

Artificial and Computational Intelligence in Games: AI-Driven Game Design

Edited by

Pieter Spronck¹, Elisabeth André², Michael Cook³, and
Mike Preuß⁴

1 Tilburg University, NL, p.spronck@gmail.com

2 Universität Augsburg, DE, andre@informatik.uni-augsburg.de

3 University of London, GB, mike@gamesbyangelina.org

4 Universität Münster, DE, mike.preuss@wi.uni-muenster.de

Abstract

With the dramatic growth of the game industry over the past decade, its rapid inclusion in many sectors of today's society, and the increased complexity of games, game development has reached a point where it is no longer humanly possible to use only manual techniques to create games. Large parts of games need to be designed, built, and tested automatically. In recent years, researchers have delved into artificial intelligence techniques to support, assist, and even drive game development. Such techniques include procedural content generation, automated narration, player modelling and adaptation, and automated game design. This research is still very young, but already the games industry is taking small steps to integrate some of these techniques in their approach to design.

The goal of this seminar was to bring together researchers and industry representatives who work at the forefront of artificial intelligence (AI) and computational intelligence (CI) in games, to (1) explore and extend the possibilities of AI-driven game design, (2) to identify the most viable applications of AI-driven game design in the game industry, and (3) to investigate new approaches to AI-driven game design. To this end, the seminar included a wide range of researchers and developers, including specialists in AI/CI for abstract games, commercial video games, and serious games. Thus, it fostered a better understanding of and unified vision on AI-driven game design, using input from both scientists as well as AI specialists from industry.

Seminar November 19–24, 2017 – <http://www.dagstuhl.de/17471>

1998 ACM Subject Classification I.2.1 Artificial Intelligence - Games

Keywords and phrases dynamical systems, entertainment modeling, game design, multi-agent systems, serious games

Digital Object Identifier 10.4230/DagRep.7.11.86

1 Executive Summary

Pieter Spronck

Elisabeth André

Michael Cook

Mike Preuß

License  Creative Commons BY 3.0 Unported license

© Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß

The video game industry has been developing rapidly in the past decade. Whereas ten years ago video games were almost exclusively aimed at entertainment, nowadays they are used



Except where otherwise noted, content of this report is licensed under a Creative Commons BY 3.0 Unported license

Artificial and Computational Intelligence in Games: AI-Driven Game Design, *Dagstuhl Reports*, Vol. 7, Issue 11, pp. 86–129

Editors: Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß



Dagstuhl Reports

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

in a variety of places in everyday life. All kinds of organizations now use video games for simulation and training. Educational institutes use video games to enrich and replace parts of courses. Governmental and health care agencies use video games to educate people and stimulate them to lead more productive lives. On top of that, the entertainment-focused games industry continues to grow and is a major industry both culturally and financially.

Two parallel developments can be observed in the games industry. On the one hand, the high-profile entertainment games (“triple-A games”) see a steady increase of time and financial resources invested in their development, to keep up with technological advances and to be able to compete in a tough market. On the other hand, the number of smaller, special-purpose games in development (including so-called “serious games”) increases dramatically too, in particular in research, training, and education. Moreover, as the pervasiveness of video games increases, so does the number of people involved in creating them. The job of creating games is no longer limited to specialist programmers and artists. Instead, those who need to use the games become heavily involved in their creation.

The serious-games domain poses additional challenges to game development beyond all the challenges already posed by games for entertainment, namely the need for a strong relation with the “real world”. Serious games often have a purpose in training, which entails that the game worlds must be a realistic depiction of the actual environment in which user functions, in particular where “behaviors” are concerned. The big data revolution means that huge quantities of data about the real world are becoming available along with the means of processing them, which may offer the possibility to automatically construct games on the basis of such data. This is a particularly enticing notion, given the financial constraints for constructing serious games, which means that professional content producers might not be available.

Furthermore, new computer games are expected to much better incorporate the different needs of a wide variety of customers, to provide more alternative modes, solution paths, incentives, emotional states, and difficulty levels. Game design, and especially balancing, must take this into account. However, this increases the complexity of design and production considerably, such that AI-based tools that can assist the human developer or even partly automatize processes are more desired than ever.

Summarizing, we note the following four trends in modern game development:

1. Technological advances have lead to an increased challenge in developing modern video games, even for expert game developers
2. There is an increased need for non-experts to be able to design and develop games
3. There is an increased need for realism in the virtual world behaviors, in particular in the area of serious games
4. A greater variety of players and a better availability of data about players leads to the need for more variable and better customizable games, which require a more complex development process.

A solution for each of these issues can be found in the application of artificial-intelligence (AI) techniques to drive the design and development of games. From the perspective of AI-driven game design, AI supports or even takes over the role of the human game developer in creating particular elements of a game, and even complete new games.

While the game industry tends to use a small selection of well-known algorithms to generate elements of game worlds (in particular where graphics and animation are concerned), the use of AI to create new environments, new behaviors of virtual characters, new narratives, new game rules, or even new gameplay mechanisms is at present limited to a very small

number of researchers. We see it, however, as the main direction in which innovation in game design and development can be found.

AI-driven game design sees applications in the design of virtual worlds, virtual characters, narratives, and game mechanics. Moreover, it can be used to assist in the human design process, and to adapt games automatically after publication. Finally, it can support the automated analysis of generated game elements. Each of these topics is a research domain by itself, which requires an interdisciplinary approach which borrows from computer science, psychology, cognitive science, and even the creative arts. A common ground is found in artificial intelligence techniques, in particular machine learning.

For this seminar, we brought together computer scientists and creative experts with the common goals of gaining a deeper understanding of various aspects of games, and of further improving games, in particular by using AI-techniques used to generate games and game elements. The goal was to look beyond what is currently possible and in use, and take steps towards the future of AI-driven game development. Besides theoretical discussions, part of the seminar was spent on trying to achieve first practical results.

Reports on the discussions and results achieved are found on the following pages. All in all, the organizers and participants deemed the seminar a great success, and are eager to continue into some of the directions that were focused on during the week at Schloss Dagstuhl.

2 Table of Contents

Executive Summary

Pieter Spronck, Elisabeth André, Michael Cook, and Mike Preuß 86

Overview of Talks

Some Observations on Automated Strategy Game Design
Cameron Browne 91

Working groups

Game Search Space Design and Representation
Dan Ashlock, Cameron Browne, Simon Colton, Amy K. Hoover, Jialin Liu, Simon M. Lucas, Mark J. Nelson, Diego Perez Liebana, Sebastian Risi, Jacob Schrum, Adam M. Smith, Julian Togelius, and Vanessa Volz 93

Blue Skies Research Questions in AI-driven Game Design
Peter I. Cowling, Elisabeth André, Rafael Bidarra, Pier Luca Lanzi, Mike Preuß, Spyridon Samothrakis, Pieter Spronck, Shoshannah Tekofsky, and Georgios N. Yannakakis 95

Some Industry Interests in AI-driven Game Design
Peter I. Cowling, Sander C.J. Bakkes, Pier Luca Lanzi, Adam M. Smith, Pieter Spronck, and Shoshannah Tekofsky 97

Machina Ex Machina (Part 1)
Reynald Francois, Matthew J. Guzdial, Marc Erich Latoschik, Antonios Liapis, Alexander Nareyek, Emily Short, and David Thue 100

Machina Ex Machina (Part 2)
Reynald Francois, Joshua Allen McCoy, Alexander Nareyek, and G. Michael Youngblood 100

What Is Machine Learning/Deep Learning?
Matthew J. Guzdial, Joshua Allen McCoy, Emily Short, Michael Treanor, R. Michael Young, and Jichen Zhu 101

AI-assisted Board Game Play
Antonios Liapis, Michael Cook, Steve Dahlskog, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Gillian Smith, Anne Sullivan, and Tommy Thompson 104

Mixed-media Game AI
Antonios Liapis, Elisabeth André, Sander C.J. Bakkes, Rafael Bidarra, Steve Dahlskog, Mirjam P. Eladhari, Ana Paiva, Mike Preuß, Gillian Smith, Anne Sullivan, Tommy Thompson, David Thue, Georgios N. Yannakakis, and R. Michael Young . 105

Emergence
Jacob Schrum, Cameron Browne, Mark J. Nelson, Ana Paiva, Julian Togelius, and Georgios N. Yannakakis 108

Human-assisted Creation of Content Within Games
Jacob Schrum, Pier Luca Lanzi, Alexander Nareyek, and Pieter Spronck 111

Play
Jacob Schrum, Cameron Browne, Amy K. Hoover, Sebastian Risi, David Thue, Julian Togelius, and Jichen Zhu 112

AI As Reflective Practice	
<i>Gillian Smith, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Adam M. Smith, Anne Sullivan, Tommy Thompson, and R. Michael Young</i>	113
The Happiness Game	
<i>Pieter Spronck, Alex J. Champandard, Peter I. Cowling, and Pier Luca Lanzi</i>	115
Social Network Modelling in Video Games (Part 1)	
<i>Shoshannah Tekofsky, Dan Ashlock, Matthew J. Guzdial, Emily Short, David Thue, Michael Treanor, and Jichen Zhu</i>	116
Social Network Modelling in Video Games (Part 2)	
<i>Shoshannah Tekofsky, Dan Ashlock, and G. Michael Youngblood</i>	117
Backstory Generation	
<i>David Thue, Mirjam P. Eladhari, Joshua Allen McCoy, Mike Preuß, Spyridon Samothrakis, Emily Short, Anne Sullivan, Michael Treanor, and R. Michael Young</i>	118
A General Language for Matching Tile Games	
<i>Julian Togelius, Cameron Browne, Simon Colton, Mark J. Nelson, and Diego Perez Liebana</i>	119
Gameplay Evaluation Measures	
<i>Vanessa Volz, Dan Ashlock, Simon Colton, Steve Dahlskog, Jialin Liu, Simon M. Lucas, Diego Perez Liebana, and Tommy Thompson</i>	122
Explainable AI for Designers	
<i>Jichen Zhu, Rafael Bidarra, Alex J. Champandard, Simon Colton, Reynald Francois, Matthew J. Guzdial, Amy K. Hoover, Antonios Liapis, Sebastian Risi, Gillian Smith, Anne Sullivan, and G. Michael Youngblood</i>	125
Panel discussions	
Evaluation	
<i>Pieter Spronck</i>	127
Participants	129

3 Overview of Talks

3.1 Some Observations on Automated Strategy Game Design

Cameron Browne (RIKEN - Tokyo, JP)

License © Creative Commons BY 3.0 Unported license
© Cameron Browne

The program LUDI, developed for my PhD studies [1], evolved combinatorial board games as symbolic expressions representing structured rule trees. The generation of games proved easy – the system could evolve thousands of candidates per second using standard genetic programming operators – but their evaluation proved much more difficult. Evaluation required approximately 30 minutes of self-play analysis per game, and the derivation of appropriate aesthetic indicators made up the bulk of the research effort.

The quality of a game, i.e. its potential to interest human players, cannot be evaluated by its rules alone. Much of a game’s appeal lies in the unexpected emergent behaviour that can occur as it is played, which may not be deducible from the rules. For example, the most popular game evolved by LUDI, which it named Yavalath [2], contains two apparently contradictory rules: players win by making 4-in-a-row of their colour but lose by making 3-in-a-row of their colour beforehand. This combination produces an interesting forced move mechanism when played, adding tension and drama to the game and making it interesting for players.

Such emergent strategies can be not just entertaining for players, but crucial for their comprehension of the game. For example, in the recently invented game Omega [3], players’ scores are based on the product of group sizes, requiring a degree of calculation that made the game mentally exhausting and difficult to plan ahead. Players found the game opaque and typically made uniformed moves until forced to count the score at the end, removing any tension and making it initially unpopular. It was not until Omega was implemented in the Axiom game system [4] that an emergent strategy was observed. The system’s Monte Carlo tree search (MCTS) [5] move planner made moves consistent with a strategy of forming its own pieces into groups as close to size 3 as possible, while forming enemy pieces into groups as far from size 3 as possible. This was later proven to be an optimal strategy for the game [3].

This optimal strategy of “form groups of size 3” provided a convenient meme for players that made the game more comprehensible and outlined a concrete plan of action. It also revealed the game to fundamentally be both a connection game and an anti-connection game [6], as players sought to connect enemy groups into larger configurations while blocking their own groups from being so extended, which imported a whole slew of implicit sub-strategies. Players immediately found Omega more accessible and enjoyable through a simple change of perception, and the discovery of this simple strategy saved the game.

Lantz et al. introduce the notion of the strategy ladder [7], in which players learn increasingly complex strategies relevant to a game as they play it, that build upon each other. They posit that the most interesting games are those with a constant and linearly increasing strategy ladder. This makes good sense as such games would give players both something to play towards (the strategies they know) and something to learn (the strategies they don’t know). Games in which winning strategies are trivially learnt and applied would be too simple to be of interest to players, while those in which even the simplest strategies are excessively difficult to learn would be too intractable to be of interest to players. This resonates with the observation by Allis et al. that games which survive do so because they provide an intellectual challenge at a level which is neither too simple to be solvable, nor too complex to be incomprehensible [8].

The question then arises: when automatically evaluating games for their potential to interest human players through AI self-play, what level of playing strength will best capture an authentic experience of the game as played between human players? Random play will obviously not simulate the experience of the game as played between intelligent players. Conversely, superhuman AI play could go too far to the other extreme, and give an equally unrepresentative experience of the game as played by human players. For example, Draughts is drawish when played at even the human champion level – international tournaments have ended with a whimper when finalists drew all 20 games in the final [9] – but remains an engaging and hugely popular game for the average player worldwide.

There is constant and understandable pressure in the game AI research community to strive for superhuman results in all cases. However, I argue that capping the playing strength of AI agents at a lower (strong human) level for the purposes of game evaluation, is more likely to capture a realistic “human” experience of the game. But even estimating what constitutes “strong human level” for a given game remains an open question.

