A Panorama of Artificial and Computational Intelligence in Games

Georgios N. Yannakakis, *Member, IEEE*, and Julian Togelius, *Member, IEEE*

Abstract— **This paper attempts to give a high-level overview of the field of artificial and computational intelligence (AI/CI) in games, with particular reference to how the different core research areas within this field inform and interact with each other, both actually and potentially. We identify ten main research areas within this field: NPC behavior learning, search and planning, player modeling, games as AI benchmarks, procedural content generation, computational narrative, believable agents, AI-assisted game design, general game artificial intelligence and AI in commercial games. We view and analyze the areas from three key perspectives: (1) the dominant AI method(s) used under each area; (2) the relation of each area with respect to the end (human) user; and (3) the placement of each area within a human-computer (player-game) interaction perspective. In addition, for each of these areas we consider how it could inform or interact with each of the other areas; in those cases where we find that meaningful interaction either exists or is possible, we describe the character of that interaction and provide references to published studies, if any. We believe that this paper improves understanding of the current nature of the game AI/CI research field and the interdependences between its core areas by providing a unifying overview. We also believe that the discussion of potential interactions between research areas provides a pointer to many interesting future research projects and unexplored subfields.**

Keywords: games, artificial intelligence, computational intelligence

I. Introduction

The field of artificial and computational intelligence in games (game AI/CI) has seen major advancements and several success stories in the roughly 10 years it has existed as a separate research field. During this time, the field has seen the establishment and growth of important yearly meetings including the IEEE Conference on Computational Intelligence and Games (CIG) and the AAAI Artificial Intelligence and Interactive Digital Entertainment (AIIDE) conference series as well as the launch of the IEEE Transactions on Computational Intelligence and AI in Games (TCIAIG). In a recent Dagstuhl Seminar on Artificial and Computational Intelligence in Games[1] dozens of the most prominent game AI researchers were invited to identify and discuss future advancements of the key areas in the field [1], [2]. That seminar is also the origin of this paper. The seminar resulted in several papers providing surveys of current research and promising research directions for individual research areas or topics; in contrast, this paper attempts to provide a *panoramic*[2] view of these key areas. By this, we mean that we see all of the areas together and especially their interrelationships; we describe existing influences between the areas but also discuss the potential of a few unexplored areas of key importance for future game AI/CI.

The ten game AI/CI areas identified during the Dagstuhl seminar and covered in this paper are as follows:

1) *Non-player character (NPC) behavior learning*
2) *Search and planning*
3) *Player modeling*
4) *Games as AI benchmarks*
5) *Procedural content generation*
6) *Computational narrative*
7) *Believable agents*
8) *AI-assisted game design*
9) *General game AI*
10) *AI in commercial games.*

The main motivations for us writing this paper are equivalent to the aims of the Dagstuhl seminar: to help active game AI/CI researchers understand how their particular area or topic relates to other areas within this field, how they can benefit from knowledge created in these other areas and how they can make their own research more relevant for the other areas, so that we collectively can advance the state of the art in game AI/CI. We aim to facilitate and foster synergies across active research areas through placing all key research efforts into a taxonomy with the hope of developing a common understanding and vocabulary within the field of AI/CI in games. While our target audience is researchers that are active within the field, we hope that the paper can also serve as an introduction to research on AI/CI as applied to games for readers with some background in AI or CI.

As mentioned above, the follow-up volume to Dagstuhl seminar 12191 contains several surveys of individual research areas, and topics [1], [2]. In addition to that, there have recently been survey and vision papers published in journals and conferences about several areas with the general field, for example Monte-Carlo Tree Search [3], procedural content generation [4], [5], player modeling [6], computational narrative [7], AI for game production [8], AI for games on mobile devices [9] and game AI at large [10]. Some previous papers have attempted broader surveys, for example on evolutionary computation in games [11] and CI in games [12]. In contrast to those papers, the current paper focuses on the structure of the whole research field from variant perspectives. Moreover, it does not mainly

GNY is with Institute of Digital Games, University of Malta, Msida 2080, Malta. JT is with Center for Computer Games Research, IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark. Emails: georgios.yannakakis@um.edu.mt, julian@togelius.com

[1] The details of the seminar, a report and a follow-up volume are available at: `http://www.dagstuhl.de/12191`

[2] Panorama is formed by the Greek words πᾶν (“all’’) and ὅραμα (“sight’’), and can be translated as “seeing everything”; thus, in our context “panoramic” is a synonym to “holistic”.

describe the various areas themselves, but the interaction between them. It also provides an updated and relatively comprehensive bibliography. The paper starts by providing a general overview of the ten areas with respect to player-game interaction, to the dominant AI methods used and the areas' association to different end users within game research and development. The paper then proceeds with a detailed analysis of each area and its corresponding connections to other areas and ends with a high level summary of our analysis, a discussion about the potential of each game AI/CI area and an outline of core unexplored interconnections that present great promise.

