Guest Editorial: General Games

I. General Games Research

A general game player (GGP) is a computer program that plays a range of games well, rather than specializing in any one particular game. Such programs have potential benefits for AI and CI research, where the creation of artificial general intelligence remains one of the open grand challenges.

GGPs were first proposed in the 1960s [1] and became a reality in the 1990s with the Metagame system [2] and commercial program Zillions of Games [3], both of which focused on general *Chess*-like games. The specification of the official game description language (GDL) and annual AAAI GGP competitions followed in the first decade of this century, providing a platform for serious academic study into this topic [4]. The recent advent of Monte Carlo tree search (MCTS) methods has allowed the development of truly competitive GGP agents [5], and there is current research interest in applying GGP principles to general video games [6].

In this special issue, we focus on three key aspects of general games research, that are relevant to any GGP:

1) representation: the description and encoding of games define a system's scope (and limitations);
2) interpretation: games must be interpreted correctly and efficiently by any resulting AI agent;
3) exploitation: agents should ideally harness learned knowledge to improve behavior in new contexts.

The eight papers accepted for publication in this special issue of the IEEE Transactions on Computational Intelligence and AI in Games (T-CIAIG) investigate these three key aspects. The aim of this special issue it to provide a snapshot of current work in general games research, and to identify potential launching points for future research into the next generation of general game systems.

A. Representation

Any given GGP will typically have an associated game description language that defines the scope of games that it supports. This issue of representation is probably the most important factor in determining the generality of a given player, and remains an obstacle to true universality while different players exist within their own conceptual frames.

Game description languages tend to be declarative in nature, and can be characterized as being low level or high level (or a mixture of both). A low-level game language describes games in terms of simple, general instructions that update the game state, e.g., using first-order logic. Low-level game descriptions are more general but can be verbose, and the rules of the game are not always obvious from the description. A high-level game language, on the other hand, encapsulates useful game-specific knowledge into conceptual chunks, which are more efficient and convenient to work with at the expense of generality. High-level game languages can be structured to describe games as human designers might conceptualize them, hence can be more comprehensible, and can also include low-level functionality for the best of both worlds.

The first paper in this special issue, "The game description language is Turing complete" by Saffidine [7], shows that the official GDL used in the AAAI GGP competitions is Turing complete, and proposes strengthening the recursion restriction of the original GDL specification to ensure that resulting games will be decidable. This paper finally answers a question raised previously in the literature, which was generally assumed to be true but never proven until now.

Schaul then describes an alternative game description language in his paper "An extensible description language for video games" [8]. His Python-based framework provides an extensible approach for quickly prototyping 2-D video games, to facilitate game design and the benchmarking of experimental learning algorithms. This system combines the benefits of both the low-level and high-level approaches, as it allows the user to work with code fragments at the conceptual level, but also interactively program new behavior at runtime at the code level.

Schmidt describes the extension of the Zillions of Games program in his paper "The axiom general purpose game playing system" [9]. Schmidt's Axiom system overcomes limitations in Zillions of Games to broaden the range of possible games through an extensible game description language based on Forth, in another example of a programming language used to supplement a game description language.

B. Interpretation

GGPs must be able to interpret games within their domain correctly and execute them efficiently. This is especially important in the context of general games, where standard techniques used to optimize dedicated single game programs cannot typically be applied across the board.

Schiffel and Björnsson describe how the choice of interpreter can be critical for GDL players, in their paper "Efficiency of GDL reasoners" [10]. They provide an overview of open-source GDL reasoners currently available, their pros and cons, and relative performance on a number of test cases, and highlight the performance implications when moving from dedicated single game players to general game players.

While most GGPs operate on abstract mathematical models of games, Hausknecht *et al.* take a visual approach to playing general video games based on feedback from the raw pixels on the screen, in their paper "A neuroevolution approach to general Atari game playing" [11]. Their HyperNEAT system, based on neuroevolution techniques, has learned to play 61 Atari video games directly from the console with considerable success, even surpassing human performance in some of the games.

