ECS7017P - Computational Game Design (Coursework 3)

Design of Game Description Languages.

Rohan Ajay Deuskar – EC22054

1. Introduction:

A formal method of conceptualizing games using logic, regular expressions, or other methods to define notions of rules, state transitions and so on is called a Game description language. The first evidence of a game description language is found in 1996 the Metagame System [1] before being formally called a game description language, the Metagame System was a general game player from strategic chess-like games which included a grammatical representation of the rules of a game (See Figure 1). Later in the year 2005 when AAAI started the General Game Playing field of research and standardised 'GDL' as their language used for describing games. [2]

Since the inception of GDL, many other game description languages have been created some for general game playing while some for other areas of academic research, some even for commercial use. The scope of implementation has been broadened to many different areas.

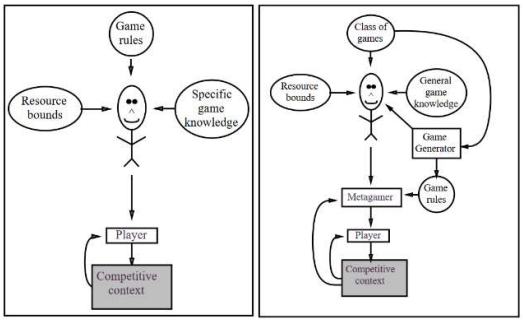


Figure 1: Metagame System describing a game player (left) vs a meta-game player(right) [1]

Primarily, Game Description Languages were used in General Game Playing, an area of research where researchers can test and evaluate General Game AI agents and search algorithms. Alternatively, the use of these languages using general game playing has also led to research in discovering and measuring new game dimensions for evaluating games.

Another use of Game Description Languages has been procedural generation. This can be further classified into procedural generation in games themselves and procedural generation in game content. Since game description languages define the setup, game rules and legality concisely, it is possible to let AI algorithms change, evaluate, and produce new content using these. This has been done not only within games to produce newer content in those games, but also in a niche of game-types to produce new games itself.

Other than academic success, game description languages have also had commercial success, from developing and selling board games produced using these languages to selling softwares to build our own games using simple implementations and customisation of existing games in the system.

GDL had a lot of success in developing General Game Players, after AAAI's competition, AI agents like CadiaPlayer, WoodStock and Galvanise had incredible success in it [2]. Other languages like Regular Board Games RBG tried to optimise the system that GDL had by changing the language basis from logic programming to – In RBG's case – regular expressions. The Digital Ludeme project was brought forth with languages like Ludi and Ludii as a tool for AI researchers to model strategy games of the past and generated board games commercially as well [3], [4]. Another note-worthy mention is of VGDL which brought the concept of game description languages from board games to video games. VGDL has had success in not only designing 2D video game AI [5] but also with procedural game content generation [6]. ANGELINA is another example of this and has been able to now generate 3d games which were enough to not be realised to be AI made by judges in a game jam competition. [7]

Since the advancement in AI and search algorithms like reinforcement learning and Monte Carlo Tree search, the use of game description languages to create general game players has been improved. Languages like RBG, Ludii have promising implementations of RL agents and have been design in a way to be efficient yet balance human-readability [4], [8]. An extension proposed as a future work in VGDL was a complete fine-grained score system with intermediate rewards which could help RL approaches in the system.

RBG's compiler optimisations and efficiency reasoning in the recent study published by researchers also emphasize the impact of benchmarks in General Game playing using description languages[9].

The Digital Ludeme Project and its objectives to create a study of research called Digital Archæoludology using Ludii to map the historical strategy games using evolution and other strategies allow ancestral state reconstruction to shed light on the gaps in our partial knowledge about the history and mathematical representation between games.

2. Employment of Game Description Languages

As described earlier, game description languages can be used over several different areas. The design of these languages can also be different depending on the intended area of their employment. Some languages can also be used over multiple usages and thus would be considered more versatile than others in terms of usage.