A. *Scope and limits of the taxonomy*

The list of research areas identified at Dagstuhl should not be regarded complete and inclusive of all potential areas of game AI/CI research. It could also be argued that this list of areas is arbitrary and that there is overlap between them. However, this could likely be said of any research field in any discipline. (In software engineering, software design overlaps with software requirements analysis, and in cognitive psychology, memory research overlaps with attention research.) While it might be possible to perform an analysis of this research field so that the individual areas have none or minimal overlap, this would likely lead to a list of artificial areas that do not correspond to the areas game AI researchers perceive themselves to be working in.

It could also be argued that we are omitting certain areas; for example, we are not discussing the areas of *pathfinding* in games and AI for *social simulations*, also identified at Dagstuhl. This is because pathfinding is a relatively isolated area with restricted interaction with the other research areas, and social simulation overlaps very much with believable agents and computational narrative. The scope of this paper is not to provide an inclusive survey of all ten game AI/CI areas but rather a roadmap of interconnections between them via representative references.

As research progresses in our field, new research questions will pop up and new methods be invented, and other questions and methods recede in importance. We believe that all taxonomies of research fields are by necessity tentative. Consequently, the list of areas defined above should not be regarded as fixed. We expect that the list will look somewhat different already at the next Dagstuhl seminar. Hopefully, some researchers will use this paper as part of an argument for the novelty of their research, by showing how their approach does not fit into any of the areas we discuss. In any case, it is important to remember that this is *one* possible conceptualization of the field; others are possible.

A further note on the terminology in this paper is that the title uses the expression "artificial and computational intelligence in games" to refer to the entire field. This reflects the dual roots of the field in artificial intelligence (AI) and computational intelligence (CI) research, and the use of these terms in the names of the major conferences in the field (AI-IDE and CIG) and the flagship journal (TCIAIG, explicitly targeting both CI and AI research). There is no agreement on the exact meaning of the terms AI and CI. Historically, AI has been associated with logic-based methods such as reasoning and planning, and CI has been associated with biologically-inspired methods such as neural networks and evolutionary computation. However, there is considerable overlap and strong similarities between these fields. Most of the methods proposed in both fields aim to make computers perform tasks that have at some point been considered to require intelligence to perform, and most of the methods include some form of heuristic search. The field of machine learning intersects with both AI and CI, and many techniques could be said to be part of either field.

In the rest of the paper we will use the terms "AI in games" and "game AI" to refer to the whole research field, including those approaches that originally come from the CI and machine learning fields. There are the two reasons for this: readability, and that we think that the distinction between CI and AI is not useful for the purposes of this paper or indeed the research field it describes. Our use of these terms is not intended to express any prejudice towards particular methods or research questions. (For a non-exclusive list of methods we believe are part of "AI" according to this definition, see Section II-A.)

B. *Outline of the paper*

The structure of the paper is as follows: In Section II, we start by holistically analyzing the ten game AI areas within the game AI field and we provide three alternative views over game AI: one with respect to the methods used, one with respect to the end users within game research and development and one where we outline how each of the research areas fits within the player-game interaction loop of digital games. In turn, Section III, which constitutes the main part of the paper, digs deeper into the ten research areas and describes each of them. With the subsection describing each area, there is a short description of the area and a paragraph on the possible interactions with each other areas for which we have been able to identify strong, weak or potential influence. The paper ends with a section containing our key conclusions and visions for the future of the field.

II. Three panoramic views of game AI research

Analyzing any research field as a composition of various subareas with interconnections and interdependencies can be achieved in several different ways. In this section we view game AI research from three high-level perspectives that focus on the computer (i.e. the AI methods), the human (i.e. the potential end user of game AI) and the interaction between the key end user (i.e player) and the game. Instead in Section III we outline each game AI area and present the interconnections between the areas.

