A SURVEY OF PROCEDURAL CONTENT GENERATION TOOLS IN VIDEO GAME CREATURE DESIGN

NUNO BARRETO

CISUC / Department of Informatics Engineering
University of Coimbra
Portugal

nbarreto@dei.uc.pt

LICÍNIO ROQUE

CISUC / Department of Informatics Engineering
University of Coimbra
Portugal

lir@dei.uc.pt

Keywords: Creature Generation in Video Games, Interactive Design Tools, Procedural Content Generation

In this paper we provide an analysis of the participation of Humans and Computers in generating creative content, in the context of video games. More specifically, we analyze the role Procedural Content Generation can play as a means to empower designers to produce artifacts in a more efficient manner. We instantiate this analysis by focusing on the creation of creatures as an area that requires further research. In order to make this analysis, we surveyed the state of the art of Procedural Content Generation and Interactive Design Tools, all with the specific mindset of mapping techniques applied to generating of otherwise helping the production of game creatures and the Human participation involved in the process.

1. INTRODUCTION

Over the years video game development has been made accessible for a broad audience. Nowadays both independent and AAA studios work to produce video games hoping they are played by millions of players. However, this development can be either lengthy or costly (Hendrikx, *et al.* 2013; Togelius, Yannakakis, *et al.* 2011) or, in some cases both, and result in an artifact that may not succeed as developers would expect. Because of this, AAA teams, who work with million dollar budgets and teams with hundreds of people, tend to recycle working formulas with minor improvements so that the risk of failing is minimized. As a consequence, this results in a cultural stagnation of video games. Independent teams, on the other hand, can take more risky approaches as they work with smaller teams and budgets. Nevertheless, these teams are occasionally only composed of enthusiasts with less technical and/or artistic expertize required to develop a video game which may hinder, or stop, the development process.

Hendrikx, *et al.* (2013), Togelius, Yannakakis, *et al.* (2011) and Nareyek (2007) argue that *procedural content generation* (PCG) can solve the first problem, and Merrick, Isaacs and Barlow (2013) state that it may solve the second. PCG is an automated means to create game content. Previously used to overcome hardware restrictions, PCG is now employed as a way to increase game content without increasing development costs and automate some development processes. It is even argued, that it may "ultimately yield designs that compete with human imagination" (Togelius, Yannakakis, *et al.* 2011). Later we will argue that, instead of competing with human imagination, PCG could be used to empower the designer. Moreover, it could be used as a design tool to stimulate the designer's creativity.

Merrick, Isaacs and Barlow (2013) state that there is a potential, yet poorly explored, link between PCG and computational models of creativity that could ultimately contribute to the designer's creative process. In fact, according to McCormack and d'Inverno (2012), computers "can transform and inspire human creativity in significantly different ways than any other artificial or human made device". We suggest this link can be explored further to bring forth ways to help designers in a cooperative environment involving human and PCG techniques.

There are also approaches proposing the creation of *interactive design tools* (IDT). Built upon evolutionary algorithms, *interactive evolutionary computation* (Takagi 2001) differs from regular evolutionary algorithms in the sense that a fitness function is replaced by human evaluation. This kind of approaches can be integrated alongside computational models of creativity and PCG as they share a common ground. Nonetheless, there are other methods for IDT that act as suggestion providers (Chaudhuri and Koltun 2010; Marin, Bignon and Lequay 2008) that simulate user's creativity.

In this paper, we provide an analysis of the state of the art in game content creation approaches. More specifically, we instantiate the abstract notion of content design to that of creature design, an area not sufficiently explored as expressed in (Hendrikx, *et al.* 2013). With this specific intent, we survey PCG techniques to later find a common link between both disciplines in the context of IDT.