References

- 1 C. Browne. *Automatic Generation and Evaluation of Recombination Games*. Evergreen Terrace PhD thesis, QUT, Brisbane, 2008
- 2 C. Browne. *Evolutionary Game Design*. Springer, 2011
- 3 N. Romeral Andres. From Mathematical Insight to Strategy. In *Game & Puzzle Design*, 1(2):53–54, 2015
- 4 G. Schmidt. The Axiom General Purpose Game Playing System. In *IEEE Transactions on Computational Intelligence and AI in Games (T-CIAIG)*, 6(4):332–342, 2014
- 5 C. Browne et al. A Survey of Monte Carlo Tree Search Methods. In *IEEE Transactions on Computational Intelligence and AI in Games (T-CIAIG)*, 4(1):1–43, 2012
- 6 C. Browne. *Connection Games: Variations on a Theme*. AK Peters, 2005
- 7 F. Lantz, A. Isaksen, A. Jaffe, A. Nealen, and J. Togelius. Depth in Strategic Games. In *The AAAI-17 Workshop on What’s Next for AI in Games? (WS-17-15)*, 967–974, 2017
- 8 L. V. Allis, H. J. van den Herik, and I. S. Herschberg. Which Games Will Survive? In *Heuristic Programming in Artificial Intelligence 2: The Second Computer Olympiad*, Ellis Horwood, 232–243, 1991
- 9 C. Freeling. Dameo: A New Step in the Evolution of Draughts? In *Game & Puzzle Design*, 1(2):28–37, 2015

4 Working groups

4.1 Game Search Space Design and Representation

Dan Ashlock (University of Guelph, CA), Cameron Browne (RIKEN - Tokyo, JP), Simon Colton (Falmouth University, GB), Amy K. Hoover (NJIT - Newark, US), Jialin Liu (Queen Mary University of London, GB), Simon M. Lucas (Queen Mary University of London, GB), Mark J. Nelson (Falmouth University, GB), Diego Perez Liebana (University of Essex - Colchester, GB), Sebastian Risi (IT University of Copenhagen, DK), Jacob Schrum (Southwestern University - Georgetown, US), Adam M. Smith (University of California - Santa Cruz, US), Julian Togelius (New York University, US), and Vanessa Volz (TU Dortmund, DE)

License © Creative Commons BY 3.0 Unported license

© Dan Ashlock, Cameron Browne, Simon Colton, Amy K. Hoover, Jialin Liu, Simon M. Lucas, Mark J. Nelson, Diego Perez Liebana, Sebastian Risi, Jacob Schrum, Adam M. Smith, Julian Togelius, and Vanessa Volz

Automatically generating games, including rules, environments, and boards, is a difficult type of automatic content generation [2, 5, 6, 8]. There are many possible levels on which Automatic Game Design (AGD) can happen. One of the simpler types of AGD is to take an existing game with a large number of parameters and set those parameters so that the resulting game is fun or interesting [4]. This group discussed the possibilities for making the problem of automatic game design easier by designing the search space, while assuming that only playable games are of interest.

The manifesto the group decided on was to locate search spaces in which most points were good games, or at least search spaces that were enriched with good games. The simple example of parameter tuning to locate a good game shows that even the simplest case of automatic game design may need to assume some degree of automatic tuning. This means that a highly enriched search space may be one that assumes that the fitness measure is *Baldwinian*, it uses the object being evolved as the starting point for a local optimization and returns the fitness of the optimized object.

Tools for search space design and enrichment

An approach that has been effective in the past is to carefully design the representation for search. A simple example of this would be to create a collection of level fragments that include a matching rule for how they may be linked, and then evolve the choices of which fragments to use. This amounts to decomposing the level design problem into a fragment design problem and a fragment assembly problem, both of which are easier. This approach is synergistic with automatic parameter tuning.

A more general approach that includes decomposition is that of search space transformation. Abstract the essential details that define multiple versions of a puzzle and then search the abstraction. The principle of transforming the search space can be realized in a number of ways and might make a good target for a special session at a conference.

Another potentially important element is to create a search space for games that embeds common user experience measures within the design or which leaves hooks or entry points for measuring or sampling user behavior. This speaks not so much to the space of games defined, but to effective design in which the game designed, whatever its nature, is relatively easy to evaluate from a user experience perspective. This sort of design criteria can include creating games that are easy to crowd-source the gathering of user experience data.

Many game design spaces are *ad hoc* with the parameters to be tuned arising from code that plays an initial version of the game. While being able to tune whatever game someone sends you source code for is a valuable skill, designing search spaces of games may be made easier by employing formal semantics. Evolutionary computation can search the realizations of a grammar without difficulty. Incorporating formal semantics into the design space for a category of games employs a natural and well developed set of tools within computer science and is a method for both search space decomposition and transformation. It also would help to formalize the process of including user experience data gathering hooks.

Another possible approach is to discover structures that are highly evolvable. If we have a game specification for a game that is evolutionarily close to other good games, then at least the part of the search space where these games lie is an enriched search space. This idea echos the idea of exploitation in evolutionary computation: having found a good area, examine it more closely. This, in turn, raises the issue of representation. Turning the problem on its head, an effective representation is one in which good games are often close to one another. In this case the metric used to define “close” is evolutionary time needed to discover one object when the other is available in the evolving population.

The *mutual fertility* of two members of an evolving population is the expected fitness of their offspring, excluding offspring that are in effect clones of the parents. Searching for high fertility parents and then using these as population seeds is an automatic way of locating a rich search space. If the final phases of a design process are human-in-the-loop evaluation, locating high fertility starting points may be a way of increasing the efficiency of human-in-the-loop systems by avoiding user fatigue. It is worth noting that the fertility of parents is an algorithm-dependent notion.

Anchoring a rich system

The group proposed the following method of anchoring and evaluating a rich search space. It should be possible to express existing games in the representation used for search and use such games as anchors and as genetic seeds. The definitive quality measure for such a game search space is the rate at which good, novel games are discovered. Requiring that existing games be possible is a way of placing the search space in what is already known to be good territory. There may be a price paid in degree of novelty and the group thought this might be an excellent area for additional research.

A subgroup continued to investigate this thought further, following the idea of using a Generative adversarial network (GAN) to encode the anchor. Generative adversarial networks are a popular algorithm in machine learning usually applied to image generation, where impressive results were obtained in terms of the imitation of the style of the training image set [3]. The group therefore investigates how GANs can be applied to PCG using the MarioAI framework as an example. The developed approach is detailed in the following.

We use a GAN to generate Mario levels represented as a matrix of integers that encode the different tiles and enemies in the level. The training corpus are the levels for the original Super Mario Bros. game taken from the *Video Game Level Corpus* [7]. After the GAN has concluded training, we have obtained a model that is able to generate different Mario levels based on a random input vector. Next, we optimise the input vector (based on [1]) in order to obtain playable Mario levels using a CMA-ES. Our evaluation function for the playability of a level is based on importing the generated levels into the MarioAI framework and running the (A* agent) on them.

We were able to observe very promising results with our first prototype in terms of the ability of the GAN to reproduce a basic level structure. However, almost none of the generated levels were playable due to the height of the required jumps. We therefore started experimenting with different versions of GANs as well as representations.

References

- 1 P. Bontrager, W. Lin, J. Togelius, and S. Risi. Deep interactive evolution. In *Proceedings of Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt 2018)*, 2018.
- 2 Cameron Browne. *Evolutionary Game Design*. Springer, 2011.
- 3 Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 2014 Conference on Neural Information Processing Systems (NIPS 2014)*, pages 2672–2680, 2014.
- 4 Kamolwan Kunanusont, Raluca Gaina, Jialin Liu, Diego Perez-Liebana, and Simon Lucas. The n-tuple bandit evolutionary algorithm for game improvement. In *Proceedings of the Congress on Evolutionary Computation*, pages 2201–2208, 2017.
- 5 Mark J. Nelson, Julian Togelius, Cameron Browne, and Michael Cook. Rules and mechanics. In *Procedural Content Generation in Games*, pages 99–121. Springer, 2016.
- 6 Thorbjørn S. Nielsen, Gabriella A. B. Barros, Julian Togelius, and Mark J. Nelson. Towards generating arcade game rules with VGDL. In *Proceedings of the 2015 IEEE Conference on Computational Intelligence and Games*, 2015.
- 7 Adam James Summerville, Sam Snodgrass, Michael Mateas, and Santiago Ontañón. The VGLC: The video game level corpus. *Proceedings of the 7th Workshop on Procedural Content Generation*, 2016.
- 8 Julian Togelius and Jürgen Schmidhuber. An experiment in automatic game design. In *Proceedings of the 2008 IEEE Symposium On Computational Intelligence and Games*, pages 111–118, 2008.

4.2 Blue Skies Research Questions in AI-driven Game Design

Peter I. Cowling (University of York, GB), Elisabeth André (Universität Augsburg, DE), Rafael Bidarra (TU Delft, NL), Pier Luca Lanzi (Polytechnic University of Milan, IT), Mike Preuß (Universität Münster, DE), Spyridon Samothrakis (University of Essex - Colchester, GB), Pieter Spronck (Tilburg University, NL), Shoshannah Tekofsky (Spirit AI - London, GB), and Georgios N. Yannakakis (University of Malta - Msida, MT)

License © Creative Commons BY 3.0 Unported license

© Peter I. Cowling, Elisabeth André, Rafael Bidarra, Pier Luca Lanzi, Mike Preuß, Spyridon Samothrakis, Pieter Spronck, Shoshannah Tekofsky, and Georgios N. Yannakakis

The aim of this group on the second day was to generate high-risk-high-return research directions to stimulate research in the games AI research community. The questions discussed and a brief commentary are given below. Some of these ideas were developed further in the workshop and will be discussed in other reports. The titles are deliberately short – since the groups initial task was to take 10 minutes to come up with a title of at most 6 words – this seemed to stimulate creativity.

Individual Turing Test (Shoshanna): Can we create an AI that learns from player data for both a group of players and an individual, that can pass the Turing test of having a behavioural profile in the game which is hard to distinguish from the individual player. Having effective AI Natural Language Processing would be a game changer for this – but there is much we can do without advances in NLP.

Paths of inevitability for persistent worlds (Spyros): Games and simulation environments tend to reach a point of stagnation where player actions, interactions and perceptions tend to repeat. It is desirable to provide a revolutionary shake-up of game worlds from

time to time to maintain game and player interest. The question was to use AI to find the minimal change “meme” which would snowball into large-scale change without any stage of the change feeling unnatural to the player (consider the disruptive effect of mobile phones on society rather than that of wholesale population uprising).

Non-paradoxical entertaining tuned challenge (Pieter): An AI which acts as a director in a game, tuning the game world to an individual player. This might be seen in the context of an AI which chooses human opponents/team members with similar levels of skill to the player.

ML (Georgios): This project proposal involved extracting information about player interaction with a game – and used small amounts of data about an individual and transfer learning to learn what an individual player might like. Understanding and providing critique of what the ML is doing is a key.

Positive Computing (Elisabeth): There has been much interest in the HCI community as to value-centred design – for example designing games to make players happier, rather than simply to make them absorbed/addicted. The Wild Divine Game was cited as an interesting example in this area. Also replacing humans by AIs “just because we can” was not felt universally positive.

Science of Happy (Peter): Games, interactive and social media reach around 50% of planet earth, and through interaction have the potential to be a major positive influence on society. The research questions were about games with the purpose of making the player happy, through encouraging players to exercise, be mindful, connect with others, reflect, set goals etc. Habitica was cited as an interesting example where habits and action lists were combined into a game. We can conceive of AI mentors (possibly data-driven – Amazon recommenders know what you might want to buy – an AI mentor recommender might propose some action you could take to encourage positivity and positive self talk). Approaches such as Cognitive Behaviour Therapy might be embedded. . . There are psychology models that could help – such as the “big 5” dimensions of personality (Extraversion, Agreeableness, Conscientiousness, Neuroticism, Openness).

The questions above related to happiness and positivity were taken further by two groups (that later merged) to consider multi-agent economic models which also consider happiness, and whether policy or environment change could be investigated in such models – for example by the introduction of an effective meme. Work is needed to understand the work in the economic and sociology literature already undertaken in this area. It seems likely contributions using the expertise of the Game AI research community would be in embedding significant intelligence in the agents, and having an AI “government” which investigated the space of “political systems” for leading the system of agents. AI is also needed for analysis and understanding of the data from the multi-agent system.

While it is challenging to consider “AI for Games to change the world” it was useful to reflect and work together on this question of huge scale and potential impact.

4.3 Some Industry Interests in AI-driven Game Design

Peter I. Cowling (University of York, GB), Sander C.J. Bakkes (Tilburg University, NL), Pier Luca Lanzi (Polytechnic University of Milan, IT), Adam M. Smith (University of California - Santa Cruz, US), Pieter Spronck (Tilburg University, NL), and Shoshannah Tekofsky (Spirit AI - London, GB)

License © Creative Commons BY 3.0 Unported license

© Peter I. Cowling, Sander C.J. Bakkes, Pier Luca Lanzi, Adam M. Smith, Pieter Spronck, and Shoshannah Tekofsky

The group convened in the first session of the week to discuss the research questions collected by Peter Cowling before the workshop (given below). The goals of the group were to:

- Find low-hanging fruit – research questions from the games industry where approaches which are well known to the games AI research community might be used.
 - Need to talk to industry about potentially useful techniques from research – these may be used “deeply ingrained problems” such as bot behaviour where only a limited range of approaches is currently used in industry.
 - Given many examples of X, generate more examples of X (where X is art/music/-levels/...) – felt to be a low-hanging fruit with the ready availability of ML and analytics tools.
 - A “button monkey” for games testing, but which more effectively explored the possibility space via MCTS/ML, was also felt to be low-hanging.
- Understand how things function in games companies, and how research ideas might be introduced, and how best to talk to games companies.
 - It was mentioned that amateur/hobbyist games developers are sometimes willing to give full source code for a game
 - Probably useful to both academic and industry communities if models and examples for academic-industry collaboration were made available
 - “Assist a designer” is more useful than “automatic/semi-automatic generation”
 - Academics with good industry contacts, and games industry folks who regularly attend academic research events, can provide an important brokerage service (as had been done here). Also attendance and talks at industry conferences such as the GDC AI summit and nucl.ai were felt promising. Attendance at research events at labs where there is significant industry input (e.g. DC Labs/IGGI at York) was also felt useful
 - Persuading larger games companies to sponsor prizes could be a useful way for them to get prototypes and recognise potential employees
 - Game jams also work well
 - PhD and masters students consultancies/placements provide good opportunities to embed technologies in games companies
- Understand our values – what does the games AI research community want from relationships with the games industry – where are the win-wins?
 - AI for back story was felt to be an interesting, difficult challenge
- Understand some of the “blockers” to getting research AI into industry
 - AI techniques such as ML might be perceived as “too risky” by the games industry, hence risk mitigation is likely to make research AI proposals more acceptable to industry
 - AI will be rated in terms of industry producers in terms of the enhanced experience for the player – there may be a perception of a small return for better AI compared to more levels or better art

Overall we had a positive and interesting discussion with much food for thought to take further later in the week...