Game AI is composed of a (set of) methods, processes and algorithms in artificial intelligence as those are applied to, or inform the development of, games. Naturally, game AI can be analyzed through the *method* used by identifying the dominant AI approaches under each game AI area (see Section II-A). Alternatively, game AI can be viewed from the game domain perspective with a focus on the *end users* of each game AI area (see Section II-B). Finally game AI is, by nature, realized through systems that entail rich human computer interaction (i.e games) and, thus, the different areas can be mapped to the interaction framework between the player and the game (see Section II-C).

A. The methods (computer) perspective

The first panoramic view of game AI we present is centered around the AI methods used in the field. As the basis of this analysis we first list the core AI methods mostly used in the game AI field. The six key methodology areas identified include evolutionary computation, reinforcement learning, supervised learning, unsupervised learning, planning and tree search. For each of the 10 game AI areas investigated we have identified the AI methods that are *dominant* or *secondary* in the area. While the dominant methods represent the most popular techniques used in the literature, secondary methods represent techniques that have been considered from a substantial volume of studies but are not dominant.

We have chosen to group methods according to what we perceive as a received taxonomy. While it would certainly be possible to classify the various methods differently, we argue that the proposed classification is compact (containing solely six key methodology areas), it follows standard method classifications in AI and is representative of methods identified and discussed in the Dagstuhl Seminar on Artificial and Computational Intelligence in Games. To clarify the methods we discuss here, supervised learning refers to learning a model that maps instances of datasets to target values such as classes; target values are necessary for supervised learning. Common algorithms used here are ID3 (decision tree learning), backpropagation (neural nets) and support vector machines. Unsupervised learning refers to algorithms that find patterns (e.g. clusters) in datasets that do *not* have target values. This includes methods such as k-means, self-organizing maps and Apriori. Reinforcement learning refers to methods that solve reinforcement learning problems, where a sequence of actions is associated with positive or negative rewards, but not with a “target value” (the correct action). The paradigmatic algorithm here is TD-learning. Evolutionary computation refers to population-based global stochastic optimization algorithms such as genetic algorithms, evolution strategies or particle swarm optimization. Planning refers to any method that builds plans, i.e. paths from a start state to an end state; these include the STRIPS system as well as Partial Order Planning. Finally, tree search refers to methods that search the space of future actions and build trees of possible action sequences, often in an adversarial setting; this includes the Minimax algorithm, its very common variation Alpha-Beta, and Monte Carlo Tree Search.

While this taxonomy is commonly accepted, the lines can be blurred. In particular evolutionary computation, being a very general optimization method, can be used to perform either supervised, unsupervised or reinforcement learning (more or less proficiently). The model-building aspect of reinforcement learning can be seen as a supervised learning problem (mapping from action sequences to rewards), and the commonly used tree search method Monte Carlo Tree Search can be seen as a form of TD-learning. The result of any tree search algorithm can be seen as a plan, though it is often not guaranteed to lead to the desire end state. That the various methods have important commonalities and some overlap does not detract from the fact that each of them is clearly defined.

Table I illustrates the relationship between game AI areas and corresponding methods. It is evident that evolutionary computation, planning and tree search appear to be of dominant or secondary use in most game AI areas: six in total. Evolutionary computation is a dominant method for NPC behavior learning, player modeling, procedural content generation and AI assisted game design and has been considered in believable agents research and in general game AI. Planning and tree search are, unsurprisingly, almost entirely overlapping with respect to the areas AI they are used the most; tree search, however, finds a dominant use in general game AI where planning has not been considered yet. Supervised learning is of moderate use across the game AI areas and appears to be dominant in player modeling and believable agents research. Finally, reinforcement learning and unsupervised learning find limited use across the game AI areas, respectively, being dominant only on NPC behavior learning and player modeling.

Viewing Table I from the game AI areas perspective (table columns) it seems that player modeling, believable agents and AI assisted game design define the three game AI areas with the most diverse and richest palette of AI methods. On the contrary, procedural content generation is solely dominated by evolutionary computation and planning to a secondary degree. It is important to state that the popularity of any AI method within a particular area is closely tight to the task performed or the goal in mind. For example, evolutionary computation is largely regarded as a computationally heavy process which is mostly used in tasks associated with offline training. As PCG so far mainly relies on content that is generated offline, evolutionary computation offers a good candidate method and the core approach behind search-based PCG. On the contrary, if online learning is a requirement for the task at hand other methods (such as reinforcement learning or pruned tree-search) tend to be preferred.