2. PROCEDURAL CONTENT GENERATION IN GAMES

PCG is an automated means to create game content. Togelius, Yannakakis, *et al.* (2011) presents a taxonomy and survey for PCG techniques and categorizes them according to the techniques' morphology and application. Hendrikx, *et al.* (2013) provides a different approach to classifying PCG. Instead of grouping PCG techniques, Hendrikx, *et al.* developed a layered taxonomy for game content in which PCG can be used. Nonetheless, both these classifications are not mutually exclusive and they can complement each other.

Khaled, Nelson and Barr (2013) proposed yet another means to analyze PCG techniques. In their work, Khaled, Nelson and Barr 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; game-design goal solving PCG algorithms known as Designers; and Experts, monitors and analysts of gameplay data. The authors argue that these metaphors can help better understand and develop PCG.

From the several approaches to PCG including Cellular Automata, Perlin Noise or even Markov Chains, Togelius, Yannakakis, *et al.* (2011) introduce a family of algorithms called search-based PCG. Search-based algorithms gener-

ate content candidates (composed of a genotype, a phenotype and its respective genotype/phenotype mapping function) according to an evaluation function and they include, but are not limited to, evolutionary algorithms. Togelius, Yannakakis, *et al.* state that search-based PCG is a valuable tool but it should not be used to solve every existing PCG problem.

Smith and Mateas (2011) introduced a different approach: by defining the design space as a range of answer sets, they can generate content through the means of answer set programming. Both previously mentioned types of algorithms are further explored in (Togelius, Justinussen and Hartzen, 2012b). Here, Togelius, Justinussen and Hartzen 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 algorithm and furthermore, it is a framework that supports humans as part of the content creation.

PCG, according to Nareyek (2007) and Yannakakis (2012), is the next logical step for Artificial Intelligence (AI) applied to game development. They claim that Non-Playable Character behavior, a former significant challenge in video game AI, has reached a point where it is practically solved and that AI developers should focus instead on using AI techniques to improve PCG and, consequently, use it to aid game designers. Craveirinha, Roque and Santos (2013) also adapt this notion that AI techniques can help develop PCG to assist human designers and introduce an author-centric architecture for PCG.

2.1. PROCEDURAL CONTENT GENERATION OF CREATURES

Current state of the art dictates that a creature is a composition of several elements which include a mesh, or 3D model, animations and behaviors with some initial work being done in sound. This assumes that creatures, as an artificial being, when used in contexts such as simulations or video games are introduced as this composition as opposed to each individual element. Creatures are inserted in the “game bits” layer of the game elements taxonomy presented in (Hendrikx, *et al.* 2013) which they state that their generation is complex and still requires further research.

The following subsections illustrate state of the art approaches to generate meshes, animations and behaviors. It is clear that most of the literature tackles each of these creature generation sub-problems individually and combining them with a set of dependencies could yield interesting results. However, that is beyond the scope of this paper.

2.1.1. ANIMATION

There are some accounts of procedural animation being used for creatures/animals'. Sims (1994) work encompasses an Artificial Life environment in which populations of geometric creatures compete to survive. These creatures, or individuals, are composed of a Neural Network controller that decides which forces are applied to generate movement. These controllers are evolved in an evolutionary process to fit a given environment.

Sims' work served as inspiration for the motion synthesis tool Euphoria (Champandard 2008). Champandard argues that Euphoria does not achieve realistic results as people expect and it is best used when motion capture is not an option. Another work derived by Sims' artificial life project is that of Wampler and Popovic (2009). Here optimization methods are used to generate gaits and to optimize existing morphologies of legged animals. Wampler and Popovic concluded that while their method does not capture all gaits, such as a horse gallop, it can be used to aid an animation designer by optimizing certain aspects of a gait while the designer authors the remaining aspects.

Hecker, *et al.* (2008) employed a system in the video game *Spore* that applies animations to unknown morphologies. Here, animations are created beforehand by designers using semantic information and are then generalized so that they can be used on an arbitrary morphology. This way, it is not necessary to create an animation for each morphology. This system, however, still has some flaws pointed out by Hecker, *et al.* and most importantly still relies on human testing to ensure that the resulting animations are working on a given morphology.

