

Ludii as a Competition Platform

Matthew Stephenson, Éric Piette, Dennis J.N.J. Soemers and Cameron Browne

Department of Data Science and Knowledge Engineering

Maastricht University

Maastricht, The Netherlands

{matthew.stephenson, eric.piette, dennis.soemers, cameron.browne}@maastrichtuniversity.nl

Abstract—Ludii is a general game system being developed as part of the ERC-funded Digital Ludeme Project (DLP). While its primary aim is to model, play, and analyse the full range of traditional strategy games, Ludii also has the potential to support a wide range of AI research topics and competitions. This paper describes some of the future competitions and challenges that we intend to run using the Ludii system, highlighting some of its most important aspects that can potentially lead to many algorithm improvements and new avenues of research. We compare and contrast our proposed competition motivations, goals and frameworks against those of existing general game playing competitions, addressing the strengths and weaknesses of each platform.

Index Terms—General Game Playing, Artificial Intelligence, Competitions, Ludii, Ludemes, Board games

I. INTRODUCTION

Game-based research and analysis has, and continues to be, one of the most popular and long running approaches for developing, evaluating and comparing new artificial intelligence algorithms [1]. Since the early days of AI research, games have been used to demonstrate the abilities and potential of AI for solving complex combinatorial problems [2]. Playing games effectively often requires multiple skills, such as understanding the game rules, evaluating different situations, planning several moves ahead, predicting the behaviour of other players, selecting one of many possible actions, and combining all these aspects together to determine an optimal strategy.

A. Game Playing Agents

Many algorithms have been developed for playing specific games with the goal of being able to outperform human experts, achieving super-human performance. This includes many board or card games such as Chess [3], Go [4], Shogi [5], Checkers [6], Backgammon [7], Hanabi [8] and Poker [9], as well as popular video games such as Super Mario Bros. [10], Starcraft [11], Angry Birds [12], Ms. PacMan [13], Doom [14] and Unreal Tournament [15]. Many of these games are also associated with competitions that hope to provide a benchmark comparison between the best approaches currently being developed [16]. Such competitions are a great way to encourage collaborations and friendly rivalries between different teams and AI techniques.

Apart from being highly enjoyable to watch, developing agents that can play traditional or modern games effectively has many immediate benefits and real-world applications [1].

Research on games has led to new and improved AI techniques that are applicable across multiple scenarios. Examples include the many recent advances in Deep Learning and Monte Carlo Tree Search (MCTS) promoted by agents such as AlphaGo and AlphaGo Zero for the game of Go [17], [18], the use of counterfactual regret minimization and subgame abstraction for playing both limit and no-limit Texas hold'em poker respectively [19], [20], and the use of parallelized Alpha-Beta pruning for playing Chess [21]. Such research can also be applied to many non-game related problems [22], validating the use of games as a well defined and safe benchmark task for evaluating new AI techniques.

B. General Game AI

While many of these game specific agents and techniques can be very effective for their intended domain, developing agents that are only capable of playing a specific or limited subset of games has its downsides. Many of these game playing algorithms tend to over-specialise on the game they are designed for, as domain specific knowledge and highly tailored heuristics can often give a substantial performance advantage [23], [24]. Because of this, such agents are typically unable to transfer their abilities between different games. A world class poker agent is unlikely to be able to play Chess or Go with anywhere near the same level of skill that it previously exhibited. This is a significant limitation of current AI agents compared to most human players, as many of us are likely able to play a wide range of games and can learn new ones relatively quickly. The field of general game playing attempts to address this problem by requiring developed agents to work well across many different, and sometimes unknown, game possibilities. This has become one of the grand challenges of AI: Creating general intelligence that is capable of solving many different tasks, or in the case of general game playing, capable of playing many different games.

Apart from simply playing a given game as well as possible, many alternative AI problems focused around general games also exist.

1) *Human-like AI*: Instead of optimal performance, some researchers try to develop agents that achieve human-like game playing abilities [25]. These agents try to mimic the playing style of human players, who often are not as adept as AI in some game aspects, such as checking many possible scenarios or performing complex mathematical calculations, but are likely to be better or equal in other areas. Such agents

typically require fewer computational resources than those needed to achieve super-human performance, and would likely be much better at evaluating how enjoyable or well-balanced a particular game might be for a human player [26].

2) *Procedural Content Generation*: Another popular area of general game research is that of procedural content generation (PCG). Most PCG algorithms are developed specifically for generating a particular game element (level, weapon, vehicle, objective, etc.) for a specific type of game [27]–[29]. General PCG takes these same concepts but applies them to a much wider range of applicable games types. This can also include generating entirely new games from scratch, or generating certain game aspects given other pre-set requirements. Developing level or rule generators that can successfully create well-designed and enjoyable content for a wide range of different games creates many new difficulties beyond that of standard PCG [30], [31]. General game evaluation metrics should also be able to assess the quality of multiple games for a variety of different factors, such as enjoyment, content variety, strategic depth, expected playtime, etc.