Clearly the possibility space for future implementations of AI methods under particular game AI areas seems rather large. While particular methods have been traditionally dominant in specific areas for good reasons (e.g. planning in computational narrative) there are equally good reasons to believe that the research in a game AI area itself has been heavily influenced by (and limited to) its corresponding dominant AI methods. The empty cells of Table I indicate potential areas for exploration and offer us an alternative view of promising new intersections between game AI areas and methods.

TABLE I

Dominant (●) and secondary (○) AI methods for each of the ten AI areas. NPC is NPC behavior learning, S&P is Search and Planning, PM is Player Modeling, CN is Computational Narrative, BA is Believable Agents, AI-Ass. is AI-Assisted Game Design, GGAI is General Game AI and Com. AI is AI in Commercial Games. The total number of areas a method is used appears at the rightmost column of the table. Note that the "AI as game benchmarks'' area is omitted from this table as all methods are applicable

	NPC	S&P	PM	PCG	CN	BA	AI-Ass.	GGAI	Com. AI	Total (Dominant)
Evolutionary Computation	●		●	●		○	●	○		6 (4)
Reinforcement Learning	●					○		○		3 (1)
Supervised Learning	○		●			●			○	4 (2)
Unsupervised Learning			●		○		○			3 (1)
Planning		●		○	●	●	●		●	6 (5)
Tree Search		●			○	○	○	●	●	6 (3)
Total (Dominant)	3 (2)	2 (2)	3 (3)	2 (1)	3 (1)	5 (2)	4 (2)	3 (1)	3 (2)	

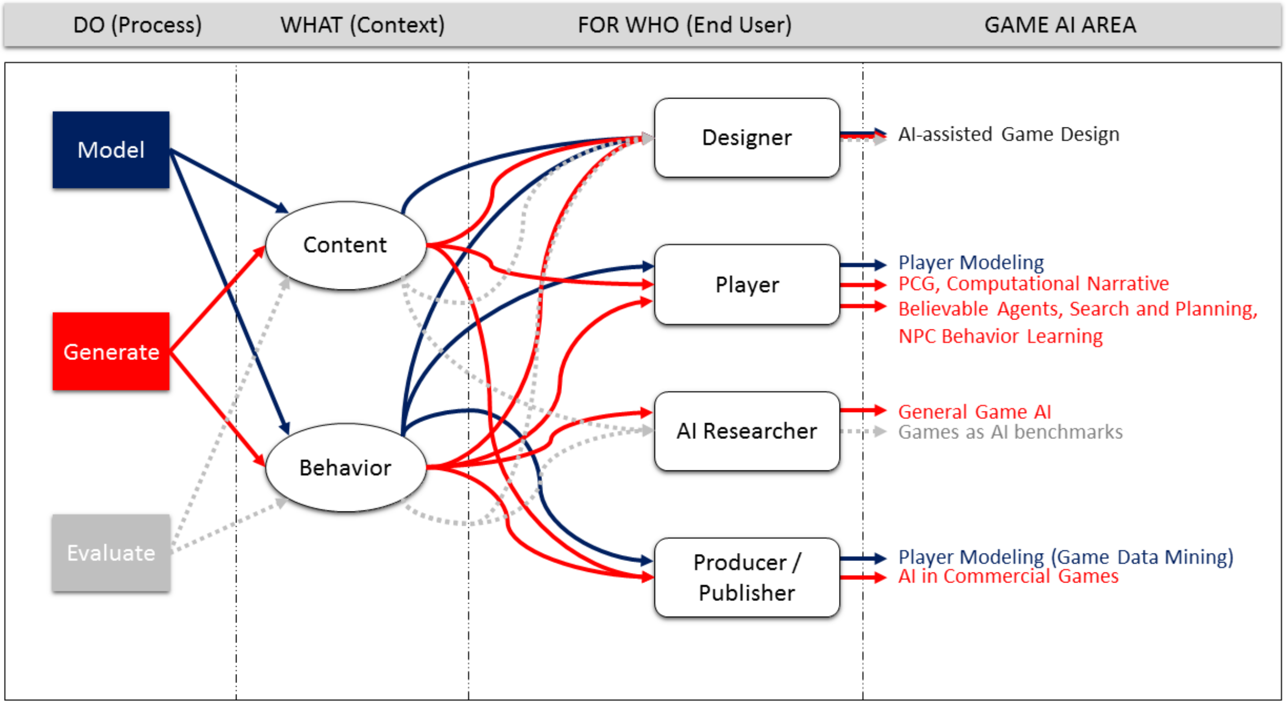

Fig. 1. The end user perspective of the identified game AI areas. Each AI area follows a *process* (model, generate or evaluate) under a *context* (content or behavior) for a particular *end user* (designer, player, AI researcher or game producer / publisher). Dark gray (blue in color print), light gray (red in color print) and ultra light gray (dotted) arrows represent the processes of modeling, generation and evaluation, respectively.