There are other approaches for procedural animation that focus on specific morphologies. Cenydd and Teahan (2012) developed a system for arthropod animations. Cenydd and Teahan argue that this system achieves realistic locomotion behaviors on spiders and other insects

yet they believe that it can be applied on other animals such as mammals. Coros, *et al.* (2011) provided a means to generate locomotion for quadrupeds. By using an abstract skeletal model from which quadrupeds are derived and in conjunction with gait controllers that apply virtual forces to the skeleton, several gaits were achieved. These were then validated through comparison with existing motion data and evaluated for their robustness, i.e., how the model behaves when applied external unexpected forces. From the results, Coros, *et al.* concluded that although their system generated gaits similar to those of a German Shepherd dog there were some motion subtleties that were produced to which Coros, *et al.* argue that can be achieved if complementing their system with a biomechanical model.

2.1.2. MESH

Another part of Sims' work involved the evolution of morphologies of geometric 3D models. In a nutshell, these morphologies were described as a directed graph which went through the evolutionary process alongside a corresponding animation controller. Hudson (2013) also developed an evolutionary process to generate rigged creatures. Using Houdini, a tool built for a PCG game development pipeline, Hudson evolves structures in Houdini's API to create sets of creatures from which the user can choose to evolve further and even crossover with other creatures.

Using conceptual blending, Ribeiro, *et al.* (2003) uses Divago to generate novel creatures from a preexisting database of decomposable models. Divago, much like the tool proposed by Hudson, can be fed by the generated creatures to develop even more creations. Nevertheless, during the time of writing of the article, some modules were not yet tested as it was still a work in progress.

Inspired by computer generated architecture shape grammars, a technique used to generate buildings, Ilcík, *et al.* (2010) developed an extension to these grammars for skeletal generation and consequently, posing models. This extension was used to develop various objects and even organic creatures, including humanoids. Ilcík, *et al.* conclude that there are still some problems, regarding the posing mechanism, that need to be addressed in these extended grammars, including preservation of mass, extreme positions and collisions resolution.

Finally, Simmons, Wilhelms and Van Gelder (2002) present a mechanism to generate animal models that result in the variation of a preexisting model (or canonical as defined by Simmons, Wilhelms and Van Gelder). By using a horse as a canonical model, Simmons, Wilhelms and Van Gelder, built other variations including a pony. A main disadvantage of this model regarding the generation of arbitrary creatures is that it requires the canonical model to be thought of *a priori*, which may not be the case. Another disadvantage, as pointed by the authors, is that some components, such as the muscle model, had to be handmade instead of automatized.

2.1.3. BEHAVIORS

There are two main approaches to develop behaviors for agents in video games: one, named Agent Centric or Bottom Up method, involves the creation of agents as independent entities that have their own goals and objectives that may, or may not, conflict with other agents' goals and objectives. Initial work in this approach was made by Epstein and Axtell (1996). In their work, they provide several, although not procedural, frameworks for emerging social behaviors. The other approach in modelling agent behaviors is known as Master Plan or Top Down method. Essentially, it relies on the creation of controllers that give orders to a collection of agents in order to fulfill some sort of strategic plan or common goal. This latter approach, however, is beyond the scope of this paper as the Agent Centric approach is generally used in the context of creature behavior modelling.

Mateas and Stern (2004) and Young, *et al.* (2004) developed a similar approach in the sense that both set of authors created a means to generate behaviors according to sets of predefined actions. The differences rely on the fact that whilst Young, *et al.* developed an architecture, Mimesis, that can be integrated into game engines so that engines generate plans according to a given goal, Mateas and Stern developed a language, ABL, that supports the definition of behaviors that can even be run in parallel, all of which to achieve a given goal. Both approaches were developed with believable agents in story driven environments in mind.

