The Mario AI Championship 2009–2012

Julian Togelius, Noor Shaker, Sergey Karakovskiy and Georgios N. Yannakakis

Abstract

We give a brief overview of the Mario AI Championship, a series of competitions based on an open source clone of the seminal platform game *Super Mario Bros*. The competition has four tracks. The gameplay and learning tracks resemble traditional reinforcement learning competitions, the Level generation track focuses on the generation of entertaining game levels, and the Turing Test track focuses on human-like game-playing behaviour. We also outline some lessons learned from the competition and its future. The paper is written by the four organisers of the competition.

1 Origins

AI and machine learning are in constant need of better benchmarks. In reinforcement learning, the choice has long stood between simplistic toy problems such as pole balancing and the Mountain Car, and complex, slow and non-replicable robot problems. Within the CI/AI in Games community, a series of competitions has grown up where competitors submit controllers for modified or reconstructed versions of existing computer games. Using existing computer games as AI benchmarks brings several benefits, the most important being that the games are almost guaranteed to contain interesting AI challenges by virtue of being popular among human players. (One of the most important reasons games are engaging to humans is that they provide learning challenges [8]). Almost as important is that good scoring mechanisms are available, that the visual aspects of the games make it easy to compare and characterise the performance of the controllers, and that it is easy to engage both students and the general public in the competition. Several recently introduced competitions are based on games such as Ms. Pac-Man [10], the first-person shooter Unreal Tournament [6], the real-time strategy game StarCraft and the car racing game TORCS [9].

In 2009, the first and the third author of this paper set out to create a benchmark for game AI controllers based on *Infinite Mario Bros* (IMB). IMB is an open source clone (created by Markus Persson, who later went on to create Minecraft) of Nintendo's platform game *Super Mario Bros* (SMB), which has been one of the world's most influential games since its release in 1985. The core gameplay task in IMB, like in SMB, is to guide the player character Mario from the start to the end of a two-dimensional world without getting killed by enemies or falling down gaps, and while collecting coins and power-ups. Unlike SMB,

1

IMB features in-game procedural generation of levels, thus the word "Infinite" in its title.

Creating the first version of the Mario AI Benchmark involved significant re-engineering of the core loops of the game, making all timing optional (so that the benchmark can run several thousands times faster than the original game on a modern computer) and removing all sources of stochasticity. It also involved creating an interface for Mario controllers. In this interface, the controller receives a $22 * 22$ array representing the area around Mario plus some additional state variables, and returns one of 16 possible actions at each time step, where a time step is 40 ms.

2   The Gameplay Track

The first competition was run in August 2009, and constituted only what later became known as the Gameplay track. In this track, controllers are scored proportionally to how far towards the goal they get on 40 levels generated by the game's level generator. The initial publicity for the competition was picked up by several international news media, such as *New Scientist*, *Le Monde* and *MSNBC*. This naturally led to great interest in the competition, not only from academic researchers. One of the competitors, Robin Baumgarten from London's Imperial College, released a video of his submission on YouTube, where it gathered around a million views [1]. The video, showing Baumgarten's controller playing through a level at breakneck speed and executing several move sequences that would have been masterful had they been performed by a human player, had the dual effect of both attracting further attention to the competition and dissuading some competitors as they thought there was no way they could beat Baumgarten's controller (a screenshot from the video can be see in Figure 1). Interestingly, all that this controller did was to search through state space using A*. The competition attracted 15 submissions and Baumgarten went on to win, managing to clear all the levels. Though there were several submissions based on various learning techniques, including evolutionary computation and neural networks, none of them performed remotely as well as techniques based on search in state space. The first year's competition, along with all submitted controllers, is described further in [18].

The 2010 Mario AI Championship ran in association with three different conferences (EvoStar, IEEE CEC and IEEE CIG), and all competition events included the Gameplay track. In order to keep the competition relevant we needed to increase the challenge, so that there was a real difference between the top-ranking controllers. We observed that none of the controllers were able to handle levels that included "dead ends", i.e. where there is more than one path, not all paths are possible to continue to the end of the level along, and it is not possible to decide in advance which path is the right one. Choosing the wrong path at such a junction forces the player to backtrack and choose another path. While an A* agent would in theory be able to find the right path

---

[1]http://www.youtube.com/watch?v=DlkMs4ZHHr8

Figure 1: Robin Baumgarten's winning Mario-playing agent in action. The red lines represent possible future paths.

given infinite time, in practice any straightforward implementation encounters a combinatorial explosion of possible paths and times out.

