General General Game AI

Julian Togelius
Tandon School of Engineering
New York University
New York, New York
julian@togelius.com

Georgios N. Yannakakis
Institute of Digital Games
University of Malta
Msida, Malta
georgios.yannakakis@um.edu.mt

Abstract—Arguably the grand goal of artificial intelligence research is to produce machines with general intelligence: the capacity to solve multiple problems, not just one. Artificial intelligence (AI) has investigated the general intelligence capacity of machines within the domain of games more than any other domain given the ideal properties of games for that purpose: controlled yet interesting and computationally hard problems. This line of research, however, has so far focused solely on one specific way of which intelligence can be applied to games: playing them. In this paper, we build on the general game-playing paradigm and expand it to cater for all core AI tasks within a game design process. That includes general player experience and behavior modeling, general non-player character behavior, general AI-assisted tools, general level generation and complete game generation. The new scope for *general* general game AI beyond game-playing broadens the applicability and capacity of AI algorithms and our understanding of intelligence as tested in a creative domain that interweaves problem solving, art, and engineering.

I. INTRODUCTION

By now, an active and healthy research community around computational and artificial intelligence (AI)[1] in games has existed for more than a decade — at least since the start of the IEEE Conference on Computational Intelligence and Games (CIG) and the Artificial Intelligence and Interactive Digital Entertainment (AIIDE) conference series in 2005. Before then, research has been ongoing about AI in board games since the dawn of automatic computing. Initially, most of the work published at IEEE CIG or AIIDE was concerned with learning to play a particular game as well as possible, or using search/planning algorithms to play a game as well as possible without learning. Gradually, a number of new applications for AI in games and for games in AI have come to complement the original focus on AI for playing games [1]. Papers on procedural content generation, player modeling, game data mining, human-like playing behavior, automatic game testing and so on have become commonplace within the community. There is also a recognition that all these research endeavors depend on each other [2]. Games appear to be an ideal domain for realizing several long-standing goals of AI

including affective computing [3], computational creativity [4] and ultimately *general* intelligence [5], [6].

However, almost all research projects in the game AI field are very *specific*. Most published papers describe a particular method — or a comparison of two or more methods — for performing a single task (playing, modeling, generating etc.) in a single game. This is problematic in several ways, both for the scientific value and for the practical applicability of the methods developed and studies made in the field. If an AI approach is only tested on a single task for a single game, how can we argue that is an advance in the scientific study of artificial intelligence? And how can we argue that it is a useful method for a game designer or developer, who is likely working on a completely different game than the method was tested on?

Within AI focused on playing games, we have seen the beginnings of a trend towards generality. The study of general artificial intelligence through games — general game playing — has seen a number of advancements in the last few years. Starting with the General Game Playing Competition, focusing on board games and similar discrete perfect information games, we now also have the Arcade Learning Environment and General Video Game AI Competition, which offer radically different takes on arcade video games. Advancements vary from the efforts to create game description languages suitable for describing games used for general game playing [7], [8], [9], [10] to the establishment of a set of general video game AI benchmarks [7], [11], [12] to the recent success of deep Q-learning in playing arcade games with human-level performance just by processing the screen's pixels [13].

While the general game playing is studied extensively and constitutes one of the key areas of game AI [2] we argue that the focus of generality solely with regards to the performance of game-playing agents is *very narrow* with respect to the spectrum of roles for general intelligence in games. The types of general intelligence required within game development include game and level design as well as player and experience modeling. Such skills touch upon a diverse set of cognitive and affective processes which have until now been ignored by general AI in games. For general game AI to be truly general, it needs to go beyond game playing while retaining the focus on addressing more than a single game or player.

In other words, we are arguing that we need to extend the generality of general game playing to all other ways in which

---

[1] In the article, we will mostly use the terms "AI in games" or "game AI" to refer to the whole research field, including the various techniques commonly thought of as computational intelligence, machine learning, deep learning etc. "AI" just rolls off the tongue more easily.

AI is (or can be) applied to games. More specifically we are arguing that the field should move towards methods, systems and studies that incorporate three different types of generality:

1) *Game* generality. We should develop AI methods that work with not just one game, but with any game (within a given range) that the method is applied to.

2) *Task* generality. We should develop methods that can do not only one task (playing, modeling, testing etc) but a number of different, related tasks.

3) *User/designer/player* generality. We should develop methods that can model, respond to and/or reproduce the very large variability among humans in design style, playing style, preferences and abilities.

We further argue that all of this generality can be embodied into the concept of *general game design*, which can be thought of as a final frontier of AI research within games. We assume that the challenge of bringing together different types of skillsets and forms of intelligence within autonomous designers of games not only can advance our knowledge about human intelligence but also advance the capacity of general artificial intelligence. The paper briefly reviews the state of the art within each of the major roles an AI can take within game development, broadly following the classification of [2]. In particular AI can (1) take a non-human-player (*non-player*) role and either play games (in lieu of a human) or control the behavior of non-player characters (see Section II), (2) *model player* behavior and experience (see Section III), (3) *generate* content such as levels or complete games (see Section IV), or 4) *assist* in the design process through both the modeling of users (designers) and the generation of appropriate content (see Section V). For each of these roles we argue for the need of generality and we propose ways that this can be achieved. We conclude with a discussion on how to nudge the research field towards addressing general problems and methods.

