

ISSN 2186-7437

NII Shonan Meeting Report

No.130

Artificial General Intelligence in Games: Where Play Meets Design and User Experience

Ruck Thawonmas
Julian Togelius
Georgios N. Yannakakis

March 18–21, 2019



National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Artificial General Intelligence in Games: Where Play Meets Design and User Experience

Organizers:

Ruck Thawonmas (Ritsumeikan University, Japan)

Julian Togelius (New York University, USA)

Georgios N. Yannakakis (University of Malta, Malta)

March 18–21, 2019

Plain English summary (lay summary): Arguably the grand goal of artificial intelligence (AI) research is to produce machines that can solve multiple problems, not just one. Until recently, almost all research projects in the game AI field, however, have been very specific in that they focus on one particular way in which intelligence can be applied to video games. Most published work describes a particular method—or a comparison of two or more methods—for performing a single task in a single game. If an AI approach is only tested on a single task for a single game, how can we argue that such a practice advances the scientific study of AI? And how can we argue that it is a useful method for a game designer or developer, who is likely working on a completely different game than the method was tested on? This Shonan meeting aims to discuss three aspects on how to generalize AI in games: how to play any games, how to model any game players, and how to generate any games, plus their potential applications. The meeting consists of 17 discussions on relevant topics. Findings of this meeting can be found in the discussions’ abstracts, which include overviews of respective topics, highlights of research questions, their potential answers, and future directions.

Arguably the grand goal of artificial intelligence (AI) research is to produce machines with general intelligence or artificial general intelligence: the capacity to solve multiple problems, not just one. Video (or computer) games are one of the most promising research platforms for the study of general intelligence [1], which was pointed out as early as May 2012 by a group of participants at Dagstuhl Seminar 15051 “Artificial and Computational Intelligence in Games: Integration” (cf. their report entitled General Video Game Playing¹). Almost seven years after the aforementioned seminar, the first Dagstuhl Seminar related to AI in video games, we brought to Shonan Meeting prominent researchers and

¹<http://drops.dagstuhl.de/opus/volltexte/2013/4337/pdf/8.pdf>

rising-star young colleagues in relevant areas to discuss research topics related to the meeting theme: Artificial General Intelligence in Games: Where Play Meets Design and User Experience. The meeting was inspired by and based on the vision paper *General General Game AI* [2].

This meeting followed a typical Dagstuhl Seminar’s group-discussion style where in each day topics were raised by participants in the morning and after that multiple topics were selected and individually discussed by a group of participants who showed their interest in the topic. During the meeting, 17 topics emerged whose findings were presented; the majority of groups reviewed the state of the art in an area and theorized on new ideas and potential future directions. The topics can be divided, with a certain degree of overlap, into three main categories according to the required AI ability as well as its potential applications:

General Game Playing (P): The ability to play games well across any context and game (seen or unseen).

General Player Models (M): The ability to recognize general socio-emotional and cognitive/behavioral patterns of humans while playing any game.

General Game Generation (G): The ability to create game content generators equipped with general creative capacities across games and creative tasks.

Below we enlist the topics and their associated categories, sorted in alphabetical order, for each day of the seminar.

Mar 18 (Monday)

- AI as Curators, Critics and Best Friends (**P**)
- Game Style Corpora (**G**)
- Learning Forward Models (**P**)
- Team Sports for AI Benchmarking Revisited (**P & M**)
- Universal Player Models (**M**)
- Which games should we (AI) explore next? (**P**)

Mar 19 (Tue)

- AI for Playground Games (**P & G**)
- Designing a Crowd-Sourced, Emergent AI Game (**G**)
- Game Analytics Theory-based models X Data-based models (**M**)
- Game-based AI Benchmarks (**P**)
- Learning Abstract Forward Models (**P**)

Mar 20 (Wed) and 21 (Thu) Mornings

- Challenges of the combined creation of narrative and world – The Quiet Year (Live Demo) (**G**)
- Game Complexity vs Strategic Depth (**P**)
- Game research and the real world (**P**)
- Games and Societies Are National Game Preferences Predictable? (**M**)
- General AI Game Commentary (**G**)
- Optimization of Game Design with Parameter Tuning, Content Generation with Psychological Player Model (**G**)

The meeting's outcomes include ideas, candidate answers, and future directions for the following research questions.

General Game Playing: What are some other contexts, besides playing for winning, in which games are played? What are the challenges of building AI that can perform or assist this work? How to learn forward models or use existing forward models to learn abstracted versions? What is it about some specific games that makes them so popular in research? Are there other games and other challenges that communities pay less attention to that may pose greater AI-challenges? Which new game-based AI benchmarks need to be developed in the future in order to continue using games to drive progress in AI research? How exactly do we measure the complexity of a game? How do we quantify its strategic depth objectively? How does game AI research relate to, and could contribute to, the world outside games?

General Player Models: How can we create computational models of player behaviour? How should we explore the possibility and advantages of joining two alternative approaches for modeling players' behaviors: data-based models that represent players' tendencies based on data and theory-based models that depict psychological aspects? How can we predict the popularity of games across different countries and cultures?

General Game Generation: How can we construct game descriptions automatically? How can we design a new genre of video games that combines entertainment value with furthering scientific progress in AI? What does it take to incrementally build up a story together with a possible world where it takes place? What are the challenges and their corresponding solutions for both let's play and live-streaming commentary to implement General AI Game Commentary? How can game content be automatically optimized?

General Game Playing & Generation: What might general game playing and level generation look like in a radically different context such as in playground games?

General Game Playing & Models: What are the roles of team-based sports games in the landscape of AI research, their possible impact and interesting problems for further analysis?

An abstract of each topic can be found in the remainder of this report. From the quality of these extended abstracts, we anticipate that longer versions of these reports should be available soon in the form of vision papers, perspective reviews, or technical studies in academic journals or conferences. We would thank all the 29 participants (listed below), committee members and staff of NII Shonan Meetings, Shonan Village Center staff, and the two volunteer guides from Kanagawa Systematized Goodwill Guide Club for their valuable contributions to this fruitful meeting.

References

- [1] Yannakakis, Georgios N and Togelius, Julian: Artificial intelligence and games, Springer Nature, 2018
- [2] Togelius, Julian and Yannakakis, Georgios N: General General Game AI. In: 2016 IEEE Conference on Computational Intelligence and Games (CIG), pp. 1–8, (2016)

List of Participants (Non-Organizers)

- Rafael Bidarra, Delft University of Technology (Netherlands)
- Paolo Burelli, IT University Of Copenhagen (Denmark)
- Michael Cook, Queen Mary University of London (UK)
- Mirjam Eladhari, Sodertorn University (Sweden)
- Daniele Gravina, University of Malta (Malta)
- David Ha, Google Brain (Japan)
- Tomohiro Harada, Ritsumeikan University (Japan)
- Amy K. Hoover, New Jersey Institute of Technology (USA)
- Hiroyuki Iida, Japan Advanced Institute of Science and Technology (Japan)
- Ahmed Khalifa, New York University (USA)
- Kyung-Joong Kim, Gwangju Institute of Science and Technology² (South Korea)
- Luiz Bernardo Martins Kummer, Pontifical Catholic University of Parana (Brazil)
- Antonios Liapis, University of Malta (Malta)
- Diego Perez Liebana, Queen Mary University of London (UK)
- Simon Mark Lucas, Queen Mary University of London (UK)
- Maxim Mozgovoy, University of Aizu (Japan)
- Tomoharu Nakashima, Osaka Prefecture University (Japan)
- Mark Jason Nelson, American University (USA)
- Mike Preuss, Leiden University (Netherlands)
- Jochen Renz, Australian National University (Australia)
- Florian Richoux, JFLI, CNRS, NII (Japan)
- Ana Fernández del Río, Yokozuna Data (Japan)
- Sebastian Risi, IT University of Copenhagen (Denmark)
- Joon-Hong Seok, NCSoft (Korea)
- Gillian Smith, Worcester Polytechnic Institute (USA)
- Matthew Stephenson, Maastricht University (Netherlands)
- Shoshannah Tekofsky, Freelance Researcher in AI and Video Games (Netherlands)
- Sila Temsiririkkul, Japan Advanced Institute of Science and Technology (Japan)
- Alexander Zook, Blizzard Entertainment (USA)

²with Sejong University (South Korea) at the time of participation

Contents

| | | |
|----|---|----|
| 1 | AI as Curators, Critics and Best Friends | 7 |
| 2 | Game Style Corpora | 9 |
| 3 | Learning Forward Models | 11 |
| 4 | Team Sports for AI Benchmarking Revisited | 13 |
| 5 | Universal Player Models | 15 |
| 6 | Which games should we (AI) explore next? | 17 |
| 7 | AI for Playground Games | 20 |
| 8 | Designing a Crowd-Sourced, Emergent AI Game | 22 |
| 9 | Game Analytics: Theory-based models X Data-based models | 24 |
| 10 | Game-based AI Benchmarks | 26 |
| 11 | Learning Abstract Forward Models | 27 |
| 12 | Challenges of the combined creation of narrative and world – The Quiet Year (Live Demo) | 29 |
| 13 | Game Complexity vs Strategic Depth | 30 |
| 14 | Game research and the real world | 32 |
| 15 | Games and Societies: Are National Game Preferences Predictable? | 33 |
| 16 | General AI Game Commentary | 35 |
| 17 | Optimization of Game Design with Parameter Tuning, Content Generation with Psychological Player Model | 37 |

1 AI as Curators, Critics and Best Friends

Participants: Michael Cook, Mirjam Eladhari, Ahmed Khalifa, Ana Fernández del Río, Hiroyuki Iida, Gillian Smith, and Matthew Stephenson

When researchers or game developers design AI systems to play games, they typically focus on *playing to win*. In terms of milestones in the history of AI and games, the best-known and most influential systems were designed to beat humans at a particular game, such as *Deep Blue* for Chess, *AlphaGo* for Go, or *OpenAI Five* for DOTA 2. There are many reasons for this, the most important of which is that the problem is usually extremely well-defined. Yet this history of playing to win has ignored many kinds of game, and many ways of playing games. Speedrunners, quality assurance testers, journalists, critics, researchers, curators and more all play games in different ways, and look out for different things when they do so. To frame game-playing simply as playing to win ignores all of these different subcultures and tasks, as well as ignoring the many different ways everyday play can manifest, such as subversive play. The aim of this working group was to identify some other contexts in which games are played, and to discuss the challenges of building AI that can perform or assist this work.