The following question was asked by Peter Cowling to various companies which work in game development: **Research in AI-Driven Games Design should consider specific research questions and areas such as...** The following answers were given:

From a **variety of people at one of the world's largest games companies**:

- Machine Learning assisted art production:
 - Make tools that can create music using deep learning (see for example <http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/>), and can generate the music on the fly adapting to the current situation the player is.
 - Develop a library that can learn the art style of a game to help generate more assets following that style, similar to deep learning demos that can transform a picture to match a well-known painter style (<https://qz.com/495614/computers-can-now-paint-like-van-gogh-and-picasso/>)
 - The previous two example were based on audio and 2d data, but can a similar system be created for 3D assets, allowing to help the creation of 3D models within the same art style of the rest of the game.
 - e.g. the game developer wants to give the ML assisted art production tool examples of things they want in their game and say 'make more of this please'.
- Machine Learning for QA:
 - Test which improves itself (in terms of coverage, scope, severity or performance, etc.) within a set of boundary rules.
- Interpreting inputs with Machine Learning:
 - Visual inputs for computer vision
 - Audio object recognition
 - User inputs - voice, face, eye, hand, gesture, movement, brain (e.g. gaze tracking, body tracking)
 - The basic problem is to 'create context from inputs'. (Is the system understanding what the user is doing? 'semantics', the meaning of the input from the system's pov.) (Is the system understanding what the user is feeling - 'emotion'.)
- Intelligent Agents:
 - Conversational interface – being done in Bots and Intelligent Assistants?
 - Intelligent agents that support multiple users (e.g. multiple users' voice input)

From a **CEO of Company making Games With a Purpose**: "Google just integrated AI-APIs into Android. Intel and Qualcomm both have AI chips being manufactured for mobile. What are researchers doing about these?"

From a **Head of Technology at a medium-sized games company**: "One interesting problem that has landed on my desk recently is the design of a system that can automatically generate UI based on a hierarchical tree of information. This information isn't limited to simple check boxes and input fields. It is more of a mind map of all the components that go to make something as complex as a car, from the valve specification right up to the colour and shape the bonnet. Each component has descriptive text along with links to child components that lie within that system or feature. Each component needs UI to allow fields to be changed and customised. Generating a good comprehensive UI for such a set of data is a huge task. An AI helper could be of use in this space.

UI design is an often overlooked feature of any game. It is very hard to get right and is often a lot more complex than it would appear at first glance.

It is also however the first experience that a player gets of any game. Like the lobby of a hotel, it needs to look and feel ‘right’. An AI that can generate good UI look, feel and flow from loose descriptive data of the content and a style guide would be interesting.

Following on from this would be the first time user experience (FTUE). As developers we’re often too close to the product to ‘see’ it as a first time user would. From game boot to 1st play session. Ideally focus testing helps in this space. This then has the problem that you need to find new users continually, and then assess how consistent their responses are from group to group. An AI that can give an assessment of FTUE in a consistent and measureable manner could be helpful.”

From an **Academic Archaeologist with successful museum exhibitions (50K visitors) using VR**: “I am interested in how place based games can help people to engage with museums and cultural heritage sites (e.g. country houses). Nobody has any doubt that museums and old buildings have personalities, I want to know whether AI can help give these personalities a voice. How can people have conversations with old buildings? We can learn from the past, can the past learn from us?”

From a **Tech/programmer for a major games service provider**: “How reinforcement learning agents can be applied for testing across multiple games?”

From an **Academic researcher in game AI**: “Co-operative and / or partially observable games. See Ms. Pac-Man Vs Ghost Team as an example of taking a game and designing a PO version. Needs good AI for AI driven design which the competition is providing ready for the next step.”

From a **CEO of a small games company**: “Procedural generation of back story for NPCs in RPGs – where do they come from and why does this cause them to behave in a particular way – also talked about memory for in-game agents so that even if they have limited responses they make the best response – not just the same response every time.”

From a **CEO of a micro VR/games company**: “AI to test immersion in a VR game – immersion and realism are very different things. Using VR to visualise (and “play with”) complex datasets.”

From a **Programmer/researcher working in a (high tech non-games) SME and in a games lab**: “When games are used for serious purposes on human players, such as psychoanalysis, skill analysis, behaviour analysis, establishing baselines (such as the navigation abilities of the general population in Sea Hero Quest) or even treating medical/psychological conditions, then they need to be realistic to elicit realistic behaviour from the players under analysis. Hence, an obvious use case is employing AI to create realistic environments, realistic in-game characters, realistic metrics and realistic game play. Also, at the moment, such games can be quite prescribed so it would be good if they were more adaptive to the players – again for realism. However, this requires transparency of the algorithms – need to know what the algorithm is “doing” throughout (and preferably why it is doing it) to allow for thorough, systematic, transparent and effective analyses. I guess the question is how do you model and adapt the complexities of environments, characters and behaviour while maintaining transparency and authenticity, and achieving the desired outcome (serious games and analyses)?”

From a **CEO of a small games company** : “Superfast prototypes for game concepts, levels and terrains - to allow rich conversations with clients. Level design (my take – show it a bunch of levels and it makes new candidate levels).”

From a **CEO of a small games company**: “Level design – given some hand-crafted levels – come up with some similar editable candidates.”

From a **CEO of a small games company**: “Convincing organic shapes that can move on their own – without the need for humans to generate pieces – as was done in Spore and No Man’s Sky. Could focus primarily on locomotion or on form – or ideally on both. Believable worlds – using the back story and terrain to influence architecture, road layout, NPC occupants etc.”

4.4 Machina Ex Machina (Part 1)

Reynald Francois (Ubisoft - Düsseldorf, DE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Marc Erich Latoschik (Universität Würzburg, DE), Antonios Liapis (University of Malta - Msida, MT), Alexander Nareyek (Singapore, SG), Emily Short (Oxford, GB), and David Thue (Reykjavik University, IS)

License © Creative Commons BY 3.0 Unported license

© Reynald Francois, Matthew J. Guzdial, Marc Erich Latoschik, Antonios Liapis, Alexander Nareyek, Emily Short, and David Thue

The premises of the workshop were to lay the groundwork regarding whether AI systems could be used to profile players in order to create a model of their play-related needs. With this model, the director AI (dAI) could control the different aspects of a game in order to further satisfy the needs of the player and enhance the experience a given player has when playing the game.

The workshop was split in two topics:

Group 1 covered the definition of the player drives and the dAI as an “experience manager”. By defining the boundaries of the field of research, the research group aimed at identifying multiple elements:

- The **field of action** the dAI would have on the experience (e.g. player motivation, direct manipulation of the game environment...
- The **tools** dAI needs to perform its task (verbs accessible to the dAI (create, remove, modify, etc...), player modeling, gamer & player interaction history...)
- The **experience metrics** (player satisfaction, temporality, game state VS optimal state...)

The ultimate deliverable from this part of the workshop was to provide a proposal for a dAI-to-game-to-player interaction model that would serve as a base for Group 2, explained next.

4.5 Machina Ex Machina (Part 2)

Reynald Francois (Ubisoft - Düsseldorf, DE), Joshua Allen McCoy (University of California - Davis, US), Alexander Nareyek (Singapore, SG), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

License © Creative Commons BY 3.0 Unported license

© Reynald Francois, Joshua Allen McCoy, Alexander Nareyek, and G. Michael Youngblood

Group 2 continued where Group 1 left off (see above), and focused on the central tool to the dAI performance: **Player Modelling**, and its relevance or portability across multiple experiences.

The goal of the research was to demonstrate that properties inherited from the modelling of the player in one game can be carried over to have an impact on a chosen metric (in this case: player satisfaction) in another game.

The experimental protocol focused on creating a rudimentary player model containing in this case a single information: the risk-averse factor of a player, a property defining how much risks a player will take when playing games designed around high risk / high gains mechanics.

Two simple games based on the mechanic were designed (revolving around dice throwing); a human participant would take the role an hypothetical dAI would have and manipulate the game parameters to try to enhance the metric (satisfaction) and a protocol was drafted as following:

1. Have the player play game 1 and rate his experience
2. Create a player model mapping the risk-adverse factor of the player
3. Simulate the intervention of the dAI on game 1 by modifying the rules to be more risk-prone or adverse depending on the player profile
4. Have the player play game 1 again and rate his experience comparatively to the first playthrough
5. Have the player play game 2 and rate his experience
6. Simulate the intervention of the dAI on game 2 by modifying the rules to be more risk-prone or adverse depending on the player profile with the game 1 data
7. Have the player play game 2 again and rate his experience comparatively to the first playthrough
8. Assess whether the model created following game 1 data impacts favorably the selected metric (satisfaction) when the simulated dAI “enhances” game 2

4.6 What Is Machine Learning/Deep Learning?

Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Joshua Allen McCoy (University of California - Davis, US), Emily Short (Oxford, GB), Michael Treanor (American University - Washington, US), R. Michael Young (University of Utah - Salt Lake City, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

License © Creative Commons BY 3.0 Unported license

© Matthew J. Guzdial, Joshua Allen McCoy, Emily Short, Michael Treanor, R. Michael Young, and Jichen Zhu

Machine learning (ML) has advanced considerably in recent years, especially the subset of machine learning referred to as deep learning (DL) [2]. Public response to these advances has exceeded the advances themselves, impacting academic, economic, and journalistic fields. These factors put pressure on academic practitioners of non-ML artificial intelligence to apply ML to their research. However, there exists insufficient resources to train these academics, given that most educational materials target novices or experts. We identify the need for ML/DL educational resources aimed at this group: AI academics without existing ML/DL training. We focus on deriving introductions to ML/DL appropriate to this audience and identifying their common misconceptions and concerns.

To introduce ML we focus on the development of an accessible definition to separate what ML is and is not, and a classification of ML approaches. We settled on a definition of ML as function approximation. Given this definition, to apply machine learning to a problem one must have some desired input and output and must make the assumption

that there exists some function mapping these two classes. These input/output pairs could be states to actions (as in reinforcement learning), elements to categories (as in clustering problems), or images to labels (as in image recognition problems). We divided machine learning approaches according to this definition into three camps: discrete classification (decision trees, reinforcement learning, etc.), continuous classification (linear regression, neural networks), and categorization (clustering). Given this definition we next identify the borders of what machine learning can and cannot do. We identify three five major boundaries of modern machine learning: (1) local vs global coherency, (2) completeness of data representation, (3) amount and nature of training data, (4) existence and handling of edge cases, and (5) explainability.

Local vs global coherency identifies the domains and problems to which modern ML can achieve reasonable success. ML performs well in domains where local coherency is sufficient over global coherency. As an example consider the success of ML approaches at image recognition and generation compared to its relative failure in textual domains. An image of a cat with one pixel altered is still an image of a cat, while a paragraph with a single word altered can have a vastly different meaning. This is also reflected in the second of our identified boundaries: “completeness of data representation”, by which we indicate the extent to which the form of representation chosen for data encodes that data’s meaning. For example again consider the differences between images and text, an image of a cat more closely aligns with a human understanding of a cat, than do the letters “c”, “a”, “t”.

The next two identified boundaries are related: amount and nature of training data and the existence and handling of edge cases. By their nature, most modern ML approaches fit some distribution to the available training data. This means that the amount and nature of training data will impact the distribution learned, with extremely varied or small amounts of training data leading to worse performance.

Explainability in this context means the ability for an AI or ML system to explain its decisions or understanding to a non-expert human user, typically in natural language. Modern ML, particularly deep learning, has been described as a “black box” system, with decisions being difficult to explain outside of large sets of numbers. While there has been some work in using machine learned-features as a framework for explanation [3], in general the extent to which a domain calls for explanation will impact the success of applying ML.

We note that beyond a concrete definition and boundaries, many expert AI researchers without ML experience share concerns informed by this lack of experience. We identify three major concerns: (1) whether and how AI and ML can be integrated, (2) whether ML can be applied to cognitively-inspired CS research, and (3) whether ML can be applied to game systems.

In terms of the integration of AI and ML systems, we first note that most deployed ML systems make use of traditional AI approaches to fill in the gaps of modern ML. For example, using rules-based systems to translate a predicted distribution into discrete actions. We derive a three-part framework for describing integration of machine learning with traditional or classic artificial intelligence: AI before ML, AI during ML, or AI after ML. In AI before ML systems have input that is first parsed by a hand-authored AI system, such as tagging chat input with a rules-based system before constructing a reply with a ML system. In AI during ML, the AI and ML systems take turns or act concurrently at some task, perhaps a storytelling system with an AI planner handling global planning while an ML system handles sentence-to-sentence transitions. Lastly in AI after ML, AI is used to transform or clarify the output of an ML system, as in the earlier example of translating a distribution over potential actions to a single action.