C. Exploitation

The driving motivation behind general games research is the development of artificial general intelligence, and the creation of AI agents able to reason and function in new contexts. To

Digital Object Identifier 10.1109/TCIAIG.2014.2369009

this end, the exploitation of knowledge learned in one context (i.e., game) to other contexts (i.e., other games) is of the utmost interest. The final three papers in this special issue examine different techniques for deriving such learned knowledge and harnessing it in new situations.

In "Self-adaptation of playing strategies in general game playing" [12], Świechowski and Mańdziuk describe how their GGP agent MINI-Player maintains a pool of likely strategies and automatically selects appropriate combinations of these to play a given game. These strategies include concepts such as distance to goal, material count, mobility, and so on. This approach is similar in concept to earlier advisor-based systems [2], [13], but in this case the strategies are used to bias MCTS simulations to good effect.

Similarly, Benbassat and Sipper evolve evaluation functions for games in their paper "EvoMCTS: A scalable approach for general game learning" [14], and use these to guide the playout strategy of their MCTS player. Their approach is scalable and surpasses handcrafted AI players using little domain knowledge and no expert domain knowledge, for a number of abstract strategy games.

Finally, Tak *et al.* investigate domain-independent decay methods for improving simulation-based playout strategies in their paper "Decaying simulation strategies" [15]. They demonstrate that three simple decay methods (called move decay, batch decay, and simulation decay) can significantly improve the performance of MCTS players when used in appropriate combinations, for both turn-taking and simultaneous-move games.

II. Future Directions

To date, general games research has focused on the definition, interpretation, and playing of games. The next logical step is to incorporate a generational component to produce complete general game systems (GGSs) that also evaluate and design games, as procedural content generation (PCG) [16] becomes an increasingly important research topic for both the academic and commercial game sectors. While preliminary studies have demonstrated successful cases of computer-aided game design [2], [17] and fully automated game design through the evolution of rule sets [13], we are yet to see a fully integrated GGS that harnesses its general game expertise for the intelligent design of new games.

Finally, we would like to conclude by again emphasizing the point that each game description language imposes a representational frame around its associated general game player or system. It would be interesting to explore ways to transcend these frames, to allow the cross-pollination of strategic and design concepts between different systems, different types of games, and even between game and nongame domains.

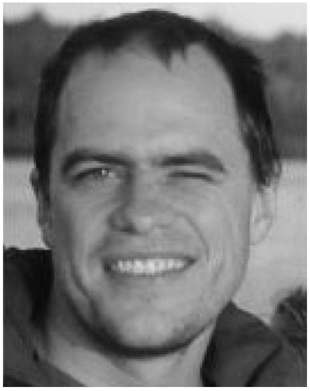
Cameron Browne, *Guest Editor*
Science and Engineering Faculty
Queensland University of Technology
Brisbane, Qld. 4000 Australia
c.browne@qut.edu.au

Julian Togelius, *Guest Editor*
Center for Computer Games Research
IT University of Copenhagen
Copenhagen, 2300 Denmark
julian.togelius@gmail.com

Nathan Sturtevant, *Guest Editor*
Department of Computer Science
University of Denver
CO, 80210 USA
sturtevant@cs.du.edu