C. General Game Platforms

Several research platforms currently exist for promoting the development of general AI for games [32], [33]. The most notable examples of this include the General Game Playing (GGP) system [34], which focuses primarily on board and abstract games, the General Video Game AI (GVGAI) framework [35], which focuses on simple video games, and the Arcade Learning Environment [36], [37], which allows agents to play a large number of Atari 2600 games. Both the GGP and GVGAI platforms also have competitions associated with them that promote and compare work done using their respective systems, and which are discussed in greater detail in the following section.

Ludii is an upcoming ludemic general game system that hopes to provide a large number of exciting opportunities for general game AI research [38]. Ludii has so far been shown to possess many improvements in efficiency, simplicity and generality beyond those of other prior general game systems, and the ludemic approach it takes for describing games allows for many new research possibilities. We describe in this paper some of the competitions that we intend to organise and run using it. While some of these described competitions overlap the structures and goals of existing alternatives, we believe that the benefits of our system, combined with the novel and varied ideas we present, will make Ludii one of the most attractive and popular competition platforms in future years.

D. Paper Overview

The remainder of this paper is organised as follows: Section II describes some existing general game competitions, the different tracks they offer, as well as their respective strengths and weaknesses. Section III describes our new proposed Ludii system in greater detail, highlighting some of its most important and appealing aspects that make it suitable for a wide range of competitions. Section IV describes some of the

agent-based competitions that we intend to run with Ludii. Section V describes some alternative PCG-based competitions that can either run separately or alongside the agent-based competitions. Section VI provides some possible competitions which do not strictly feature agents or PCG, instead delving into other areas of AI research such as data mining, player analysis, and information visualisation. Section VII finishes by concluding and summarising the ideas presented within this paper, as well as addressing any potential challenges or shortcomings that could occur.

II. EXISTING GENERAL GAME COMPETITIONS

The main competitions around developing general game AI are the International General Game Playing (IGGP) Competition and the General Video game AI (GVGAI) Competition. While some other general game competitions exist, such as the Tiltyard Open competition,¹ these two are easily the most popular and well-documented examples. In this section we discuss both of these competitions, describing the types of games they focus on, how the competitions are structured, and the different tracks they offer.

A. IGGP Competition

The IGGP competition was one of the first general game playing competitions, organised by the Logic group of the Stanford University in 2005 [39]. This competition is hosted annually at the AAAI conference, and centers around the creation of agents that can play games defined using the Game Description Language (GDL) [40]. Games written in GDL are described in terms of simple instructions based on first-order logic clauses. This restricts the types of games that can be written in the base GDL language to deterministic games with perfect information. While extensions to this language have been proposed that allow for imperfect information and epistemic games [41], [42], these are currently not included as part of the base GDL game repository² and are not used in the IGGP competition. This repository is typically updated with a few extra games each year.

Compared to other alternative languages, GDL provides a low-level description of games. Even the simplest of games can take hundreds or thousands of lines to define, and making small modifications to existing games, such as changing the size or shape of the board, often requires many lines of code to be changed or added. Presenting only the underlying logic for the game, rather than encapsulating common game concepts within more human-understandable terms, makes game descriptions difficult for humans to understand. Processing these descriptions can also be computationally expensive due to the logic resolution required, and GDL agents must construct their own internal representation of each game's mechanics in order to play effectively. Many medium to high complexity games either require a very large amount of time to model (e.g. Go) or are rendered unplayable due to the large computational costs involved (e.g. Chess). Even more complex board games

¹<http://tiltyard.ggp.org/>

²GGPBASE: <https://github.com/ggp-org/ggp-base>

such as Arimaa [43], which was designed specifically to be difficult for brute force tree search algorithms, would be almost impossible to describe in GDL, likely requiring thousands of lines of code.

The IGGP competition has a single track, that focuses exclusively on playing games. Agents submitted to the competition are evaluated on a pre-determined collection of several previously unseen GDL games. Most of the games written in GDL, as well as the majority of games used in the competition, tend to be variants of well known board or puzzle games. Competition entrants do not know which types of games their agent will be required to play beforehand. During the competition, each participating agent initially receives the GDL description of each game it is required to play, often between 30 and 120 seconds beforehand, and is typically given between 5 and 30 seconds of decision time for each move that it makes. The competition consists of an initial preliminary round that is open to everyone, after which the top 8 teams proceed to a series of one on one playoff matches to determine the eventual winner. Each playoff match typically consists of three different games, with the best performing agent across all games winning the match. More details on the competition specifics, including how the playoff rounds are structured, how performance on each game is measured, and other rule technicalities, can be found here [34], [39].