Traditionally, AI has a strong link to cognitive science. In ML, including deep learning, these algorithms could be used to expose biases in datasets, an area for potential further research. We further identify a framework adapted from neural network research in cognitive science in which deep learning could represent a subconscious or fast processing cognitive process. Guzdial et al. [1] composed an abstract art system in which a convolutional neural net (a type of deep neural network) identifies emotional meaning from images, trained based on human tags of emotion names on flickr image uploads. A traditional AI search process utilizes this ML system as a heuristic or reward in trying to achieve certain visual emotional depictions (using the AI during ML framework as above). This abstract system represents a hypothesis of human abstract artist cognition, which could be experimentally interrogated.

In addition, we sketch out a few abstract frameworks to apply ML to game systems. For example, placing the player in a training role of a reinforcement learner, giving the player indirect control as that agent made its way through levels. Alternatively one could imagine placing the player in the role of an ML expert, tuning parameters to achieve certain model outputs for simplified puzzles. Lastly, we consider a game in which players alter training data for classifiers to seek certain performance.

We conclude by collecting some guidelines for best practice when applying machine learning to a problem. We include this as some AI experts consider ML tuning “black magic”. We identify AB testing as a key practice, varying only a single parameter and checking performance. Further we identify a process for applying ML, and especially deep learning, to problems. First, cut down your data representation to only the information you personally would need to map input to output. Second, adapt existing architectures or start with the bare minimum (earn your complexity). Lastly, when tuning the architecture, create hypotheses for what could be going wrong, and vary appropriate parameters to test these hypotheses.

References

- 1 Matthew Guzdial, Duri Long, Christopher Cassion, and Abhishek Das. Visual procedural content generation with an artificial abstract artist. In *Proceedings of ICCG Computational Creativity and Games Workshop*, 2017.
- 2 Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- 3 Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.

4.7 AI-assisted Board Game Play

Antonios Liapis (University of Malta - Msida, MT), Michael Cook (University of London, GB), Steve Dahlskog (Malmö University, SE), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), and Tommy Thompson (Anglia Ruskin University - Cambridge, GB)

License © Creative Commons BY 3.0 Unported license

© Antonios Liapis, Michael Cook, Steve Dahlskog, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Gillian Smith, Anne Sullivan, and Tommy Thompson

Over the last decade, an increasing amount of tabletop board games use digital media to replace or enhance elements of gameplay. As an example, the first edition of *Mansions of Madness* [6] required one of the players to take the role of a Keeper as an opponent to the remaining players, handling monster movement and revealing previously hidden information. The second edition of *Mansions of Madness* [8] “is now guided by a companion app that removes the need for a Keeper player, makes for a fully cooperative game, allows for single player gaming, and makes setup quick and easy”¹. A plethora of other board games use primarily mobile phone applications for time management as in *One Night Ultimate Werewolf* [1], initializing and revealing hidden information when appropriate as in *Alchemists* [4], a combination of both as in *X-COM* [7] and *Space Alert* [3], and enhancing the atmosphere of the game e.g. via the sound effects in *Stop Thief!* [10].

Since board games are seeing a resurgence of popularity and a desire to innovate thematically, mechanically and technologically [5], it is the perfect time to analyze how digital game technologies and artificial intelligence can be applied to enhance analog play. Technology, especially if enhanced with AI, has many possible uses for board games in the near future. On the one end, simple enhancements such as saving or transmitting a game state can allow for games to be played in multiple sessions, or over the Internet. Assistive technology can be used to simulate the board game in a forward model digitally, allowing players to see the results of their actions one or more steps ahead: feedback can be provided to players in abstract form (e.g. whether this move is beneficial or not) or showing the full digitized board in a future state. In such a forward model, opposing players’ moves can also be simulated using methods as simple as minimax [11] and as complex as a computational model of each player based on their previous decision-making strategies [9]. Beyond assistive technology for players, games in which a player acts as the game master can benefit from similar technology both for managing non-player characters (e.g. automating or semi-automating decision-making of individual NPCs digitally so that the game master confirms and moves the figures on the board) but also for providing inspiration and assistance during game preparation. On the more ambitious end of the spectrum, board game play mediated by technology can allow for customization of the game’s rules: this can be done based on human choice (e.g. checking which optional rules to use from a list), or automatically based on past game data in this player group or a global player community. Automated rule adaptation can be done when the game starts (ensuring that all players understand and learn the rules of play which remain consistent throughout the session) or in real-time as the game progresses (considering play data in the current game for dynamic difficulty adjustment [12]).

¹ Official Frequently Asked Questions: <https://www.fantasyflightgames.com/en/products/mansions-of-madness-second-edition/>

In light of the plethora of options for integrating technology and AI into board game play, a set of 33 usage patterns were identified. The usage patterns largely focused around increasing “time to fun” (e.g. by simulating the effects of a player action as a tutorial), enhancing the atmosphere (e.g. by adding sound effects or by offering additional rendering options through projectors), increasing depth of play (e.g. by managing hidden information or logistics beyond what a human game master can do), or changing the ruleset (e.g. by creating new rules or adapting existing rules). Using the format for describing design patterns followed by Bjork and Holopainen [2], four example usage patterns were chosen and further detailed: (a) tech for creativity support, (b) tech for atmosphere, (c) tech provides player flavoring, (d) tech adapts/creates rules. With this initial set of usage patterns in place, all 33 usage patterns are planned to be mapped in future research on the topic.

References

- 1 Bézier Games. One Night Ultimate Werewolf, 2014.
- 2 Staffan Bjork and Jussi Holopainen. *Patterns in Game Design*. Charles River Media, 2004.
- 3 Czech Games. Space Alert, 2008.
- 4 Czech Games. Alchemists, 2014.
- 5 Owen Duffy. Board games’ golden age: sociable, brilliant and driven by the internet. <https://www.theguardian.com/technology/2014/nov/25/board-games-internet-playstation-xbox>, 2014. The Guardian. Online. Accessed 9 January 2018.
- 6 Fantasy Flight Games. Mansions of Madness, 2011.
- 7 Fantasy Flight Games. X-COM, 2015.
- 8 Fantasy Flight Games. Mansions of Madness, 2nd ed., 2016.
- 9 Christoffer Holmgård, Antonios Liapis, Julian Togelius, and Georgios N. Yannakakis. Evolving models of player decision making: Personas versus clones. *Entertainment Computing*, 16:95–104, 2016.
- 10 Restoration Games. Stop Thief!, 2017.
- 11 Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd ed.)*. Upper Saddle River, 2003.
- 12 P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma. Difficulty scaling of game ai. In *Proceedings of the International Conference on Intelligent Games and Simulation*, 2004.

4.8 Mixed-media Game AI

Antonios Liapis (University of Malta - Msida, MT), Elisabeth André (Universität Augsburg, DE), Sander C.J. Bakkes (Tilburg University, NL), Rafael Bidarra (TU Delft, NL), Steve Dahlskog (Malmö University, SE), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Ana Paiva (INESC-ID - Porto Salvo, PT), Mike Preuß (Universität Münster, DE), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), Tommy Thompson (Anglia Ruskin University - Cambridge, GB), David Thue (Reykjavik University, IS), Georgios N. Yannakakis (University of Malta - Msida, MT), and R. Michael Young (University of Utah - Salt Lake City, US)

License © Creative Commons BY 3.0 Unported license

© Antonios Liapis, Elisabeth André, Sander C.J. Bakkes, Rafael Bidarra, Steve Dahlskog, Mirjam P. Eladhari, Ana Paiva, Mike Preuß, Gillian Smith, Anne Sullivan, Tommy Thompson, David Thue, Georgios N. Yannakakis, and R. Michael Young

Over the last decades, digital technologies have moved away from the personal computer (PC) into cloud computing, ubiquitous computing, intelligent robots and smart devices.

From wearable technologies to remote-controlled household items and from sensors for crowd control to personal drones, a growing range of technologies provides an opportunity for games and artificial intelligence. While artificial intelligence (AI) is already a big part of the Internet of Things, raising concerns in terms of ethics and politics [14], research in game AI has been relatively confined to more traditional computer and video games. Relevant work on wearable technologies as game controllers [13], mixed- or virtual-reality rendering [8], technology-enhanced play in playgrounds [5] or board games [6], or social robots for games [3], has not taken full advantage of AI for controlling or mediating the experience. Using the term *mixed-media* to refer broadly to any media, digital or otherwise, aside from the game loop within a PC or a game-specific database, this working group attempted to map out the broad topic of mixed-media in terms of its applications for game AI.

Potentially, AI could assist games in taking advantage of mixed-media in two principal ways: use input from non-game sources within the game, and move elements of play or gameplay outcomes into non-PC outputs. The usefulness of non-game information as *input* to gameplay is primarily in the identification of *context*. Context can be personal (e.g. information relevant to this particular player) or broader and anonymous (e.g. number of people in the player's vicinity or trending topics on Twitter). Context can come from the actual environment of a player via sensors (e.g. temperature sensors on a player's mobile phone, or surveillance logs in public spaces such as museums), from a variant game controller (e.g. speech detection or face detection as an explicit part of gameplay), from social relationships (e.g. based on social media profiles, real life proximity, or a history of social interaction), from game histories (e.g. past gameplay habits in other games), from cultural histories (e.g. players' demographic data linking to cultural heritage databases), or from temporal context (e.g. the current time of day or date, or the time passed since the user interacted with the game).

On the other hand, the *output* of such applications (playful or not) can also be delivered in non-traditional media beyond digital screens of a PC or mobile device. Closer to traditional digital outputs, mixed- or virtual-reality devices could be considered, along with environmental projections (such as large screens or projections on different parts of a building, in the case of multiplayer games played in a shared environment). On a similar (and familiar) direction, output or intermediate states of a long-lasting game can be shared on social media, potentially soliciting other users' feedback as additional input (to provide even more context, as discussed above). A game's final output, such as a drawing in a collaborative drawing game [11] or an AI-assisted drawing tool [4], could be printed out in paper, fabric [1], or via 3D printing [9, 10], or sonified into a musical piece [12]. More ambitiously, the intermediate states of such a game could be used to control robots, wearable actuators (worn by players), or smart home devices. Obviously, the further the output is from traditional game output, the more challenging the design problem of making such an application a seamless, playful experience while ensuring that the game output is understood as such and –when moving into the real world– remains safe to use and respects players' privacy.

Following an initial mapping of the possibility space for mixed-media game AI, including possible audiences, purposes and challenges, the working group focused on three specific use cases. The first use case focused on textile input to physical output for MEND, a platform where people contribute to a communal art piece (projected on a wall) by scanning embroidered physical objects, one person at a time. The second use case focused on physical input to virtual output, for a game designed around the use of a sensor able to detect laughter in groups [7] as a mediator for when a game session is won and by whom. The final use case problematized the topic of input more broadly, trying to identify context within player's

game data; the main question revolved around whether there are signals or log data which can be assumed to be independent of the game context [2] but able to capture the player's context in terms of experience. The broad nature of mixed-media game AI was thus mapped out, and a small part of the design and problem space it can offer was explored.

References

- 1 Lea Albaugh, April Grow, Chenxi Liu, James McCann, Jen Mankoff, and Gillian Smith. Threadsteading: Playful interaction for textile fabrication devices. In *Installation in Interactivity at ACM Conference on Computer-Human Interaction*, 2016.
- 2 Elizabeth Camilleri, Georgios N. Yannakakis, and Antonios Liapis. Towards general models of player affect. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2017.
- 3 Filipa Correia, Patrícia Alves-Oliveira, Nuno Maia, Tiago Ribeiro, Sofia Petisca, Francisco S Melo, and Ana Paiva. Just follow the suit! trust in human-robot interactions during card game playing. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 507–512. IEEE, 2016.
- 4 Nicholas Davis, Yanna Popova, Ivan Sysoev, Chih-Pin Hsiao, Dingtian Zhang, and Brian Magerko. Building artistic computer colleagues with an enactive model of creativity. In *Proceedings of International Conference on Computational Creativity*, 2014.
- 5 Die Gute Fabrik. Johann Sebastian Joust, 2014.
- 6 Fantasy Flight Games. Mansions of Madness, 2nd ed., 2016.
- 7 Simon Flutura, Johannes Wagner, Florian Lingenfelder, Andreas Seiderer, and Elisabeth André. Laughter detection in the wild: demonstrating a tool for mobile social signal processing and visualization. In *Proceedings of the ACM International Conference on Multimodal Interaction*, 2016.
- 8 Alex Hern. Will 2017 be the year virtual reality gets real? <https://www.theguardian.com/technology/2016/dec/27/will-2017-be-the-year-virtual-reality-gets-real>. The Guardian. Online. Last accessed 9 Jan 2018.
- 9 Britton Horn, Gillian Smith, Rania Masri, and Janos Stone. Visual information vases: Towards a framework for transmedia creative inspiration. In *Proceedings of the International Conference on Computational Creativity*, 2015.
- 10 Joel Lehman, Sebastian Risi, and Jeff Clune. Creative generation of 3D objects with deep learning and innovation engines. In *Proceedings of the International Conference on Computational Creativity*, 2016.
- 11 Antonios Liapis, Amy K. Hoover, Georgios N. Yannakakis, Constantine Alexopoulos, and Evangelia V. Dimaraki. Motivating visual interpretations in iconoscope: Designing a game for fostering creativity. In *Proceedings of the Foundations of Digital Games Conference*, 2015.
- 12 Marco Scirea, Gabriella A. B. Barros, Noor Shaker, and Julian Togelius. SMUG: Scientific Music Generator. In *Proceedings of International Conference on Computational Creativity*, 2015.
- 13 Joshua Tanenbaum, Karen Tanenbaum, Katherine Isbister, Kaho Abe, Anne Sullivan, and Luigi Anzivino. Costumes and wearables as game controllers. In *Proceedings of the International Conference on Tangible, Embedded, and Embodied Interaction*, 2015.
- 14 The Future of Life Institute. An open letter to the united nations convention on certain conventional weapons. <https://futureoflife.org/autonomous-weapons-open-letter-2017/>, 2017. Online. Accessed 9 January 2018.