Motivations

The working group discussion began by discussing the motivation for the work, and the individual subgoals within this large problem space that were personally relevant to us. Overall, there was an understanding that this would lead to a more holistic view of games, that took into account the cultural, social, personal, emotional and other contexts in which games exist and are experienced. Not all of these would be equally easy for an AI to access, and in some cases accessing them at all might raise complicated issues (for example, an AI critically evaluating a game about trauma or grief). Yet merely acknowledging they exist would be a large step forward for many aspects of game AI.

In many cases, our discussion revealed a lot of links to existing research work, which supports the idea that these issues are not simply fringe experimentation but also deeply relevant to the current problems the field is engaging with. For example, automated game design systems would benefit from the ability to learn design knowledge from other games, to explain how to play games to a prospective player, or to place their creations in the context of games history. Game recommendation, a huge part of the data-driven games industry, would greatly benefit from richer recommendation systems that took into account the dynamics and aesthetics of the game being recommended, rather than the text used to describe its store entry.

Personal Experience

Each working group member recommended a game to the group, and we noted down the variations in recommendation, the way personal experience influenced the process, and the distinct types of understanding that were being demonstrated. For example, some recommendations came simply from personal interest, while others were tailored to the audience. Some recommendations came with anecdotes about what it was like to play the game; others explained a personal connection to the game's design; and some tried to convey math-

ematical properties of the game in question. In one case, the person making the recommendation had not played the game before, but were able to sell the game's concept and the interesting features it had.

It's important to note that we are not seeking to build an AI system that can replicate these processes, or have the same experiences as a human, but it is vital to understand the expectations and the social function of recommendations in order to build systems that can better exhibit them. It also demonstrated how interconnected many of these features are, which reinforced our decision to combine them together under a single group theme – recommendations often were a form of curation, or referred to specific design criticism. This reinforces the idea of a 'holistic' approach to game playing, which takes into account all of these different evaluative lenses through which a game can be viewed.

Thought Experiments

As a final exercise we sketched out three hypothetical systems, each of which represented one of the three areas we had considered during the group discussion. This activity helped us conceptualise some of the features these systems would need, and some of the challenges that might arise in trying to build them.

For **curation** we imagined a system that could curate a large community of digital creators, such as *itch.io*. We imagined one or more automated curation bots which explore the site and create exhibitions on a regular basis, selecting content from the large catalogue of works uploaded to *itch.io* and presenting these collections, with curator's statements, on the website's front page. For **recommendation** we imagined a system that could pull in vast amounts of knowledge from many different sources, and decide on relevance at large and small scales, from personal to general. This system would be able to adapt recommendations to highly specific moments in time (waiting at a bus stop in the rain) as well as longer-term phases in a person's life (becoming a parent, growing older). For **critique** we imagined a system that has internalised the total history of game development and design, and was able to draw on this knowledge to provide analysis of new game ideas. In particular, the focus here was on considering the kinds of knowledge required to offer this advice – not just playing a game, but its history on social media, reading its evaluation in the press, understanding the response from other designers.

Conclusions

Our discussion was lively and full of ideas, and this topic seems full of open research questions, exciting next steps, and promising future work. Our immediate plans are to write a vision paper for a future conference, and contribute towards the founding of an AI/bot design festival to encourage people to come together and experiment in this new area.

2 Game Style Corpora

Participants: Antonios Liapis, Ruck Thawonmas, and Shoshannah Tekofsky

With the fast-paced developments in machine learning, a plethora of data repositories have become available. Digital games have already applied deep learning methods in a number of tasks for agent control [3; 5] and content creation [7; 1; 2]. An issue raised by [7] is the lack or limited availability of large corpora for data-heavy learning tasks in games. While there is a repository for functional aspects of arcade levels [8], only recently [11] has attention been drawn to data repositories which incorporate non-functional, aesthetic aspects of games.

The working group focused on a specific task which would require such a game style corpus to exist: constructing game descriptions (GDs) automatically. The outcomes of such a task would have a variety of applications. A straightforward application would be for analysis of a game ecosystem based on games’ similarities in some features or their differences in others. Such a granular description language would allow for games to be clustered together based on functional, thematic and visual aspects; [11] describe a similar clustering task via visual output alone. A more ambitious application would be to directly use the discovered GDs to generate new game content either by modifying some sort of higher-order representation [7], or to evaluate how generated content for existing or new games adhere to the overall intended style of the game [4]. Finally, a challenging but valuable application would be to map discovered GDs to the human skills required to play such a game. The language used for the automatically constructed GDs could consist of machine readable patterns, or a linked data structure as in ontologies [6]; more ambitiously, however, a human-readable interpretation and perhaps explanations [12] would be ideal.

The process through which such a corpus of GDs could be automatically built was an important topic of discussion within the working group. In terms of the inputs which could be used, the most important ones are playtraces from humans or from artificial agents, introductory or promotional videos, text-based tutorials, guides, reviews or wikias, screenshots, “let’s play” videos, as well as maps and levels of the game. Based on this input, the ideal algorithms to process them would revolve around Computer Vision (including object detection and classification, optical character recognition and others), signal processing (especially for game audio), natural language processing (for text-based tutorials and reviews) and sequence mining for discovering key moments in both text-based tutorials and video playthroughs. The discovered style patterns, taking the form of GDs, could be useful for designers attempting to find related games clusters, for recommender systems for large game distributors, for players in terms of their skill summary and matchmaking (if the GD identifies which skills are needed from its players).

Admittedly, the goal of automatically extracting GDs is an ambitious one which could easily be an AI-complete problem. To overcome some of the challenges, a number of shortcuts have been identified by the group: (a) using prior knowledge such as existing ontologies as a structure but also in terms of content (such as SKOS [9]), (b) pre-defining meta-characteristics of the screen input (e.g. properties of viewpoints, HUD), (c) using real-world data sources or models trained on such (e.g. for object detection of real-world objects and letters as in labelled objects in [10]), (d) applying pre-existing game knowledge in the

form of user tags, genre in a supervised learning fashion.

References

- [1] Guzdial, M., Sturtevant, N., Li, B.: Deep static and dynamic level analysis: A study on infinite mario. In: Proceedings of the AIIDE workshop on Experimental AI in Games (2016)
- [2] Karavolos, D., Liapis, A., Yannakakis, G.N.: Using a surrogate model of gameplay for automated level design. In: Proceedings of the IEEE Conference on Computational Intelligence and Games (2018)
- [3] Kempka, M., Wydmuch, M., Runc, G., Toczek, J., Jaskowski, W.: ViZ-Doom: A Doom-based AI research platform for visual reinforcement learning. In: Proceedings of the Computational Intelligence in Games Conference (2016)
- [4] Liapis, A., Yannakakis, G.N., Nelson, M.J., Preuss, M., Bidarra, R.: Orchestrating game generation. *IEEE Transactions on Games* **11**(1), 48–68 (2019)
- [5] 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., Hassabis, D.: Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015), <http://www.nature.com/nature/journal/v518/n7540/full/nature14236.html>
- [6] Navigli, R., Velardi, P.: Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics* **30**(2), 151–179 (2004)
- [7] Summerville, A., Snodgrass, S., Guzdial, M., Holmgard, C., Hoover, A.K., Isaksen, A., Nealen, A., Togelius, J.: Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games* **10**(3), 257–270 (2018)
- [8] Summerville, A.J., Snodgrass, S., Mateas, M., n’ou Villar, S.O.: The vglc: The video game level corpus. In: Proceedings of the 7th Workshop on Procedural Content Generation (2016)
- [9] W3C: Skos simple knowledge organization system rdf schema. <https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html> (2008)
- [10] Wydmuch, M., Kempka, M., Jaśkowski, W.: Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games* (2018), accepted
- [11] Zhang, X., Zhan, Z., Holtz, M., Smith, A.M.: Crawling, indexing, and retrieving moments in videogames. In: Proceedings of the Conference on the Foundations of Digital Games (2018)
- [12] Zhu, J., Liapis, A., Risi, S., Bidarra, R., Youngblood, G.M.: Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. In: Proceedings of the IEEE Conference on Computational Intelligence and Games (2018)

3 Learning Forward Models

Participants: Simon M. Lucas, David Ha, Sebastian Risi, Mark J. Nelson, Diego Perez-Liebana, and Daniele Gravina

This report is the first of two at this Shonan workshop that cover the topic of learning forward models (FMs). The FM allows simulating possible future game states given an initial state and a sequence of actions to execute. FMs are essential for Statistical Forward Planning (SFP) methods, such as Monte Carlo Tree Search or Rolling Horizon Evolutionary Algorithms. SFP algorithms can help provide an explainable type of AI, since decisions can be related to their expected and observable consequences. Additionally, FMs can be used as a limitless source of training data for reinforcement learning algorithms. FM learning is an active subject of study e.g. involving several approaches, notably deep learning and rule induction (see Lucas et al for more references [2]).

In our group we investigated a local approach to learning forward models [2], starting with a simple example of making a Game out of Conway’s Game of Life (GoL), and then Sokoban. Learning forward models is in general a hard problem. The local approach attempts to decompose a complex global state transition function into a set of simpler functions that model only the inputs that each entity depends on. While for grid-based games such as GoL and Sokoban the local interaction properties are obvious, the approach may work well across a wide range of games even if the local properties are less apparent. For GoL we note that other authors have previously used a somewhat similar approach by using convolutional neural networks with weight sharing, and learning a single output given the input neighbourhood [3] [1].