Kadlec (2008) presented a framework for genetic programming used to evolve behavior trees. By using a tool called Pogamut in the game *Unreal Tournament 2004*,

Kadlec evolved agents, known as bots, to participate in the game's Deathmatch and Capture the Flag modes and ultimately compete with humans. Results indicated that the bots learned simple gameplay rules.

Santos and Roque (2010) provided some initial work in procedural behaviors with Petri Nets, where they created an architecture in which Petri Nets, modelling behaviors of agents in a *Pacman*-based game, are reorganized after each game in order to give birth new behaviors. In their work, Santos and Roque compare a strictly random method to reorganize behaviors with a data-fed technique that changes the nets' topology according to gameplay metrics. Results indicated that, in general, Petri Nets have potential for behavior modeling and, in particular, the latter technique ensued a better game experience.

Finally, in Grand, Cliff and Malhotra (1996) illustrated the architecture behind the agents in *Creatures*. These agents, or Norns, are comprised of a multilobe neural network. Each lobe is responsible for a given task such as decision, perception and concepts and they are fed by an artificial hormone system and sensors which allows Norns to develop instinctive behaviors. Additionally, this artificial system also permits Norns to learn experience-driven behaviors. Grand, Cliff and Malhotra argue that emergent social behaviors were apparently observed though they state it could have been an anthropomorphism in the eye of the beholder.

3. INTERACTIVE DESIGN TOOLS

An IDT is software which aims to aid a user's design process through human-computer interaction. Takagi (2001) surveyed a type of approach called *interactive evolutionary computation*, a family of evolutionary algorithms in which the fitness function is replaced by human evaluation. This brings forth a cooperative environment as both user and computer aspire to reach a global, or local, optimum in the intended design space. The potential of this framework is expressed in Nishino, *et al.* (2001) through the case study of a design tool for 3D modeling for both novice and expert users with promising results. However, Takagi stresses that *interactive evolutionary computation* may yield human fatigue as computers cannot get tired. Takagi presents solutions for this problem including faster fitness convergence, fitness prediction and visual

feedback of the search process. These solutions are supported by conducted subjective tests with apparent positive results.

Other approaches for IDT include *suggestion providers*. These tools aim to stimulate a user's design process by giving ideas using the user's work as basis. Marin, Bignon and Lequay (2008) use genetic algorithms to evolve 3D models using what the user is creating as initial population. The resulting candidates, i.e. deformations of the original model, are then presented to the user as a means to inspire him. Although not built upon genetic algorithms, the tool presented in (Chaudhuri and Koltun 2010) also intends to aid the design of 3D models. However, the approach relies on suggesting shapes from a given database that are ranked according to the user's current model. The main problem with this approach is that the tool is constrained by the quality of the database.

It is clear that PCG techniques can be used as IDT as stated in (Craveirinha, Roque and Santos 2013; Khaled, Nelson and Barr 2013; Yannakakis 2012; Togelius, Yannakakis, *et al.* 2011) but, it is also evident that *creative computing* can be introduced to enhance content generation. Indeed, there are some accounts of works that combine *creative computing* with PCG. Dormans and Leijnen (2013) provide a PCG technique using a creativity model as they assume that PCG that provides creative artifacts are preferred to those who don't. Dormans and Leijnen do this by splitting the algorithm into two in which one serves to creative novel solutions and the other to evaluate their usefulness.

Merrick, Isaacs and Barlow (2013) argues 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". The framework presented in (Merrick, Isaacs and Barlow 2013) also accounts for an IDT as it requires an initial design made by a human. Finally, Ribeiro, *et al.* (2003), as described in section 2.1.2, also uses creativity models for PCG.