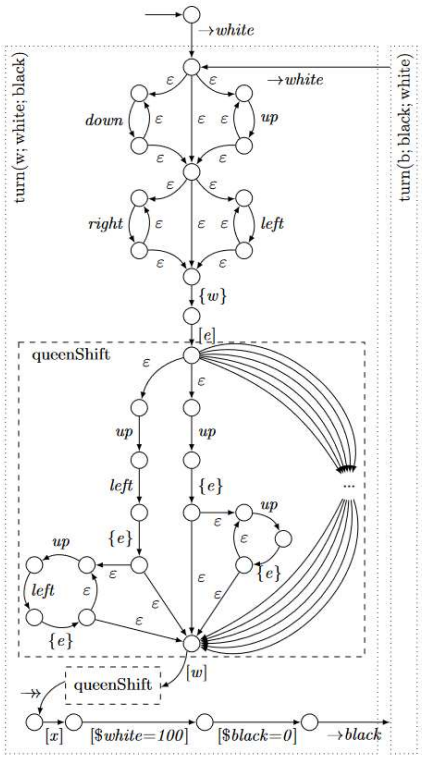
2.1. General Game Playing

Association for the Advancement of Artificial Intelligence sponsored the General Game Playing project by Stanford University firstly in Twentieth National Conference on Artificial Intelligence which later became a yearly competition [2]. General Game Playing was a concept which got birth due to the idea that an AI agent designed to be good at one game cannot be good at another genre of games. The purpose of general game player or GGP was to make an agent which would be just as good at one game as it was in another game.

GDL was introduced for this competition as a standard language for their system. As a logic-based language it used declarative statements to define game states and logical statements to define game rules and at that point of time could only support complete information and deterministic games. Due to that, scope of games that could be coded in GDL at this point was rather limited. Later versions of GDL – GDL-II [10] and GDL-III [11] – introduced incomplete information and epistemic game support to the system as well.

Up until GDL-II was introduced, GDL could not support games like Backgammon or Poker, the reason being there was no method of introducing randomness or hidden information in the system, the games described had to be perfect-information. This was fixed by GDL-II which brough forward the concepts of randomness and hidden information by extending the vocabulary of the language to support them. However, GDL-II still could not support puzzles like Cheryl's birthday as they required epistemic states and introspection. GDL-II could either hide information from everyone or no-one, GDL-III was introduced to combat this issue and provided keywords to support these concepts. The main constrain of GDL was the explicit declaration of game states, changing the size of a game board or deck of cards would require significant re-writing of the code, or designing complex games like GO, Amazons was not a feasible task.

In 2018, Regular Boardgames (RBG) was proposed [8]. The concept was to use regular language rather than GDL's logic language. This idea came from a previous work on encoding movements of chess-like simplified board games using regular language [12] and RBG extended that idea to include any deterministic board game. RBG uses a high-level language for human readability and concision. It then translates this to a low-level language for the system and agents playing the game, this provides efficiency of the system as well as the ability to write complex games using it. An experiment was performed to assess the efficiency of the system in comparison to GDL and the results proved that although they had similar performance in simple games like Connect 4, RBG was the superior system when it came to complex games like Amazons or Reversi. The experiments proved that RBG was capable of fixing constrains that GDL had faced.




```
#players = white(100), black(100)
#pieces = e, w, b, x
#variables = // no variables
#board = rectangle(up,down,left,right,
          [e, e, e, b, e, e, b, e, e, e]
          [e, e, e, e, e, e, e, e, e, e]
          [e, e, e, e, e, e, e, e, e, e]
          [b, e, e, e, e, e, e, e, e, b]
          [e, e, e, e, e, e, e, e, e, e]
          [e, e, e, e, e, e, e, e, e, e]
          [w, e, e, e, e, e, e, e, e, w]
          [e, e, e, e, e, e, e, e, e, e]
          [e, e, e, e, e, e, e, e, e, e]
          [e, e, e, w, e, e, w, e, e, e])
#anySquare = ((up* + down*)(left* + right*))
#directedShift(dir) = (dir {e} (dir {e})*)
#queenShift = (
    directedShift(up left) + directedShift(up) +
    directedShift(up right) + directedShift(left) +
    directedShift(right) + directedShift(down left) +
    directedShift(down) + directedShift(down right)
  )
#turn(piece; me; opp) = (
    ->me anySquare {piece} [e]
    queenShift [piece]
    queenShift
    ->> [x] [$ me=100, opp=0]
  )
#rules = (turn(w; white; black) turn(b; black; white))*
```