B. GVGAi Competition

The GVGAi competition is another general game playing competition, organised by the Game Intelligence Group of the University of Essex in 2014 [44], although many of the ideas and framework behind the competition were first published the previous year [45]. While this competition was originally only concerned with game playing, additional tracks that instead focus on rule and level generation have also been run. Games used in this competition are defined using the Video Game Description Language (VGDL), an alternative to GDL that allows for the creation of 2D arcade-style video games. This language has been significantly extended over the years to allow for new types of video games, such as the addition of a rudimentary physics engine in 2017 [46]. Several new games are often added each year to the official GVGAi game collection that is included with the available system code,³ but many additional games have been created by other research groups to investigate certain specific hypotheses [47].

VGDL is designed primarily for describing real-time games that can contain controllable avatars, stochastic effects and hidden information [48]. While GDL defines games using logical rules, VGDL describes a game based on the entities and interactions that can occur within it [35]. The dynamics of each component's movement, behaviour and abilities is based on its defined type, which are programmed separately into the GVGAi framework. If a new game requires a component that functions differently to those previously defined, then this must first be added into the system itself. While this approach

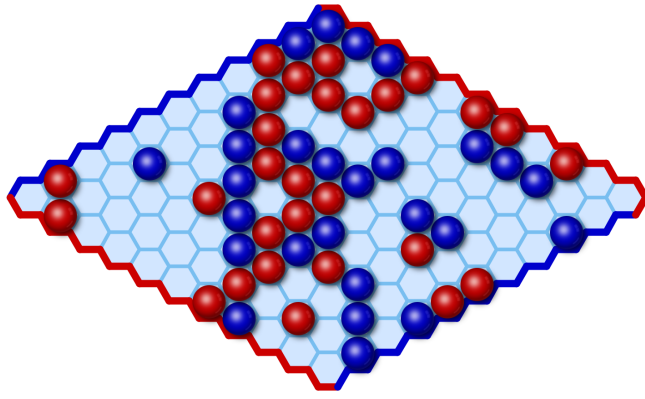
abstracts out some of the core logic and reasoning from each game's description, the result is a language that is much more flexible and which can ultimately produce a greater set of possible games. Game descriptions defined in VGDL are also typically much shorter and more human-understandable than those in GDL. Each game is also accompanied by one or more level descriptions, that defines how the specified VGDL components should be initially arranged.

The GVGAi competition currently has five distinct tracks, three of which focus on game playing and the other two on content generation. For game playing, there is the single-player planning track, the two-player planning track, and the single-player learning track. The two-player track involves games that can be played by two separate agents, and can be either cooperative or competitive in nature. In the planning tracks, future game states can be simulated to help an agent plan its next actions. The learning track removes this forward model, requiring agents to determine themselves how certain components behave and function. During the competition, agents do not have access to the VGDL descriptions for each game, and receive information only about the current game state. The agent must respond with a discrete action every 40ms, after which the next game state is computed. In addition to these, tracks focusing on level generation and rule generation are also available. In the level generation track, submitted generators must create enjoyable levels for a provided VGDL game description. In the rule generation track, submitted entries must create the rules for a given game, given the available components and level description. More information about the GVGAi competition and each of its available tracks can be found here [33], [44], [49]–[52].

III. THE LUDII SYSTEM

Games can often be broken down into conceptual units of game-related information, known as ludemes [53]. Describing games in this manner is referred to as the ludemic representation. An important benefit of this representation is that it encapsulates key game concepts and gives them meaningful labels, which allows them to be expressed in a compact and human-understandable manner. Ludii is a new and complete general game system [54], that can model games as a collection of ludemes using a class grammar approach [55]. This approach allows the game description language to be directly derived from the library class hierarchy, giving a guaranteed 1:1 mapping between the source code and the grammar. Ludii effectively makes its programming language (Java) the game description language, and as a result, it can theoretically support any rule, equipment or behaviour that can be programmed in Java. The exact details of this implementation are hidden from the user, who only deals with the simplified grammar which summarises the code to be called. Games defined using this language can be described or modified in very few lines of code, providing many possibilities for procedural game generation algorithms. As an example, Figure 1 shows the Ludii visuals and ludemic description for the board game Hex, with additional examples available here [56]–[58]

³GVGAi Software: <http://www.gvgai.net/software.php>



```
(game "Hex"
  (mode 2 (addToEmpty))
  (equipment {
    (HexBoard 11)
    (ball Each)
    (region P1 (edge NE)) (region P1 (edge SW))
    (region P2 (edge NW)) (region P2 (edge SE))
  })
  (rules
    (play (to (empty)))
    (end
      (connect (mover))
      (result (mover) Win)
    )
  )
)
```

Fig. 1: A completed game of Hex, played out on the Ludii system, along with the its ludeme-based game description.

The number and variety of games that can and will be implemented within Ludii also goes far beyond those presented by any previous general game system. Ludii can successfully model the full range of GDL games, as well as stacking games, boardless games and even games with hidden information, along with many other possibilities. This, coupled with the high efficiency achieved by Ludii compared to other general systems [56], makes it one of the largest and most research friendly collections of benchmark games for agents.