B. The end user (human) perspective

The second panoramic view of the game AI field puts an emphasis on the human end user of the AI technology or general outcome (product or solution). Towards that aim we investigate three core dimensions of the game AI field and classify all ten game AI areas with respect to the *process* AI follows, the game *context* under which algorithms operate and, finally, the *end user* type that benefits most from the resulting outcome. The classes identified under the above dimensions are used as the basis of the taxonomy we propose.

The first dimension (phrased as a question) refers to the AI process: *In general, what can AI* ***do*** *within games?* We identify three potential classes in this dimension: AI can *model*, *generate* or *evaluate*. For instance, an artificial neural network can model a playing pattern, a genetic algorithm can generate game assets and AI tools or benchmarks can be used to evaluate anything that is modeled or generated. Given that AI can model, generate or evaluate the second dimension refers to the context: ***What*** *can AI methods model, generate or evaluate in a game?* The two possible classes here are *content* and *behavior*. For example AI can model a players' affective state, generate a level or evaluate a designer's sketch. Finally, the third dimension is the end user: *AI can model, generate or evaluate either content or behavior; but,* ***for who****?* The classes under the third dimension are the *designer*, the *player*, the *AI researcher*, and the *producer*

/ publisher.

Note that the above taxonomy serves as a framework for classifying the game AI areas according to the end user and is, by no means, inclusive of all potential, processes, context and end users. For instance one could claim that the producer's role should be distinct from the publisher's role and that a developer should also be included in that class. Moreover, game content could be further split into smaller sub classes such as narrative, levels etc. Nevertheless, the proposed taxonomy provides distinct roles for the AI process (model vs. generate vs. evaluate), clear cut classification for the context (content vs. behavior) and a high level classification of the available stakeholders in game research and development (designer vs. player vs. AI researcher vs. producer / publisher).

Figure 1 depicts the relationship between the ten game AI areas and the end user in game research and development. AI-assisted game design is useful for the designer and entails all possible combinations of processes and context as both content and behavior can be either modeled, generated or evaluated for the designer. The player is benefited by the most game AI research areas (6 out of 10) compared to the other stakeholders. In particular the player and her experience is affected by research on player modeling which results from the modeling of behavior; research on procedural content generation and computational narrative, as a result of generation of content; and studies on believable agents, search and planning, and NPC behavior learning resulting from the generation of behavior. The general game AI and the games as AI benchmarks areas provide input to the AI researcher primarily. The first through the generation of behavior and the second through the evaluation of both behavior (e.g. StarCraft competition) and content (e.g. platformer AI level generation track). Finally, the game producer / publisher is affected by results on behavioral player modeling, game analytics and game data mining as a result of behavior modeling. Game producers / publishers are also benefited by progress on AI in commercial games through the generation of both content (e.g. SpeedTree) and behavior (e.g. believable NPCs).

C. The player-game interaction perspective

The third and final panoramic perspective of game AI presented in this section couples the computational processes with the human end user within a game and views all game AI areas through a human computer interaction (HCI) lens — or, more accurately, a player-game interaction lens. The analysis builds on the findings of Section II-B and places the six game AI areas that concern the player as an end user on a player-game interaction framework as depicted in Fig. 2. Putting an emphasis on player experience, player modeling directly focuses on the interaction between a player and the game context. Game content is influenced primarily by research on procedural content generation and computational narrative. In addition to other types of content, most games feature NPCs, the behavior of which is controlled by some form of AI. NPC behavior is informed by research in NPC behavior learning and believable agents. Finally, search and planning affect advancements on the game as a whole (including content and NPCs).