Figure 2: An automata representation of Amazons game (left) and the language representation in RBG (right) [9]

The Digital Ludeme Project (DLP) was a project started to model the world's traditional strategy games in a single playable database, for this purpose they needed a game description language. Using GDL was not feasible as it is difficulty in encoding complexity, in response, they created Ludii as a system which used their concept of ludemes from the Ludi GDL and the ability for integration to external applications. Ludemes are atomic objects of a game that are labelled in the language, this allowed them to use class grammar approach for designing games. This also allows for a 1:1 mapping from the language to source code.

Researchers at DLP published an empirical evaluation between Ludii, RBG and GDL which brought forth results showing Ludii to be far superior in human readability and understanding compared to the other two with on average needing a third of the tokens needed by RBG [13]. The study also compared the efficiency of the systems in which Ludii was shown to be outperforming GDL as well as RBG in most of the tested games. However, soon after, the researchers behind RBG came forward with a note to the study describing the flaws of improper implementation of games and unequal testing grounds for the study which fell in favour of Ludii [14]. They performed their own experiments with mentioned limitations and found that RBG was outperforming GDL and Ludii in most cases instead. Thereafter they have published another comparison between Ludii, RBG, GDL, and Ai Ai with addressed limitations of Ludii and Ai Ai being closed source [9]. This study also concluded in RBG having the superior efficiency on average.

```
wwwwwwwwwwwww
wA          w   w
w   w           w
w     w     w +ww
www  w1   wwwww
w           w G w
w 1             ww
w       1       ww
wwwwwwwwwwwww
```

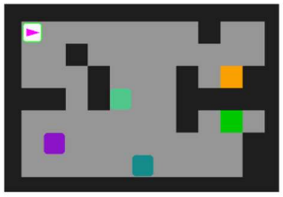


Figure 3: Representation of a game state in VGDL [15].

Taking inspiration from Arcade Learning Environment (ALE) [16] and following a proposal of creating a general-purpose video game description language, Tom Schaul created the Video Game Description Language (VGDL) [15]. The design of the language is based on defining locations and dynamics for simple building blocks, and the interaction effects when such objects collide. This was the first concrete, ambitious abstract language able to encode a subset of video games. Subsequently, the General Video Game Artificial Intelligence (GVGAI) came into

existence. VGDL provided an open-ended method of testing GVGAI as any game encoded in VGDL could be played by an agent written on the system. However, VGDL could only support Atari-like 2D single player games and had around eighty game implementations in it.

2.2. Game and Content Generation

As the rule set is defined and the games can be tested using self-play by the GGP, researchers figured out that this could be used to evolve new games and evaluate them.

Zillions of Games was a proprietary software for Windows95 [17] which was described as a universal game engine which could intelligently play any game once read in, this was because it used a custom language called Zillions Rule File (ZRF) as its game description language, and any game designed in Zillions could be played by the GGP in the engine. Using this, Vincent Hom and Joe Marks proposed the Automatic Game Design (AGD) field of study [18]. They used Zillions to evolve games randomly and using a genetic algorithm. The pool of games included 32 traditional games like Reversi, Tic-Tac-Toe, Checkers and other. The main constrain of the study was not having access to the General Game Player, since ZoG was a commercial software, the source code was not available. The game search was also limited and not dynamic as ZRF could not allow game tree search due to its design.

One of the key research objectives of DLP was to not only model the games but to also reconstruct missing knowledge of these games. For this purpose, the researchers created Ludi GDL which was a predecessor of the previously described Ludii system. Ludi GDL was also based on the concept of ludemes, and thus was considered a tree-based language. This proved to be a vital necessity for the purpose of their experiments – Evolutionary Game Design – as tree-based rulesets were much simpler to evolve compared to logic-based statements [3]. A total of 1389 games were evolved from a sample set of seventy-nine games and 19 of those were deemed viable. Two of these games, Yavalath and Ndengrod (Pentalath) were also commercialised and have had more than 50 sales each [19], [20]. The study, however, was limited to 2-player combinatorial board-games.