In the following sections we propose several future competitions and tracks that we intend to organise and run using the Ludii system. Many of these competition ideas could also be open for humans participants as well as AI. While we intend to run some human exclusive competitions, for the sake of brevity in this paper, we concentrate primarily on competitions that involve the submission of an AI-based agent or content generator.

IV. AGENT-BASED COMPETITIONS

This section describes competitions that focus on playing games. Submitted entries to these competitions must be able to play any game defined within the Ludii system without any external assistance. Rather than describing each of the separate tracks that could be run, we instead describe the different competition requirements and frameworks that may be possible. We expect to run a large number of tracks, each of which constitutes a unique combination of these properties. For example, one track might focus on a particular type of game, where agents are given certain information, and where the competition is structured in a certain format. Another track might change one or many of these competition aspects, giving an entirely new challenge. It is also possible for a competition track to feature several options for a single property, or to randomly select the properties for each individual match.

A. Game Source

A large number of Ludii games, defined using Ludii game description language, will be publicly available on the official

Ludii repository. The games used in any competition may be sourced from this or may be created especially for it.

1) *Known Games*: The evaluation process will be conducted using a set of games from the official Ludii repository. Participants will know before the competition begins which games will be used during it.

2) *Unknown Games*: Same as above, but competition participants do not know the exact games that will be used from the Ludii repository.

3) *New Games*: Entirely new games or game variants are used to evaluate agents. Participants will not have access to these games before the competition.

B. Types of Games

Games can often fit into different families or categories, and agents may be required to play games from some of these.

1) *No holds barred*: Any type of strategic game is possible.

2) *Specific Games*: All games will belong to one, or maybe several, game categories. The exact game classification scheme is yet to be decided, but likely categories could include:

- Deterministic / Stochastic Games
- Complete / Hidden Information Games
- Board / Card / Dice / Tile Games
- Boardless Games (e.g. Dominoes)
- Stacking Games (e.g. Lasca)
- Simultaneous Move Games (e.g. Chinook)
- Graph Games (e.g. Dots and Boxes)
- Team Games (discussed further in the next section)

C. Number of Agents

Games can also vary in terms of the number and dynamics of the players involved.

1) *Multi-player competitive games*: The competition uses games that involve two or more agents competing against each other. The goal is to win more games than your opponent(s), or to achieve the highest average ranking across multiple games. This can also include team games that involve several groups of agents competing against each other.

2) *Multi-player cooperative games*: The competition uses games that involve two or more agents cooperating with each other to achieve some joint objective. Agents may also have their own individual goals that they are also trying to complete. Points are awarded to each agent based on the number of completed objectives, over several different games and team combinations.

3) *Single-player puzzles*: The competition uses games / puzzles that involve only one player. The goal is either to be the first agent to solve the puzzle, or to find the shortest possible solution in a given time (if the puzzle has multiple solutions).

D. Information Provided

Agents may be provided with certain information about the games they are required to play.

1) *Forward model*: Agents have access to the forward model for each game. This forward model can be used to evaluate future states. Random game elements within these forward models can be simulated automatically or specified by the agent to achieve a desired outcome.

2) *Game Description*: Agents have access to the Ludii game descriptions. Agents must infer each game's moves, rules and victory conditions from this.

3) *Going in Blind*: Agents do not have access to either the game description or forward model, and must infer them as they play.

E. Learning Time

While all public games on the official Ludii repository will be available to all participants, if the games used to evaluate the agents are not known or available beforehand, then a set amount of time may be allocated for the agents to learn about the games they will be required to play. This is independent of whether the agents have access to the game description, forward model, or neither.

1) *No Learning Time*: Agents do not get any time to train on the games before the evaluation phase.

2) *Learning Time for the same games*: Agents have access to the evaluation games for a short period of time (training phase) before they are evaluated. The exact length of time that agents will have access to these games can vary considerably, from a few minutes, to several hours, or even a couple of days.

3) *Learning Time for different games*: Agents will be provided with a training phase before the evaluation phase, but the games available in each phase will be different in some way. Games provided during the training phase will likely be minor variants of those used in the evaluation phase, (e.g. differed board size/shape, different starting positions, etc.). Agents will need to use transfer learning capabilities to generalise the information obtained from the training phase into the evaluation phase.

F. Competition format

The first two competition formats we propose here are short-running and intended for one off events held over a few hours

or days. The second two formats are long-running and are expected to take place over many weeks or months. We intend to run both annual competitions, and continuous rankings that are updated all year round. Note that for cooperative games or multi-player games with more than two agents, additional scoring considerations will likely need to be made.

1) *Elimination*: Agents play against each other in a series of knockout style rounds. Each round of the competition selects a new set of games that all remaining agents are then evaluated using. The best performing agents from each round then proceed to the next, until only one agent remains. If the initial number of agents is very high, then a benchmark set of games may be used to select a smaller number of best performing agents that then proceed to the main tournament rounds.

