Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

General Video Game AI: Competition, Challenges, and Opportunities

Diego Perez-Liebana
University of Essex
Colchester CO4 3SQ, UK
email: dperez@essex.ac.uk

Spyridon Samothrakis
University of Essex
Colchester CO4 3SQ, UK
email: ssamot@essex.ac.uk

Julian Togelius
New York University, 2 Metrotech Center
Brooklyn, 11201 New York
email: julian@togelius.com

Simon M. Lucas
University of Essex
Colchester CO4 3SQ, UK
email: sml@essex.ac.uk

Tom Schaul
New York University
715 Broadway, 10003, New York
email: schaul@gmail.com

Abstract

The General Video Game AI framework and competition pose the problem of creating artificial intelligence that can play a wide, and in principle unlimited, range of games. Concretely, it tackles the problem of devising an algorithm that is able to play any game it is given, even if the game is not known a priori. This area of study can be seen as an approximation of General Artificial Intelligence, with very little room for game-dependent heuristics. This short paper summarizes the motivation, infrastructure, results and future plans of General Video Game AI, stressing the findings and first conclusions drawn after two editions of our competition, and outlining our future plans.

Introduction

Games have long been popular benchmarks for Artificial Intelligence. Many researchers have studied algorithms and techniques that try to approximate optimal play in computer games as different as Chess, Go, Car Racing games, Ms. PacMan, Real-Time Strategy (RTS) games and Super Mario Bros. Sometimes these research themes are accompanied by some sort of competition that puts different approaches to a test in an unified benchmark. Research on games has enabled some interesting advances in algorithmic AI, such as the use of parallelized Alpha-Beta pruning (in Chess), or the progress seen in one of the most popular algorithms in Game AI, Monte Carlo Tree Search (MCTS), in the game of Go.

Though the contributions made by game-specific research are truly important, game specific competitions have a caveat because of their inherent structure: most of the solutions proposed tend to over-specialize to the domain they are applied in. In other words, the nature of the challenge, or even the struggle to win the contest, encourages the participants to provide the algorithm with highly tailored heuristics that would work only in the game that is being used for the challenge. For example, a world-champion StarCraft agent cannot play Pac-Man at all, raising the question how much it advances the state of the art in AI.

Some attempts have been made to tackle this issue, such as the General Game Playing (GGP) competition organized by the Logic group of the Stanford University (Genesereth and Love 2005). This contest promotes the creation of players that could play any board game, defined in a Game Description Language (GDL). During the competition, the agents initially receive the description of the game and subsequently at least $1s$ of decision time for each move.

General Video Game Playing (GVGP), however, proposes a framework for artificial general intelligence where the time to decide a move is measured in milliseconds, and the game rules are not given to the agents. In this respect, our approach is similar to, but more open-ended than the Arcade Learning Environment (ALE), developed by Bellamare et. al. (Bellemare et al. 2013) and used by e.g. Google DeepMind (Mnih et al. 2015). ALE includes games from the Atari 2600 emulator, which are simple 2D arcade games with rudimentary graphics. ALE controllers are given as input the raw screen capture and a score counter, and they must be able to indicate the set of buttons that determines the next action to make in the game.

The GVGAI Framework

While ALE works on a limited set of games (provided by the Atari 2600 emulator) and uses screen capture recognition to extract the game state, our framework provides this information in an object-oriented manner, and employs a Video Game Description Language (VGDL; (Ebner et al. 2013; Schaul 2013)) to define games in a more general way.

VGDL is inspired by GDL (Love et al. 2008), which is used for the GGP competitions. However, while GDL describes games based on logical rules (or *true facts*), VGDL defines the entities and interactions that can take place in the game. The dynamics of each one of these components, such as movements and behaviour, are programmed in the framework. Albeit this requires that a given functionality is implemented in the system when creating new games, it provides more flexibility and a larger space of games that can be designed. While GDL is restricted to board games, VGDL enables the definition of a wide range of arcade and puzzle games.

VGDL allows for the definition of 2-dimensional single-player games and levels with text files. Different types of sprites can be defined, parametrized, given a 2D position and a rectangular size. Interactions between sprites occur when they collide with each other, and the consequences are defined in the game rules. For a full description of how games and levels are defined in VGDL, see (Schaul 2013). Currently about 80 games are implemented in VGDL, including versions of classic eighties games such as Pac-Man, Boulder Dash, and Space Invaders.

The GVGAI framework, written in Java, provides an object-oriented interface for creating agents that can play in any game defined in VGDL. As defined in the initial track of the competition (*Planning* track), an agent suitable to work in this framework must be able to indicate moves in a real-time fashion. After a short initialization phase, the agent receives calls at every game step and must return a discrete action to apply in no more than $40ms$.