It is important to note that we are not arguing that more focused investigations into methods for single tasks in single games are useless; these are often important as proofs-of-concept or industrial applications and they will continue to be important in the future, but there will be increasing need to validate such case studies in a more general context. We are also not envisioning that everyone will suddenly start working on general methods. Rather, we are positing generalizations as a long-term goal for our entire research community.

II. GENERAL NON-PLAYERS

A large part of the research on AI for games is concerned with building AI (i.e a non-(human)player) for playing games, with or without a learning component. Historically, this has been the first and for a long time only approach to using AI in games. Even before the beginning of AI as a research field, algorithms were devised to play games effectively. For instance, Turing himself (re)invented the Minimax algorithm to play Chess even before he had a working digital computer [14].

For a long time, research on game-playing AI was focused on classic board games, and Chess was even seen as "the drosophila of AI" [15] — at least until we developed software capable of playing Chess better than humans, at which point Chess-playing AI somehow seemed a less urgent problem. The fact that Chess became a less relevant problem once humans had been beaten itself points to the need for focusing on more general problems. The software that first exhibited superhuman Chess capability, Deep Blue, consisted of a Minimax algorithm with numerous Chess-specific modifications and a very highly tuned board evaluation function; the software was useless for anything else than playing Chess [16]. This led commentators at the time to argue that Deep Blue was "not really AI" after all [17]. The same argument could be made about AlphaGo, the AI that finally conquered the classic board game *Go* [18].

Even nowadays a large part of game AI research focuses on developing AI for playing games — either as effectively as possible, or in the style of humans (or a particular human), or with some other property [2]. Much of the research on playing videogames is organized around a number of competitions or common benchmarks. In particular, the IEEE CIG conference series hosts a respectable number of competitions, the majority of which focus on playing a particular game; popular competitions are built around games such as TORCS (a car racing game), *Super Mario Bros* (Nintendo, 1985), *Ms. Pac-Man* (Namco, 1982), *Unreal Tournament 2004* (Epic Games, 2004) and *StarCraft* (Blizzard Entertainment, 1998). These competitions are typically won by the submitted AI agent that plays the game best. What can be observed in several of these competitions, is that when the same competition is run multiple years there is indeed an improvement in performance, but not necessarily an improvement in the sophistication of the AI algorithms or their centrality to the submitted agent. In fact, the opposite trend can sometimes be discerned. For example, the simulated car racing competition started out with several competitors submitting car-driving agents which were to a large extent based on machine learning (e.g. neuroevolution). In subsequent years, however, these were outperformed by agents consisting of a large amount of hand-coded domain-specific rules, with any learning relegated to a supporting role [19], [20]. Similarly, the *StarCraft* competition has seen a number of AI-based agents performing moderately well, but the winner in several rounds of the competition consists almost entirely of hand-crafted strategies with almost no presence of what would normally be considered AI algorithms, and certainly no applicability outside *StarCraft* [21].

A. Gameplaying

Interestingly, the problem of playing games is the one that has been most generalized so far. There already exist at least three serious benchmarks or competitions attempting to pose the problem of playing games *in general*, each in its own imperfect way. The first of these is the General Game Playing Competition, often abbreviated GGP [7]. This competition has been running for more than ten years, and is based on a special-purpose game description language useful for encoding board game-like games, with discrete world state and in most cases perfect information. The submitted agents get access to the complete source code of the games they are tested on.

Every time the competition is run a few games are hand-crafted for testing new games. The Arcade Learning Environment (ALE) is instead built on an emulator of the Atari 2600 game console and includes a library of classic games [12]. Agents are only given a feed of the raw screen output. Compared to GGP, ALE has the advantage of using real video games, but the disadvantage that all games are known and creating new games requires very considerable effort. The General Video Game AI Competition (GVGAI) combines the focus on video games (from ALE) with the approach to build the competition games in a description language which allows new games to be created for each competition [9], [8], [11]. Currently, around 80 games are implemented, and for every competition 10 new games are implemented. Most games are adaptations of classic 80's arcade games, but the long-term goal is for new games to be generated automatically [22]. Agents are given access to a partial game state observation and a complete forward model.

The results from these competitions so far indicate that general purpose search and learning algorithms by far outperform more domain-specific solutions and "clever hacks". Somewhat simplified, we can say that variations of Monte Carlo Tree Search perform best on GVGAI and GGP, and for ALE (where no forward model is available so learning a policy for each game is necessary) reinforcement learning with deep networks performs best [13]. This is a very marked difference to the results of the game-specific competitions, which as discussed above tend to favor domain-specific solutions.

While these are each laudable initiatives and currently the focus of much research, in the future we will need to expand the scope of these competitions and benchmarks considerably, including expanding the range of games available to play and the conditions under which gameplay happens. We need game playing benchmarks and competitions capable of expressing any kind of game, including puzzle games, 2D arcade games, text adventures, 3D action-adventures and so on; this is the best way to test general AI capacities and reasoning skills. We also need a number of different ways of interfacing with these games — there is room both for benchmarks that give agents no information beyond the raw screen data but give them hours to learn how to play the game, and those that give agents access to a forward model and perhaps the game code itself, but expects them to play any game presented to them with no time to learn. These different modes test different AI capabilities and tend to privilege different types of algorithms. It is worth noting that the GVGAI competition is currently expanding to different types of playing modes, and has a long-term goal to include many more types of games [22].