2) *Round-Robin*: Each agent plays against all other agents on a single set of games. Points are awarded for each match-up based on the "Three points for a win" system (win = 3 points, draw = 1 point, loss = 0 points). The final point ranking may be used to calculate the winner of the competition, or to determine which agents qualify for a separate elimination format tournament.

3) *Leaderboard*: Agents can be submitted to a centralised server that keeps a continuously updated ranking of each agent's average performance across all games in the public Ludii repository. Individual performances are recorded for each game, as well as for each game category and across the entire game set. New agents or versions can be submitted at any time, to compare their performance against other existing agents. Humans are also able to be ranked on this leaderboard alongside these agents, as a benchmark measure of human vs AI ability. Performance values for each agent will likely be based on their percentage number of wins. Several rating systems for this are possible, such as Glicko-2 which is used by the GVGAI two-player track or AGON which is used for GGP.

4) *League*: Agents can be entered to compete in a league that runs over a certain period of time, likely a few months. This league operates similar to those of most sports, where all entrants play against each other over a series of matches. This would be conducted in a manner very similar to that of the described Round-Robin format, but where a greater number of games are played less often and over a longer period of time. Development teams might also be able to adjust their agent between games, to fix issues or tailor strategies towards an upcoming opponent. New games are expected to be played each week and will be live streamed online for people to watch. Some of the best performing human players may also be invited to take part in these competitions, to provide a comparison with human capabilities.

V. PCG-BASED COMPETITIONS

This section describes competitions that focus on generating games, or specific aspects of games. Similar to the previous section, we describe several different options for certain competition properties. Several adjustable properties of PCG-based

competitions, specifically “Types of Games” (under which the intended “Number of Agents” could also be included), “Information Provided” and “Competition Format”, include the same possibilities as agent-based competitions, and are therefore not repeated.

A. Generatable Content

What content the generator is expected to create. Many different game-related aspects can be designed automatically by content generators, spanning from a complete game, including rules, board, pieces, etc., to certain game elements, and even material for indirectly enhancing the game experience.

1) *Games*: Generators must create a complete and valid game. The generated game should be described using the Ludii game description language, and must compile successfully on the Ludii system.

2) *Puzzles*: Generators are given the Ludii description for a puzzle and must generate an initial state. In other words, the generator must create one or more puzzles for a given set of rules (e.g. Sudoku or Minesweeper). This idea of defining the initial setup for a set of game rules could also apply to multi-player games, but is much more applicable to single-player games and puzzles.

3) *Rules*: Generators are given the equipment that is available for a game (board, pieces, dice, cards, etc.) and must generate a viable ruleset, including game setup, turn process and victory conditions. Generators may be required to use all of the equipment available, or can ignore some if they wish.

4) *Instructions / Tutorials*: Generators are given the Ludii descriptions for a set of games, and must translate each of these into a plaintext English description. This may be extended to other languages as well, if there is sufficient demand. Similar work on this has been done for VGDL descriptions [59], but to our knowledge no competition around this has yet been run. This can also be extended to the generation of full tutorials, with accompanying pictures, videos or game scenarios that demonstrate key rules.

B. Generation Requirements

Requirements and goals for the content generated.

1) *Free Form*: Generators have no, or very few, requirements on what can be generated.

2) *Specified ludemes*: Generated games or rules must include a certain set of ludemes. Predefined groupings of ludemes (ludemeplexes), that define aspects such as the game board, specific rules, victory conditions, etc., may also be specified as a generation requirement.

3) *Partial Evidence*: This requirement is similar to that of specifying ludemes, but is based on actual archaeological and anthropological evidence. Generators will be provided with a description of the equipment and rule information that is known for a game, along with the historical and cultural background surrounding it, and must generate a possible game using this information.

4) *Deceptive Games*: The term deceptive games has been coined before in relation to several types of games, and can be used to refer to any games that exploit biases and weaknesses within current agent techniques [60], [61]. Several previous games, such as Arimaa, have already been designed with the goal of being comparatively difficult for AI compared to humans. We propose a competition centered around this, where generators try to design games that current agents perform poorly on, relative to the ability of human players (and vice versa). If successful, such games may help us to better understand the limitations of current AI techniques for general game playing.

C. Evaluation Procedure

How the generated content should be compared and ranked.

1) *Judging Panel*: Generated content is evaluated by a panel of human judges. This can be accompanied by computational testing to remove impossible or infeasible possibilities prior to human evaluation. Each judge will be presented with all or some of the generated content, and will be asked to rank them based on some desired metrics. This could include overall enjoyment, novelty, difficulty, complexity, historical accuracy, etc., depending on the competition specifications.