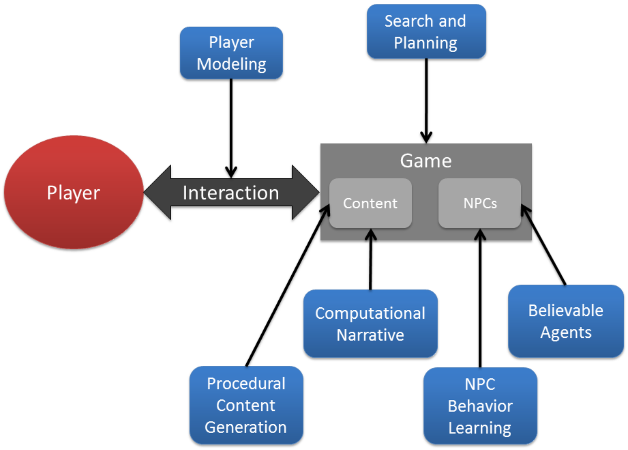

Fig. 2. The panorama of AI research viewed from a player-game interaction perspective.

Looking at the player-game interaction perspective of game AI it is obvious that the player modeling area has the most immediate and direct impact to the player experience as it is the only area linked to the player-game interaction directly. Search and planning influences the game and thereby, the player experience indirectly. Finally, from the remaining areas, PCG influences player experience the most as all games have some form of environment representation and mechanics. Believable agents and NPC behavior learning are constrained to games that include agents or non-player characters whereas computational narrative affects the player experience when a form of narrative is incorporated in the game.

The four areas not considered directly in this game AI perspective affect the player rather remotely. Research on general game AI primarily concerns game AI researchers but findings could potentially be transfered to NPC agents resulting to improved player experience. AI tools assisting game designers improve the game's quality as a whole and in retrospect the player experience since designers tend to maintain a *second order player model* [13] while designing. AI in commercial games focuses primary on the interaction between the player and the game and, thus, also affects the player remotely. Finally, game AI benchmarks are offered for both testing the content and the NPC behaviors of a game but also for the interaction between the player and the game (via e.g. player experience competitions) but are mainly directed to AI researchers (see Fig. 1).

III. How the key game AI research areas inform each other

In this section, we outline the ten most active research areas within game AI research, and discuss how they inform or influence (the terms are used interchangeably) each other.

All research areas could be seen as potentially influencing each other to some degree; however, making a list of all such influences would be impractical ($10^2 - 10 = 90$) and the result would be uninteresting. Therefore we only describe *direct* influences. Direct influences can be either *existing and strong* (represented by a • next to the corresponding influence in the following lists), *existing and weak* (represented by a ○) or *potentially strong* (represented by a ⋆). If input from the informing research area is necessary for the informed research area the link is considered strong. We do not list influences we do not consider potentially important for the informed research area, or which only go through a third research area.

The sections below list *outgoing* influence. Therefore, to know how area A influences area B you should look in the section describing area A. Some influences are mutual, some not. The notation $A \rightarrow B$ in the subsection headings of this section denotes that "A influences B''.

In addition, each section provides a figure representing all outgoing influences of the area as arrows. Black, dark gray and light gray colored areas represent, respectively, *existing and strong*, *existing and weak* and *potentially strong* influence. Areas with white background are not influenced by the area under consideration. The figures also depict the incoming influences from other areas. Incoming *existing and strong*, *existing and weak* and *potentially strong* influences are represented, respectively, with a thick solid line, a solid line and a dashed line around the game AI areas that influence the area under consideration. Note that the description of the incoming influence from an area is presented in the corresponding section of that area.

A. NPC behavior learning

Research in learning NPC behavior focuses on using reinforcement learning techniques such as temporal difference (TD)-learning or evolutionary algorithms to learn policies/behaviors that play games well. From the very beginning of AI research, reinforcement learning techniques have been applied to learn how to play board games (see for example Samuel's Checkers player [14]). Basically, playing the game is seen as a reinforcement learning problem, with the reinforcement tied to some measure of success in the game (e.g. the score, or length of time survived). As with all reinforcement learning problems, different methods can be used to solve the problem (find a good policy) [15] including TD-learning [16], evolutionary computation [11], competitive coevolution [17], [18], [19], [20], simulated annealing [21], other optimisation algorithms and a large number of combinations between such algorithms [22]. In recent years a large number of papers that describe the application of various learning methods to different types of video games have appeared in the literature (including several overviews [23], [11], [24], [25]). Research in NPC behavior learning impacts game AI at large as six key game AI areas are directly affected; in turn, four areas are directly affecting NPC behavior learning (see Fig. 3).