4. CONCLUSIONS

To conclude, we surveyed the state of the art approaches for PCG, focusing on creature generation, as it is argued to be not as explored as other areas in PCG. While, each creature's components' generation including meshes,

animations and behaviors have been tackled mostly individually, as shown by the literature, we believe that the generation of creatures can benefit from a composite generation of components as they are linked in the final result. Further research on how animations are dependent on behaviors and the mesh's morphology would enable the characterization of the implications of these relations, to improve the generation of high-level content.

Additionally, we surveyed work done for IDT as we conjecture that PCG can be used to enhance designers' work, as demonstrated in some projects. In fact, IDT have been employed for various applications with positive results. Lastly we propose that combining PCG with creativity models and use those as IDT could yield powerful tools that increase designers' productivity. It is not clear, however, how well can IDT reduce the time it takes to design an artifact and, as such, this could be an interesting research area, as some authors have noticed that unexperienced designers have become more efficient when using IDT.

Since creatures are predominant in videogames, their generation could help reduce development costs. More specifically, IDT, with underlying creativity models, for creature creation could increase the richness of artifacts produced as well as potentially speed their conception. Moreover, research made in this area could set the foundation for other less explored PCG areas such as characters which not only include aforementioned elements present in creatures, but also components tied-in with narrative such as background.

ACKNOWLEDGEMENTS

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

REFERENCES

Cenydd, Llyr ap, and Bill Teahan. "An embodied approach to arthropod animation." *Computer Animation and Virtual Worlds*. 2012. pp. 65–83.

Champandard, Alex J. "NaturalMotion's Euphoria Technology: The Honeymoon is Over." *AIGameDev*. March 14, 2008. http://aigamedev.com/open/editorial/naturalmotion-euphoria/ (accessed October 22, 2013).

Chaudhuri, Siddhartha, and Vladlen Koltun. "Data-Driven Suggestions for Creativity Support in 3D Modeling." *ACM Transactions on Graphics (TOG)*, 2010.

Coros, Stelian, Andrej Karpathy, Ben Jones, Lionel Reveret, and Michiel van de Panne. "Locomotion Skills for Simulated Quadrupeds." *ACM Transactions on Graphics (TOG)*, 2011.

Craveirinha, Rui, Licínio Roque, and Lucas Santos. "An Author-Centric Approach to Procedural Content Generation." *10th International Conference, ACE 2013*. Boekelo: Springer, 2013. pp. 14–28.

Creature Labs. *Creatures*. Mindscape, 1996.

Dormans, Joris, and Stefan Leijnen. "Combinatorial and Exploratory Creativity in Procedural Content Generation." *Fourth Procedural Content Generation in Games workshop*. Crete, 2013.

Epic Games. *Unreal Tournament 2004*. Atari Inc., 2004.

Epstein, Joshua M, and Robert Axtell. *Growing Artificial Societies: Social Science from the Bottom Up.* Washington: Brookings Institution Press, 1996.

Grand, Stephen, Dave Cliff, and Anil Malhotra. "Creatures: Artificial Life Autonomous Software Agents for Home Entertainment." *Autonomous Agents and Multi-Agent Systems 1*, 1996: pp. 39–57.

Hecker, Chris, Ryan W. Enslow, Bernd Raabe, John DeWeese, Jordan Maynard, and Kees van Prooijen. "Real-time Motion Retargeting to Highly Varied User-Created Morphologies." *ACM Transactions on Graphics (TOG)*, 2008.

Hendrikx, Mark, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. "Procedural Content Generation for Games: A Survey." *Multimedia Computing, Communications, and Applications.* New York, 2013.

Hudson, Jon. *Creature Generation using Genetic Algorithms and Auto-Rigging*. Msc. Thesis, NCCA, 2013.

Ilcík, Martin, Stefan Fiedler, Werner Purgathofer, and Michael Wimmer. "Procedural Skeletons: Kinematic Extensions to CGA-Shape Grammars." *Spring Conference on Computer Graphics 2010*. Comenius University, Bratislava, 2010. pp. 177–184.