2) *Agent Evaluation*: Generated content is evaluated by a collection of agents. This is unlikely to be effective on its own, but can be combined with human evaluation to provide a more detailed evaluation profile. Agents and other AI techniques can also evaluate games in terms of some defined quality metrics, such as fairness, game length, strategic depth, rule complexity, drawishness, etc.

3) *Public Vote*: Generated content is rated by members of the general public. The exact format for this evaluation, online poll, live voting, user survey, etc., is still to be decided. Like other evaluation procedures, this can be combined with agent-based evaluation and expert judges.

VI. OTHER COMPETITIONS

This section describes competitions that focus on topics related to both or neither of the previous two sections.

A. Turing Test

Turing test tracks could be run for both the Agent and PCG based competitions. For the Agent-based competitions, agents would need to play in a human-like and believable manner. Human judges would then play a number of games against each opponent, and attempt to determine if the adversary is an agent or human. The same principle can also apply for PCG-based competitions, where human judges would guess if some content was created by a human or generator. A competition of this nature was held previously as part of the Mario AI competition [10] and some preliminary research has been done for GVGA games [62].

B. Generate and Play

Competitions that combine Agents and PCG could be also held. Each competitor (human or AI) could provide or generate a Ludii game, which are then combined to make up the set of agent evaluation games. Generators or human designers would need to create games that would play to the strengths of their own agent, but which other agents would likely perform worse on.

C. Stab in the Dark

A designer presents a game that human participants must then play without seeing the rules. These participants play against an AI to learn the game, and try to beat it in the smallest amount of time or number of moves. Participants may be told some small amount of information about the game, such as the possible movements of a selected piece, or may be told nothing at all. The designer who produces the simplest game, which uses the fewest ludemes or has the smallest state/action space, that confuses the most people for the longest time wins.

D. Needle in a Haystack

Participants submit a game evaluation system that is then given 1,000 to 1,000,000 randomly generated games or rule sets to test. These evaluation systems are then given a certain period of time to rank these games and to select those which it believes are best, according to their own criteria. Expert judges or members of the general public will then play and rate the best games selected by each agent, to decide the competition winner.

E. QR-code design

The file size for a game defined in Ludii is so small, that each game description can likely fit onto a single QR code. Inspired by similar work on designing custom QR codes artwork,⁴ we propose a competition around designing the most visually appealing QR code for a playable Ludii game (e.g. Chess with a picture of a piece, or Hnefatafl with an image of a viking).

VII. CONCLUSIONS AND CHALLENGES

In this paper we have described the current area of General Game Playing, and presented Ludii as an alternative system with many new ideas for game AI research and future competitions. Ludii provides many interesting problems and challenges for game researchers. Ludii offers a sophisticated system and language, as well as a comprehensive database of traditional strategy games. Ludii is also efficient, understandable, modifiable, and convenient to work with. It provides an ideal platform for playing, generating and testing games, that is fully compatible with many AI techniques. Apart from a purely academic platform, Ludii has a lot of potential for being an important game design tool that can assist human game designers with the creation, analysis and play testing of their games

With such a large range of potential competitions it is important to focus on those that will have the largest development and research base, and which can also provide the most interesting results for AI research. It will therefore be impossible to cover every possible competition possibility, but we hope to identify the most popular and promising tracks over the next few months. Other challenges include internally developing and testing the large number of game implementation needed. While it will be possible for anyone to create their own games using Ludii for testing purposes, the official Ludii game repository must be subject to a rigorous quality standard. Running such a large number of competitions will require substantial hardware and organising time, likely resulting in new competitions being slowly rolled out over the next few years rather than all at once.

Future work over the next few years will primarily involve developing more games and additional functionality for the Ludii system, as well as organising and running a large number of the competitions proposed in this paper. As well as traditional strategy games, we hope to develop many other types of games for Ludii. This may include many modern board game ideas, simple video games and serious games. Looking even further into the future, we anticipate branching out into other non-game domains and applying Ludii to problems such as protein folding, physics simulations or predicting chemical reactions.

ACKNOWLEDGMENT.

This research is part of the European Research Council-funded Digital Ludeme Project (ERC Consolidator Grant #771292) run by Cameron Browne at Maastricht University's Department of Data Science and Knowledge Engineering.

REFERENCES

- [1] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018, <http://gameaibook.org>.
- [2] A. M. Turing, "Digital computers applied to games," *Faster than thought* (ed. B.V. Bowden), pp. 286–297, 1953.
- [3] T. Romstad, M. Costalba, J. Kiiski, and G. Linscott, "Stockfish: Strong open source chess engine," <https://stockfishchess.org/>, 2008.
- [4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [5] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.
- [6] J. P. Patist and M. Wiering, "Learning to play draughts using temporal difference learning with neural networks and databases," in *Proc. 13th Belgian-Dutch Conf. Mach. Learn.*, 2004, pp. 87–94.
- [7] G. Tesauro, "Temporal difference learning and td-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, 1995.
- [8] M. Eger, C. Martens, and M. A. Cordoba, "An intentional AI for hanabi," in *2017 IEEE Conf. Comput. Intell. Games*, 2017, pp. 68–75.
- [9] "Computer poker: A review," *Artif. Intell.*, vol. 175, no. 5, pp. 958–987, 2011.
- [10] S. Karakovskiy and J. Togelius, "The mario AI benchmark and competitions," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 55–67, 2012.