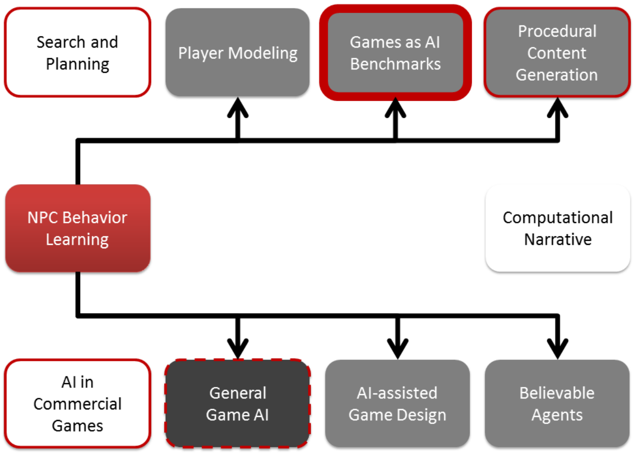

Fig. 3. NPC behavior learning: influence on (and from) other game AI research areas. **Outgoing** influence (represented by arrows): black, dark gray and light gray colored areas reached by arrows represent, respectively, *existing and strong*, *existing and weak* and *potentially strong* influence. **Incoming** influence is represented by red (in color print) lines around the areas that influence the area under investigation (i.e. NPC behavior learning in this figure): *existing and strong*, *existing and weak* and *potentially strong* influences are represented, respectively, by a thick solid line, a solid line and a dashed line.

- ○ **NPC behavior learning → Player modeling:** Though computational player modeling uses learning algorithms, it is only in some cases that it is the behavior of an NPC that is modeled. In particular, this is true when the in-game behavior of one or several players is modeled. This can be done either using reinforcement learning techniques, or supervised learning techniques such as backpropagation or decision trees. In either case, the intended outcome for the learning algorithm is not necessarily an NPC that plays as well as possible, but one that plays in the style of the modeled player [26], [27].
- ○ **NPC behavior learning → Games as AI benchmarks:** Most existing game-based benchmarks measure how well an agent plays a game — see for example [28], [29], [30]. Methods for learning NPC behavior are vital for such benchmarks, as the benchmarks are only meaningful in the context of the algorithms. When algorithms are developed that "beat" existing benchmarks, new benchmarks need to be developed. For example, the success of an early planning agent in the first Mario AI Competition necessitated that the software be augmented with a better level generator for the next competition [28], and for the Simulated Car Racing competition, the performance of the best agents on the original competition game spurred the change to a new more sophisticated racing game [31], [32].
- ○ **NPC behavior learning → Procedural content generation:** Having an agent that is capable of playing a game proficiently is useful for *simulation-based testing* in procedural content generation, i.e. the testing of newly generated game content by playing through that content

with an agent. For example, in an program generating levels for the platform game *Super Mario Bros*, the levels can be tested by allowing a trained agent to play them; those that the agent cannot complete can be discarded [33]. Browne's Ludi system, which generates complete board games, evaluates these games through simulated playthrough and uses learning algorithms to adapt the strategy to each game [34].

- ○ **NPC behavior learning → Believable agents:** An agent cannot be believable if it is not proficient. Being able to play a game well is in several ways a precondition for playing games in a believable manner though well playing agents can be developed without learning (e.g. via top-down approaches). In recent years, successful entries to competitions focused on believable agents, such as the 2k BotPrize and the Mario AI Championship Turing Test track, have included a healthy dose of learning algorithms [35], [36].
- ○ **NPC behavior learning → AI-assisted game design:** Just as with procedural content generation, many tools for AI-assisted game design rely on being able to simulate playthroughs of some aspect of the game. For instance, the Sentient Sketchbook tool for level design uses simple simulations of game-playing agents to evaluate aspects of levels as they are being edited by a human designer [37].
- • **NPC behavior learning → General game AI:** The research program to use games to develop artificial general intelligence (AGI) builds on the idea that games can be useful environments for learning algorithms to learn complex and useful behavior in, and NPC behavior learning is therefore essential to AGI-games research. Some submissions to the Stanford General Game Playing competition are based on learning algorithms, e.g. Reisinger et al.'s NEAT-based general game player is based on neuroevolution [38].