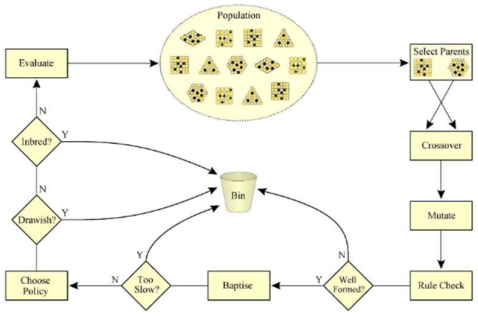


Figure 4: Game Lifecycle representation in Ludi [3].

Gcardgame [21] was a language introduced as a system capable of generating novel card games. Stochastic simulations were used both to verify the implementation of well-known games like Uno, Blackjack and Texas Hold' em Poker, and to evaluate the results of new game rules derived from the grammar. Gcardgames defines it games in a set of basic axioms such as players, card location, tokens, etc. and then a ruleset for the game. Using these the language can define any card game. Initial study produced randomly generated but playable card games and the research indicates that there is a possibility of improving the generation of card games using search and optimizing techniques.

VGDL and the GVGAI framework also had research in the field of generating content. One such study generated four hundred games randomly and using a genetic algorithm from a set of 13 games as input to them [5]. They evaluated 54 of the fittest games from the outcome and found some great game design content, however, no games of reasonably high quality were found. The key issues addressed were tied to computational expenses and lack of human timescale for responses. No issues were described about the language itself or its optimisations.

The study of search-based content generation in games has a lot of potential still and we will likely be seeing more study in this area soon.

Most of the content generation has been based around combinatorial two player board games or single player 2D video games, although there has not been much other content generated with respect to game description languages in this area of research, one study which tackled 3d space and more diverse set of games is ANGELINA[22]. ANGELINA does not necessarily use game description languages, but we can draw parallels of game description languages and abstraction of the

ANGELINA system. The initial version of ANGELINA used Rule Set, Level and Layout as abstract designs which were used in the Co-operative Co-evolution. [23]

2.3. Evaluating and/or Discovering Game Dimensions.

Another area of research that game description languages have been used is measuring game dimensions.

For categorising and evaluating generated games using Ludi, researchers had to experiment with figuring out automated game measurement to rank the games as well. They identified a total of 17 aesthetic criteria as predictors which later came down to 16 after dropping the puzzle criteria resulting in a significant speed advantage and negligible loss to accuracy [3]. These criteria were evaluated against human labelled values for the set of games and was able to predict them well. The results proved that there were indicators to measure game quality atleast in the scope of what Ludi GDL can produce.

Another interesting study in the terms of using game description languages to find game dimensions was done using a system called Modular Game Library MOGAL – later built on and called Ai Ai – to find the measure the concepts of elegance and Shibui – Japanese concept of measuring beauty [24]. The results were promising for the concept of Shibui as a potential new dimension of measuring game aesthetics.

An additional study of interest that falls into these uses is a game engine called LUDOCORE which is based on ASP and Event Calculus and is used to model videogames [25]. LUDOCORE acts more on the lines of computational support for game testing and designing but it does describe the model of the entire game, hence it can be considered to make comparisons with game description languages as well.

Overall, although there is not much research in this field, there is a lot of potential. Finding measures to evaluate games not only provides us with a way of ranking a set of games, but also allows gives us a measurement which can be used in concepts like evolutionary game design to have automated game evaluations and fitness functions for genetic algorithms used in it.

2.4. Commercial Use

The initial use of game description languages commercially was with Zillions of Games (ZoG) [17]. ZoG was developed in 1997 and supported over 48 – Later went up to 5000 – games including Chess, Checkers, Go, Reversi, Shogi, and others. The reason for its success lies in the existence of a general game player and the game description language. ZoG supported a custom description LISP-like language which would be written in Zillions Rule File (ZRF) format. Games written in this language could be added to Zillions and be playable by the AI of the system.

Another incredible system available was Ai Ai – previously called MoGaL – developed by Stephen Tavenere [26]. The software was originally called Modular Game Library which was a PhD project. Ai Ai currently has 480+ games available to play in the software. Ai Ai provides a lot of analysis tools and functions. The game description in Ai Ai can be written in Modular Game Language which is a JSON-like scripting language or can directly be written in Java as well.