The agent receives information about the game state via a Java object. This object allows the agent to query the game status (winner, time step, score), the player's state (position, orientation, resources), history of events or collisions during the game, and position of the different sprites in the level, identified only by an integer *id* for its type. Note that the rules of the game, the dynamics of the present sprites, or the victory requirements are never provided to the player. It is the agent's responsibility to discover the game mechanics while playing. Additionally, one of the main resources the agents have to reason about the environment is the forward model provided by the framework. This tool allows the agent to simulate actions and roll the game forward to one of the possible next states (the transition may be stochastic).

The GVGAI Competitions

Currently only the *Planning* track of the competition is running. It works as follows: competitors use on one or more sets of available games to train on (all games are provided with 5 different levels). Additionally, there is a competition server[1] set up to receive submissions and run the controllers on an unknown set of 10 games for validation. Participants can submit multiple times to this set of games, and a ranking table is provided on the website. The contestants do not have access to these secret games nor the levels bundled with them. The final rankings are obtained from the execution of all submissions in a third (secret) set of 10 games. For more details on the rules and sets of games, see (Perez et al. 2015).

Two editions of the GVGAI competition have taken place at the time of this writing. The first, held during the IEEE Conference on Computational Intelligence and Games (CIG) in 2014, received 14 submissions, and it was won by Adrien Couëtoux, with an approach inspired by Hierarchical Open-Loop Optimistic Planning. The second edition, was associated with three 2015 conferences: ACM GECCO, IEEE CIG and IEEE CEEC. The total number of submissions received for this edition was higher than 70, with different approaches such as tree search methods, evolutionary algorithms (EA) and combinations of different techniques.

The winner of the GECCO 2015 leg, *YOLOBOT*, put forth a combined approach between MCTS, Breadth First Search and targeting heuristics. The winning entry at CIG 2015, *Return42*, submitted a multiple heuristic controller based on an evolutionary algorithm, random walks and A*. Finally, the winners of the CEEC 2015 leg, *YBCriber*, combined reactive avoidance of hazards with Iterative Widening (IW) in their tree search algorithm.

[1] www.gvgai.net

Conclusions: Challenges and Opportunities

It seems clear that the challenge posed by GVGAI is far from solved. To begin with, the competitions received multiple entries with very different approaches, and no one particular algorithm dominates the rest. In fact, it seems that trying to analyze the type of game played, and selecting one or another algorithm in sequence (as a sort of *Hyper-Heuristic)* improves the results obtained. Additionally, the average of games won by the different winners across all editions of the contest is around 50%. This indicates that the best controllers lose approximately 1 out of every 2 games.

The future of the GVGAI competition, however, is not limited to the *Planning* track. The following tracks are currently in preparation:

- 2/N-Player track: Enhancement of VGDL and the GVGAI framework to allow multi-player games for competition and/or collaboration between agents.
- Learning track: A track with no forward model, where the agents will play each game a determined (large) number of times before being evaluated.
- Procedural Content Generation track: the submission will consist of a general level generator, able to automatically create levels for any game it is given.

Other modifications in mind are games with continuous action spaces, analysis of the game state via screen capture and games with partial observability. GVGAI has proven to be an interesting challenge, hence it has captured the interest of many researchers. Our current and future plans for this framework and competition will open and maintain several lines of interesting research for the game AI community.

References

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: an evaluation platform for general agents. *Journal of Artificial Intelligence Research* 47(1):253–279.

Ebner, M.; Levine, J.; Lucas, S.; Schaul, T.; Thompson, T.; and Togelius, J. 2013. Towards a Video Game Description Language. *Dagstuhl Follow-up* 6:85–100.

Genesereth, M., and Love, N. 2005. General Game Playing: Overview of the AAAI Competition. *AI Magazine* 26:62–72.

Love, N.; Hinrichs, T.; Haley, D.; Schkufza, E.; and Genesereth, M. 2008. General game playing: Game description language specification.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Perez, D.; Samothrakis, S.; Togelius, J.; Schaul, T.; Lucas, S.; Couëtoux, A.; Lee, J.; Lim, C.-U.; and Thompson, T. 2015. The 2014 General Video Game Playing Competition. *IEEE Transactions on Computational Intelligence and AI in Games* (to appear) DOI: 10.1109/TCIAIG.2015.2402393.

Schaul, T. 2013. A Video Game Description Language for Model-based or Interactive Learning. In *Proceedings of the IEEE Conference on Computational Intelligence in Games*, 193–200. Niagara Falls: IEEE Press.