We therefore modified the level generator to enable the generation of levels with dead ends, which from the perspective of the controller also amounted to diminishing the observability of the environment. As expected, all controllers that were based on pure search in state space performed worse in the 2010 edition of the competition. While none of the nine submitted controllers were able to clear all of the generated levels, the winner of the competition was the REALM agent by Slawomir Bojarski and Clare Bates Congdon of the University of Southern Maine. REALM uses an interesting hybrid architecture, where a set of rules governing high-level behaviour is created by an evolutionary algorithm (the ruleset is evolved "offline" and frozen before submission) but the execution of these rules is done by state space search using A* [1].

In 2011 and 2012, the Gameplay track saw very few entrants, falling below the five submission minimum we set for calculating an official result. This is somewhat puzzling, as the Mario AI Benchmark software as used in the Gameplay track has seen a good uptake as an AI benchmark, with a number of papers published using this software by people who did not participate in the competition (for example, papers have been published on dimensionality reduction for behaviour learning [5, 12], evolving behaviour trees [11], learning multimodal networks [13], and entertainment measurement [4], as well as on various hybrid game-playing approaches [17, 16, 19, 3, 2]). The situation is to some extent our own fault, as we had not provided a reliable infrastructure for submission and public record-keeping for the Gameplay and Learning tracks. This also means that REALM is still the state of the art for efficient Mario playing. The Gameplay and Learning track of the 2009 and 2010 competitions are described in more detail in [7].

3 The Learning Track

Seeing that controllers that employed any form of online learning performed poorly in the 2009 competition, we created the Learning track. This track is designed to reward controllers that learn to play a single level as well as possible. While it should in principle be possible to clear any level without having seen it before, clearing it in the shortest time possible and with the highest score usually requires knowing it in advance. Therefore, each controller submitted to the Learning track plays the same level 10000 times in order to have a chance to adapt its strategy, and is then scored (using the same scoring as in the Gameplay track) on its 10001st playthrough. 2010's championship only saw four entrants, and the winner was a version of the REALM agent which used its evolutionary algorithm "online" to keep adapting its ruleset after submission. No official results were calculated in 2011 or 2012 due to lack of submissions.

4 The Turing Test Track

One of the outstanding features of the viral video of Baumgarten's controller was how un-humanlike its behaviour was. For example, it always ran, always jumped off a platform at the very last pixel, and performed moves that relied on superhuman precision and timing. In computer game development, it is often more important for non-player characters to be believable than to be well-playing (the computer game can usually "cheat" with impunity). The problem of believable or human-like behaviour is currently understudied in game AI. Some would argue that generating human-like behaviour is just a question of "dumbing down" AI behaviour, but this is contraindicated by the characteristically machinelike behaviour of characters in many games, and the difficulty of creating controllers that behave in a human-like manner in the 2k BotPrize [6]. Ultimately, it is an empirical question what is involved in creating believably human-like behaviour in a game such as SMB.

To address this problem, we created a track of the Mario AI Championship dedicated to human-like game-playing. The idea was to perform a form of spectator-only Turing test for the controllers. Each submitted controller was played on three different levels of varying difficulty, and a video recorded of each playthrough. Videos were also recorded of human players. About a hundred spectators were shown selections of these videos via a web interface, and asked which of the players they thought were humans and which they thought were computers (each page presented a randomly selected pair of videos).

We received three entries specifically for the Turing test track of the 2012 championship. While none of the controllers were as human-like as the most human-like human, the best controller managed to convince 26% of spectators that it was a human player. The winning entry was submitted by a team led by Stefan Johansson at Blekinge Institute of Technology, and was based on artificial potential fields. A qualitative analysis of videos and comments from spectators indicate that signs of hesitation such as standing still and cancelling attempted

actions were seen as particularly human-like behaviour. A description of the methodology and participants for this track can be found in [14].

5   The Level Generation Track

The fourth track of the Mario AI Championship is based on the same software, but differs quite drastically in that competitors do not submit controllers that play the game but rather level generators that generate levels for it. Procedural content generation, where algorithms are used to create game content such as levels, maps, quests and rules, is an emerging research direction within game AI research, answering to a strong need within the game industry to control development costs and enable new forms of adaptive games. As the young field of procedural content generation lacks good benchmarks, the Level Generation track of the Mario AI Championship was designed as the first competition focused on level generation. Entrants submit level generators that generate playable levels for particular players. The evaluation consists of an event where human players first play a tutorial level, and their performance on that level is logged and sent to two different submitted level generators. Each player then plas two levels generated specifically for him/her by the two generators, and selects which one he/she prefers.