⁴<https://research.swtch.com/qr/draw>

- [11] S. Ontan, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game AI research and competition in starcraft," *IEEE Trans. Comput. Intell. AI Games*, vol. 5, no. 4, pp. 293–311, 2013.
- [12] M. Stephenson, J. Renz, X. Ge, and P. Zhang, "The 2017 AIBIRDS competition," *CoRR*, vol. abs/1803.05156, 2018.
- [13] P. Rohlfshagen and S. Lucas, "Ms pac-man versus ghost team CEC 2011 competition," in *2011 IEEE Congr. Evol. Computation*, 2011, pp. 70–77.
- [14] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, "Vizdoom: A doom-based AI research platform for visual reinforcement learning," *CoRR*, vol. abs/1605.02097, 2016.
- [15] S. Jacobs, A. Ferrein, and G. Lakemeyer, "Controlling unreal tournament 2004 bots with the logic-action language golog," in *Proc. 1st AAAI Conf. Artif. Intell. Interactive Digital Entertainment*, ser. AIIDE'05. AAAI Press, 2010, pp. 151–152.
- [16] J. Togelius, "How to run a successful game-based AI competition," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 1, pp. 95–100, 2016.
- [17] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.
- [18] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.
- [19] M. Bowling, N. Burch, M. Johanson, and O. Tammelin, "Heads-up limit hold'em poker is solved," *Science*, vol. 347, no. 6218, pp. 145–149, 2015.
- [20] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2018.
- [21] S. D. G. V. Akanksha Kumari, Shreya Singh, "Parallelization of alpha-beta pruning algorithm for enhancing the two player games," *Int. J. Advances Electronics Comput. Sci.*, vol. 4, pp. 74–81, 2017.
- [22] J. Togelius, "AI researchers, video games are your friends!" in *7th Int. Joint Conf. Comput. Intell. (IJCCI)*, vol. 2, 2015, pp. 5–5.
- [23] A. Silva and M. C. Gombolay, "Protonets: Neural-encoding human experts' domain knowledge to warm start reinforcement learning," *CoRR*, vol. abs/1902.06007, 2019.
- [24] A. Stiegler, K. P. Dahal, J. Maucher, and D. Livingstone, "Symbolic reasoning for hearthstone," *IEEE Trans. Games*, vol. 10, no. 2, pp. 113–127, 2018.
- [25] J. Mandziuk, "Towards cognitively plausible game playing systems," *IEEE Comput. Intell. Magazine*, vol. 6, no. 2, pp. 38–51, 2011.
- [26] J. Togelius, R. De Nardi, and S. M. Lucas, "Towards automatic personalised content creation for racing games," in *2007 IEEE Symp. Comput. Intell. Games*, 2007, pp. 252–259.
- [27] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.
- [28] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, pp. 1:1–1:22, 2013.
- [29] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 172–186, 2011.
- [30] X. Neufeld, S. Mostaghim, and D. Perez-Liebana, "Procedural level generation with answer set programming for general video game playing," in *7th Comput. Sci. Electronic Eng. Conf.*, 2015, pp. 207–212.
- [31] T. S. Nielsen, G. A. B. Barros, J. Togelius, and M. J. Nelson, "Towards generating arcade game rules with VGDL," in *2015 IEEE Conf. Comput. Intell. Games*, 2015, pp. 185–192.
- [32] M. Świechowski, H. Park, J. Mańdziuk, and K.-J. Kim, "Recent advances in general game playing," *The Scientific World Journal*, 2015.
- [33] D. Perez, J. Liu, A. Abdel Samea Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, "General video game AI: a multi-track framework for evaluating agents, games and content generation algorithms," *IEEE Trans. Games*, 2019, in press.
- [34] M. R. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the AAAI competition," *AI Magazine*, vol. 26, no. 2, pp. 62–72, 2005.
- [35] D. Perez-Liebana, S. Samothrakis, J. Togelius, S. M. Lucas, and T. Schaul, "General video game ai: Competition, challenges, and opportunities," in *Proc. 30th AAAI Conf. Artif. Intell.*, ser. AAAI'16, 2016, pp. 4335–4337.
- [36] 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, no. 1, pp. 253–279, 2013.
- [37] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling, "Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents," *J. Artif. Intell. Res.*, vol. 61, pp. 523–562, 2017.
- [38] C. Browne, "Modern techniques for ancient games," in *IEEE Conf. Comput. Intell. Games*, 2018, pp. 490–497.