Lucas et al [2] introduced player actions to make GoL into a single or two-player game, they also separated out the player actions from learning the rules of the game, and showed the relationship between the accuracy of the learned model and its effect on the performance of an SFP game-playing agents. One result was that agents could be tuned to better cope with the effects of inaccurate models. In this work we restricted ourselves to learning the forward models by extracting the transition data into supervised learning datasets.

When the local 3×3 neighbourhood is taken in to account there are just 512 possible binary patterns to learn, which can be represented as a truth table with that many rows, and also of course considered as a supervised learning problem. We evaluated the performance of several learning algorithms on this data, and found that the features used were critical to their generalisation performance. Since it is important to learn all the data and the dataset is small, we varied the size of the training set but included all the training data in the test set.

Using a number of classifiers we were able to learn perfectly accurate models with a variable number of samples depending on the input features used. We experimented with Random Forests, Support Vector Machines (SVMs) with Radial Basis Function (RBF) kernels and neural networks with hidden layers (hidden layers are essential due to the non-linear nature of the function to be learned). In the GoL rule, the sum of the inputs is an important feature, and using this greatly speeded up learning. Without it, nearly all samples were necessary to learn a perfect model using any of the classifiers. With the sum included, Figure 3.1 shows the learning performance for a Random Forest and an SVM with an RBF kernel. Slightly faster learning was achieved by the MLP

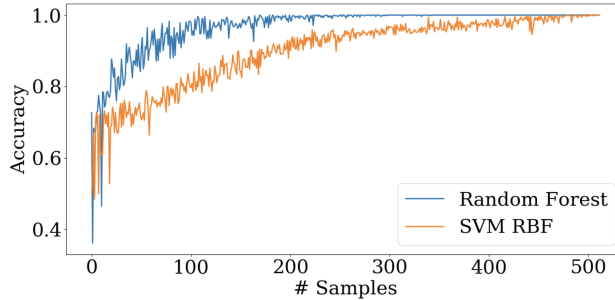


Figure 3.1: Accuracy plotted against the number of samples used for training.

with a single hidden layer, often learning a perfect model given less than 100 samples.

Using Genetic Programming with significant inductive bias (giving the integers 3 and 4, together with the sum of all inputs as IN9) we were able to quickly learn a perfectly accurate function: `add(eq(add(3, IN4), IN9), eq(3, IN9))`. Here anything non-zero is interpreted as a '1', and the centre cell is IN4. This was learned from just 50 of the 512 patterns.

The main outcomes of this are that i) the features used have an important impact on successfully learning a model; and ii) posing the FM learning problem as a supervised local learning problem has a dramatic effect on how easily the FM can be learned. Doing this transforms learning the FM for GoL from being intractable for large grids into being learnable from only a handful of state transitions. During the group work we also tried this for Sokoban, but only got as far as collecting and organizing the data, noting that a different sampling pattern was needed (a 5 x 5 cross pattern) but that it should still be possible to learn an accurate model albeit given many more state transitions. Also interesting is to explore active learning in this context, where an agent is allowed to “play” with the forward model by setting up arbitrary states and observing the subsequent state transitions.

References

- [1] Gilpin, W.: Cellular automata as convolutional neural networks. arXiv (2018)
- [2] Lucas, S.M., Dockhorn, A., Volz, V., Bamford, C., Gaina, R.D., Bravi, I., Pérez-Liébana, D., Mostaghim, S., Kruse, R.: A local approach to forward model learning: Results on the game of life game. CoRR **abs/1903.12508** (2019)
- [3] Wulff, N.H., Hertz, J.A.: Learning cellular automaton dynamics with neural networks. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) *Advances in Neural Information Processing Systems 5*, pp. 631–638. Morgan-Kaufmann (1993)

4 Team Sports for AI Benchmarking Revisited

Participants: Maxim Mozgovoy, Mike Preuss, Rafael Bidarra, Tomoharu Nakashima, and Tomohiro Harada

Background

Team sports game have been a subject of AI research for a long time. One salient example is RoboCup, a game where teams of physical or virtual robots compete in a soccer-like environment. Regular RoboCup competitions are held since 1997, and address numerous AI-related problems ranging from computer vision and robot modeling to team coordination and goal-driven behavior [1].

While RoboCup events enjoy consistent popularity, there are still many interesting research questions that are not in the agenda of most RoboCup participants. Our initial motivation to discuss RoboCup was fueled with the arrival of publicly available datasets of digitized real soccer recordings [2]. These recordings consist of sequences of frames, containing coordinates of the ball and all players on the soccer field, taken at regular time intervals. Thus, these datasets can provide insights into playing strategies of actual soccer teams, and thus can be of interest to virtual team sports games AI researchers.

The discussion of relevance of human-generated data for virtual sports teams (such as RoboCup teams) further motivated us to revisit the role of team-based sports games in the landscape of AI research, their possible impact and interesting problems for further analysis.

Topics for Further Analysis

1. What humans can learn from robots and vice versa. We recognize that any computer sports game is a very different experience comparing to a real physical activity. It grasps only certain aspects of the game while greatly downplaying other aspects. However, we still believe that the datasets of real-life recordings should be analyzed to reveal the differences in team tactics of real and virtual teams, and understand the reasons for this differences. In particular, we still do not know with certainty whether virtual or real teams follow more efficient goal-scoring strategies. Theoretically, a collection of past game recordings can also assist game situation scoring: some AI solutions assess the quality (“goodness”) of a given game situation by employing heuristic rule-based algorithms, while past games can provide real examples of situations that actually led to scoring goals.

2. What makes a team team. While we call soccer and similar games “team-based”, it is still not entirely clear what constitutes team behavior, and what are the characteristics of successful teams. It is possible that team behavior can be defined in terms of goal-driven decision making where team goals (scoring) take precedence over individual goals (such as demonstrating particular players’ skills). However, real teams possess other important traits, such as adaptability to opponent counter-actions, efficient repetition of the same successful patterns or adjusting strategies on the go. Team strategies in real-life soccer have been evolving during the whole past century [3].

3. Emergent and stable/reliable team behavior. Individual players of team sports games have to rely on imperfect information about their surroundings. They do not see the whole playing field and other players all the time, and they have very limited possibilities to communicate with their teammates

and coaches. However, good teams are still able to exhibit clearly identifiable team behavior patterns, such as attacking combinations or quick regrouping in case of player injury or removal from the field. Thus, it is worth to analyze the relation between individual and team behavior, possibly in relation to the problem of obtaining team behavior by relying primarily on local (and noisy) context rather than on perfect knowledge of the whole game field situation.

4. Is RoboCup harder than Dota? RoboCup at a glance might seem like a relatively simple stripped down game of simple goals and limited choice of players. However, it might be possible that designing a good AI for RoboCup or similar games is harder than designing AI for seemingly more complex games, such as Dota. It might be worthy to investigate a related question: why soccer is (almost certainly) not a fun multiplayer game? Why there are online multiplayer real-time strategies, and there are no online multiplayer soccer games? One possible answer might be related to a fact that Dota-like games are deliberately designed to be fun for all participants regardless of their role in a team. However, soccer players often have to follow strategies that are necessary for their teams to win, even when it means performing somewhat boring or unpleasant activities. In turn, it might mean that in soccer-like games the space of reasonable winning strategies is higher, and in many cases scoring a goal requires complex team coordination, backed with numerous prior training sessions. However, this question needs further analysis.

References

- [1] Ferrein, A., Steinbauer, G. 20 Years of RoboCup. *KI-Künstliche Intelligenz*, 2016, vol. 30(3-4), pp. 225–232.
- [2] STATS Artificial Intelligence.
Available: <https://www.stats.com/artificial-intelligence>
- [3] Wilson, J. Inverting the Pyramid: The History of Football Tactics. Orion, 2010, 480 p.

5 Universal Player Models

Participants: Paolo Burelli, Luiz Bernardo Martins Kummer, Kyung-Joong Kim, and Georgios N. Yannakakis

Creating computational model of player behaviour is an essential process in many aspects of game development. Models of players can be used to aid the design of new futures in a current game, to drive marketing initiatives and analyze the game’s profitability [1; 4] or as an integral part of the games in aspects such as procedural content generation or adaptation [8].

Modelling player is a complex task due to the complexity of the human nature and of the interaction between the player and the game; this means that, in most cases, the models developed are reduced to capture one or few specific aspects of the player experience and they are tailored to a specific game.

These conditions reduce the possibility of reusing the models developed and makes the models less resilient to changes. Addressing one of these limitations by developing player models that can be generalized over multiple games could potentially allow, game companies to build models that describe and/or predict the behaviours of their players base across their whole game portfolio, allowing the company to treat their players coherently throughout their lifetime, regardless of whether they switch to a different game or the game they are playing receives some major update.

A few works have investigated how to produce cross-game player models. Martinez et al. [5] compared players’ physiological signals between two different games to identify common predictors of reported player experience. Shaker et al. [7] investigated how to generalize in-game behaviour descriptors so that they could represent coherent features across games. Similarly, Cammilleri et al. [2] conducted an experiment to compare the generalizability of a set of meta-features describing players in-game behaviour. In addition, an analogous approach that deals with the measurement of game thrills is the Game Refinement Theory, initially proposed by Iida et al., [3].

Inspired by these works and by the representation of human emotions introduced by Russel [6] (valence, arousal, dominance), we propose a common game-play representation that can effectively represent player behaviour across games and be effectively employed to produce *universal player models*. In this representation, each in-game event is represented using a three-dimensional representation base on three axis: *contribution*, *intensity* and *agency*. The *contribution* dimension describes how much, either positively or negatively, players’ actions contribute to the achievement of their goal in the game. The *intensity* of an action describes how frequently a given action happens in a given time slot. The last dimension, *agency*, describes how much a given event is due to a choice of the player or not.

Our hypothesis is that the proposed representation can be expressive enough to convey all necessary information to describe the player experience, while, at the same time, it should generalize over any game. To be able to answer whether this hypothesis is correct, a number of open questions remain open:

- How do actions and events in different games map to these three dimensions? One importation aspect that eludes this model is currently in-game social interactions; how can they be represented?

