), 353-372.

40. Smelik, R. M., Kraker, K. J. D., Groenewegen, S. A., Tutenel T. and Bidarra, R. A Survey of Procedural Methods for Terrain Modelling. In *Proc. 3AMIGAS*, 2009.

41. Smethurst, P. Artificial Creativity. *Pi Media 44*, 2012, 24.

42. Smith A. M. and Mateas, M. Answer Set Programming for Procedural Content Generation: A Design Space Approach. *IEEE Transactions on Computational Intelligence and AI in Games 3*, 3(2011), 187-200.

43. Smith, G., Whitehead J. and Mateas, M. Tanagra: A Mixed-Initiative Level Design Tool. In *Proc. FDG '10*, ACMPress(2010), 209-216.

44. Smith, G., Whitehead J. and Mateas, M. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. *IEEE Transactions on Computational Intelligence and AI in Games 3*, 3(2011), 201-215.

45. Smuts, A. Are Video Games Art?. *Contemporary Aesthetics 3*, 2005.

46. Togelius, J., Champandard, A. J., Lanzi, P. L., Mateas, M., Paiva, A., Preuss M. and Stanley, K. O. Procedural Content Generation: Goals, Challenges and Actionable Steps. In *Proc. Artificial and Computational Intelligence in Games*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik(2013), 61-75.

47. Togelius, J., Justinussen T. and Hartzen A. Compositional procedural content generation. In *Proc. PCG '12*, ACMPress(2012), 16.

48. Togelius, J., Kastbjerg, E., Schedl, D. and Yannakakis, G. N. What is Procedural Content Generation? Mario on the borderline. In *Proc. PCGames '11*, ACM Press(2011).

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

50. Toy, M., Wichman, G., Arnold K. and Lane, J. Rogue, 1980.

51. Yannakakis, G. N. Game AI Revisited. In *Proc.CF '12*, ACM Press(2012), 285-292.

52. Yannakakis, G. N., Liapis A. and Alexopoulos, C. Mixed-Initiative Co-Creativity. In *Proc. FDG 2014*, 2014.

53. Wiggins, G. A. Searching for Computational Creativity. *New Generation Computing 24*, 3(2006), 209-222.