We also need to differentiate away from just measuring how to play games optimally. In the past, several competitions have focused on agents that play games in a human-like manner; these competitions have been organized similarly to the classic Turing test [23], [24]. Playing games in a human-like manner is important for a number of reasons, such as being able to test levels and other game content as part of search-based generation, and to demonstrate new content to players. So far, the question of how to play games in a human-like manner *in general* is mostly unexplored; some preliminary work is reported in [25]. Making progress here will likely involve modeling how humans play games in general, including characteristics such as short-term memory, reaction time and perceptual capabilities, and then translating these characteristics to playing style in individual games.

B. Non-player Behavior

Many games have non-player characters (NPCs), and AI can help in making NPCs believable, human-like, social and expressive. Years of active research have been dedicated on this task within the fields of affective computing and virtual agents. The usual approach followed is the construction of top-down agent architectures that represent various cognitive, social, emotive and behavioral abilities. The focus has traditionally being on both the modeling of the agents behavior but also on its appropriate expression under particular contexts. A popular way for constructing a computational model of agent behavior is to base it on a theoretical cognitive model such as the Belief-Desire-Intention agent model [26] and the OCC model [27], [28], [29] which attempts to effect human-like decision making, appraisal and coping mechanisms dependent on a set of perceived stimuli. The use of such character models has been dominant in the domains of intelligent tutoring systems [30], embodied conversational agents [29], and affective agents [31] for educational and health purposes. Similar types of architectures for believable and social agents exist in games such as *Facade* [32], *Prom Week* [33], *World of Minds* [34] and *Crystal Island* [35]. Research in non-player character behavior in games is naturally interwoven with research in computational and interactive narrative [36], [32], [37], [38] and virtual cinematography [39], [40], [41].

One would expect that characters in games would be able to perform well under any context and game (seen or unseen) in similar ways humans do. Not only would that be a far more effective approach for agent modeling but it would also advance our understanding about general emotive, social and behavioral patterns. However, as with the other uses of AI in games, the construction of agent architectures for behavior modeling and expression is heavily dependent on particular game contexts and *specific* to (and optimized for) a particular game. While a number of studies within affective agents focus on domain-independent (general) emotive models [31] we are far from obtaining general, context-free, "plug-n-play" computational models for agents that are applicable across games, game genres and players. The vision here is that we could create general NPCs, that could easily be dropped into any given game and adapt (autonomously and/or with designer guidance) to the requirements of a particular game, so that they can behave believably and effectively in their new context.

Clearly there are general patterns of rational and (socially) believable behavior that can be detected across games and players. An NPC in *Prom Week*, for instance, should be able to transfer aspects of its social intelligence to e.g *Facade*; but how much of such patterns are relevant for a platformer of a first-person shooter NPC? This is an open research

question. To make a paradigm shift towards generality we argue that we need to focus on aspects of a top-down agent architecture that, by nature, are more general than others. Personality, for instance, can be modeled in an abstract, domain-independent way [42] while moods — compared to emotions — define longer lasting and less specific notions [43] that could be modeled in a context-independent way [28]. Emotions are more specific and domain-dependent aspects of a computational agent architecture, as they are heavily context-dependent. It has been suggested, however, that personality and emotion are heavily interlinked and only differentiated by time and duration since personality can be viewed as the seamless expression of emotion [44]. In that regard general emotive patterns across games and players can be identified if general context-independent features that characterize general behavior can be extracted and used for modeling NPC behavior; these include generic features such as winning, loosing, achieving rewards as well as progression or tension curves across games. These features can be either manually designed or machined learned from annotated player data.

III. GENERAL PLAYER EXPERIENCE MODELING

It stands to reason that general intelligence implies (and is tightly coupled with) general emotional intelligence [45]. The ability to recognize human behavior and emotion is a complex yet critical task for human communication. Throughout evolution, we have developed particular forms of advanced cognitive, emotive and social skills to address this challenge. Beyond these skills, we also have the capacity to detect affective patterns across people with different moods, cultural backgrounds and personalities. This generalization ability also extends, to a degree, across contexts and social settings.

Despite their importance, the characteristics of social intelligence have not yet been transferred to AI in the form of general emotive, cognitive or behavioral models. While research in affective computing [3] has reached important milestones such as the capacity for real-time emotion recognition [46] — which can be faster than humans under particular conditions — all key findings suggest that any success of affective computing is heavily dependent on the domain, the task at hand and the context in general. This *specificity* limitation is particularly evident in the domain of games [47] as most of work in modeling player experience focuses on particular games, under particular and controlled conditions within particular small sets of players (see [48], [49], [50], [51] among many).

For AI in games to be general beyond game-playing it needs to be able to recognize general emotional and cognitive-behavioral patterns. This is essentially AI that can detect context-free emotive and cognitive reactions and expressions across context and builds general computational models of human behavior and experience which are grounded in a general golden standard of human behavior. So far we have only seen a few proof-of-concept studies in this direction. Early work in the game AI field focused on the ad-hoc design of general metrics of player interest that were tested across different prey-predator games [52], [53]. A more recent example is the work

of Martinez et al. [54] in which physiological predictors of player experience were tested for their ability to capture player experience across two dissimilar games: a predator-prey game and a racing game. The findings of that study suggest that such features do exist. Shaker et al. [55] later used the same approach with different games.