- How can play sessions represented using this model be compared between game having different time scales?
- If the representation proves to be general enough to be able to describe game-play in multiple game, how can its effectiveness as a description of player experience be evaluated?
- Using this representation, how can we aggregate multiple play sessions to describe a player?
- How the Iida et al.'s [3] theory can be applied to the proposed approach?

To answer these questions we plan a number of experiments: first, given a description of a number of players in different games, we plan to perform a cluster analysis on the different games and analyze to which clusters the players belong in each game. In a second experiment we will attempt to predict player retention/engagement in different games using the same representation of past player behaviour. In a third experiment we will investigate how the representation can be used to analyse player skill between games.

References

- [1] Burelli, P.: Predicting Customer Lifetime Value in Free-to-Play Games. In: Data Analytics Applications in Gaming and Entertainment. Taylor & Francis Group (2019)
- [2] Camilleri, E., Yannakakis, G.N., Liapis, A.: Towards general models of player affect. In: 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII). vol. 2018-Janua. IEEE (2017)
- [3] Iida, H., Takeshita, N., Yoshimura, J.: A metric for entertainment of boardgames: its implication for evolution of chess variants. In: Entertainment Computing, pp. 65–72. Springer (2003)
- [4] Lee, E., Jang, Y., Yoon, D., Jeon, J., Yang, S.i., Lee, S., Kim, D.W., Chen, P.P., Guitart, A., Bertens, P., Perianez, A., Hadiji, F., Muller, M., Joo, Y., Lee, J., Hwang, I., Kim, K.J.: Game Data Mining Competition on Churn Prediction and Survival Analysis using Commercial Game Log Data. IEEE Transactions on Games (2019)
- [5] Perez Martínez, H., Garbarino, M., Yannakakis, G.N., Martinez, H.P., Garbarino, M., Yannakakis, G.N.: Generic physiological features as predictors of player experience. In: International Conference on Affective Computing and Intelligent Interaction (ACII). vol. 6974 LNCS, pp. 267–276 (2011)
- [6] Russell, J.A.: A circumplex model of affect. Journal of personality and social psychology **39**(6), 1161 (1980)
- [7] Shaker, N., Shaker, M., Abou-zleikha, M.: Towards Generic Models of Player Experience. In: Proceedings, the Eleventh Aai Conference on Artificial Intelligence and Interactive Digital Entertainment (aiide-15). pp. 191–197 (2015)
- [8] Yannakakis, G.N., Togelius, J.: Modelling Players. In: Artificial Intelligence and Games. Springer International Publishing (2018)

6 Which games should we (AI) explore next?

Participants: Amy K. Hoover, Julian Togelius, Florian Richoux, Joon-Hong Seok, Sila Temsiririkkul, and Alex Zook

While games like Chess, Checkers, and Go, Starcraft, Atari, Montezuma’s Revenge, Mario Bros, and Defense of the Ancient often receive a significant amount of research attention, other games may also prove relevant to the general AI community. For instance, there is a generative AI competition for Minecraft at the International Conference on Computational Creativity, an AI-based game inspired by the design of Where in the World is Carmen Sandiego, and World of Warcraft has been the subject of an epidemiological study paper. The question is then, what is it about these games makes them so popular in research? Are there other games and other challenges that communities pay less attention to that may pose grander AI-challenges?

A four-part approach developed to explore these questions. To begin, the group made a list of individual games that are interesting and understudied discussed in section 6.1. Then, the group listed characteristics of these games that contribute to difficulty in section 6.2. The third question explored are what are the cognitive capabilities not currently demanded by games in general, and concluded with what it means for a game to be a good benchmark for AI.

6.1 List of Games that Interesting and Understudied

The group began by each proposing games that seem interesting and understudied with a brief description about the challenges represented by them that could potentially be interesting to AI researchers.

The first was character growth in MMO RPGs (e.g. Lineage, World of Warcraft). There are often many ways to play these games (e.g. follow the main story line, explore, maximize skills like cooking or fishing), and therefore many ways to measure growth. However, perhaps in some ways this growth is relatively simple to measure. Exploring could be measured by the amount of time spent in an area and the number of areas to explore. Growth along the main story line may be measurable simply through the numbers of quests completed. What aspects of growth do these metrics and measurements fail to capture?

Some of the activities in MMO’s (e.g. grinding) may relate to clicker-type games. Taking a larger problem of getting to the next level in character development (e.g. from level 10 to level 11) or in clickers the next mini achievement (e.g. Cookie Clicker). However, some clicker games complete transform from the initial game proposed: Mysterybox, Candybox, Frog Fractions, PaperClips, and a Dark Room.

However, maybe a large part of learning games lies in learning its mechanics. In a Dark Room, not all mechanics are known a priori.

Some mechanics known ahead of time, may significantly alter gameplay. For instance permadeath encourages players to minimize the amount of risk they are willing to take like in Mystery Dungeon and Chocobo’s Mystery Design.

Some games may be more about communication. One of the authors has studied communication in Call of Duty, where players develop specialized action languages to communicate with each other when voice communication is too slow or ineffective. Journey (can’t talk or hurt each other)

Nonseparable problems: You can't do something without other people
The Sims (create your own story), Overcooked (verbal collaboration) Eleusis
(Guess rules of game, game master gains points for good rules), Dixit

6.2 Difficulty Features

The following features, or characteristics, of a game can affect how it can be played by an AI: **Rewards**

- Long time horizon
- Sparse rewards
- Deceptive rewards

Mechanics

- Stochasticity

Information

- Hidden information
- Partial information
- Unknown content, rules
- Discrete vs continuous space/time

Players

- Multiagent
- Symmetric vs not
- Team structure (1v1, free-for-all, team vs team, team vs team vs team)

Game Structure

- Multiple stages
- Permadeath
- Turn structure (simultaneous, alternating)

Game representation

We identified the following aspects of how the game is represented to the AI, which affect how it can be played:

- Perception only (ex: visual)
- Access to game code
- Forward model
- Communication

6.3 A Panoply of Problems: Challenges and Games

We identified the following challenges, as exemplified by individual games:

- Infinite Time Horizon (World of Warcraft)
- Multiple Goals (World of Warcraft)
- Unclear Goals (The Sims)
- Multistage (Hearthstone)
- Collaboration (Overcooked)
- Collab./Competitive (Ultimate Chicken Horse)
- Unknown Mechanics (A Dark Room)
- Unknown Rules (Eleusis, Dixit)
- Agency over rules (Pandemic Legacy)
- Multimodality (Pokemon Go)
- Text input/output (Zork)
- Self Directed (Minecraft)
- Appreciation (Rock Band)
- Ends < - - - - - > Means

6.4 Cognitive Capabilities

We can also look at games from the cognitive angle, and see which cognitive capabilities they require (e.g. following CHC Theory):

- Perception
- Planning
- Memory
 - Short-term
 - Long-term
- Language
- Attention
- Communication

6.5 One Does not Simply Apply

There are various techniques that are possible, and which one is most useful will generally depend on the characteristics of the game and its representation:

Learning

- Supervised learning
- Evolutionary computation
- Temporal difference learning
- Policy gradients

Planning

- Minimax
- MCTS
- A*
- Evolutionary computation

7 AI for Playground Games

Participants: Gillian Smith, Mike Cook, Ahmed Khalifa, Kyung-Joong Kim, and Sila Temsiririrkkul

General video game playing has, thus far, largely focused on a particular set of game genres such as arcade games, retro console games, and puzzle games. In this group, we set out to discuss what general game playing and level generation might look like in a radically different context. Instead of digital games, what about physical games? Instead of games you play to win, what about games you play to experience? Instead of games that have strict rule sets, what about games whose rules morph over time? The outcomes from this group were: a set of game properties interesting to explore in future general game AI research, a list of games from multiple cultures that meet these criteria (sourced from Shonan participants), and an early sense for what a formal description language for playground games might look like.

Playground Games Playground games are an exciting potential area for Game AI research because they are unlike any other digital game studied thus far. In discussing our interest in playground games for general playing and generation, we identified several common design properties of these games that make them especially interesting and challenging:

- **Audience.** Unlike many of the games used as testbeds for game AI research, playground games are played predominantly by young children at varying stages of physical, emotional, and social development.
- **Number of Players.** Some playground games are played by a single person; others by an indeterminate and ever-changing collection of adults and children.
- **Changing Rules.** Playground games often morph over time, based on player preferences and sometimes collaborative decision-making.
- **Inconclusive End Conditions.** Some playground games have clear endings, with winners and losers, but others end when a new game begins or when children get tired.
- **Experience-Driven Play.** Playground games are not fun because of winning or losing, but because of the experience *during* play.
- **Team Selection Criteria.** Part of the playground experience is not just playing a game, but also choosing team members. Participants shared many different mechanisms for team selection, often culturally situated, ranging from leaders choosing teams to random team selection.

Each of these design properties offers opportunities for Game AI research, such as modeling player developmental stages, or realtime evolution of games over time.

Example Games

We collected many examples of playground games from Shonan participants by inviting everyone to share their favorite playground game from their childhood. It quickly became clear that many games are common across cultures, though may have regional variants and different names. We collected the names of common games; participants translated most game names to English: *Simon Says* (UK), *Four Square* (New Zealand), *Tag* (USA)/*Tig* (UK)/*Perched Cat* (France), *Bulldog* (UK)/*Sparrowhawk* (France), *The Long Donkey* (Greece)/*Horse Rider* (Korea), *Red Light Green Light* (US)/*Flower Blossom* (Korea), “1, 2, 3, Sun!” (France), *The English Hideout* (Spain). Participant discussion about playground games also resulted in commentary about the social values and common lessons that are taught through these games: social interaction, sharing spaces respectfully, language development, cooperation, conflict resolution, and sportsmanship.

Playground Game Description Language?