Kadlec, Rudolf. *Evolution of intelligent agent behaviour in computer games*. Msc. Thesis, Prague: Charles University, 2008.

Khaled, Rilla, Mark J Nelson, and Pippin Barr. "Design metaphors for procedural content generation in games." *SIGCHI Conference on Human Factors in Computing Systems* . New York: ACM, 2013. pp. 1509–1518.

Marin, Philippe, Jean-Claude Bignon, and Hervé Lequay. "A Genetic Algorithm for Use in Creative Design Processes." *28th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*. Minnesota, 2008. pp. 332–339.

Mateas, Michael, and Andrew Stern. "A Behavior Language: Joint Action and Behavioral Idioms." In *Life-Like Characters: Tools, Affective Functions, and Applications*, by Helmut Prendinger and Mitsuru Ishizuka, pp. 135–163. Springer, 2004.

Maxis. *Spore*. Electronic Arts, 2008.

McCormack, Jon, and Mark d'Inverno. "Computers and Creativity: The Road Ahead." In *Computers and Creativity*, by Jon McCormack and Mark d'Inverno, pp. 421–424. Berlin: Springer, 2012.

Merrick, Kathryn E., Amitay Isaacs, and Michael Barlow. "A Shape Grammar Approach to Computational Creativity and Procedural Content Generation in Massively Multiplayer Online Role-Playing Games." *Entertainment Computing*. 2013. pp. 115–130.

Namco. *Pac-Man*. Namco, 1980.

Nareyek, Alexander. "Game AI Is Dead. Long Live Game AI!" *Intelligent Systems, IEEE*. 2007. pp. 9-11.

Nishino, Hiroaki, Hideyuki Takagi, Sung-Bae Cho, and Kouichi Utsumiya. "A 3D Modeling System for Creative Design." *15th International Conference on Information Networking*. Beppu City, 2001. pp. 479-486.

Ribeiro, Paulo, Francisco Pereira, Bruno Marques, Bruno Leitão, and Amílcar Cardoso. "A Model For Creativity In Creature Generation." *GAME-ON'03*. 2003.

Santos, Sérgio, and Licínio Roque. "Ensaio de Reescrita de Comportamentos em Videojogos com Base no ." *IX Simpósio Brasileiro de Games e Entretenimento Digital.* Florianópolis, 2010.

Simmons, Maryann, Jane Wilhelms, and Allen Van Gelder. "Model-based Reconstruction for Creature Animation." *ACM SIGGRAPH Symposium on Computer Animation* (SCA '02). ACM Press, 2002. pp. 139-146.

Sims, Karl. "Evolving 3D Morphology and Behavior by Competition." *Artificial Life*, 1994: pp. 353-372.

Smith, Adam M., and Michael Mateas. "Answer Set Programming for Procedural Content Generation: A Design Space Approach." *Computational Intelligence and AI in Games*. 2011. pp. 187-200.

Takagi, Hideyuki. "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation." *IEEE*. 2001. pp. 1275-1296.

Togelius, Julian, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. "Search-Based Procedural Content Generation: A Taxonomy and Survey." *Computational Intelligence and AI in Games*. 2011. pp. 172-186.

Togelius, Julian, Tróndur Justinussen, and Anders Hartzen. "Compositional procedural content generation." *FDG Workshop on Procedural Content Generation* (PCG). 2012.

Wampler, Kevin, and Zoran Popovic. "Optimal Gait and Form for Animal Locomotion." *ACM Transactions on Graphics (TOG)*, 2009.

Yannakakis, Georgios N. "Game AI Revisited." *9th conference on Computing Frontiers*. New York, 2012. pp. 285-292.

Young, R Michael, Mark O. Riedl, Mark Branly, Arnav Jhala, R J. Martin, and C J. Saretto. "An architecture for integrating plan-based behavior generation with interactive game environments." *Journal of Game Development*, 2004: pp. 51-70.