Another study by Martinez et al. on deep multimodal fusion can be seen as an embryo for further research in this direction [51]. Various modalities of player input such as player metrics, skin conductance and heart activity, have been fused using deep architectures which were pretrained using autoencoders. Even though that study was rather specific to a particular game, its deep fusion methodology can be expanded across variant data corpora — such as the DEAP [56] and the platformer experience [57] datasets — and player metrics datasets that are openly available such as the game trace archive [58]. Discovering entirely new representations of player behavior and emotive manifestations across games, modalities of data, and player types is a first step towards achieving general player modeling. Such representations can, in turn, be used as the basis for deriving the *ground truth* of user experience in games.

IV. GENERAL CONTENT GENERATION

The study of procedural content generation (PCG) [59] for the design of game levels has reached a certain extent of maturity and is, by far, the most popular domain for the application of PCG algorithms and approaches (e.g. see [60], [2], [48] among many). In this section we first discuss this popular facet of computational game creativity and then connect it to the overall aim of general complete game generation.

A. Level Generation

Levels have been generated for various game genres such as dungeon-crawlers [61], [62], horror games [63], space-shooters [64], first-person shooters [65], [66], and platformers [49]. Arguably the platformer genre — through the Mario AI Framework, which builds on a clone of Super Mario Bros [67], [68] — can be characterized as the "drosophila of PCG research". A number of approaches such as constructive methods, search-based PCG [60], experience-driven PCG [48], solver-based PCG [69], data-driven PCG [70], [71], [72] or mixed-initiative PCG [73], [74] have been used for the creation of platformer levels in the Mario AI Framework with algorithms varying from simple multi-pass processes [75] to evolving grammars [76], exhaustive search on crowdsourced models of experience [77], and constraint solvers such as answer set programming [73]. What is common in all of the above studies is their *specificity* and strong dependency of the representation chosen onto the game genre examined. In particular for the Mario AI Framework, the focus on a single level generation problem has been very much a mixed blessing: it has allowed for the proliferation and simple comparison of multiple approaches to solving the same problem, but has also led to a clear overfitting of methods. Even though some limited generalization is expected within game levels of the

same genre the level generators that have been explored so far clearly do not have the capacity of *general* level design.

As with the other sub-tasks of game design discussed in this paper we argue that there needs to be a shift in how level generation is viewed. The obvious change of perspective is to create *general level generators* — level generators with general intelligence that can generate levels for any game (within a specified range). That would mean that levels are generated successfully across game genres and players and that the output of the generation process is a that is meaningful and playable well as entertaining for the player. Further, a general level generator should be able to coordinate the generative process with the other computational game designers who are responsible for the other parts of the game design.

To achieve general level design intelligence algorithms are required to capture as much of the level design space as possible at different representation resolutions. We can think of representation learning approaches such as deep autoencoders [78] capturing core elements of the level design space and fusing variant game genres within a sole representation — as already showcased by a few studies in the PCG area e.g.in [79]. A related effort is the Video Game Level Corpus [80] which aims to provide a set of game levels across multiple games and genres which can be used for training level generators for data-driven procedural content generation.

The first attempt to create a benchmark for general level generation has recently been launched in the form of the Level Generation Track of the GVGAI competition. In this competition track, competitors submit level generators capable of generating levels for unseen games. The generators are then supplied with the description of several games, and produce levels which are judged by human judges [81]. Initial results suggest that constructing competent level generators that can produce levels for any game is much more challenging than constructing competent level generators for a single game.

B. Game Generation

While level generation, as discussed above, is one of the main examples of procedural content generation, there are many other aspects (or "facets") of games that can be generated. These include visuals, such as textures and images; narrative, such as quests and backstories; audio, such as sound effects and music; and of course the generation of all kinds of things that go into game levels, such as items, weapons, enemies and personalities [82], [59]. However, an even greater challenge is the generation of complete games, including some or all of these facets together with the rules of the game.

There have been several attempts to generate games, including their rules. These include approaches based on artificial evolution [83], [84], [85], [86], [87], [88], attempts based on constraint satisfaction [89], [69], [90] and attempts based on matching pattern databases [91], [92]. Some of these attempts have "only" generated the game rules, whereas others have included some aspects of graphics or theming of the game. We are not aware of any approach to generating games that tries to generate more than two of the facets of games listed

above. We are also not aware of any game generation system that even tries to generate games of more than one genre. Multi-faceted generation systems like *Sonancia* [63], [93] co-generate horror game levels with corresponding soundscapes but do not cater for the generation of rules.

It is clear that the very domain-limited and facet-limited aspects of current game generation systems result from intentionally limiting design choices in order to make the very difficult problem of generating complete games tractable. Yet, in order to move beyond what could be argued to be toy domains and start to fulfill the promise of game generation, we need systems that can generate multiple facets of games at the same time, and that can generate games of different kinds.

A few years ago, something very much like general game generation was outlined as the challenges of "multi-content, multi-domain PCG" and "generating complete games" in another vision paper co-authored by a number of researchers active in the game AI and PCG communities [94]. It is interesting to note that there has not seemingly been any attempt to create more general game generators since then, perhaps due to the complexity of the task. Currently the only genre for which generators have been built that can generate high-quality games is abstract board games; once more genres have been "conquered", we hope that the task of building more general level generators can begin.

V. GENERAL AI-ASSISTED GAME DESIGN TOOLS

The area of AI-assisted game design tools has seen significant research interest in recent years [2] with contributions mainly on the level design task [74], [73], [95], [96], [97], [98], [99]. All tools however remain *specific* to the task they were designed for and their underlying AI focuses on understanding the design process [74], on the generation of a specific facet (or domain) within a game [73] or on both [95].