This track was run in 2010 and again in 2012. In 2010, six teams entered the competition showcasing vastly different approaches to level generation. The competition was won by Ben Weber of the University of California, Santa Cruz, who used a simple technique where levels were generated by "scanning" in multiple passes along the level, in each pass adding a new type of level item. In general, there was a negative correlation between the complexity of the approach and players' appreciation of the results, with those submissions that attempted to adapt levels to detected playing styles using optimisation techniques finishing at the bottom of the league. Information about the methodology and entrants can be found in [15].

In 2012, we had 11 submissions and the winners were Ya-Hung Chen, Ching-Ying Cheng and Tsung-Che Chiang from National Taiwan Normal University who used gameplay data to calculate the player's score along multiple dimensions such as interacting with coins, enemies and blocks and used these scores to generate the levels by alternating between different types of zones.

6   Lessons learned

Four years of running the Mario AI Championship has taught us a few things about what to do and what not to do when running a game-based AI competition. Let's start with what we did right. Basing the competition on a version of a famous game and keeping an active presence in social media helped getting attention for the competition. More importantly, we keep all of the software open source and encourage all competitors to open-source their entries. We also

went to great lengths to ensure that the API is very simple to use – the target was for anyone familiar with Java to be able to have a controller up and running within five minutes – and that the framework is computationally lightweight and lacks external dependencies. These factors together seem to be responsible for the impressive adoption of the software for both research (we have lost track of the research teams and papers using the software) and teaching (the software is used for assignments in dozens of AI classes worldwide).

However, most of the controllers and level generators developed for research purposes have not been submitted to any track of the competition, and after 2010 the Gameplay and Learning tracks have effectively stagnated. We believe the main reason for this is our failure to keep a central, coordinated source of information for the competition. While there is a portal web page[2] and a mailing list[3], there has been confusion regarding which version of the software and which settings have been used for particular competition events, and it has not always been easy to find updated information about results or the submitted controllers themselves. This has diminished the value of the software as a benchmark. A better example to follow is the Ms. Pac-Man competition, which evaluates controllers automatically via a web interface and keeps an updated league table at all times.

7   The Future of the Competition

The Mario AI Championship is currently being relaunched for 2013 under the name "The Platformer AI Competition", with reworked graphics and sounds from the open-source game *SuperTux* [4]. The name and graphics change is meant to avoid problems associated with Nintendo's ownership of the name "Mario" and the Mario graphics, but also to signify a reinvigoration of the competition taking into account seveal of the lessons we've learned while running the Mario AI Championship. In particular, the new competition will adopt a better approach to making canonical benchmark code available for each competition event, and making the code of competition submissions available. Initially we will concentrate on the Turing Test and Level Generation tracks, given that these contain what we see as the currently most fertile research challenges and as they seem to draw the most interest from the academic community. The first event of the new competition will be held at the IEEE CIG conference, August 11-13 in Niagara Falls, Canada.

The authors

Julian Togelius is an Associate Professor at the IT University of Copenhagen. He works on various topics in computational intelligence and games. His current

---

[2] http://www.marioai.org
[3] https://groups.google.com/group/mariocompetition
[4] http://platformerai.com

main research directions involve search-based procedural content generation in games, automatic game design, and fair and relevant benchmarking of game AI through competition. Togelius holds a BA from Lund University (2002), an MSc from the University of Sussex (2003), and a PhD from the University of Essex (2007).

Noor Shaker is a postdoc at the IT University of Copenhagen. She received a 5-year BA in IT Engineering (2007) from Damascus University, an M.Sc. in Artificial Intelligence (2009) from Katholieke Universiteit Leuven, and a PhD from the IT University of Copenhagen (2013). Her research interests include player modelling, procedural content generation, affective computing and player behaviour imitation.

Sergey Karakovskiy is a PhD student at St. Petersburg State University. He received a 5-year diploma (2008) in Applied Mathematics and Engineering from St. Petersburg State University and an M.Sc. degree in Informatics from the University of Lugano (2010). His research interests include artificial intelligence, computational intelligence in games, neuroevolution, theories of interestingness and artificial curiosity, reinforcement learning and brain-computer interfaces.