References

[1] J. Pitrat, "Realization of a general game-playing program," in *Proc. Int. Fed. Inf. Process. Congr.*, 1968, pp. 1570–1574.
[2] B. Pell, "A strategic metagame player for general chesslike games," in *Proc. 12th Nat. Conf. Artif. Intell.*, Seattle, WA, USA, 1994, vol. 2, pp. 1378–1385.
[3] J. Mallett and M. Lefler, "Zillions of Games—Unlimited board games and puzzles," 1998 [Online]. Available: http://www.zillions-of-games.com
[4] M. R. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the AAAI competition," *AI Mag.*, vol. 26, no. 2, pp. 62–72, 2005.
[5] H. Finnsson and Y. Björnsson, "Simulation-based approach to general game playing," in *Proc. Assoc. Adv. Artif. Intell.*, Chicago, IL, USA, 2008, pp. 259–264.
[6] M. J. Hausknecht, P. Khandelwal, R. Miikkulainen, and P. Stone, "Hyperneat-GGP: A hyperneat-based Atari general game player," in *Proc. Genetic Evol. Comput. Conf.*, Philadelphia, PA, USA, 2012, pp. 217–224.
[7] A. Saffidine, "The game description language is Turing complete," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2354417.
[8] T. Schaul, "An extensible description language for video games," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2352795.
[9] G. Schmidt, "The axiom general purpose game playing system," *IEEE Trans. Comp. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2302303.
[10] S. Schiffel and Y. Björnsson, "Efficiency of GDL reasoners," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2308362.
[11] R. M. M. Hausknecht, J. Lehman, and P. Stone, "A neuroevolution approach to general Atari game playing," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2294713.
[12] M. Świechowski and J. Mańdziuk, "Self-adaptation of playing strategies in general game playing," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2275163.
[13] C. Browne, "Automatic generation and evaluation of recombination games," Ph.D. dissertation, Faculty Inf. Technol., Qld. Univ. Technol. (QUT), Brisbane, Qld., Australia, 2008.
[14] A. Benbassat and M. Sipper, "EvoMCTS: A scalable approach for general game learning," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2306914.
[15] M. J. W. Tak, M. H. M. Winands, and Y. Björnsson, "Decaying simulation strategies," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, Dec. 2014, DOI: 10.1109/TCIAIG.2014.2310782.
[16] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 172–186, Sep. 2011.
[17] I. Althöfer, "Computer-aided game inventing," Faculty Math. Comput. Sci., Friedrich-Schiller Univ., Jena, Germany, Tech. Rep., 2003.



Cameron Browne (M'11) received the Ph.D. degree in computer science from the Queensland University of Technology (QUT), Brisbane, Qld., Australia, in 2008, winning the Dean's Commendation for Outstanding Thesis.

He is currently a Senior Research Fellow in the Science and Engineering Faculty at QUT. He holds a Vice-Chancellor's Research Fellowship for the project *Games Without Frontiers*, in which he is developing the AUTOGAME general game system. He is the author of the books *Hex Strategy: Making the Right Connections* (Natick, MA, USA: A K Peters/CRC Press, 2000), *Connection Games: Variations on a Theme* (Natick, MA, USA: A K Peters/CRC Press, 2005), and *Evolutionary Game Design* (New York, NY, USA: Springer-Verlag, 2011), which won the 2012 GECCO "Humies" award for human-competitive results in evolutionary computation.

Dr. Browne is a Section Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES and the *International Computer Games Association Journal*.

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

He is an Associate Professor at the Center for Computer Games Research, IT University of Copenhagen, Copenhagen, Denmark. He works on all aspects of computational intelligence and games and on selected topics in evolutionary computation and evolutionary reinforcement learning. His current main research directions involve search-based procedural content generation in games, game adaptation through player modeling, automatic game design, and fair and relevant benchmarking of game AI through competitions.

Dr. Togelius is a past chair of the IEEE Computational Intelligence Society (CIS) Technical Committee on Games, and an Associate Editor of the IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND GAMES.

Nathan Sturtevant received the B.Sc. degree in electrical engineering and computer science from the University of California Berkeley, Berkeley, CA, USA, in 1996 and the M.Sc. and Ph.D. degrees in computer science from the University of California Los Angeles (UCLA), Los Angeles, CA, USA, both in 2003.

He was a Postdoctoral Researcher at the University of Alberta, Edmonton, AB, Canada. He has been an Assistant Professor in the Computer Science Department, University of Denver, Denver, CO, USA, since 2010. His primary research involves state–space search techniques, including combinatorial search, path planning, and adversarial search. He is particularly interested in large-scale parallel search, and also works on learning in agent-centered search. He has applied this research to many application games, such as the game *Dragon Age: Origins*, as well as other domains.