Guest Editorial: Procedural Content Generation in Games

I. Procedural Content Generation

CREATING games is in essence a nontechnical task and, accordingly, creating computer games involves much more than just programming. First, there needs to be a game design. And then, depending on the game genre, there needs to be character models, maps, levels, boards, racing tracks, trees, rocks, weapons, textures, sound effects, quests, clouds, and so on. In many contemporary game productions, creating all this *game content* requires a significantly larger effort and expense than the actual programming of the game. Therefore, having access to tools that automate some of this content production would be of great and direct benefit to game developers. In addition, such techniques would also indirectly benefit game scholars, as advances in generative techniques are likely to improve our understanding of the content that is designed and of design methods in general.

Fortunately, there are many methods (originating, e.g., from AI, computational intelligence, computer graphics, modeling, and discrete mathematics) that are eminently applicable to generating a large variety of game content. Conversely, efficiently generating proper game content poses a number of interesting challenges to such methods, which can spur the development of new algorithms. Therefore, the field of procedural content generation (PCG) covers a wide set of problems and methods that are of high interest to both game developers and academic researchers.

The call for this special issue mentioned many different aspects of PCG for games, which was clearly reflected in a total of 22 manuscripts being submitted to the IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG) targeted at this special issue. Between November 2010 and June 2011, all these manuscripts underwent a thorough peer-review process, leading to considerable improvement and sharpening of their contributions. Eventually, the present eight papers stood out as specially representative of the current state of the field and were selected for inclusion in the special issue.

II. The Papers

The selected papers present a good combination of surveys, conceptual frameworks, innovative methods, and applications. Together, they cover a considerable range of methods (including constraint solving, stochastic search, planning, and grammars) and of content types (including game rules, buildings, and quests). However, there is a slight bias towards evolutionary computation as a method and game levels as a content domain.

Starting off with "Search-based procedural content generation: a taxonomy and survey," Togelius *et al.* present a taxonomy of PCG in general, and survey work in PCG that employs a "search-based" approach, i.e., that uses evolutionary or other stochastic search/optimization algorithms. A number of published studies are discussed and put into context using the proposed taxonomy. The paper ends with a discussion of open research topics within search-based PCG, and suggestions for how and when to take this approach to a PCG problem.

Next, in "Answer set programming for procedural content generation: A design space approach," Smith and Mateas propose the use of answer set programming (ASP) as a means of formalizing content generation problems, and solving them using the SAT-solving algorithms that underlie ASP solvers. The authors also consider PCG problems as essentially about finding good solutions in a search space, but the representations and search techniques proposed are radically different than those by Togelius *et al.* A couple of examples of this approach are reviewed, and a worked example is given of how to solve a problem using ASP that had previously been solved using evolutionary computation.

In "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," Smith *et al.* describe a system for creating platform game levels. This is a mixed-initiative system, aimed at augmenting the capabilities of human level designers rather than replace them. In Tanagra, the user edits part of a game level, and the system responds by reshaping the adjacent parts of the level so as to fit in with the newly edited segment. The underlying methods used are reactive planning and constraint solving, which might be seen as a form of search.

One PCG technique that is definitely not a form of search is grammar rewriting, as used by Dormans and Bakkes in "Generating missions and spaces for adaptable play experiences." The authors propose a theory of game design as model transformation, and use grammars to generate both levels and missions for *Zelda*-style action adventures. Additionally, some strategies are presented for how to make grammar-based level generation adapt to the needs and preferences of individual players using player models.

Returning to the search-based approach, in "A generic approach to challenge modeling for the procedural creation of video game levels," Sorenson *et al.* present a method for evolving levels for platform games. Key components are the use of a form of player models, as well as the FI-2Pop constraint satisfaction evolutionary algorithm to enforce that the generated levels be playable.

In "Automatic track generation for high-end racing games using evolutionary computation," Lanzi *et al.* describe a working system based on genetic algorithms for generating tracks for an advanced 3-D car racing game. They use two fitness functions: one rewards diversity in terms of track shape, while the other is simulation based (involving an AI driving the

Digital Object Identifier 10.1109/TCIAIG.2011.2166554

car on the track that is being evaluated) and rewards diversity in driving speed.

In "Search-based procedural generation of maze-like levels," Ashlock *et al.* describe a technique for evolving maze levels, similar to those common in game genres such as roguelikes and action adventures. Key techniques of the proposed method combined direct and indirect representations and fitness functions based on dynamic programming.

In "Generating consistent buildings: A semantic approach for integrating procedural techniques," Tutenel *et al.* propose a generic framework aimed at integrating different PCG techniques, so that all disparate content generated by each of them seamlessly combines to form consistent environments, as, e.g., complex buildings (including façade, floor plan layout, furniture, etc.). The integration process is coordinated by a moderator that uses a library of semantic classes and constraints.