Georgios N. Yannakakis is an Associate Professor at the University of Malta. He received an M.Sc. (2001) in Financial Engineering from the Technical University of Crete and a Ph.D. in Informatics from the University of Edinburgh (2005). His research interests include user modelling, neuroevolution, computational intelligence in games, cognitive modelling and affective computing, emergent cooperation and artificial life.

References

[1] Slawomir Bojarski and Clare Bates Congdon. Realm: A rule-based evolutionary computation agent that learns to play mario. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 83–90, 2010.

[2] Nobuto Fujii, Yuichi Sato, Hironori Wakama, and Haruhiro Katayose. Autonomously acquiring a video game agents behavior: Letting players feel like playing with a human player. In *Advances in Computer Entertainment*, pages 490–493. Springer, 2012.

[3] Sergiu Goschin, Ari Weinstein, Michael L Littman, and Erick Chastain. Planning in reward-rich domains via pac bandits. In *Workshop on Reinforcement Learning*, page 25, 2012.

[4] Zahid Halim, A Rauf Baig, and Hasan Mujtaba. Measuring entertainment and automatic generation of entertaining games. *International Journal of Information Technology, Communications and Convergence*, 1(1):92–107, 2010.

[5] Hisashi Handa. Dimensionality reduction of scene and enemy information in Mario. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1515–1520, 2011.

[6] Philip Hingston. A new design for a Turing test for bots. In *IEEE Conference on Computational Intelligence and Games (CIG)*, pages 345–350, 2010.

[7] Sergey Karakovskiy and Julian Togelius. The Mario AI Benchmark and Competitions. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):55–67, 2012.

[8] Raph Koster. *A theory of fun for game design*. Paraglyph Press, 2005.

[9] Daniele Loiacono, Pier Luca Lanzi, Julian Togelius, Enrique Onieva, David A. Pelta, Martin V. Butz, Thies D. Lönneker, Luigi Cardamone, Diego Perez, Yago Saez, Mike Preuss, and Jan Quadflieg. The 2009 simulated car racing championship. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2):131–147, 2010.

[10] Simon M Lucas. Ms Pac-Man competition. *ACM SIGEVOlution*, 2(4):37–38, 2007.

[11] Diego Perez, Miguel Nicolau, Michael O'Neill, and Anthony Brabazon. Evolving behaviour trees for the Mario AI competition using grammatical evolution. In *Applications of Evolutionary Computation*, volume 6624 of *Lecture Notes in Computer Science*, pages 123–132. Springer Berlin / Heidelberg, 2011.

[12] Stephane Ross and J. Andrews Bagnell. Efficient reductions for imitation learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[13] Jacob Schrum and Risto Miikkulainen. Evolving multimodal networks for multitask games. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):94–111, 2012.

[14] Noor Shaker, Julian Togelius, Georgios N. Yannakakis, Likith P. K. Satish, Vinay S. Sathiraj, Stefan J. Johansson, Robert G. Reynolds, Leonard Kinnaird-Heether, Tom Schumann, and Marcus Gallagher. The turing test track of the 2012 Mario AI championship: Entries and evaluation. In *IEEE Conference on Computational Intelligence and Games (CIG) (in press)*, 2013.

[15] Noor Shaker, Julian Togelius, Georgios N. Yannakakis, Ben Weber, Tomoyuki Shimizu, Tomonori Hashiyama, Nathan Sorenson, Philippe Pasquier, Peter Mawhorter, Glen Takahashi, Gillian Smith, and Robin Baumgarten. The 2010 Mario AI championship: Level generation track. *IEEE Transactions on Computational Intelligence and Games*, 2011.

[16] Shunsuke Shinohara, Toshiaki Takano, Haruhiko Takase, Hiroharu Kawanaka, and Shinji Tsuruoka. Search algorithm with learning ability

for Mario AI–combination A* algorithm and Q-learning. In *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD)*, pages 341–344. IEEE, 2012.

[17] Ereck R Speed. Evolving a Mario agent using cuckoo search and softmax heuristics. In *International IEEE Consumer Electronics Society's Games Innovations Conference (ICE-GIC)*, pages 1–7. IEEE, 2010.

[18] Julian Togelius, Sergey Karakovskiy, and Robin Baumgarten. The 2009 Mario AI competition. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2010.

[19] Jyh-Jong Tsay, Chao-Cheng Chen, and Jyh-Jung Hsu. Evolving intelligent Mario controller by reinforcement learning. In *International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 266–272. IEEE, 2011.