These commercially available bundles and their languages had a very distinct design difference compared to academically used GDL, these languages used board descriptions along with visual elements being a part of the languages, as they are used to provide playable entertaining games to the buyer. A good example of this would be the image token used in Zillions which specified the image to be used for boards and pieces.

3. Conclusion

Game description languages provide a generalised method of describing games. This has been particularly helpful in the field of general game playing where agents can learn the game rules and legality of any game described in the language. The class-method approach or regular language approach have also provided the innovative possibility of generating game content and games themselves. Not only has this provided a new area of research but has also increased the efficiency of these languages.

The use of game description languages was initially restricted to general game playing however researchers saw the potential of other areas such as procedural content generation, using RL agents, game evaluation, etc.

There has also been a significant rise aimed at targeting the efficiency of the languages. With GDL systems evolving from GDL reasoners like prop nets [27] to languages like Ludii and RBG which are high-level languages which are then translated to low-level languages for efficiency yet still allow human-readability for the description language itself.

Game description languages have also evolved to allow more complex and varied games to be defined, with GDL-II [10] and GDL-III [11] allowing GDL to include epistemic hidden information games. Ludii introducing system of ludemes which allows for complex 2-player combinatorial games [4]. RBG in similar grounds by introducing macros in the language to allow increased complexity in the definitions [8]. VGDL introducing game description to 2-D arcade games and broadening the concept of game description languages to video games [15]. Gcardgames with potential for card game procedural generation using genetic algorithms is another future work possible in generating novel card games using a game description language [21].

This study has also shown the various fields apart from research where game description languages have been used and how they differ from academic purposes. ZoG providing entertainment value in its era [17] and Ai Ai providing a library of playable and measurable games in the same context of entertainment [26].

Future research in game description languages will show more efficient systems built on the languages, introduction of RL agents to train and test novel games. Procedural content generation mainly Procedural content generation using Machine Learning (PCGML) based on these languages. Adaption of more 2D video game genres using VGDL or another language. Not only 2D, but also the field of 3D games and their abstraction is an interesting concept for exploration.

Despite its limitations and challenges, these languages have proven to be a valuable tool for studying and generating games. With the ongoing development of AI algorithms like Reinforcement learning and other genetic evolution techniques like NEAT, game description languages are likely to continue to play a significant role in the future of game development and research.

4. References:

[1] B. Pell, "A STRATEGIC METAGAME PLAYER FOR GENERAL CHESS-LIKE GAMES," *Comput. Intell.*, vol. 12, no. 1, pp. 177–198, Feb. 1996, doi: 10.1111/j.1467-8640.1996.tb00258.x.

[2] M. Genesereth, "General Game Playing: Overview of the AAAI Competition".

[3] C. Browne and F. Maire, "Evolutionary Game Design," *IEEE Trans. Comput. Intell. AI Games*, vol. 2, no. 1, pp. 1–16, Mar. 2010, doi: 10.1109/TCIAIG.2010.2041928.

[4] É. Piette, D. J. N. J. Soemers, M. Stephenson, C. F. Sironi, M. H. M. Winands, and C. Browne, "Ludii -- The Ludemic General Game System." arXiv, Feb. 21, 2020. doi: 10.48550/arXiv.1905.05013.

[5] T. S. Nielsen, G. A. B. Barros, J. Togelius, and M. J. Nelson, "Towards generating arcade game rules with VGDL," in *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, Aug. 2015, pp. 185–192. doi: 10.1109/CIG.2015.7317941.

[6] D. Perez, "Tutorial I: Video Game Description Language (VGDL) and the challenge of creating agents for General Video Game Playing (GVGP)," in *2015 IEEE Conference on Computational Intelligence and Games (CIG)*, Aug. 2015, pp. 20–20. doi: 10.1109/CIG.2015.7317661.

[7] C. Donlan, "Can an AI win a game jam?", [Online]. Available: https://www.eurogamer.net/can-an-ai-win-a-game-jam?utm_source=social_sharing&utm_medium=CopyLink&utm_campaign=social_sharing

[8] J. Kowalski, M. Mika, J. Sutowicz, and M. Szykuła, "Regular Boardgames." arXiv, Nov. 13, 2018. doi: 10.48550/arXiv.1706.02462.