Finally, we closed with a discussion of what would need to go into a description language for playground games. Some of the variables considered include:

- Number of players
- Number of teams
- Minimum and maximum number of players per team
- Roles for players
- Game termination condition
- Rules (some of them conditional)
- Game variants

We also discussed the role that space plays in playground games: some games are played on an open at plane, while others depend upon play structures or landmarks that can be labeled. A separate space description file, similar to a level description file for a game, would be necessary for such games as well.

Discussion about the format for a playground game description language brought up several fundamental questions about how games are defined. At what point does a regional variation of a game become a new game in itself? Should games be defined by the roles players take? What is the distinction between a “team” and a “role”?

Conclusions

Discussion about Game AI for playground games resulted in more questions than answers, but overall excitement about the way that thinking about playground games reframed the way we think about game AI. We began thinking about games more as experience than as something to be won. It also quickly became clear how culturally situated game play is, as even when the same game is played in different communities, the experiences can be wildly different as well. Though their difficulty to define means it is unlikely that playground games will be the next domain for general game AI, it is still useful to reflect upon how playground games reveal the biases and assumptions about generality currently made in general game AI research.

8 Designing a Crowd-Sourced, Emergent AI Game

Participants: Shoshannah Tekofsky, Rafael Bidarra, Mirjam P. Elad-hari, Daniele Gravina, David Ha, and Georgios N. Yannakakis

Crowd-sourced, emergent AI games would constitute a new genre of video games that combine entertainment value with furthering scientific progress in AI. The concept is a fusion of Marvin Minsky’s Society of Mind [1], the scientific breakthrough game FoldIt [2], and modern MMO gaming communities. The genre has the potential to push the boundaries of content-generation, collective narrative generation, hybrid human/artificial intelligence, implementing anti-fragility, and increasing the performance of AI in ‘subjective’ fields such creativity, deception, and ‘cuteness’.

The novelty of the approach lies in three elements: 1) *Emergence* - Intelligence emerges unpredictably from the design of the game and the user’s input. The emergence property is only present if forms of intelligence develop that the designers had not foreseen. This is contingent on the game allowing for intelligence to recombine and grow beyond its explicit design. 2) *Crowd-sourcing* - The fitness function, architecture, learning algorithm and/or I/O of the AI should be crowd-sourced for two reasons. First, crowd-sourcing elements of the AI’s design or training allow it to potentially surpass the complexity that could be generated by a limited team of engineers and contributors. Secondly, crowd-sourcing is the conditional element to generate the emergence property of the AI by allowing for rich and uncontrolled contributions to its design and training. 3) *Game* - The AI is integrated into a game to support the crowd-sourcing element. By gamefying the experience of creating and training the AI, non-expert users can be enticed to spend time and resources on developing the AI. Additionally, game environments allow for limited and clear fitness functions for the AI to optimize.

The above concept contains four major challenges. First of all, crowd-sourcing is sensitive to **trolling (a)** and other perverse meta-incentives. In order to generate useful results, the game would have to de-incentivize trolling or introduce a robust moderation mechanic. Secondly, it is unclear what would constitute the **game play, world, and AI entities (b)** in such a game. This question is further expanded upon below by offering an example game design. Thirdly, AI creation is inherently effortfull and not **fun (c)** for non-experts. Lastly, the **complexity (d)** of the data structure of the AI needs to balance fun with depth. If it is too complex then players would need considerable technical knowledge to engage with the game. If it is too simple, then no interesting intelligence can emerge from the game. The challenge is to balance complexity with fun such that players enjoy the game while still allowing for complex AI to emerge.

Players versus Programmers

The following is a short expansion on one possible game design for a crowd-sourced, emergent AI game: *Players versus Programmers*. It is an asymmetrical multiplayer game. Programmers create AI with *Complex AI Tool Sets* to learn complex behavior. Players create *levels* with *simple directed co-evolution / human computation* to teach complex behavior. This creates a flow-like tension between world complexity (challenge) created by the players and AI complex-

ity (ability) by the programmers. Together, the programmers and players are challenged to keep the AI within its own 'flow' channel. This is conceptualized as its optimal learning trajectory. The *Programmers vs Players* game concept tackles the four challenges of the genre as follows.

Trolling (a) is handled with a leaderboard mechanic. The programmer AI is ranked by how many levels it can solve. The player level is ranked on how difficult or useful it is for the current AI. In this manner, AI and level are ranked on how closely they approximate the 'flow' channel for the AI's learning trajectory.

The **game play, world and AI (b)** could be an MMO-like environment. By design, the programmers create the AI itself while the players create the world (levels and objectives). The game designers are left with the challenge of generating the *game space* - data structure of the AI, the elements that can be used to design the levels, and the elementary actions that can be performed in the world. The challenge for the designers is to create the most unbounded experience they can achieve for the programmers and players.

To introduce **fun (c)** into the game experience, players and programmers can see the AI's traverse levels live while they are in the creation process for either levels or AI's. This design element would offer an experience reminiscent of the old Lemmings games.

Lastly, to tackle the **complexity (d)** issue, programmers and players will have asymmetrical tool kits. Programmers will be given access to semi-technical, out-of-the-box AI algorithms that they can string together with I/O connections. Players will be offered a simple WYSIWYG experience through a visual co-evolution interface [3] for different elements of their level. For instance, the geometry of a level will evolve according to an evolutionary algorithm. The player is offered a grid of possible outcomes and selects the parents for the next generation of geometry. This allows for generative, emergent properties in the levels, while also providing a simple, action-based game play to the players. During the selection process, the AI's are still progressing through the level and so selection and evolution speed play into the game experience.

Overall, *Programmers versus Players* is an example of a crowd-sourced, emergent AI game that would allow players to create and teach AI's that may possibly grow into something smarter and more versatile than any one contributor could have foreseen.

References

- [1] Minsky, M.: Society of mind. Simon and Schuster, 1988.
- [2] FoldIt Portal, <https://fold.it/portal/>. Last accessed 15 April 2019
- [3] Isted, P. T., Ma. B., Risi, S.: Interactive evolution of levels for a competitive multiplayer FPS. 2015 IEEE Congress on Evolutionary Computation (CEC). IEEE (2015).

9 Game Analytics: Theory-based models X Data-based models

Participants: Luiz Bernardo Martins Kummer, Tomoharu Nakashima, Ruck Thawonmas, and Joon-Hong Seok

Introduction

This study aims at exploring the possibility and advantages of joining two analogous approaches that deal with the same subject: the modeling of players' behaviors. On the one hand, there are Data-based models that represent players' tendencies based on data, on the other hand, there are Theory-based models that depict psychological aspects. Two hypotheses were proposed regarding the approach conception and its appliance. After this initial work, intended results for the hypotheses were suggested together with the proposition of experiments and new open questions.

Hypotheses

Analyzes over the Data-based models may hide some risk situations as the behavior presented on data may not portray all the possible motivational stages of players, as depicted by Theory-based models. It means that Data-based models may present faulty predictions when players change their interest in continuing playing. In view of it, the following hypotheses are proposed.

- H1. The association of psychological models to players data improves the identification of risk situations in the usage lifecycle of games (e.g., churn).
- H2. The use of psychological models can suggest the best moment to release a game upgrade.

Proposed Approach

The chosen model for this work is the Zhu et al. model [2], which points four motivational stages of players (i.e., Try, Tasting, Retention, and Abandonment). The studies were performed on usage data from an MMORPG called Blade&Soul³[1], assuming the following behavioral borders of each Zhu et al. stage: (**Try**) from the first play until the max level achievement (including the tutorial); (**Tasting**) based on the players' objectives, to identify when all of them were achieved; (**Retention**) after a player completing all his/her objectives, there is an increase in social interactions; and (**Abandonment**) the churn occurrence.

The intended result for H1 is shown in Table 9.1, where the white columns regard the Data-based information and the blue one the Theory-based information. In the H2 perspective, we suggest a new metric which has a range between 0 and 100, where 100 means the best moment to release a game upgrade. Table 9.2 presents its intended result.

Open Questions

- How to identify the players' objectives?

³For more information about Blade&Soul, please visit this website: <https://www.bladeandsoul.com>.

Table 9.1: Intended result for H1

| Player ID | Time-span | Zhu et al.s Stage | | Churn |
|-----------|-----------|-------------------|-----|-------|
| 1 | 1 | Tasting | ... | No |
| 1 | 2 | Retention | ... | Yes |
| ... | ... | ... | ... | ... |

Table 9.2: Intended result for H2

| 1st month | 2nd month | 3rd month | 4th month | 5th month | 6th month |
|-----------|-----------|-----------|-----------|-----------|-----------|
| 10 | 20 | 40 | 70 | 97 | 80 |

- When to release a new game upgrade?
- What is the ideal balance between profitable and non-profitable players?
- Do game producers release games at the best moment?

An extended version of this study can be found at: https://www.researchgate.net/publication/332167977_Game_Analytics_Theory-based_models_X_Data-based_models

References

- [1] Kim, K.J., Yoon, D., Jeon, J., Yang, S.i., Lee, S., Lee, E., Jang, Y., Kim, D.W., Chen, P.P., Guitart, A., et al.: Game data mining competition on churn prediction and survival analysis using commercial game log data. arXiv preprint arXiv:1802.02301 (2018)
- [2] Zhu, L., Li, Y., Zhao, G.: Exploring the online-game life cycle stages. In: E-Business and E-Government (ICEE), 2010 International Conference on. pp. 2436–2438. IEEE (2010)

10 Game-based AI Benchmarks

Participants: Jochen Renz, Tomohiro Harada, Amy Hoover, Hiroyuki Iida, Julian Togelius, Maxim Mozgovoy, and Matthew Stephenson

This group continued investigating the question of which new game-based AI benchmarks need to be developed in the future in order to continue using games to drive progress in AI research. As the previous day’s investigations had mostly focused on the properties of the games themselves, we now changed the perspective to focus on the capacities that these games demanded from their players.