To illustrate the problems with game-specificity: The authors have demonstrated AI-assisted game design tools to game developers outside academia numerous times, and the feedback have often been that the developers would want something like the demonstrated tools—for their own games. For example, *Ropossum* [96], [100] can greatly assist in designing levels for *Cut the Rope*, but that game is already released and has plenty of levels. Meanwhile, a game developer working on another game, even another physics puzzler, is not helped by *Ropossum* and might not have the time or knowledge to implement the ideas behind it for their own game.

General intelligence is required for tools as much it is required for the other areas of game artificial intelligence. Tools equipped with general capacities to assist humans across game tasks (such as level and audio design) and games genres but also learn to be general across design styles and preferences can only empower the creative process of game development at large. To be general a tool needs to be able to recognize general tasks and procedures during the game design process. An obvious direction towards designer-general tools is through the computational modeling of designers [95] across differebt tasks for identifying general design patterns as well

as personal aesthetic values, styles and procedures. That can be achieved to a degree through imitation learning and sequence mining [101] techniques resulting in designer models that are general across users of the tool. Tools can be general across games too. A level design assistive tool, for instance, can be trained to identify common successful design patterns across game levels of variant genres; in this case levels can to be represented as 2D or 3D maps enriched with item placement and playtrace information that can be fused using e.g. deep autoencoders which has been proved a successful method to combine modalities of different resolution and type (such as time series vs. discrete events) [102], [51]. Finally tools can be general across game design tasks. For instance, a level tool would be able to recommend good sound effects for the level it just co-designed with a human designer if it is equipped with a transmedia (level to audio) representation as e.g. in the work of Horn et al. [103]. Such representations could potentially be machine-learned from existing level-audio patterns.

One path towards building general AI-assisted game design tools is to build a tool that works with a general framework, being able to work on any game expressed in that framework. Such an effort is currently underway for the General Video Game AI framework, for games expressed in VGDL; early work on this project explores how design patterns can be recommended between games [104].

VI. THE ROAD AHEAD AND HOW TO STAY ON PATH

In this paper we have argued that the general intelligence capacity of machines needs to be both explored and exploited in its full potential (1) across the different *tasks* that exist within the game design and development process, including but absolutely no longer limited game playing; (2) across different *games* within the game design space and; (3) across different *users* (players or designers) of AI. We claim that, thus far, we have underestimated the potential for general AI within games. We also claim that the currently dominant practice of only designing AI for a specific task within a specific domain will eventually be detrimental to game AI research as algorithms, methods and epistemological procedures will remain specific to the task at hand. As a result, we will not be manage to push the boundaries of AI and exploit its full capacity for game design. We are inspired by the general game-playing paradigm and the recent successes of AI algorithms in that domain and suggest that we become less specific about all subareas of the game AI field including player modeling, emotive expression, game generation and AI-assisted design tools. Doing so would allow us to detect and mimic different general cognitive and emotive skills of humans when designing games — a creative task that fuses problem solving, artwork and engineering skills.

It might be worth noting that we are not alone in seeing this need. For example, Zook argues for the use of various game-related tasks (not just game playing) to be used in artificial general intelligence research [105]. It also worth noting, again, that we are not advocating that all research within the CI/AI in games field focuses on generality right now; studies on particular games and particular tasks are still valuable, given how little we still understand and can do. But over time, we predict that more and more research will focus on generality across tasks, games and users, because it is in the general problems the interesting research questions of the future lay.

The path towards achieving general game artificial intelligence is still largely unexplored. For AI to become less specific — yet remain relevant and useful for game design — we envision a number of immediate steps that could be taken: first, and foremost the game AI community needs to adopt an *open-source* accessible strategy so that methods and algorithms developed across the different tasks are shared among researchers for the advancement of this research area. Venues such as the current game AI research portal[2] could be expanded and used to host successful methods and algorithms. For the algorithms and methods to be of direct use particular technical specifications need to be established — e.g. such as those established within game-based AI benchmarks — which will maximize the interoperability among the various tools and elements submitted. Examples of benchmarked specifications for the purpose of general game AI research include the general video game description language (VGDL) and the puzzle game engine PuzzleScript[3]. Finally, following the GVGAI competition paradigm, we envision a new set of competitions rewarding general player models, NPC models, AI-assisted tools and game generation techniques. These competitions would further motivate researchers to work in this exciting research area and enrich the database of open-access interoperable methods and algorithms directly contributing to the state of the art in computational general game design.

REFERENCES

[1] G. N. Yannakakis, "Game AI revisited," in *Proceedings of the 9th conference on Computing Frontiers*. ACM, 2012, pp. 285–292.

[2] G. N. Yannakakis and J. Togelius, "A Panorama of Artificial and Computational Intelligence in Games," 2014.

[3] R. W. Picard and R. Picard, *Affective computing*. MIT press Cambridge, 1997, vol. 252.

[4] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational game creativity," in *Proceedings of the 5th International Conference on Computational Creativity*, 2014.

[5] J. Laird and M. VanLent, "Human-level ai's killer application: Interactive computer games," *AI magazine*, vol. 22, no. 2, p. 15, 2001.

[6] T. Schaul, J. Togelius, and J. Schmidhuber, "Measuring intelligence through games," *arXiv preprint arXiv:1109.1314*, 2011.