[9] J. Kowalski *et al.*, "Efficient Reasoning in Regular Boardgames," in *2020 IEEE Conference on Games (CoG)*, Aug. 2020, pp. 455–462. doi: 10.1109/CoG47356.2020.9231668.

[10] M. Thielscher, "GDL-II," *KI - Künstl. Intell.*, vol. 25, no. 1, pp. 63–66, Mar. 2011, doi: 10.1007/s13218-010-0076-5.

[11] M. Thielscher, "GDL-III: A Description Language for Epistemic General Game Playing," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 1276–1282. doi: 10.24963/ijcai.2017/177.

[12] J. Kowalski, J. Sutowicz, and M. Szykuła, "Simplified Boardgames." arXiv, Jul. 15, 2016. doi: 10.48550/arXiv.1606.02645.

[13] É. Piette, M. Stephenson, D. J. N. J. Soemers, and C. Browne, "An Empirical Evaluation of Two General Game Systems: Ludii and RBG." arXiv, Jun. 29, 2019. doi: 10.48550/arXiv.1907.00244.

[14] J. Kowalski, M. Mika, J. Sutowicz, and M. Szykuła, "A note on the empirical comparison of RBG and Ludii." arXiv, Oct. 04, 2019. doi: 10.48550/arXiv.1910.00309.

[15] T. Schaul, "A video game description language for model-based or interactive learning," in *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, Aug. 2013, pp. 1–8. doi: 10.1109/CIG.2013.6633610.

[16] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The Arcade Learning Environment: An Evaluation Platform for General Agents," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, Jun. 2013, doi: 10.1613/jair.3912.

[17] "Zillions of Games -- Unlimited Board Games & Puzzles." https://www.zillions-of-games.com/ (accessed May 01, 2023).

[18] V. Hom and J. Marks, "Automatic Design of Balanced Board Games," *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, vol. 3, no. 1, Art. no. 1, 2007, doi: 10.1609/aiide.v3i1.18777.

[19] "Yavalath," *BoardGameGeek*. https://boardgamegeek.com/boardgame/33767/yavalath (accessed May 01, 2023).

[20] "Pentalath," *BoardGameGeek*. https://boardgamegeek.com/boardgame/51401/pentalath (accessed May 01, 2023).
[21] J. M. Font, T. Mahlmann, D. Manrique, and J. Togelius, "A Card Game Description Language," in *Applications of Evolutionary Computation*, A. I. Esparcia-Alcázar, Ed., in Lecture Notes in Computer Science, vol. 7835. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 254–263. doi: 10.1007/978-3-642-37192-9_26.
[22] M. Cook, S. Colton, and J. Gow, "The ANGELINA Videogame Design System—Part I," *IEEE Trans. Comput. Intell. AI Games*, vol. 9, no. 2, pp. 192–203, Jun. 2017, doi: 10.1109/TCIAIG.2016.2520256.
[23] M. Cook and S. Colton, "Multi-faceted evolution of simple arcade games," in *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, Aug. 2011, pp. 289–296. doi: 10.1109/CIG.2011.6032019.
[24] C. Browne, "Elegance in Game Design," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 3, pp. 229–240, Sep. 2012, doi: 10.1109/TCIAIG.2012.2197621.
[25] A. M. Smith, M. J. Nelson, and M. Mateas, "LUDOCORE: A logical game engine for modeling videogames," in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, Aug. 2010, pp. 91–98. doi: 10.1109/ITW.2010.5593368.
[26] "Ai Ai Home Page – Stephen Tavener," May 08, 2016. http://mrraow.com/index.php/aiai-home/ (accessed May 01, 2023).
[27] C. F. Sironi and M. H. M. Winands, "Optimizing Propositional Networks," in *Computer Games*, T. Cazenave, M. H. M. Winands, S. Edelkamp, S. Schiffel, M. Thielscher, and J. Togelius, Eds., in Communications in Computer and Information Science, vol. 705. Cham: Springer International Publishing, 2017, pp. 133–151. doi: 10.1007/978-3-319-57969-6_10.