The underlying assumption here is that if a game demands a particular ability from a human player, it would also require something similar from an AI player. This assumption is not necessarily watertight. For example, if a game demands long-term planning from a human, an AI agent could conceivably escape this requirement by learning a number of stored responses to particular strategies, and so play the game well without ever doing something we would recognize as planning. However, we proceeded in the belief that the assumption is mostly true, and that for the cases where the assumption is wrong, this would also be enlightening.

We chose a number of games, so chosen to be markedly different from each other in terms of which capabilities they demanded from their players. The games and AI benchmarks we investigated were Angry Birds, Montezuma’s Revenge, Chess, Kriegspiel, Texas Hold’em, Hanabi, Doom, Robocup (Simulation League), Fighting Games Competition (similar to Street Fighter), Starcraft, Mafia, Super Mario Bros, and Obstacle Tower (a procedurally generated 3D platformer). The various player capabilities we identified were Handling Noise and Non-terminism, Colaborating, Predicting Friend Actions, Predicting the Environment, Estimating Opponent Position, Estimating Friend Positions, Estimating Friend Strategies, Physics Understanding, Predicting Physical Consequences of Actions, Abstracting Action Space, Reflexes, Predicting Opponent Actions, Fast Planning, Tracking Moving Targets, Predicting Opponent Strategies, Knowing when to Act, 2D Orientation and Navigation, 3D Orientation and Navigation, Collecting Items, Long-term Lookahead and Planning, Approximating Difficulty and Stability of Positions, Recognizing Spatial Patterns, Matching and Applying Stored Responses, and Complex Visual Processing. Because of the way these capabilities were identified, were we sought specifically to disambiguate games that had similar capability requirements, no two games in the list have exactly the same requirements. The resulting table show that there are certain emergent clusterings of games and requirements, but there are also many unused combinations. For example, the game Hanabi stands out in not sharing capability requirements with many other games. It is possible that new AI benchmarks could be designed based on combining requirements from Hanabi with requirements from other games.

Following on from this discussion, we also looked at capabilities that were demanded by some games, but not by games that had been used as AI benchmarks. This list includes Team Coordination (as required by Overcooked), Commonsense Reasoning (as required by Scribblenauts), Diplomacy (as required by the eponymous Diplomacy). Games were we did not clearly identify which types abilities were required, though they are clearly different from those that have been used as AI benchmarks so far, include Pictionary, Dixit, and The Incredible Machine.

11 Learning Abstract Forward Models

Participants: Diego Perez-Liebana, Sebastian Risi, Antonios Liapis, Mike Preuss, Simon M. Lucas, Florian Richoux, Paolo Burelli, and Mark J. Nelson

Statistical Forward Planning (SFP) methods, such as Monte Carlo Tree Search (MCTS; [1]) or Rolling Horizon Evolutionary Algorithms (RHEA; [5]), require a predictive or Forward Model (FM) for decision making. The FM allows simulating possible future game states given an initial state and an action to execute. For this, a FM requires two pieces of functionality: i) **copy**, which creates an exact copy of the game state in memory; and ii) **advance**, which rolls the state forward when provided with an action. Abundant research has been directed towards the use of SFP methods in relatively small games. For instance, MCTS has become the standard algorithm for creating Go AI players and, in combination with Deep Learning, it has reached super-human performance [7]. RHEA is another family of algorithms that has recently obtained remarkable results, comparable or even surpassing MCTS in certain domains [3].

The application of SFP algorithms to larger and more complex games poses some hazards in terms of the efficiency of these methods. These approaches operate iteratively searching the space of possible solutions to the decision making problem; the more iterations, the better the action suggestions will be. In large games (such as Civilization VI; see Figure 11.1) where the action and state spaces are considerably large, the **copy** and **advance** procedures become computationally expensive. In order to provide an action for the game in a sensible time scale, the number of possible iterations needs to be reduced which consequently hinders the performance of the algorithms. One potential solution to this problem is to *learn* forward models. FM learning is an active subject of study (especially when these models are not available), given the reactive and flexible capabilities of SFP methods. Examples of FM learning can be found in racing and first person shooter games [4], puzzle games [6] and General Video Game Playing [2]. In this seminar, we investigated how to use an *existing* forward model to learn an abstracted version that can be used by an SFP algorithm to plan at a higher level, by learning the consequences of using macro-actions.

Figure 11.2 shows the game (CityWars) we implemented to this end (see github.com/SimonLucas/KotlinTest). In this game, two factions compete to gain control of the opponent's base. The atomic actions consist of sending x units from one cell to a contiguous one. When a destination cell contains units of the opposing faction, the final count of units will be the difference between the two. Consequently, sending troops to an enemy location will result in unit losses for both sides. The game is designed to offer a relatively low complexity at micro-management level; however, atomic actions require coordination and strategy to be effective. This aspect makes the game an ideal test-bed to investigate how to model macro-actions; in this case, the coordinated movement of as group of units to a target location. Rather than micro-managing the step by step movement of units, the abstract forward model should be able to determine, just by advancing the state once, the percentage of the started group of units that will arrive at the destination and at what time.

We propose that this can be achieved by using the existing forward model to generate data to learn from. By generating a large amount of play-traces (i.e. by a RHEA agent that controls the atomic action decision), one could learn the outcome of these parameterised macro-actions. The abstract FM could eventually be queried for any instantiation of this macro-action. The resultant system would count with two different levels-of-detail FM: one for macro and one for atomic actions. Different learning methods can be used to this end; one possibility is using Convolutional Neural Networks, where different layers of the game are provided as input (i.e. presence of units and bases) and the predicted percentage of troops arriving as the output. As a result of the working group we have an experimental setup that can produce plentiful training data: the next step is to investigate the quality of the predictions that can be made at the macro level and then to observe the effects on a game-playing agent.

References

- [1] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez-Liebana, D., Samothrakis, S., Colton, S.: A Survey of Monte Carlo Tree Search Methods. IEEE TCIAIG 4(1), 1–43 (2012)



Figure 11.1: Example of a game (Civilization VI - Firaxis, 2016) with large action and state spaces.

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 11 | 77 | 2 | 7 | 2 | 11 | 76 | 143 | 2 | 0 |
| 14 | 27 | 28 | 368 | 16 | 1 | 3 | 127 | 5 | 0 |
| 72 | 12 | 11 | 37 | 28 | 25 | 55 | 32 | 7 | 198 |
| 1 | 39 | 7 | 47 | 188 | | 126 | 41 | 17 | 1 |
| 68 | 98 | 8 | 21 | 288 | 3 | 42 | 68 | 41 | 5 |
| 14 | 188 | 29 | 29 | 6 | | 18 | 72 | 1 | 2 |
| 142 | 98 | 65 | 11 | 152 | 1 | 114 | 126 | 18 | 198 |
| 59 | 2 | 311 | 62 | 448 | | 11 | 48 | 337 | 194 |
| 14 | 187 | 15 | 0 | 34 | 136 | 7 | 182 | 379 | 9 |
| 1 | 7 | 58 | 188 | 28 | 47 | 8 | 6 | 14 | 1 |

Figure 11.2: CityWars, the game implemented to serve as benchmark to learn abstract FM.

- [2] Dockhorn, A., Apeldoorn, D.: Forward Model Approximation for General Video Game Learning. In: IEEE Computational Intelligence and Games (2018)
- [3] Gaina, R.D., Liu, J., Lucas, S.M., Perez-Liebana, D.: Analysis of Vanilla Rolling Horizon Evolution Parameters in General Video Game Playing. In: EvoStar. pp. 418–434. Springer (2017)
- [4] Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
- [5] Perez-Liebana, D., Samothrakis, S., Lucas, S., Rohlfshagen, P.: Rolling Horizon Evolution versus Tree Search for Navigation in Single-player Real-time Games. In: ACM Conference on Genetic and Evolutionary computation. pp. 351–358 (2013)
- [6] Racanière, S., et al.: Imagination-augmented agents for deep reinforcement learning. In: Advances in neural information processing systems. pp. 5690–5701 (2017)
- [7] Silver, D., et al.: Mastering the Game of Go without Human Knowledge. Nature **550**(7676), 354 (2017)

12 Challenges of the combined creation of narrative and world – The Quiet Year (Live Demo)

Participants: Rafael Bidarra, Michael Cook, Amy Hoover, Kyung-Joong Kim, and Gillian Smith

There have been several (digital) tools and methods proposed for helping create either a narrative or a virtual world, but none to do both simultaneously. Hence our starting research question: what does it take to incrementally build up a story together with a possible world where it takes place?

We centered around the challenges raised by the game 'The Quiet Year' (TQY), a role-playing game in which a community of players creates and evolves both their world map and their storylines. They do this in turns, by taking actions, decisions and reactions, stimulated and/or constrained by the cards they pick from a deck (each card corresponding to one week of the year), which may bring good or bad news, thus leading players to reshape or revise their goals, beliefs and/or expectations.

We discussed and identified various challenges an AI would likely have to face to participate in TQY, either as a player or even just as a player's assistant. Among them we can point out the following:

- how to assess the extent to which other players' actions corroborate your present scheme of values and goals?
- how to choose/advise how to react to a card event? (establish associations based on an ontology of the world so far; derive/anticipate plausible consequences using commonsense reasoning;)
- how to allow an AI player to do theory of mind (mind reading) from the interaction with other human or AI players?
- how could an AI player perform imitation learning from the logs/transcripts of TQY human players?
- how could an AI competition be designed around TQY? For example, it would require a way to evaluate AI players behaviors and outcomes from the game play.

In order to give people an insight into what these challenges involve, rather than giving a presentation, we opted to run a live mini-demo of The Quiet Year for the whole plenary group.

13 Game Complexity vs Strategic Depth

Participants: Matthew Stephenson, Diego Perez-Liebana, Mark Nelson, Ahmed Khalifa, and Alexander Zook

The notion of complexity and strategic depth within games has been a long-debated topic with many unanswered questions. How exactly do you measure the complexity of a game? How do you quantify its strategic depth objectively? This seminar answered neither of these questions but instead presents the opinion that these properties are, for the most part, subjective to the human or agent that is playing them. What is complex or deep for one player may be simple or shallow for another. Despite this, determining generally applicable measures for estimating the complexity and depth of a given game (either independently or comparatively), relative to the abilities of a given player or player type, can provide several benefits for game designers and researchers.