4.9 Emergence

Jacob Schrum (Southwestern University - Georgetown, US), Cameron Browne (RIKEN - Tokyo, JP), Mark J. Nelson (Falmouth University, GB), Ana Paiva (INESC-ID - Porto Salvo, PT), Julian Togelius (New York University, US), and Georgios N. Yannakakis (University of Malta - Msida, MT)

License © Creative Commons BY 3.0 Unported license

© Jacob Schrum, Cameron Browne, Mark J. Nelson, Ana Paiva, Julian Togelius, and Georgios N. Yannakakis

Well designed games often produce emergent behaviour that makes them more than the sum of their parts. The aim of this workshop was to explore this phenomenon of emergence in game design. While it is not strictly possible to design for emergence – it is by definition the occurrence of the unexpected – the contribution of this workshop was to explore what emergence means in this context and ways it might be encouraged in future designs.

Emergence occurs within games in various ways. Higher-order constructs emerge from the interaction of lower-level entities, and such emergence is often a desirable outcome in games, because it allows for new player experiences, and can spare programmers from meticulously micromanaging every aspect of a game’s design. In this session, various forms of emergence in video games were discussed, including the emergence of game dynamics, complex strategies, player goals, and social dynamics.

Game dynamics can emerge from low-level rules, such as how physics are simulated. If a game engine allows certain types of in-world objects to interact physically, then player actions can set off chain reactions that are ultimately the result of physical rules. Interesting examples include the gravity gun in *Half-Life 2*, and the physical simulation in *World of Goo*. Complex strategies can emerge from the low-level rules dictating how the game is won or how points and resources are acquired. For instance, in the abstract board game *Omega*, a player’s score is the product of the sizes of each cluster of pieces in their color. This low-level definition of player score happens to lead to an emergent strategy, in which maximizing the number of groups of size three that one possesses will in turn maximize game score. A more complex example is the deck-building game *Hearthstone*, in which various rules about how different cards interact allow for a wide range of emergent player strategies.

Player goals can emerge when a game gives players the freedom to explore a world and perform a wide-variety of interesting actions. The primary example of such goal emergence that was discussed was *Minecraft*, because certain modes of play do not have any preset goal. Rather, it is up to players to simply survive, and do as they please. Players typically choose to build elaborate structures, but what they build and how is completely up to them. Player goals can also emerge in Massively Multiplayer Online Role-Playing Games (MMORPGs), in which individuals and groups decide where to go in the world and what goals they want to achieve.

Such games also allow for the emergence of social dynamics, due to the fact that human players are allowed to interact. The interactions that a game allows between players, be they combat interactions or economic transactions, can lead to the emergence of interesting social dynamics within the community. However, even in a game where someone primarily plays alone, social dynamics can emerge from the ability to share the fruits of one’s labors, such as the ability to share one’s creations in *Minecraft*, or the fact that anyone can share their experiences in any game via the video game streaming service Twitch.

Despite identifying these various forms of emergence, participants struggled to find a firm, and general definition that would allow for concrete measurement of the degree of emergence

within a game. However, many proposals were considered. One prevalent notion was that a game with elegant rules could be seen to support emergence if it gave rise to complex and/or unexpected player strategies. Such emergence is certainly desired in certain types of games. However, examples of extremely complex games that allow emergent strategy also exist, such as the aforementioned *Hearthstone*, and *Pokemon*. These games are laden with many intricate details, and the details of how cards interact in *Hearthstone* and how different traits and abilities interact in *Pokemon* actually lend even more emergent complexity to these games.

Participants also realized that care should be taken when incorporating the concept of “unexpectedness” into the definition of emergence. For example, if a game’s design allows for interesting behavior that is unexpected from the point of view of the players, then such an outcome might be an interesting example of emergence. However, the unexpected quickly becomes expected when it can be replicated, thus making such a definition highly subjective. Furthermore, a highly stochastic game could be argued to be full of unexpected outcomes, but such outcomes would likely not be considered interesting, and might even be frustrating to some. If we are considering emergence as a desirable property, then it would not make sense to include such outcomes under the umbrella of emergence.

However, examples of emergence that resulted in negative player experiences were also discussed. For example, the ability to kill other players in *Diablo* leads to the emergence of Player vs. Player (PVP) style gameplay, but such gameplay could be very undesirable to certain players. In order to foster an enjoyable player experience, many MMORPGs only allow PVP behavior in designated areas or on designated servers, to prevent this emergent behavior from harming the player experience. Unusual glitches, bugs, and exploits can also be considered to emerge from unexpected game rules and design choices, but game designers typically want to avoid this type of emergence.

A useful and actionable idea that was considered for measuring emergence was to tie it to the concept of strategic depth. Specifically, if a game could be designed in such a way that additional mental or computational effort led to a variety of different ways to succeed in the game, then that game could be argued to have emergent strategies. However, computational effort is strongly tied to the feature representation that is used to play a game, so it is unclear how general this idea would be in practice.

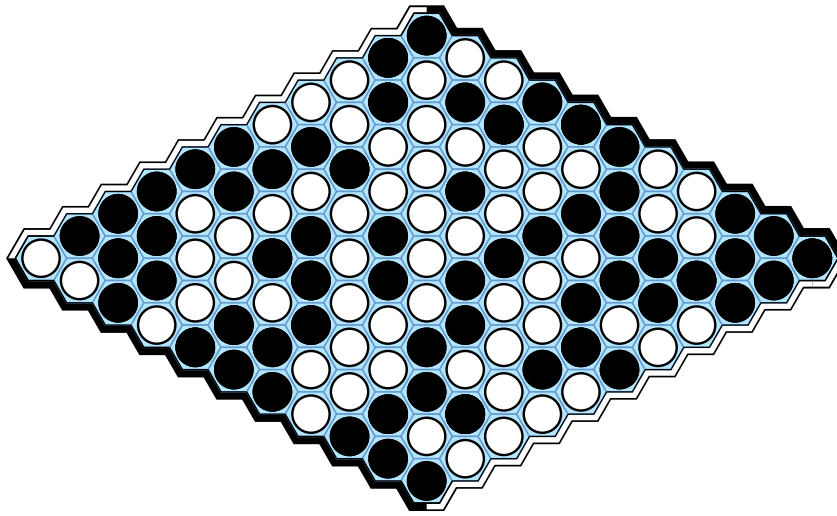
Note that computational effort and mental effort by the (human) player can be quite different things in practice. For example, consider the following hypothetical case, based on the game of Hex.

Figure 1 shows a game of Hex, in which Black has won by connecting the black sides of the board with a connected group of black pieces. Now consider two versions of this game:

1. Hex: Players win by connecting their sides of the board with a connected group of their pieces.
2. HexP: Players win by connecting their sides of the board with an even-sized connected group of their pieces, but lose by connecting their sides of the board with an odd-sized connected group of their pieces.

On a superficial reading of the rules, the games do not seem that different. From the computational resources viewpoint there is also little difference; the odd/even group size calculation will be of similar complexity to the connectivity test, and depending on implementation could be handled by code that is nearly the same.

However, from the human player’s perspective, there is a world of difference between these two games. The human brain is adept at spotting connections, and it is clear at a glance that Black has established a cross-board connection to win the game of Hex shown in



■ **Figure 1** A game of Hex won by Black.

Figure 1. However, in order to determine the winner of the corresponding game of HexP for this same board state, players must count the size of the connecting group (33 pieces) to realise that Black has lost this game by connecting the black sides with an odd-sized group of black pieces. Hex is a brilliant game with clear rules and objectives that allow players to engage in deep strategic planning, whereas HexP would be a dreadful game with a confusing objective in which players must spend most of their mental effort simply counting and recounting group sizes, as groups of pieces grow and coalesce throughout the game. The additional group parity rule might introduce some new strategies to the game, but these are far outweighed by the added complexity that comes with them.

This example highlights that added complexity does not necessarily encourage the emergence of anything interesting. In fact, we could even go further to distinguish between “good emergence”, i.e. the emergence of surprising, interesting and beneficial phenomena, and “bad emergence”, i.e. the emergence of confusing, annoying and detrimental behaviour. It is not typically possible to predict which form of emergence a given set of conditions will produce. However, we can suggest a simple rule of thumb that might help identify potential discrepancies between computational cost and mental effort: Any rule that involves arithmetic of any sort will typically make a game harder for human players.

Another concrete idea proposed in the workshop for use in competitive games was to measure the size of a Pareto archive of player strategies. Many games are viewed to be interesting if there are multiple ways to win, or if the path to victory strongly depends on the play style of one’s opponents. Thus, one could collect play strategies that are the only successful way to defeat other play strategies, even if those strategies were themselves susceptible to yet other strategies. A large archive of such strategies would, in some cases, indicate a game rich in strategic options. However, one could also imagine a degenerate example of a scaled up game of rock-paper-scissors (with more than three options), in which the Pareto archive would be very large, but few interesting player strategies would be available. Once again, it is difficult to distinguish between good emergence and bad emergence with a raw measurement of this sort.

Furthermore, for the proposed ways of measuring emergence, even in the best cases they would measure the strategic complexity of a game more than the actual emergence of that

complexity. Short of looking at the code, it is not clear how one would distinguish between a game full of meticulously programmed, individualized responses to various situations, and one in which the same apparent complexity was present, but was the result of comparatively simple programming. However, finding more meaningful ways to measure complexity first could open the door to a more meaningful study of emergence in the future.

4.10 Human-assisted Creation of Content Within Games

Jacob Schrum (Southwestern University - Georgetown, US), Pier Luca Lanzi (Polytechnic University of Milan, IT), Alexander Nareyek (Singapore, SG), and Pieter Spronck (Tilburg University, NL)

License © Creative Commons BY 3.0 Unported license
© Jacob Schrum, Pier Luca Lanzi, Alexander Nareyek, and Pieter Spronck

Artificial Intelligence (AI) can be used in various ways to create content for video games. A commercial example is the racing game Forza Horizon 2, in which neural networks learn to emulate specific players to serve as opponents for humans. Procedural content generation is also currently used in many modern games, to create levels, among other things. Some prominent examples from academia are based on neuroevolution, such as the evolved weapon firing patterns in Galactic Arms Race, and the selectively bred flowers in the social game Petalz.

Outside of video games, many AI techniques have been developed for creating interesting artistic content. Some techniques that were discussed included basic search and optimization, case-based reasoning, generative/creative adversarial networks, and evolution of various representations, including Compositional Pattern Producing Networks (CPPNs). CPPNs have proven to be particularly flexible, in that they can produce 2D images, 3D shapes, animations, audio tones, soft body robots, and the flowers in the aforementioned Petalz game.

Because of the flexibility of CPPNs, a specific content-creation game was envisioned that could use evolved CPPNs in numerous ways: The Infinite Art Museum. This game concept would allow a player to explore a dynamically generated world filled with artistic artifacts, in which the locations the player chooses to explore would act as an implicit selection function influencing the creation of new art within the world. The geometry of the world would not need to adhere to real-world physical rules, which would allow a high degree of branching of hallways for players to explore. The textures on walls could be generated by CPPNs, allowing players to explore an interesting variety of abstract art. This basic concept could be further elaborated to allow the world to be populated by 3D objects generated by CPPNs, and even CPPN generated sounds (CPPNs can generate basic tones that can be used as “instruments” to play MIDI files) and animations (by querying CPPNs across time). It may even be interesting to allow for some degree of cross pollination between art mediums, so that users can hear CPPN generated sounds based on their 2D texture preferences, or see animations based on the 3D shapes they like the most. An element of action and conflict could even be introduced into the game by using CPPN generated soft robots as enemies.

If the basic design of the Infinite Art Museum described so far could be implemented, it would provide an interesting proof-of-concept demonstrating the appeal of such features in a video game.

4.11 Play

Jacob Schrum (Southwestern University - Georgetown, US), Cameron Browne (RIKEN - Tokyo, JP), Amy K. Hoover (NJIT - Newark, US), Sebastian Risi (IT University of Copenhagen, DK), David Thue (Reykjavik University, IS), Julian Togelius (New York University, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

License © Creative Commons BY 3.0 Unported license

© Jacob Schrum, Cameron Browne, Amy K. Hoover, Sebastian Risi, David Thue, Julian Togelius, and Jichen Zhu

Humans and non-human animals play, and human enjoyment of games seems to be tied to our desire to play in various forms. Playful activities have some evolutionary benefits, such as aiding the development of motor and planning skills. Play also provides organisms opportunities to develop social skills such as the ability to cooperate and negotiate. Because of the benefits that play imparts to real-world organisms, it is natural to ask whether artificial agents in games can benefit from, or even exhibit, playful behavior.

Programming an artificial agent to exhibit playfulness first requires understanding playfulness. To an outside observer, an agent whose behavior and/or goals seem to shift randomly can appear to be playful, and some amount of randomness may indeed be an intrinsic aspect of playfulness. However, playful behavior can also be more focused, seemingly driven by curiosity. Therefore, searching for playful behavior can be facilitated by tools such as Novelty Search, which specifically optimizes for novel outcomes. Agents that imitate other agents are also viewed as playful in some contexts, though such imitation may be purposely imperfect or exaggerated, indicating that the imitating agent is in a sense transforming an observed sequence of actions into a slightly different sequence. Having thought about different forms of playfulness, several prototype programs were devised to make use of such notions.