- [39] M. R. Genesereth and Y. Björnsson, "The international general game playing competition," *AI Magazine*, vol. 34, pp. 107–111, 2013.
- [40] N. Love, T. Hinrichs, D. Haley, E. Schkufza, and M. Genesereth, "General game playing: Game description language specification," 2008.
- [41] S. Schiffel and M. Thielscher, "Representing and reasoning about the rules of general games with imperfect information," *J. Artif. Intell. Res.*, vol. 49, pp. 171–206, 2014.
- [42] M. Thielscher, "GDL-III: A description language for epistemic general game playing," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1276–1282.
- [43] O. Syed and A. Syed, "Arimaa - a new game designed to be difficult for computers," *ICGA Journal*, vol. 26, pp. 138–139, 2003.
- [44] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, S. M. Lucas, A. Couëtoux, J. Lee, C. Lim, and T. Thompson, "The 2014 general video game playing competition," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 3, pp. 229–243, 2016.
- [45] J. Levine, C. B. Congdon, M. Ebner, G. Kendall, S. M. Lucas, R. Mikkilainen, T. Schaul, and T. Thompson, "General Video Game Playing," in *Artif. Comput. Intell. Games*, ser. Dagstuhl Follow-Ups, 2013, vol. 6, pp. 77–83.
- [46] D. Perez-Liebana, M. Stephenson, R. D. Gaina, J. Renz, and S. M. Lucas, "Introducing real world physics and macro-actions to general video game AI," in *2017 IEEE Conf. Comput. Intell. Games*, 2017, pp. 248–255.
- [47] D. Anderson, M. Stephenson, J. Togelius, C. Salge, J. Levine, and J. Renz, "Deceptive games," in *21st Int. Conf. Applications Evol. Computation*, ser. EvoStar'18, 2018, pp. 376–391.
- [48] T. Schaul, "A Video Game Description Language for Model-based or Interactive Learning," in *Proc. IEEE Conf. Comput. Intell. Games*. IEEE, 2013, pp. 193–200.
- [49] R. D. Gaina, D. Prez-Libana, and S. M. Lucas, "General video game for 2 players: Framework and competition," in *2016 8th Comput. Sci. Electronic Eng.*, 2016, pp. 186–191.
- [50] R. D. Gaina, A. Couëtoux, D. J. N. J. Soemers, M. H. M. Winands, T. Vodopivec, F. Kirchgeßner, J. Liu, S. M. Lucas, and D. Perez-Liebana, "The 2016 two-player gvgai competition," *IEEE Trans. Games*, vol. 10, no. 2, pp. 209–220, 2018.
- [51] A. Khalifa, D. Perez-Liebana, S. M. Lucas, and J. Togelius, "General video game level generation," in *Proc. Genetic Evol. Computation Conf.*, 2016, pp. 253–259.
- [52] A. Khalifa, M. C. Green, D. Perez-Liebana, and J. Togelius, "General video game rule generation," in *2017 IEEE Conf. Comput. Intell. Games*, 2017, pp. 170–177.
- [53] D. Parlett, "What's a ludeme?" in *Game Puzzle Design*, vol. 2, 2016, pp. 83–86.
- [54] C. Browne, J. Togelius, and N. Sturtevant, "Guest editorial: General games," *IEEE Trans. Comput. Intell. AI Games*, vol. 6, no. 4, pp. 317–319, 2014.
- [55] C. B. Browne, "A class grammar for general games," in *Advances in Computer Games*, 2016, pp. 167–182.
- [56] 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," *CoRR*, vol. abs/1905.05013, 2019.
- [57] M. Stephenson, É. Piette, D. J. N. J. Soemers, and C. Browne, "An overview of the ludii general game system," in *IEEE Conference on Games*, 2019, p. to appear.
- [58] É. Piette, M. Stephenson, D. J. N. J. Soemers, and C. Browne, "An empirical evaluation of two general game systems: Ludii and rbg," in *IEEE Conference on Games*, 2019, p. to appear.
- [59] M. C. Green, A. Khalifa, G. A. B. Barros, T. Machado, A. Nealen, and J. Togelius, "Atdelfi: Automatically designing legible, full instructions

- for games,” in *Proc. 13th Int. Conf. Foundations Digital Games*, 2018, pp. 17:1–17:10.
- [60] D. Anderson, M. Stephenson, J. Togelius, C. Salge, J. Levine, and J. Renz, “Deceptive games,” in *21st Int. Conf. Applications Evol. Computation*, ser. EvoStar’18, 2018, pp. 376–391.
- [61] M. Stephenson and J. Renz, “Deceptive angry birds: Towards smarter game-playing agents,” in *Proc. 13th Int. Conf. Foundations Digital Games*, 2018, pp. 13:1–13:10.
- [62] J. Lehman and R. Miikkulainen, “General video game playing as a benchmark for human-competitive ai,” in *AAAI-15 Workshop on Beyond the Turing Test*, 2015.