There are multiple possible ways of measuring the complexity or depth of a game, each of which is likely to give a different outcome. Lantz et al. propose that strategic depth is an objective, measurable property of a game, and that games with a large amount of strategic depth continually produce challenging problems even after many hours of play [1]. Snakes and ladders can be described as having no strategic depth, due to the fact that each player's choices (or lack thereof) have no impact on the game's outcome. Other similar (albeit subjective) evaluations are also possible for some games when comparing relative depth, such as comparing Tic-Tac-Toe against StarCraft. However, these comparative decisions are not always obvious and are often biased by personal preference. As such, we cannot always say for certain which games are more complex or deep than others. As an example, consider the board games Chess and Go. Chess has more piece types, each with differing movement rules and properties, whereas Go typically has a much larger board, providing a sizeable state and action space. It is unclear how much each of these factors impacts the complexity or depth of each game. Would playing Chess on a larger board make it more strategic to play? Would adding extra rules to Go increase the game's depth or be seen as ruining a beautiful and elegant game? While increasing the complexity of a game can also increase its depth, adjusting certain gameplay factors might have more of an effect than others. Browne suggests that strategic depth should be considered relative to a game's complexity [2], and that games which are more complex than others should also possess additional strategic depth.

The number of factors that could potentially influence the complexity or depth of a game is likely to be vast. Common properties might be aspects such as the size of the state space, the branching factor (i.e. action space), the number of rules, deterministic or stochastic, discrete or continuous, the number of players, and so on. Even this small collection of properties poses some problems regarding how they are measured. When determining the number of rules for a game, what description language should be used? How do you compare single-player and two-player games? Should the response time of a human compared to that of an agent be taken into account? We do not have any answers to these questions and any individual opinions are likely to be highly subjective. This also holds for comparing the relative impact of each of these properties. One player might do very well at fully deterministic games that require long term planning, while a second can better deal with probability calculations, and a third is able to keep a straight face in bluffing games. The perceived complexity and depth of any given game is likely to vary between these players. This also applies to artificial agents depending on the AI techniques and approaches being employed. This makes it impossible to say that one game is more complex or deep than another, without taking into account the human or agent that is playing it.

While it is not yet clear how to accurately estimate the complexity or depth of games, doing so could have several benefits for game analysis and development. One application could be for identifying flaws or limitations in games. The original rules for several traditional board games, such as the ancient Viking game of Hnefatafl or the Maori game of Mu Torere, were incorrectly recorded, leading to unfairly balanced games [2]. Methods for analysing the depth of these games would allow such weaknesses to be detected and corrected. Such a case was demonstrated for the 1982 video game Q*bert, where a previously unknown glitch was discovered by a reinforcement learning agent [3]. Agents can also identify additional strategies or levels of depth not previously considered by humans, such as with DeepBlue and AlphaGo[4].

One idea for future work could be to select a suitable set of benchmark games and test how complex or deep each game is for a collection of agents and a variety of possible measures. Identifying any similarities between resource and performance curves across different game

features would allow us to be more confident of which features most impact the complexity or depth of a game, particularly if several different empirical measures broadly align. It might also be worthwhile investigating or developing games that humans find easy to play but agents currently perform poorly on, as these likely represent limitations with current AI techniques.

References

- [1] F. Lantz, A. Isaksen, A. Jaffe, A. Nealen, J. Togelius, “Depth in Strategic Games”, AAAI Workshops, North America, 2017, pp. 1-8.
- [2] C. Browne, “Modern Techniques for Ancient Games”, IEEE Conference on Computational Intelligence and Games, Maastricht, 2018, pp. 1-8.
- [3] P. Chrabaszcz, I. Loshchilov, F. Hutter, “Back to Basics: Benchmarking Canonical Evolution Strategies for Playing Atari”, IJCAI, Stockholm, 2018, pp. 1419-1426.
- [4] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. and Dieleman, S. (2016), “Mastering the game of Go with deep neural networks and tree search.” *Nature*, 529(7587), 484.

14 Game research and the real world

Participants: Julian Togelius, Hiroyuki Iida, David Ha, Jochen Renz, and Shoshannah Tekofsky

This group discussed the large and multifaceted topic of how game AI research relates to, and could contribute to, the world outside games. Games are popular testbeds for AI research, but almost only in the setting where the AI is trying to play the game. Within Game AI research, there are many more roles and perspectives for AI in games. Most games in some way model a real-life phenomenon or process, meaning that AI methods that are useful for a game may be useful for that phenomenon or process.

Some ways in which game AI methods could be useful for real-world issues include:

- Use optimization and reinforcement learning to find loopholes in the law, such as tax law. This would be a form of penetration testing that helps understand weaknesses in our systems. We all have experiences with how AI game playing methods find loopholes.
- Use game-playing agents to simulate complex real-world phenomena, to help public understanding of these phenomena.
- Systems of governance could be explored by encoding the "game mechanics of society". We could then search for new mechanisms of governance, in the same way we can use search and optimization to create new game rules and game content.

One possible conceptual framework for this discussion is the parallel between the three branches of government, according to Montesquieu: executive, legislative, and judicial, and common tasks for AI in games. Here, the executive branch would correspond to playing games, the legislative to generating content, and the judicial to assessing games and/or players. Many parallels between the workings of games, governments and societies could potentially be drawn here.

To make these ideas more concrete, we proposed an experiment where we would train deep neural networks to play SimCity. SimCity is a classic urban simulation game, released in 1989, which played an important part in creating the simulation game genre. In the game, the player builds and manages a city, and has to deal with balancing the budget, planning the city, and perhaps even dealing with earthquakes and attacks by Godzilla. Interestingly, the game has no goal as such, though many players invent goals for themselves, such as creating the largest city possible, the most content citizens, or amassing a maximum amount of wealth. Since the game came out it has been subject to numerous analyses of what its political content is. While the game was partly inspired by Jay Forrester's Urban Dynamics theory, which is commonly seen as neoliberal, SimCity creator Will Wright claims that some of the core values he sought to express in the game was that public transport is good and nuclear power is risky.

Training a network would be an interesting challenge for reinforcement learning, with the numerous actions to carry out, zooming around the map etc. It has certain similarities with playing real-time strategy games, but also differences, in particular the very long time dependencies and the challenge of macro structure in building (cities in SimCity are considerably more complex than bases in StarCraft). A key difference is also that SimCity, as noted above, does not have a unique success criterion; we therefore have several different kinds of rewards to consider.

One interesting outcome of this project would be to see what kind of cities the agent would build. Would it create high-tax or low-tax cities, with or without public transport and nuclear power, functionally separated or combined, with highways cutting through the city or a more organic structure? This could then be compared to the multitude of comments on SimCity which have tried to interpret what politics the game expresses through its procedural rhetoric. Maybe we are reading our own politics into the game as much as, or more than, the game expresses a politics?

Once we have created agents that can play SimCity, we can create new SimCity-likes, optimized to make the agents create specific kinds of cities. In other words, we could find game-simulations of society that embed political messages. This would require us to create a language for SimCity-like simulation games, which would be an interesting undertaking in itself. The goal for this extension of the project would be to algorithmically probe the ways in which we could reconfigure simulation games to send specific messages.

15 Games and Societies: Are National Game Preferences Predictable?

Participants: Mike Preuss, Mirjam P Eladhari, Antonios Liapis, Sila Temsiririrrkul, and Georgios N. Yannakakis

General Discussion

The group started with the general idea that (computer) games and societies nowadays interact in various ways. As a starting point, we enumerated different types of interactions, placing them into a general context.

- Games are made by humans and these are living in a cultural context, hence reflecting the contexts to different degrees. Games are never completely 'universal' in the sense that a game always has a cultural fingerprint that enables conclusions on where it came from (even relatively universal games as Chess and Civilization inherit some cultural identity).
- Educational games are often made from data sources such as Wikipedia, which themselves are biased towards the societies their main contributors live in. On the other hand, they also influence (educate) societies.
- Games for health are a good example of computer games actually targeted at societal effects.
- Societies and their traditions highly influence the reward systems as well as the types of cooperation and competition that are prevalent and also expected by the audience, and these are also reflected in the mechanisms used in computer games.
- Computer games are a modern type of media that is especially widespread and thus influences cultural convergence.

Cultural Differences in Games

When looking at the cultural differences we find represented in games, especially in the AI parts of games, we presume that these are most recognizable in NPC behavior and Procedural Content Generation. This lead to the question if it would be possible to look at a game and estimate where it comes from. However, very popular games are played in a cross-cultural way (e.g. Zelda, FIFA), made in one region of the world but played in many. This would probably mean that they are played, reviewed and criticized in different ways in different cultures. Games that are played worldwide with dedicated server areas (e.g. League of Legends, StarCraft II, Final Fantasy Online) show that there are differences in user preferences, choices and decisions [8]. Also rating appears to work differently with respect to the cultural environment [6]. For example, it is a common belief that where Asians have a tendency not to be exposed with their opinion, Europeans generally seem to be a bit more critical than Americans [3; 7].

Predicting Popularity of Games in Different Countries

Based on these insights, it is possible to hypothesize that if games express the different factors that make up cultural identities, and we have a model that assigns the factors to cultures in a quantitative way, we could, with access to large amounts of data on how popular different games are in different societies/cultures/nations, and can then learn to predict how popular a game will be in another cultural context.