The first prototype was a variant of Mario including an additional playful Mario agent, referred to here as Luigi. This game allows a human player to control Mario as usual, while an additional AI controller dictates the action of Luigi. In the prototype, Luigi is a variant of an A* planning agent that sets random goals, which results in it running back and forth on the screen in what can be viewed as a very simple form of playfulness. However, the introduction of such an agent opens up the possibility of several enhancements that could be pursued in the future. Specifically, Luigi could play tag with the human-controlled Mario, so that the goal of the game is to chase or escape the other agent rather than to beat the level. Another possible enhancement would be to attempt a competitive speed run with the artificial agent. The reason that this variant would encourage a playful mood within the game is that the two agents can interact, so that they are not merely performing the same actions, but frequently knocking each other out of the way and trying to control their actions despite the presence of competition. A final variant considered is one in which Luigi blithely pursues random goals, while it is the player's duty to protect Luigi from the hazards in the environment.

The second prototype presents a playful world to a human player. This prototype extends previous work in a 2D top-down car-racing simulator in which arbitrary game rules can be introduced. The human player starts by navigating with a car in a plain environment, but a disembodied agent begins introducing rules to alternately aid and thwart the player. Furthermore, another car-driving agent can interact with the human player within the world. This agent alternates between mimicking the human player and attempting to achieve its own goals, which are based on the shifting rules within the world.

The final prototype that was developed recognizes the seeming playfulness of inanimate

objects, which is in-turn the product of the human tendency to ascribe agency to such objects. Simple interactions based purely on physical rules can result in the illusion of playfulness. Specifically, the movement of balloons in the wind might appear playful to an imaginative observer, and thus this prototype attempts to create an environment in which simple physical interactions of balloon-like objects appear playful to a human player. The human controls a pin point and tries to pop the balloons, but the opponent AI agent controls the wind, and can either help or hinder the player.

Though simple, these games show promise, and indicate that the concept of playfulness can be used to generate fun games. These various prototypes could be expanded upon to more thoroughly investigate the concept of play, and its usefulness in video games, in the future.

4.12 AI As Reflective Practice

Gillian Smith (Worcester Polytechnic Institute, US), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), Adam M. Smith (University of California - Santa Cruz, US), Anne Sullivan (University of Central Florida - Orlando, US), Tommy Thompson (Anglia Ruskin University - Cambridge, GB), and R. Michael Young (University of Utah - Salt Lake City, US)

License © Creative Commons BY 3.0 Unported license
© Gillian Smith, Mirjam P. Eladhari, Matthew J. Guzdial, Emily Short, Adam M. Smith, Anne Sullivan, Tommy Thompson, and R. Michael Young

This group formed out of a common interest in exploring why the act of creating AI systems is personally satisfying. In doing so, we discussed meaning, subjectivity, and bias in AI systems, and the implementation and adoption of AI systems for artistic expression. A common theme that emerged through the conversation was that of treating AI creation as a reflective practice.

Why Do We Do AI? Though scientific curiosity is a uniting factor in why each member of our group performs research in artificial intelligence, we share several other motivations of a more personal and subjective nature. Several group members commented that building AI systems involving virtual characters is empowering: it provides a means to experiment with different interpretations of human behavior, and to learn from the gaps between simulation and reality. Others found building AI systems to be satisfying because it is creative and expressive as an artistic medium [4], and because of the excitement for AI's potential to challenge pre-existing notions of what games can be [2, 9].

AI and Meaning. There was a commonly shared sense that creating AI systems is a deeply personal act, though as scientists we are taught and often strive to minimize subjectivity. AI systems are formalizations of theories pertaining to the domain the system operates on; while they may be based on theories from academic literature in relevant disciplines, they may also be used to encode and continually iterate upon personal perspectives and theories. A major theme that emerged in our discussion on meaning, subjectivity, and bias in AI systems was that of grappling with reductionism. In order to make theories concrete enough to be operationalized algorithmically, they must be stripped of nuance and interpretation. Such reductionism runs counter to humanistic goals; ironically, even in AI systems designed to simulate virtual humans, the algorithms can be dehumanizing in their

reductionist formulation of social interactions. For example, The Sims [5] presents humans as manipulable entities with needs that are constantly eroding and must be satisfied by player intervention, reducing people and relationships to a small number of meters that must be managed.

A second theme in our discussion was that of treating AI as a means for reflection on the process of design. One group member described the act of creating a computational generative system as that of “explaining myself to an alien intelligence”. The act of specifying every detail of a desirable space of artifacts to be generated to an entity that does not “understand” can be a reflective one, as each discovery of a forgotten rule prompts its specification.

AI for Artistic Expression. Several members of the group are practicing artists as well as artificial intelligence researchers. Thus, we spent some time discussing the relationship between art and artificial intelligence, and how we use and research AI in our own creative practices. A common theme that emerged was that of embracing error: in the arts, mistakes are a natural part of the creative process, and are often not problems to be fixed but rather treated as serendipitous. Currently, “mistakes” in AI systems are often seen as bugs to be squashed. This group discussed what it would mean for AI systems to be able to make mistakes and grow from them.

Fostering Reflection through Constructing AI Systems. We have argued thus far that there is value in treating the creation of AI systems as a personal and reflective process. A question then emerges: how, if at all, can we make that process accessible to those who do not have a background in artificial intelligence or even in computer science? In the context of AI-driven game design, we discussed the potential for mixed initiative design tools (e.g. [3, 8]) that support users in deeply reflecting on what they design. Inspired by Schon’s framing of design as conversation (and, indeed, as reflective practice) [6], we discussed novel methods for engaging in that conversation, both in terms of the interface used to make design decisions, and the way in which feedback is provided to the designer. With regard to interaction design, we discussed ways in which tactile and tangible interfaces could permit a more organic and playful design exploration: pushing and pulling a system’s expressive range [7] to include (or exclude) certain kinds of designed artifacts. Providing deeper feedback to designers was centered around providing rapid feedback to users in terms of impact on expressive range, inspired by Cook’s Danesh [1], and we discussed several new potential metrics and visualizations to support reflective design.

References

- 1 Michael Cook, Jeremy Gow, and Simon Colton. Danesh: Helping bridge the gap between procedural generators and their output. In *Proceedings of the 2016 Workshop on Procedural Content Generation in Games*, co-located with FDG/DiGRA 2016.
- 2 Mirjam P. Eladhari, Anne Sullivan, Gillian Smith, and Josh McCoy. *AI-Based Game Design: Enabling New Playable Experiences*, UC Santa Cruz Baskin School of Engineering, Santa Cruz, CA, 2011
- 3 Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*, 2013
- 4 Michael Mateas. Expressive Intelligence: Artificial Intelligence, Games and New Media. In *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*. Springer Berlin Heidelberg, 2–2: 2007
- 5 Maxis. *The Sims*. PC game, 2000

- 6 D. A. Schon. *The reflective practitioner: How professionals think in action*. Basic Books, New York, NY, 1983
- 7 Gillian Smith and Jim Whitehead. Analyzing the Expressive Range of a Level Generator. In *Proceedings of the Workshop on Procedural Content Generation in Games*, co-located with FDG 2010
- 8 Gillian Smith, Jim Whitehead, and Michael Mateas. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. In *IEEE Transactions on Computational Intelligence in AI and Games* 3(3): 201–215, 2011
- 9 Mike Treanor, Alexander Zook, Mirjam P. Eladhari, Julian Togelius, Gillian Smith, Michael Cook, Tommy Thompson, Brian Magerko, John Levine, and Adam Smith. AI-based game design patterns. In *Proceedings of the 2015 Conference on the Foundations of Digital Games*, 2015

4.13 The Happiness Game

Pieter Spronck (Tilburg University, NL), Alex J. Champandard (AiGameDev.com KG - Wien, AT), Peter I. Cowling (University of York, GB), and Pier Luca Lanzi (Polytechnic University of Milan, IT)

License © Creative Commons BY 3.0 Unported license

© Pieter Spronck, Alex J. Champandard, Peter I. Cowling, and Pier Luca Lanzi

Computer games are used to entertain people, train and educate them, support social interaction, and sometimes to learn about industrial and societal processes. While these are lofty goals, in an idealistic vision one can imagine that computer games are used in a much more world-changing manner. The question that we asked is “Can computer games be used to improve the overall happiness of the human race?”

Our answer to this question is the idea that a game can be developed that contains knowledge on what makes humans happy, and allows for experimentation with different societal constructs and different political systems to see what their effect is on human happiness. As such, the game would be a realistic model of happiness, and provide convincing arguments for particular social changes. We discussed and implemented an initial version of the so-called “Happiness Game.”

The purpose of the Happiness Game is to emulate a world of agents which represent humans. The agents can interact in various ways. The agents can also be happy and unhappy to certain degrees. The main interaction between the agents is “trading,” as that is the basis for human interaction. However, agents may also exhibit behaviors that are not materially beneficial, but may improve their “lives” by increasing their own happiness, or the happiness of other agents. Agents in the system may simply emulate certain natural behaviors, but may also be designed to try to exploit the system, or to learn to interact with the systems in order to maximize their effectiveness in acquiring material wealth or happiness. We implemented various agent strategies for proposing and accepting/rejecting trades.

The initial implementation of the game functioned correctly, but also quickly demonstrated that the concepts of “happiness” and “survival” are not easy to pin down. For instance, while the system would make agents which were too unhappy die off (which seems realistic), it also allowed agents to survive just by “being happy” (which is unrealistic). The reason is that in the initial system both food and happiness were both used to track an agent’s survival state, but that either was sufficient for an agent to survive. Thus, our conclusion

was that the concept of happiness and how it would propagate in the system needs a careful design, before a second generation of the system is built.

In a second generation, we look forward to experimenting with political systems, which may be imposed upon the system to affect the existence of the agents. For instance, a socialist system may take much of the agents' wealth to redistribute it over the population, while a communist system forces the agents into particular behaviors and a capitalist system leaves the agents free to behave in whichever way they want, including exploiting other agents. The system can then be used to study the effect of political systems on the quality of life of the agents, but also experiment with new, original political systems which may lead to an overall higher quality of life for all agents. This way, the effects of social change may be envisioned before the change is actually made.

The ultimate goal of such a system would be to increase the quality of life of all humans. Naturally, building a simulation of the necessary complexity is a huge undertaking, needing tens of millions of euros. However, small, limited prototypes can be developed cheaply to examine the workings of the system and as a guide for coming up with the necessary requirements of the system, and perhaps as a convincing argument to find grant money for building the systems that the world actually needs.

4.14 Social Network Modelling in Video Games (Part 1)

Shoshannah Tekofsky (Spirit AI - London, GB), Dan Ashlock (University of Guelph, CA), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Emily Short (Oxford, GB), David Thue (Reykjavik University, IS), Michael Treanor (American University - Washington, US), and Jichen Zhu (Drexel Univ. - Philadelphia, US)

License © Creative Commons BY 3.0 Unported license

© Shoshannah Tekofsky, Dan Ashlock, Matthew J. Guzdial, Emily Short, David Thue, Michael Treanor, and Jichen Zhu


The workshop started with a quick introduction of what social network modeling is and how it may map to video games. A social network model is a graph-based data structure where players are the nodes in the graph, and the edges between players are messages, relationships, or interactions between the players. A node contains (semi-)static information about the player, such as their personality, game demographics (profile), real life demographics, and session distribution. An edge can contain information about what characterizes the connection between two given players such as social style, weight (frequency/intensity), topic, cooperation/defection, and so forth. A network is the entire set of nodes and edges. It allows for calculations to be executed to approximate concepts such as popularity (in-degree) and social influence (centrality).

Having defined a social network, we continued on to brainstorm about applications of social networks in video games. There were four major threads identified. First, a game could be created where an AI attempts to disrupt or influence the social network of humans. The gameplay element could be to identify who is the AI and who is a regular human. This is in some ways reminiscent of Werewolves. Second, a game could be made that takes a human player's social network as input and generates a narrative game that happens in a similar social space. The game would thus have reflective properties for the player. Third, a game authoring tool may be developed that takes social network motifs as input for the creation of a story. The game could adapt to disruption of social motifs crucial for the story. For instance, if the story is based on a love triangle, but the player removes one of the characters

in the love triangle, the game may recover by establishing a love triangle between different characters. Fourth, a game could be created where the human player is a "gardener" or deity that shapes a social network of fictional characters as they play out their daily lives. This is similar to the Sims franchise of games, but with a focus on social relationships instead of human needs.

4.15 Social Network Modelling in Video Games (Part 2)

Shoshannah Tekofsky (Spirit AI - London, GB), Dan Ashlock (University of Guelph, CA), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

License  Creative Commons BY 3.0 Unported license
© Shoshannah Tekofsky, Dan Ashlock, and G. Michael Youngblood


This session started with a discussion of Bruce Tuckman's Stages of Group Development (1965) and how social network modeling and analysis could be used to represent, track, understand, and manipulate (for better or worse) this process—especially in the accelerated timescale of a game. This framing led to the discussion of other social phenomena that could also be represented with similar attributes and a progressive morphology.

The two days of conversations led to the identification of Social Network Modeling in Video Games constituting a novel field of video game studies. It is the synthesis of the methods and insights from the Social Network Modeling field to the application field of Video Games. The participants resolved to write a book as a primer and intro to the field. The topics and chapter structure were discussed.

The book will consist of two parts. Part 1 covers the technical application of Social Network Analysis (SNA) to video games, by introducing the relevant concepts from graph theory, applying the relevant data mapping to the graph for the video game space, and then working through the process of how to design a network representation for a given application. Part 2 of the book introduces four major application areas for SNA in games: group performance, norm/meme diffusion, social cohesion, and social modeling. First, group performance centers on the concept of matchmaking and development of productive teams in games. Second, norm/meme diffusion describes messages spread through a social network. Third, social cohesion details how players may increase or decrease the connectedness of their social network. Lastly, social modeling expands on the narrative-based applications for SNA that were discussed in part 1 of the workshop.