[7] M. Genesereth, N. Love, and B. Pell, "General game playing: Overview of the AAAI competition," *AI Magazine*, vol. 26, no. 2, pp. 62–72, 2005.

[8] T. Schaul, "A video game description language for model-based or interactive learning," in *Proceedings of the 2013 IEEE Conference on Computational Intelligence in Games*, 2013, pp. 1–8.

[9] M. Ebner, J. Levine, S. M. Lucas, T. Schaul, T. Thompson, and J. Togelius, "Towards a video game description language," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[10] C. Martens, "Ceptre: A language for modeling generative interactive systems," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[11] D. Perez, S. Samothrakis, J. Togelius, T. Schaul, S. Lucas, A. Couëtoux, J. Lee, C.-U. Lim, and T. Thompson, "The 2014 general video game playing competition," 2015.

[2]http://www.aigameresearch.org/

[3]http://www.puzzlescript.net/

[12] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *arXiv preprint arXiv:1207.4708*, 2012.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] A. M. Turing, M. Bates, B. Bowden, and C. Strachey, "Digital computers applied to games," *Faster than thought*, vol. 101, 1953.

[15] N. Ensmenger, "Is chess the drosophila of ai? a social history of an algorithm," *Social studies of science*, p. 0306312711424596, 2011.

[16] M. Campbell, A. J. Hoane, and F.-h. Hsu, "Deep blue," *Artificial intelligence*, vol. 134, no. 1, pp. 57–83, 2002.

[17] M. Newborn, *Kasparov versus Deep Blue: Computer chess comes of age*. Springer Science & Business Media, 2012.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[19] J. Togelius, S. Lucas, H. D. Thang, J. M. Garibaldi, T. Nakashima, C. H. Tan, I. Elhanany, S. Berant, P. Hingston, R. M. MacCallum *et al.*, "The 2007 ieee cec simulated car racing competition," *Genetic Programming and Evolvable Machines*, vol. 9, no. 4, pp. 295–329, 2008.

[20] D. Loiacono, P. L. Lanzi, J. Togelius, E. Onieva, D. A. Pelta, M. V. Butz, T. D. Lönneker, L. Cardamone, D. Perez, Y. Sáez *et al.*, "The 2009 simulated car racing championship," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 2, no. 2, pp. 131–147, 2010.

[21] S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game ai research and competition in starcraft," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 5, no. 4, pp. 293–311, 2013.

[22] D. Perez-Liebana, S. Samothrakis, J. Togelius, T. Schaul, and S. M. Lucas, "General video game ai: Competition, challenges and opportunities," in *Proceedings of AAAI*, 2016.

[23] P. Hingston, "A new design for a turing test for bots," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 345–350.

[24] N. Shaker, J. Togelius, G. N. Yannakakis, L. Poovanna, V. S. Ethiraj, S. J. Johansson, R. G. Reynolds, L. K. Heether, T. Schumann, and M. Gallagher, "The turing test track of the 2012 mario ai championship: entries and evaluation," in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–8.

[25] A. Khalifa, A. Isaksen, J. Togelius, and A. Nealen, "Modifying mcts for human-like general video game playing," in *Proceedings of IJCAI*, 2016.

[26] A. S. Rao and M. P. Georgeff, "Modeling rational agents within a bdi-architecture," *KR*, vol. 91, pp. 473–484, 1991.

[27] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.

[28] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann, "Generic personality and emotion simulation for conversational agents," *Computer animation and virtual worlds*, vol. 15, no. 1, pp. 1–13, 2004.

[29] E. André, M. Klesen, P. Gebhard, S. Allen, and T. Rist, "Integrating models of personality and emotions into lifelike characters," in *Affective interactions*. Springer, 2000, pp. 150–165.

[30] C. Conati, "Intelligent tutoring systems: New challenges and directions." in *IJCAI*, vol. 9, 2009, pp. 2–7.

[31] J. Gratch and S. Marsella, "A domain-independent framework for modeling emotion," *Cognitive Systems Research*, vol. 5, no. 4, pp. 269–306, 2004.

[32] M. Mateas and A. Stern, "Façade: An experiment in building a fully-realized interactive drama," in *Game Developers Conference*, vol. 2, 2003.

[33] J. McCoy, M. Treanor, B. Samuel, M. Mateas, and N. Wardrip-Fruin, "Prom week: social physics as gameplay," in *Proceedings of the 6th International Conference on Foundations of Digital Games*. ACM, 2011, pp. 319–321.

[34] M. P. Eladhari and M. Mateas, "Semi-autonomous avatars in world of minds: A case study of ai-based game design," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. ACM, 2008, pp. 201–208.

[35] J. Rowe, B. Mott, S. McQuiggan, J. Robison, S. Lee, and J. Lester, "Crystal island: A narrative-centered learning environment for eighth grade microbiology," in *workshop on intelligent educational games at the 14th international conference on artificial intelligence in education, Brighton, UK*, 2009, pp. 11–20.

[36] D. Thue, V. Bulitko, M. Spetch, and E. Wasylishen, "Interactive storytelling: A player modelling approach." in *AIIDE*, 2007, pp. 43–48.

[37] M. O. Riedl and V. Bulitko, "Interactive narrative: An intelligent systems approach," *AI Magazine*, vol. 34, no. 1, p. 67, 2012.