Data in this domain is available to a certain extent, e.g. via the web page www.vgchartz.com. We aim at connecting this to Hofstede's cultural dimensions theory [5; 4] that attempts to explain cultural differences across nations with a numerical 6-dimensional scale. Training a machine learning model on this may enable making reasonable predictions about games that are not yet released or also point to generally close relations between national game markets. Moreover, the ordered list of most popular games in www.vgchartz.com lends itself well to preference learning [2; 1], where a trained model finds the ranking order for a new set of instances. We are aware that this approach is limited in the way that for certain games, the popularity is based on factors that are probably not to be found in Hofstede's theory, as e.g. for games that model popular sports as football. It is also important to note that this type of work inherently builds upon generalized categorizations of cultural aspects. Nevertheless, this could be an interesting and useful avenue of research.

References

- [1] Farrugia, V.E., Martínez, H.P., Yannakakis, G.N.: The preference learning toolbox. arXiv preprint arXiv:1506.01709 (2015)
- [2] Fürnkranz, J., Hüllermeier, E.: Preference learning. In: Encyclopedia of Machine Learning, pp. 789–795. Springer (2011)
- [3] Handy, C.: Tocqueville revisited. the meaning of american prosperity. *Harv Bus Rev* **79**(1), 57–63 (2001)
- [4] Hofstede, G., Hofstede, G.J., Minkov, M.: Cultures and Organizations - Software of the Mind: Intercultural Cooperation and its Importance for Survival (3. ed.). McGraw-Hill (2010)
- [5] Hofstede, G.H.: Cultures and Organizations - Software of the Mind: Intercultural Cooperation and its Importance for Survival. McGraw-Hill (1991)
- [6] Kübler, R., Pauwels, K., Yildirim, G., Fandrich, T.: App popularity: Where in the world are consumers most sensitive to price and user ratings? *Journal of Marketing* **82**(5), 20–44 (2018)
- [7] de Tocqueville, A.: De la democratie en Amerique. Saunders and Otley (1835)
- [8] Yee, N.: Chinese gamers are more competitive and completionist, more homogeneous in gaming motivations than us gamers. <https://quanticfoundry.com/2018/11/27/gamers-china-us/> (2018)

16 General AI Game Commentary

Participants: Tomoharu Nakashima, Daniele Gravina, Paolo Burelli, and Ruck Thawonmas

In recent years, there has been a growing interest in game commentary, especially with the spread of game-only video streaming services, such as Twitch.tv⁴ and Youtube Gaming⁵. Broadly we can classify game commentary in two primary categories, *let's play* and *live-streaming*. Let's play are videos where a player records the playthrough of the game while documenting the video with his/her personal experience; typically they are recorded offline and heavily edited. Different let's play videos exist, ranging from a detailed analysis of the game design system to simple reaction videos; often, they are also accompanied by a camera view of the player's face. Live-streaming are videos where online commentary is provided on a playthrough, without editing, and typically comment on live e-sports matches. This commentary happens online, during the match, they require fast-paced commentary and the ability to describe in a few words the action happening on the screen.

The use of Artificial Intelligence (AI) for game commentary is relatively underexplored, although it might benefit both professional streamers and video-makers. Previous work has addressed this area of research with different approaches. In [4] it is proposed to select the best camera view based on a machine learning approach. In [6], it is proposed to use unsupervised learning to detect the interesting highlight in playthrough videos. Cinematography is addressed in [5], where it is proposed a high-level approach where a user can decide the best camera view for the selected scene.

Given the potential advantages of an AI generated game commentary, the working group identified a number of interesting challenges for both let's play and live-streaming commentary. Specifically, depending on the application, several different problems can be identified. The first application is highlight detection. This task requires detecting the most interesting scenes from a video, based on the audience reactions or the number of viewers. Cinematography is another application. In this task, we have to identify the best way to capture what is happening on the screen and which camera view is the best to emphasize it. Finally, the last application is game commentary. This task implies an underlying story to be narrated and requires coordination between the shown action and the comments.

Motivated by the potential real-world applications and by the number of interesting open problems posed by game commentary, our working group discussed and proposed a general approach for General Game-AI commentary. Specifically, given the necessity of targeting different tasks and providing a system able to adapt to different games, we propose a modular system capable of generalizing the narrative underlying playthrough videos (see Fig. 16.1). Our system uses different input sources, such as video from streaming video services and labels (hand-made or automatically extracted from cues). Given the input and the labels, different approaches can work together to extract the most important scenes. This part is responsible for extracting the most interesting part of the video given the labels, with data-driven models (preference learning [1] or attention-based machine learning [2]) or knowledge-based models. Once this information is processed we can use the high-level information extracted to model a general narrative. Given the different challenges identified above, multiple general narrative models can be trained; in the case of live-streaming, for instance, a short and meaningful description of the scene is more important compared to a thorough analysis of the playthrough. A possible implementation may employ a partial-ordering of the narrative, as depicted in [3] or less structured solutions, depending on the purposes and sources of information used. Once this general narrative model has been obtained, we can use it for a multitude of outputs, depending on the application.

References

- [1] Yannakakis, Georgios N., Roddy Cowie, and Carlos Busso. "The ordinal nature of emotions: An emerging approach." *IEEE Transactions on Affective Computing* (2018).
- [2] Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y., "Attention-based models for speech recognition." *Advances in neural information processing systems*. (2015).

⁴www.twitch.tv

⁵www.youtube.com/gaming/

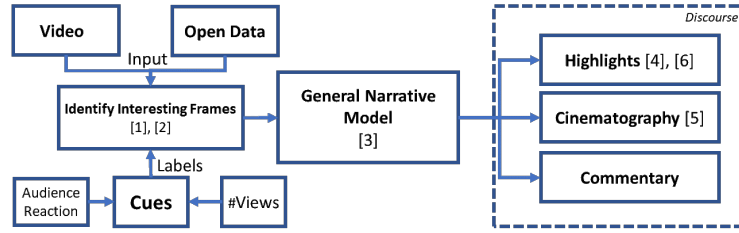


Figure 16.1: Modular system for General AI Game Commentary. This system can be divided in three major components: interesting frames identification, general narrative modeling, and output generation depending on the final application.

- [3] Magerko, B. “Story Representation and Interactive Drama.” *AIIDE* (2005).
- [4] Chen, Jianhui, Lili Meng, and James J. Little. “Camera selection for broadcasting soccer games.” *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, (2018).
- [5] Velho, Luiz, Leonardo Carvalho, and Djama Lucio. “Live Probabilistic Editing for Virtual Cinematography.” *International Conference on Entertainment Computing*. Springer, Cham, (2018).
- [6] Ringer, Charles, and Mihalis A. Nicolaou. “Deep unsupervised multi-view detection of video game stream highlights.” *Proceedings of the 13th International Conference on the Foundations of Digital Games*. ACM, (2018).

17 Optimization of Game Design with Parameter Tuning, Content Generation with Psychological Player Model

Participants: Tomohiro Harada, Luiz Bernardo Martins Kummer, Joon-Hong Seok, Sebastian Risi, Simon Lucas, Maxim Mozgovoy, and Florian Richoux

Motivation

This topic summarizes automatic optimization of game content. Content targeted for optimization includes character parameters, level components, game rules, and so on. Automatic optimization of game content can generate many contents suitable for different difficulty levels and intentions. When optimizing game content, the following three factors need to be considered.

- **Optimization algorithms:** Evolutionary computation methods such as genetic algorithm and evolutionary strategy, and machine learning methods such as neural network and reinforcement learning are considered as candidates.
- **Fitness functions:** In order to generate game content automatically, fitness functions are needed to evaluate the quality of the content. In order to design fitness functions, several methods can be used: hand coded, rule based, and AI agent based. The indicators used in the evaluation may be game performance, intrinsic motivations, novelty search. It is important to design fitness functions that combine these factors appropriately.
- **Game design / content + search space:** It is important to decide which content of the game to optimize and how to express them so that they can be handled by the optimization method. Possible targets of optimization are positions and features of items, obstacles, enemies, and rules in games. To encode them, discretized value and continuous value can be considered. For these encoding, it is necessary to decide how to determine the granularity of the discretized value and how to design the range of values in each encoding method.

Psychological models

In this topic, we discuss an example of using psychological models proposed by Bartle [1] as a content evaluation index. Bartle proposed the following four player types.

1. **Achiever:** prefers to gain “points” or “score” in games
2. **Explorer:** prefers to gain discovered areas
3. **Socializer:** prefers to play games for the social aspect, rather than the actual game itself
4. **Killer:** prefers to fight with other players, rather than the actual game itself

Example game: Cave Swing

We take Cave Swing as an example of game optimization. Cave Swing is a game that aims at reaching a preset goal while avoiding dangerous areas. The player character moves by swing using the closest anchor. Targets of optimization are the game physics: the gravity of x- and y-axes and the attraction of the anchor, and the game components: the positions of the anchors, the goal area, the dangerous area, and the size of the game stage. Cave Swing provides a player agent using Rolling Horizontal Evolution Algorithm, and parameters can be optimized by N-Tuple Bandit Evolutionary Algorithm (NTBEA) [2].

As a baseline fitness function, we consider the following AI agent based one:

$$F = (\text{Score of strong player} - \text{Score of weak player}) \quad (1)$$

The first term is the score acquired by the strong player play, while the second one is the score acquired by the weak player play. By using this fitness function, you can generate levels that require more play skills.

By adding features and fitness function considering the psychological model to Cave Swing, game content suitable for one or more psychological models can be generated. For example, by adding moving anchors, you can generate content that both Achiever and Explorer can

enjoy. Alternatively, by developing into a multiplayer game, it is possible to generate content suitable for Socializer and Killer.

An example code is available from the following GitHub link: <https://github.com/SimonLucas/KotlinTest>

Next step

As the next step in the optimization of game content, we consider a method that considers the psychological model in this discussion. Other than this, generation of game content according to desired game duration of the game designers, real-time in-game optimization using fast forward model and SFP player can be considered.

References

- [1] R. Bartle: Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD research* **1**(1), 19 pages (1996)
- [2] Kunanusont, K., Gaina, R. D., Liu, J., Perez-Liebana, D., Lucas, S. M.: The N-Tuple Bandit Evolutionary Algorithm for Automatic Game Improvement. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2201–2208 (2017)