4.16 Backstory Generation

David Thue (Reykjavik University, IS), Mirjam P. Eladhari (Södertörn University - Huddinge, SE), Joshua Allen McCoy (University of California - Davis, US), Mike Preuß (Universität Münster, DE), Spyridon Samothrakis (University of Essex - Colchester, GB), Emily Short (Oxford, GB), Anne Sullivan (University of Central Florida - Orlando, US), Michael Treanor (American University - Washington, US), and R. Michael Young (University of Utah - Salt Lake City, US)

License  Creative Commons BY 3.0 Unported license

© David Thue, Mirjam P. Eladhari, Joshua Allen McCoy, Mike Preuß, Spyridon Samothrakis, Emily Short, Anne Sullivan, Michael Treanor, and R. Michael Young

In the context of Artificial Intelligence (AI) and its applications in digital games, the ability to automatically generate narrative content offers many potential benefits. For authors of game narratives, it could provide additional leverage in the content-creation process, both multiplying the outputs of their work via parameterized generation and accelerating their workflow via AI-assisted exploration of a potential narrative space. For players of game narratives, it could support the real-time adaptation of a game’s ongoing narrative, toward providing each player with a unique, personalized experience. While a significant amount of research has investigated story generation as the challenge of generating potential story futures, substantially less work has explored the challenge of generating potential story pasts – so called “backstories”.

The need for further discussion of backstory generation became apparent on the first day of the Dagstuhl Seminar “Artificial and Computational Intelligence in Games: AI-Driven Game Design”. Two preparatory talks (given by Peter I. Cowling and Emily Short, respectively) both highlighted backstory generation as a topic that creators of both digital games and interactive fiction were keen to see developed. Combined with the proposer’s prior interest, these talks inspired the creation of a workshop on backstory generation that took place on the same day.

Given the diverse backgrounds of the workshop attendees, we began the workshop with a discussion of each others’ motivations for attending. The result was an initial mapping of the space of potential applications for backstory generation. These took the form of questions that we might be able to answer if a backstory generator could be used, such as:

- How would authors use or interact with a backstory generator?
- What characterizes potential entry points for flexible, meaningful stories?
- How might we answer player questions about the past?
- Given a set of story characters, each with specific characteristics, can we generate plausible stories that explain how those characteristics came to be?
- More generally, given any desired next state of a narrative, can we generate a plausible chain of events that explains its occurrence?
- How might a drama/experience manager maximize a situation’s narrative meaning?

Using this list as a basis, we began to build a set of insights and concerns that would be important to consider when developing technology for backstory generation. These included:

- the additional challenge of backstory generation as compared to future story generation (since some backstories could contradict existing story history, the problem is more constrained than future story generation);
- the notion that contradictions between backstories can sometimes be smoothed over by declaring that some part of the conflicting information was false (e.g., lies or faulty recollections);

- the need to develop heuristics to guide decisions during generation, as well as story metrics to use in those heuristics (e.g., how much narrative meaning will be added?);
- the need to respect the intent of authors and integrate it in the generation process (this could mean not filling in some parts of the past, because their ambiguity is intended);
- the value of using multiple levels of granularity/abstraction when representing content (while a scientific approach demands granularity, creative authoring benefits from pragmatic abstraction); and
- the realization that environmental storytelling (i.e., the way in which narrative is conveyed through the visual layout of a scene) is a form of backstory-telling, which makes environment design a rich target for backstory generation.

This work was aided by choosing an example of backstory being revealed in a popular movie (in which a character says “I am your father”) and considering how and why such a revelation might be generated automatically.

Noting the importance of respecting the existing constraints of an ongoing story, we continued the workshop with a more detailed consideration of the narrative context in which a backstory generator might be used. Much of this discussion focused on the characters in a story’s world and how they might best be modelled to support valuable backstory generation. We identified two parts of such a model as being particularly important: a way for characters to change or suspend their goals or personalities, and a way for their goals or personalities to determine what aspects of prior events they privilege when revealing those events as backstory.

To conclude the workshop, we returned to our initial discussion of what backstory generation is and why we should pursue it, toward forming a tentative synthesis. In our view, a backstory in a narrative game is part of a story that (i) the player experiences non-contemporaneously with its execution, and (ii) that demonstrates (ideally meaningful) connections between two or more elements of the larger story’s history. Backstories should be generated automatically because they can add causal richness to a story’s world in a way that offers authors additional leverage over the narrative space, and also because they can provide AI-driven story managers with additional flexibility when personalizing each player’s narrative experience. The key challenges of backstory generation involve capturing and preserving authorial intent as well as ensuring sufficient narrative meaning in the results. Further work on this topic is planned, and we welcome discussions and suggestions for collaboration from all interested parties.

4.17 A General Language for Matching Tile Games

Julian Togelius (New York University, US), Cameron Browne (RIKEN - Tokyo, JP), Simon Colton (Falmouth University, GB), Mark J. Nelson (Falmouth University, GB), and Diego Perez Liebana (University of Essex - Colchester, GB)

License © Creative Commons BY 3.0 Unported license

© Julian Togelius, Cameron Browne, Simon Colton, Mark J. Nelson, and Diego Perez Liebana

Matching tile games are a genre of casual game based on, as the name implies, grids of tiles that need to be matched in various ways to clear them and/or score points. Popular examples include Bejeweled, Tetris, and Candy Crush Saga. This working group was inspired by two attempts to formalize the variation of game mechanics in the construction of this genre of games: a paper that examines the history of how these games arose historically through

■ **Listing 1** Bejeweled-like game in a JSON file format.

```
{
  "grid": {
    "type": "Square",
    "size": 4,
    "matchtilecell": "Square",
    "blocks": "single",
    "colour": "true"
  },

  "actions": { "rotate": {"size":2} },

  "rules": {
    "spawning": {"from": "Top", "type": "random"},
    "gravity": {"direction":"down"},
    "match":
    {
      "effect":"Clear",
      "type":"minimum",
      "number": 3,
      "shapes": "OrthoLine",
      "pattern": "colour",
      "reward": {"type": "Incremental", "value":100}
    }
  },

  "terminations": { "time": {"value":100} }
}
```

adaptations and variations on existing games [4], and a lengthy blog post cataloguing the various axes of variation needed to account for the games in this space [1].

This existing work that we consulted arose from a game-studies motivation, but is formal enough that it seemed promising to adapt it towards a generative model that would allow automatic discovery of new matching-tile games, an instance of rule-focused automated game design [5]. We investigated two approaches to modeling this space of games, both of which are currently in progress.

The first approach investigated adapting the Juul/Bailey style analysis as directly as possible, into a language along the lines of the Video Game Description Language (VGDL) designed at an earlier iteration of this Dagstuhl seminar series [2]. This approach takes each of the axes of mechanic variation they identified, and makes it explicit in a modeling language for matching-tile games. A specific game in this space is loadable from a JSON file by an interpreter. An example of a Bejeweled-like game described in this language can be seen in Listing 1.

The second approach we developed was a lower-level model inspired by interactive Cellular Automata (CA). Here, tiles are modeled as CA-style tiles, and mechanics are CA-style rules specifying how the following state is produced from the current state configuration. This approach does not explicitly capture Juul/Bailey style mechanic variations in the modeling language, instead using a smaller and lower-level set of representational primitives that make fewer assumptions, from which it would be expected that those higher-level mechanics

■ **Table 1** Three in a row scoring mechanic for a Bejeweled-like game.

Purpose	Start / End	Time	Row	Column	Colour	State	Score
Three in a row scoring	Start pattern	t	x-1	y	c1	tiled	0
		t	x	y	c1	tiled	0
		t	x+1	y	c1	tiled	0
	End pattern	t+ ϵ	x-1	y	\emptyset	exploded	+10
		t+ ϵ	x	y	\emptyset	exploded	+10
		t+ ϵ	x+1	y	\emptyset	exploded	+10

■ **Table 2** Tile dropping down mechanic for a Bejeweled-like game.

Purpose	Start / End	Time	Row	Column	Colour	State	Score
Tile dropping down	Start pattern	t	x	y	\emptyset	evacuated	0
		t	x	y+1	c1	tiled	0
	End pattern	t+ ϵ	x	y	c1	tiled	0
		t+ ϵ	x	y+1	\emptyset	evacuated	0

identified in existing matching-tile games would emerge as particular types of designs. For a similar Bejeweled-like game, different game mechanics could be specified as primitives such as the ones shown in Table 1 (for scoring a match of three tiles with the same colour in a row) and Table 2 (for a tile to drop down when a gap is available beneath it).

Lines of future work include an analysis of the expressiveness of these two game representations, as well as the study of algorithms that can search in these spaces. Additionally, the development of general algorithms that can play this type of games without domain knowledge could be an interesting avenue for future research, following the trends of the General Game Playing [3] and General Video Game AI [6] competitions.

References

- 1 Match Game Mechanics: An exhaustive survey. <https://www.gamasutra.com/blogs/JonathanBailey/20150227/237544/>. Accessed: 2018-01-19.
- 2 Marc Ebner, John Levine, Simon M Lucas, Tom Schaul, Tommy Thompson, and Julian Togelius. Towards a Video Game Description Language. In *Dagstuhl Follow-Ups*, volume 6, pages 85–100, 2013.
- 3 Michael Genesereth, Nathaniel Love, and Barney Pell. General Game Playing: Overview of the AAAI Competition. *AI magazine*, 26(2):62, 2005.
- 4 Jesper Juul. Swap Adjacent Gems to Make Sets of Three: A History of Matching Tile Games. *Artifact*, 1(4):205–216, 2007.
- 5 Mark J Nelson, Julian Togelius, Cameron Browne, and Michael Cook. Rules and Mechanics. In *Procedural Content Generation in Games*, pages 99–121. Springer, 2016.
- 6 Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Simon M Lucas, and Tom Schaul. General Video Game AI: Competition, Challenges and Opportunities. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 4335–4337, 2016.

4.18 Gameplay Evaluation Measures

Vanessa Volz (TU Dortmund, DE), Dan Ashlock (University of Guelph, CA), Simon Colton (Falmouth University, GB), Steve Dahlkog (Malmö University, SE), Jialin Liu (Queen Mary University of London, GB), Simon M. Lucas (Queen Mary University of London, GB), Diego Perez Liebana (University of Essex - Colchester, GB), and Tommy Thompson (Anglia Ruskin University - Cambridge, GB)

License © Creative Commons BY 3.0 Unported license

© Vanessa Volz, Dan Ashlock, Simon Colton, Steve Dahlkog, Jialin Liu, Simon M. Lucas, Diego Perez Liebana, and Tommy Thompson

The goal of this group is to develop a framework for logging information from games in a common format that captures common-currency metrics like win/loss, score as a function of time, entropy measures on games state, and others listed subsequently. The framework provides an implementation of a number of general measures that have previously been used to describe gameplay in an abstract form. The relevant information from the game has to be extracted with game-specific code implemented by the user and can then be processed by our framework. The framework is thus capable of logging full game states at each tick of the game, but also allows users to analyse specific characteristics of gameplay. The framework will be made available at [<https://github.com/GAIGResearch/GameEvaluationMetricsAtDagstuhl>] and is intended to be extensible to include more measures. During the seminar, we have implemented a number of features within the framework and applied it to the GVGA software, which already resulted in interesting observations

Based on related publications, we have collected a list of measures / features that we envision to include or have already included in the framework. In this list, we have included a standard collection of core measures that are generic across (almost) all types of games and thus permit a broad classification of types of games. We have also included more specific features such as agent decisiveness and contextualised actions, thus enabling more sophisticated comparisons on restricted sets of games. Additionally, we have included some features that are an interpretation of a measure (e.g. drama), but are well-established in our research field along with their definition in the literature. Although we agree that cosmetic aspects of a game have a significant effect on the player, we focus here on features that describe the actual gameplay and exclude aesthetics-based features.

The features we wish to be able to log are listed in the following, those that are starred are captured in the prototype code. The annotation (D) indicates a directly loggable measure while (I) is an indirect one requiring additional processing.

Direct logging features

- Game duration* (in ticks / seconds) (D)
- Agent score*(D)
- Game outcome (win, loss, tie, fail, etc.)*(D)
- Agent actions log*(D)
- Game state log(D)
- Game events (count; frequency)* (D)
- Available actions log / Branching factor* (D)
- Hierarchical actions / Contextualised actions (D / I)
- Lead change (D)

General indirect features

- Spatial entropy (I)
- Action entropy* (I)
- State space entropy (I)
- Object density* (I)

Agent-based features

- Agent surroundings (I)
- Agent decisiveness* (I)
- Agreement of agent with player models(I)
- Agreement between multiple different agents (I)
- AI Critics (I)
- State space exploration percentage (I)
- Empowerment (I)

Interpreted features (with references)

- Outcome uncertainty (I) [3]
- Drama log * (I) [3]
- Engagement (I) [4]
- Skill depth (I) [5]

In the following, we define lesser known and complex terms in this context:

Game Events

A game event is an invocation of a rule of the game, as opposed to a low level action.

Hierarchical Actions / Contextualised Actions

Hierarchical actions are those that aggregate multiple primitive actions in a cluster made meaningful based on their context. Examples include waiting by a power pill until ghosts are close in pac man or the shuffle-shuffle-bend-hand-gesture combination that emits a fireball in Street Fighter. Detection of hierarchical features is part of the future intentions. The group suggested discovery of hierarchical features via the use of bounded time window Markov models or, in the case where the actions are known – like the two examples – can be defined within the code hook as loggable events.

Lead Change

In multiplayer games with an obvious measure of how well the players are performing in a game (such as victory points, score, etc.), the player performing best based on this measure is considered to be in the lead at a given point in time. A lead change occurs if the player who is in the lead changes. The fact that the obvious measure of which player is in the lead is often deceptive is a key feature of many games, for example Reversi/Othello.

Spatial Entropy

The nominal definition of spatial entropy is the Shannon entropy of the spatial position of the agent. This sounds simple, but there is a problem: some positions on the board may not be accessible or, worse, may be accessible based on a state conditioned event or situation. The group decided to record the spatial entropy as if all cells in the game space are accessible. This gives a common currency measure of spatial entropy that is useful for comparison within a game. The practical effect is that the normalization of game entropy will yield a maximum below one where there are inaccessible cells. If the entropy is used for comparison between players or agents this is not a problem at all; if the full, correct value is required for some purpose then calculation of the number of accessible squares can be performed as a post analysis and then the entropy value scaled accordingly.