[38] R. M. Young, M. O. Riedl, M. Branly, A. Jhala, R. Martin, and C. Saretto, "An architecture for integrating plan-based behavior generation with interactive game environments," *Journal of Game Development*, vol. 1, no. 1, pp. 51–70, 2004.

[39] D. K. Elson and M. O. Riedl, "A lightweight intelligent virtual cinematography system for machinima production," in *AIIDE*, 2007, pp. 8–13.

[40] A. Jhala and R. M. Young, "Cinematic visual discourse: Representation, generation, and evaluation," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 2, no. 2, pp. 69–81, 2010.

[41] P. Burelli, "Virtual cinematography in games: investigating the impact on player experience," *Foundations of Digital Games*, 2013.

[42] H. J. Eysenck, *Biological dimensions of personality*. Guilford Press, 1990.

[43] S. Kshirsagar, "A multilayer personality model," in *Proceedings of the 2nd international symposium on Smart graphics*. ACM, 2002, pp. 107–115.

[44] D. Moffat, "Personality parameters and programs," in *Creating personalities for synthetic actors*. Springer, 1997, pp. 120–165.

[45] J. D. Mayer and P. Salovey, "The intelligence of emotional intelligence," *Intelligence*, vol. 17, no. 4, pp. 433–442, 1993.

[46] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 1, pp. 39–58, 2009.

[47] G. N. Yannakakis and A. Paiva, "Emotion in games," *Handbook on Affective Computing*, pp. 459–471, 2014.

[48] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *Affective Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 147–161, 2011.

[49] N. Shaker, S. Asteriadis, G. N. Yannakakis, and K. Karpouzis, "A game-based corpus for analysing the interplay between game context and player experience," in *Affective Computing and Intelligent Interaction*. Springer Berlin Heidelberg, 2011, pp. 547–556.

[50] ——, "Fusing visual and behavioral cues for modeling user experience in games," *Cybernetics, IEEE Transactions on*, vol. 43, no. 6, pp. 1519–1531, 2013.

[51] H. P. Martínez and G. N. Yannakakis, "Deep multimodal fusion: Combining discrete events and continuous signals," in *Proceedings of the 16th International Conference on Multimodal Interaction*. ACM, 2014, pp. 34–41.

[52] G. Yannakakis and J. Hallam, "A generic approach for obtaining higher entertainment in predator/prey computer games," *Journal of Game Development*, 2005.

[53] G. N. Yannakakis and J. Hallam, "A generic approach for generating interesting interactive pac-man opponents," in *IEEE CIG, year=2005*.

[54] H. Perez Martínez, M. Garbarino, and G. Yannakakis, "Generic physiological features as predictors of player experience," *Affective Computing and Intelligent Interaction*, pp. 267–276, 2011.

[55] N. Shaker, M. Shaker, and M. Abou-Zleikha, "Towards generic models of player experience," in *Proceedings, the Eleventh Aaai Conference on Artificial Intelligence and Interactive Digital Entertainment (aiide-15)*. AAAI Press, 2015.

[56] S. Koelstra, C. Mühl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "Deap: A database for emotion analysis; using physiological signals," *Affective Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 18–31, 2012.

[57] K. Karpouzis, G. N. Yannakakis, N. Shaker, and S. Asteriadis, "The platformer experience dataset," in *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*. IEEE, 2015, pp. 712–718.

[58] Y. Guo and A. Iosup, "The game trace archive," in *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*. IEEE Press, 2012, p. 4.

[59] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2015.

[60] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," 2011.

[61] R. van der Linden, R. Lopes, and R. Bidarra, "Procedural generation of dungeons," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 6, no. 1, pp. 78–89, 2014.

[62] L. Johnson, G. N. Yannakakis, and J. Togelius, "Cellular automata for real-time generation of infinite cave levels," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. ACM, 2010, p. 10.

[63] P. Lopes, A. Liapis, and G. N. Yannakakis, "Sonancia: Sonification of procedurally generated game levels," in *Proceedings of the ICCC workshop on Computational Creativity & Games*, 2015.

[64] A. K. Hoover, W. Cachia, A. Liapis, and G. N. Yannakakis, "Audioinspace: Exploring the creative fusion of generative audio, visuals and gameplay," in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer International Publishing, 2015, pp. 101–112.

[65] L. Cardamone, G. N. Yannakakis, J. Togelius, and P. L. Lanzi, "Evolving interesting maps for a first person shooter," in *Applications of Evolutionary Computation*. Springer, 2011, pp. 63–72.

[66] W. Cachia, A. Liapis, and G. N. Yannakakis, "Multi-level evolution of shooter levels," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[67] J. Togelius, N. Shaker, S. Karakovskiy, and G. N. Yannakakis, "The mario ai championship 2009-2012," *AI Magazine*, vol. 34, no. 3, pp. 89–92, 2013.

[68] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, "A comparative evaluation of procedural level generators in the mario ai framework," 2014.

[69] A. M. Smith and M. Mateas, "Answer set programming for procedural content generation: A design space approach," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 187–200, 2011.

[70] A. Summerville and M. Mateas, "Super mario as a string: Platformer level generation via lstms," *arXiv preprint arXiv:1603.00930*, 2016.

[71] M. Guzdial and M. Riedl, "Toward game level generation from gameplay videos," *arXiv preprint arXiv:1602.07721*, 2016.