III. The Community

Procedural content generation has been sporadically used in games since the early 1980s. However, that has largely been relegated to niche roles and/or niche games, and most methods used in early examples were, by today's standards, rather simplistic. In addition, back then interaction was very scarce between academic researchers and those game developers using PCG in some form.

The last 10–15 years have seen a considerable increase in PCG research, although not always with an explicit application in games. This is apparent in the areas of computer graphics and modeling, witnessed, for example, by the impressive results proposed yearly at the Eurographics and ACM Siggraph conferences, addressing procedural generation of buildings, cities, road networks, vegetation, and other natural phenomena.

Another, more game-focused research thread emerged from the area of AI, when game AI finally became a respectable academic research topic (which, depending on your perspective, might have been as late as the early 2000s). Most of that initial work, though, focused on generating well-performing (occasionally good-looking or believable) strategies for computer players or behavior for nonplayer characters. More recently, however, a number of papers started appearing in conferences such as the IEEE Conference on Computational Intelligence and Games, Artificial Intelligence and Interactive Digital Entertainment, and Foundations of Digital Games, proposing AI and CI solutions to content generation problems. All together, there is clearly an increasing interest and quality in the PCG work being published yearly at these conferences, which led those researchers to become aware of each other's work, thus promoting the growing academic community focused on PCG for games.

For example, the editors of this special issue were involved in the organization of the first two workshops on PCG, which were colocated with the Foundations of Digital Games conferences in 2010 and 2011(http://pcgames.fdg2010.org and http://pcgames.fdg2011.org). We intend to keep organizing workshops in this series, as a natural outlet for new concepts and ideas in PCG research. Additionally, the PCG Task Force was formed within the IEEE Computational Intelligence Society (http://game.itu.dk/pcg/) for researchers interested in this field, with an associated mailing list/discussion group which is free to join (http://groups.google.com/group/proceduralcontent). In a separate effort, Andrew Doull maintains a PCG Wiki, including a catalog of PCG techniques and their usage in games (http://pcg.wikidot.com/).

Another outcome of the efforts to organize the PCG research community is this special issue. We were fortunate to assemble an interesting selection of high-quality papers in this issue. We hope that you will find them to be as interesting and useful as we found the effort to put this issue together.

Julian Togelius, *Guest Editor*
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, 2300 Denmark

Jim Whitehead, *Guest Editor*
Center for Games and Playable Media
University of California, Santa Cruz
Santa Cruz, CA 95064-1077 USA

Rafael Bidarra, *Guest Editor*
Computer Graphics Group
Delft University of Technology
Delft, 2628 The Netherlands

Julian Togelius received the B.A. degree in philosophy from Lund University, Lund, Sweden, in 2002, the M.Sc. degree in evolutionary and adaptive systems from University of Sussex, Brighton, U.K., in 2003, and the Ph.D. in computer science from University of Essex, Essex, U.K., in 2007.

He is currently an Assistant Professor at the IT University of Copenhagen (ITU), Copenhagen, Denmark. Before joining the ITU in 2009, he was a Postdoctoral Researcher at IDSIA in Lugano. His research interests include applications of computational intelligence in games, procedural content generation, automatic game design, evolutionary computation, and reinforcement learning; he has around 50 papers in journals and conferences about these topics.

Dr. Togelius is an Associate Editor of the IEEE Transactions on Computational Intelligence and AI in Games and is the current Chair of the IEEE CIS Technical Committee on Games.

Jim Whitehead (S'94–M'06–SM'08) received the Ph.D. degree in information and computer science from the University of California Irvine, Irvine, in 2000.

He is an Associate Professor in the Computer Science Department, University of California Santa Cruz, Santa Cruz. He was an active participant in the creation of the Computer Science: Computer Game Design major at the University of California Santa Cruz in 2006. His research interests include software evolution, software bug prediction, procedural content generation, and augmented design.

Prof. Whitehead is a member of the Association for Computing Machinery (ACM) and the International Game Developers Association (IGDA). He is the founder and chair of the Society for the Advancement of the Science of Digital Games (SASDG).

Rafael Bidarra graduated in electronics engineering from the University of Coimbra, Coimbra, Portugal, in 1987 and received the Ph.D. degree in computer science from Delft University of Technology, Delft, The Netherlands, in 1999.

He is currently an Associate Professor of Game Technology at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology. He leads the research line on game technology at the Computer Graphics Group. His current research interests include: procedural and semantic modeling techniques for the specification and generation of both virtual worlds and gameplay; serious gaming; semantics of navigation; game adaptivity and interpretation mechanisms for in-game data. He has published many papers in international journals, books, and conference proceedings.

He integrates the editorial board of several journals, and has served in many conference program committees.