Agent Surroundings

As agent surrounding, we define the subset of the game state observable by the agent at a given point in time. The intention of logging this feature is to facilitate reasoning about the agent's tactics and strategy. For a 3D game, this could be based on the agent's first-person view. For a 2D scrolling game, this could be the currently observable part of the game, but could also allow a restricting the view to the neighbourhood of the agent in order to facilitate easier reasoning / learning.

Agent Decisiveness

Measures that describe the decisiveness of an agent are of course dependent on the specific agent implemented, but a suitable general measure is an agent's convergence speed. Some agents such as MCTS also have an obvious uncertainty measure.

AI Critics [1]

AI critics are agents with a set of goal distinct from finishing the game that are supposed to simulate player preferences. Based on these goals / preferences, they can offer a critical evaluation of the gameplay. In [1], the goals are expressed as weights on a utility function.

Empowerment [2]

Empowerment is an information-theoretic measure that on the one hand expresses the influence an agent has on the game via its actions and on the other hand the extent to which this influence is observable by the agent.

References

- 1 A. Liapis, C. Holmgård, G. N. Yannakakis, and J. Togelius. Procedural personas as critics for dungeon generation. In *European Conference on the Application of Evolutionary Computation (EvoApplication 2015)*, vol. 9028, 331–343, 2015
- 2 C. Guckelsberger, C. Salge, J. Gow, and P. Cairns. Predicting player experience without the player: An exploratory study. In *Annual Symposium on Computer-Human Interaction in Play (CHI PLAY)*. ACM, New York, NY, 305–315, 2017
- 3 C. Browne, and F. Maire. Evolutionary Game Design. In *IEEE Transactions on Computational Intelligence and AI in Games* 2(1):1–16, 2010
- 4 J. Brockmyer, C. Fox, K. Curtiss, E. McBroom, K. Burkhart, and J. Pidruzny. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. In *Journal of Experimental Social Psychology* 45(4):624–634, 2009

- 5 J. Liu, J. Togelius, D. Pérez-Liébana, and S. Lucas. Evolving Game Skill-Depth using General Video Game AI Agents. In *IEEE Congress on Evolutionary Computation (CEC'17)*, 2017.

4.19 Explainable AI for Designers

Jichen Zhu (Drexel Univ. - Philadelphia, US), Rafael Bidarra (TU Delft, NL), Alex J. Champandard (AiGameDev.com KG - Wien, AT), Simon Colton (Falmouth University, GB), Reynald Francois (Ubisoft - Düsseldorf, DE), Matthew J. Guzdial (Georgia Institute of Technology - Atlanta, US), Amy K. Hoover (NJIT - Newark, US), Antonios Liapis (University of Malta - Msida, MT), Sebastian Risi (IT University of Copenhagen, DK), Gillian Smith (Worcester Polytechnic Institute, US), Anne Sullivan (University of Central Florida - Orlando, US), and G. Michael Youngblood (Xerox PARC - Palo Alto, US)

License © Creative Commons BY 3.0 Unported license

© Jichen Zhu, Rafael Bidarra, Alex J. Champandard, Simon Colton, Reynald Francois, Matthew J. Guzdial, Amy K. Hoover, Antonios Liapis, Sebastian Risi, Gillian Smith, Anne Sullivan, and G. Michael Youngblood

In response to the rapid technological success in AI, the emerging research area of Explainable AI aims to better communicate AI systems' decisions and actions to human users. The central goals of explainable AI are often to increase users' understanding, foster trust, and improve their ability to utilize the systems.

Explainable AI for designers (XAID), in particular, focuses on enhancing designers' capability to (co-)create user experiences with AI. Through the vantage point of computer games, we examine 1) the design space of explainable AI for game designers, 2) three case studies of XAIDs, and 3) design guidelines and open challenges in each case.

We identified different types of XAID techniques that can facilitate the game design and development process. In a broad stroke, they can be categorized into i) what to explain, ii) when to explain, and iii) how to explain. In terms of what to explain, XAIDs can be used to communicate many aspects of game AI. For instance, they can be used to explain the process of the chain of actions and reactions taken by game AI. Alternatively, they can simply explain the results of processing certain inputs. Regarding when to explain, the description may take place before, during, or after AI's operations, each scenario affording different types of explanations. As for how to explain, factors such as the form of explanations (e.g., as a tutorial or as justifications of specific AI actions at hand), levels of abstraction (e.g., concrete details or high-level abstraction), and the interaction model (e.g., AI as a tool for the human designers or as a co-creator with the designers) directly influences of how XAIDs should be designed.

To ground our survey of the XAID space, we conducted three case studies based on the type of AI systems (black-box or white-box) and the part of the game development process in which AI techniques are used. The case studies include XAIDs for 1) white-box procedural content generation (PCG) systems, 2) black-box PCG systems, and 3) black-box NPC behavior control.

In a white-box PCG system, we assume a system that has full knowledge of the underlying processes taking place; this allows XAID (e.g. in the form of text generation) to be embedded within the content generation code. White box PCG can hypothetically output a narrative (as a sequence of sentences), with each sentence produced following each command or function call. Ideal generative architectures for this approach are pipelines, where a number of generative processes are "chained", each producing an intermediate result which is taken by

the next process as input and producing an enhanced result as its own output [1]. In such an architecture, generation of textual explanation need not be internalized within the codebase, instead assessing the intermediate results from each process along the pipeline. Explanations in a white box PCG system can be produced at any point in the generative process and in any degree of clarity (as the explanation subsystem can have full knowledge of the underlying logic or ways in which content quality is assessed). Due to these reasons, the challenge for XAID in white box PCG is how to handle the possibly vast volume of information that can be output by such a system. Presenting a compelling and intuitive narrative to the user regarding the choices taken by the system can be done:

- **sequentially** in the order that the system makes decisions. This explanation can follow some form of story structure which simulates e.g. the generative pipeline [2]. Work on story generation can be used to enhance how the connections are made between different time slices in the generation (e.g. via natural language processing [3]) so that the narrative is coherent and causal links are made obvious. This can be achieved by post-processing the generated sequence of sentences to introduce throwbacks to past generative decisions which affect future outcomes or to foreshadow how one decision early on affects the final outcome.
- as **highlights** of the generative process by filtering out and omitting less interesting points in the generated sentence structure. For this to happen, a number of evaluation mechanisms are needed to assess interestingness (will this be interesting to a human user?), clarity (will this be understandable by a human user?), or creativity (is this point a creative milestone [4] where the design shifts?) of the text or the underlying generative commands that prompted it. Therefore, it is necessary to have the whole narrative (sequence of sentences) before the most interesting points within it are chosen in a post-processing step.
- **non-sequentially**, summarizing the explanation starting from the most important points regardless of whether those are performed first or last in the code. Indeed, it is possible to start by presenting a description (visual, textual, or otherwise) of the final artifact and backtracking some of its most interesting elements on points in the generative process where those happened. Moreover, tropes such as sports game summaries can be used as inspiration, presenting the main outcomes of the generative process first (as non-sequential highlights) followed by a longer form of the sequential narrative regarding how generation progressed from unformed to fully formed content.

An open challenge in providing useful XAID is how to fit the entire processes of the white-box system into something that is compact and yet sufficient for designers.

In a black-box PCG system, we specifically looked into what type of XAID would be useful for game designers at a AAA game studio. We determined an "AI as student" framework matched designers intuitions when it came to artificial intelligence and black box machine learning techniques. To fit this framework a potential agent would need to: (1) share a common language of design with the human designer, (2) communicate its current understanding of this language, and (3) update this understanding in response to designer feedback. We identified three areas of existing work that matched these requirements: zero and one shot learning, explainable AI, and active learning respectively. In practice this would look like a designer pre-defining content with certain tags and an AI training on these tags (zero and one shot learning), the designer interrogating the agent's output content and tags through checking the maximally activated filters (explainable AI), and giving feedback through single examples, which the AI could use to retrain (active learning). A key **open challenge** in this area is how to filter out common sense knowledge in the explanation.

Finally, in a *black-box NPC behavior* control system, we focused on agents controlled by deep neural networks (DNNs), especially deep Q-networks (DQNs). Two types of information stood out as of particular importance to be well communicated to game designers. First, given a particular gameplay context, what is the likely distribution of actions the NPC can take at this given moment. Second, given a particular NPC action, what are all the possible situations that can lead to this action. Given the large number of possible actions and/or situations, similar to highlights in white-box PCG systems, a good design guideline for XAID is to highlight the unexpected and reduce the visibility of the common. A key open challenge to provide both types of information to a human designer is how to design the reward function.

In conclusion, explainable AI for designers is crucial for advancements in AI to be fully utilized in computer games and other types of interactive experiences. Results from this XAID workshop show that the current understanding of how to communicate the underlying AI algorithms to human designers is still rudimentary. Although different types of AI algorithms place varying challenges and opportunities for the corresponding XAIDs, an emergent key challenge shared in all three cases we investigated is **salience**. That is, among the various types of information that could be provided about the underlying AI algorithms, how do we define, identify, and communicate what is noteworthy. Future work includes deeper understandings of salience grounded in the specific needs of designers, algorithmic investigations of how to procedurally identify salient features, as well as design innovation of how to communicate them to human designers.

References

- 1 John Charnley, Simon Colton, Maria Teresa Llano Rodriguez and Joseph Corneli. The FloWr Online Platform Automated Programming and Computational Creativity as a Service. In *Proceedings of the International Conference on Computational Creativity*, 2016
- 2 Simon Colton, Jakob Halskov, Dan Ventura, Ian Gouldstone, Michael Cook and Blanca Perez-Ferrer. The Painting Fool Sees! New Projects with the Automated Painter. In *Proceedings of the International Conference on Computational Creativity*, 2015.
- 3 François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. Automatic generation of textual summaries from neonatal intensive care data. In *Artificial Intelligence* 173, 7–8 (May 2009), 789–816. 2009.
- 4 Georgios N. Yannakakis, Antonios Liapis, Constantine Alexopoulos. Mixed-Initiative Co-Creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.

5 Panel discussions

5.1 Evaluation

Pieter Spronck (Tilburg University, NL)

License © Creative Commons BY 3.0 Unported license
© Pieter Spronck

At the end of the Seminar, we had a one-hour session with all attendees still present to evaluate the seminar. While the general consensus was that the Seminar had been a lot of fun and a great success, a desire was expressed to be a slightly less free-form for a future Seminar. In particular, the following points were brought up:

- Several attendees felt that they should have been primed more before arriving on the purpose of the Seminar and what was expected of them.

- To get more off-the-wall, crazy ideas, attendees should be instructed to think about and discuss them before getting to the Seminar. As it was, the “crazy” ideas started flowing only on the second day.
- Working groups could already be set up before arriving at the Seminar.
- Several attendees expressed an interest in having someone give an instructive workshop, for instance on a tool.
- It would be valuable if all working groups would report back in a structured form using sheets.

Furthermore, a discussion was held on the value of having an explicit code-of-conduct. Several arguments for and against having a code-of-conduct were brought up, but ultimately no conclusion was reached, apart from the fact that some attendees would really appreciate one as several conferences in the past (not Dagstuhl meetings) would have benefited from having one. In this respect, Dagstuhl may consider to write an explicit code-of-conduct for Dagstuhl Seminars in general.

Participants

- Elisabeth André
Universität Augsburg, DE
- Dan Ashlock
University of Guelph, CA
- Sander C.J. Bakkes
Tilburg University, NL
- Rafael Bidarra
TU Delft, NL
- Cameron Browne
RIKEN – Tokyo, JP
- Alex J. Champandard
AiGameDev.com KG – Wien, AT
- Simon Colton
Falmouth University, GB
- Michael Cook
University of London, GB
- Peter I. Cowling
University of York, GB
- Steve Dahlskog
Malmö University, SE
- Mirjam P. Eladhari
Södertörn University –
Huddinge, SE
- Reynald Francois
Ubisoft – Düsseldorf, DE
- Matthew J. Guzdial
Georgia Institute of Technology –
Atlanta, US
- Amy K. Hoover
NJIT – Newark, US
- Pier Luca Lanzi
Polytechnic University of
Milan, IT
- Marc Erich Latoschik
Universität Würzburg, DE
- Antonios Liapis
University of Malta – Msida, MT
- Jialin Liu
Queen Mary University of
London, GB
- Simon M. Lucas
Queen Mary University of
London, GB
- Joshua Allen McCoy
University of California –
Davis, US
- Alexander Nareyek
Singapore, SG
- Mark J. Nelson
Falmouth University, GB
- Ana Paiva
INESC-ID – Porto Salvo, PT
- Diego Perez Liebana
University of Essex –
Colchester, GB
- Mike Preuß
Universität Münster, DE
- Sebastian Risi
IT University of
Copenhagen, DK
- Spyridon Samothrakis
University of Essex –
Colchester, GB
- Jacob Schrum
Southwestern University –
Georgetown, US
- Emily Short
Oxford, GB
- Adam M. Smith
University of California –
Santa Cruz, US
- Gillian Smith
Worcester Polytechnic
Institute, US
- Pieter Spronck
Tilburg University, NL
- Anne Sullivan
University of Central Florida –
Orlando, US
- Shoshannah Tekofsky
Spirit AI – London, GB
- Tommy Thompson
Anglia Ruskin University –
Cambridge, GB
- David Thue
Reykjavik University, IS
- Julian Togelius
New York University, US
- Michael Treanor
American University –
Washington, US
- Vanessa Volz
TU Dortmund, DE
- Georgios N. Yannakakis
University of Malta – Msida, MT
- R. Michael Young
University of Utah –
Salt Lake City, US
- G. Michael Youngblood
Xerox PARC – Palo Alto, US
- Jichen Zhu
Drexel Univ. – Philadelphia, US