[72] S. Snodgrass and S. Ontanon, "A hierarchical mdmc approach to 2d video game map generation," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.

[73] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: A mixed-initiative level design tool," in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*. ACM, 2010, pp. 209–216.

[74] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014.

[75] N. Shaker, J. Togelius, G. N. Yannakakis, B. Weber, T. Shimizu, T. Hashiyama, N. Sorenson, P. Pasquier, P. Mawhorter, G. Takahashi *et al.*, "The 2010 mario ai championship: Level generation track," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 4, pp. 332–347, 2011.

[76] N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, and M. O. Neill, "Evolving levels for super mario bros using grammatical evolution," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 304–311.

[77] N. Shaker, G. N. Yannakakis, and J. Togelius, "Crowdsourcing the aesthetics of platform games," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 5, no. 3, pp. 276–290, 2013.

[78] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[79] A. Liapis, H. P. Martınez, J. Togelius, and G. N. Yannakakis, "Transforming exploratory creativity with delenox," in *Proceedings of the Fourth International Conference on Computational Creativity*. AAAI Press, 2013, pp. 56–63.

[80] A. J. Summerville, S. Snodgrass, M. Mateas, and S. O. Villar, "The vglc: The video game level corpus," *arXiv preprint arXiv:1606.07487*, 2016.

[81] A. Khalifa, D. Perez-Liebana, S. M. Lucas, and J. Togelius, "General video game level generation," in *Proceedings of IJCAI*, 2016.

[82] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational game creativity," in *Proceedings of the Fifth International Conference on Computational Creativity*, vol. 4, no. 1, 2014.

[83] C. Browne, "Automatic generation and evaluation of recombination games," Ph.D. dissertation, Queensland University of Technology, 2008.

[84] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *Proceedings of the 2008 IEEE Symposium on Computational Intelligence and Games*, 2008, pp. 111–118.

[85] T. S. Nielsen, G. A. Barros, J. Togelius, and M. J. Nelson, "Towards generating arcade game rules with vgdl," in *Computational Intelligence and Games (CIG), 2015 IEEE Conference on*. IEEE, 2015, pp. 185–192.

[86] M. Cook and S. Colton, "Multi-faceted evolution of simple arcade games." in *CIG*, 2011, pp. 289–296.

[87] J. M. Font, T. Mahlmann, D. Manrique, and J. Togelius, "Towards the automatic generation of card games through grammar-guided genetic programming." in *FDG*, 2013, pp. 360–363.

[88] J. Kowalski and M. Szykuła, "Evolving chess-like games using relative algorithm performance profiles," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2016, pp. 574–589.

[89] M. J. Nelson and M. Mateas, "Towards automated game design," in *AI* IA 2007: Artificial Intelligence and Human-Oriented Computing*. Springer, 2007, pp. 626–637.

[90] A. Zook and M. O. Riedl, "Automatic game design via mechanic generation." in *AAAI*, 2014, pp. 530–537.

[91] M. Treanor, B. Blackford, M. Mateas, and I. Bogost, "Game-o-matic: Generating videogames that represent ideas," in *Procedural Content Generation Workshop at the Foundations of Digital Games Conference*, 2012.

[92] M. Cook and S. Colton, "Ludus ex machina: Building a 3d game designer that competes alongside humans," in *Proceedings of the 5th International Conference on Computational Creativity*, 2014.

[93] P. Lopes, A. Liapis, and G. N. Yannakakis, "Framing tension for game generation," in *Proceedings of the Seventh International Conference on Computational Creativity*, 2016.

[94] J. Togelius, A. J. Champandard, P. L. Lanzi, M. Mateas, A. Paiva, M. Preuss, and K. O. Stanley, "Procedural content generation in games: Goals, challenges and actionable steps," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[95] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *Proceedings of the AIIDE Workshop on Artificial Intelligence & Game Aesthetics*, 2013.

[96] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels." in *AIIDE*, 2013.

[97] E. Butler, A. M. Smith, Y.-E. Liu, and Z. Popovic, "A mixed-initiative tool for designing level progressions in games," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 2013, pp. 377–386.

[98] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-initiative design of game levels: Integrating mission and space into level generation," in *Proceedings of the 10th International Conference on the Foundations of Digital Games*, 2015.

[99] B. Kybartas and R. Bidarra, "A semantic foundation for mixed-initiative computational storytelling," in *Interactive Storytelling*. Springer, 2015, pp. 162–169.

[100] N. Shaker, M. Shaker, and J. Togelius, "Evolving playable content for cut the rope through a simulation-based approach." in *AIIDE*, 2013.

[101] H. P. Martínez and G. N. Yannakakis, "Mining multimodal sequential patterns: a case study on affect detection," in *Proceedings of the 13th international conference on multimodal interfaces*. ACM, 2011, pp. 3–10.

[102] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 689–696.

[103] B. Horn, G. Smith, R. Masri, and J. Stone, "Visual information vases: Towards a framework for transmedia creative inspiration," in *Proceedings of the Sixth International Conference on Computational Creativity June*, 2015, p. 182.

[104] T. Machado, I. Bravi, Z. Wang, J. Togelius, and A. Nealen, "Recommending game mechanics," in *Proceedings of the FDG Workshop on Procedural Content Generation*, 2016.

[105] A. Zook, "Game AGI beyond Characters," in *Integrating Cognitive Architectures into Virtual Character Design*. IGI Global, 2016, pp. 266–293.