B. Search and planning

Search is one of the fundamentals of computer science, with many of its core algorithms (e.g. Dijkstra's) being search algorithms. In games, two kinds of search are particularly important: best-first search — epitomized by the A* algorithm [39] which is widely used for pathfinding — and game tree search such as MiniMax search with Alpha-Beta pruning which defines the standard algorithm for playing discrete games such as board games. While both A* and MiniMax are very old algorithms, and it might appear that basic search is essentially a solved issue, there is in fact algorithmic innovation happening at a rapid pace within the AI in games community. In particular, the Monte Carlo Tree Search (MCTS) algorithm was invented only a few years ago and is the focus of intense research; it is currently the best-performing algorithm for board games with high branching factors, such as Go [3]. And overview of search and planning in games, particularly in real-time games, can be found in [40]; for path-finding, a very common application of search in games, an overview can be found in [41].

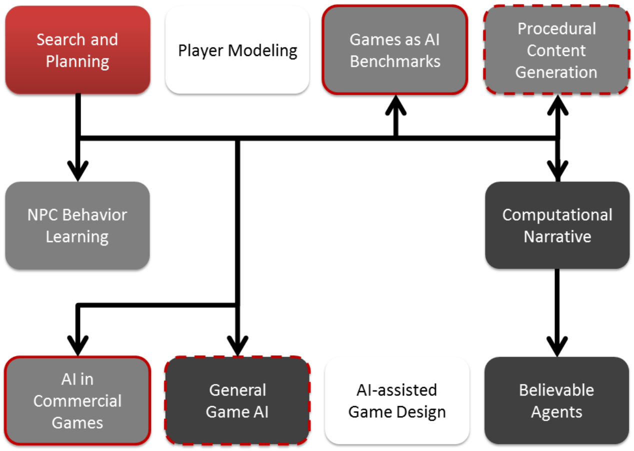

Fig. 4. Search and planning: influence on (and from) other game AI research areas.

Planning is an application of search in state space: a planning algorithm (typically a version of best-first search) searches for the shortest path from one state to another. As such, planning is an important technique for solving many game playing tasks, and has been used both for controlling NPC behavior (such as in the landmark FPS *F.E.A.R.* and acclaimed RPG *Fallout 3*) and creating and managing narratives. Research on search and planning directly influences six out of the ten key game AI areas showcasing its importance on the game AI roadmap (see Fig. 4).

- ○ **Search and planning → NPC behavior learning:** There is plenty of scope for hybrid intelligent game-playing systems that combine planning with learning techniques, and therefore for planning research to inform NPC behavior learning research. In board games, there has been much research on using temporal difference learning [16] or evolution to learn evaluation functions for tree search algorithms [42], and these techniques could very well be generalized to other types of games. Another way of combining planning with NPC learning is to use planning algorithms as primitives in learned controllers, as has been done e.g. by the REALM agent that won the 2010 Mario AI Championship through evolving rule sets that determined goals for planning algorithms to reach [43].
- ○ **Search and planning → Games as AI benchmarks:** While we are not aware of any competition explicitly geared toward planning techniques in a game context, submissions based on planning have performed well in some recent competitions. For example, a very simple planning technique (A* search in game state space using atomic game actions as operators) won the first Mario AI Competition [44].
- ○ **Search and planning → Procedural content generation:** Like learned NPCs, planning can be used to test generated content. In particular, when the game engine is computationally expensive it might make sense to do

only a partial simulation, using planning to replace the actual agent, to test the content. In the mixed-imitative PCG tool Tanagra, levels are evaluated for playability using simple heuristics of reachability which constitute a form of linear planning [45].

- **Search and planning → Computational narrative:** Computational narrative methods for generating or adapting stories of expositions are typically build on planning algorithms, and planning is therefore essential for narrative [46]. The space of stories can be represented in various ways, and the representations in turn make the use of dissimilar search/planning algorithms useful, including traditional optimisation and reinforcement learning approaches [47], [48].
- **Search and planning → Believable agents:** Research on search and, particularly, planning algorithms and methods has traditionally fueled research on believable agents. Agents that are capable of planning actions, express behavioral and emotional patterns in a sequential manner offer increased capacity with respect to believability [49], [50]. Most notably, studies on interactive virtual and conversational agents [49] as well as virtual humans [50] have been built on planning algorithms.
- **Search and planning → General game AI:** As most environments, both in games and in the real world, include challenges that require some sort of planning to solve, it would seem that some form of planning would be part of any truly general game player. In the general game playing competition, almost all of the top players use MCTS. The current best player is the CadiaPlayer, which is completely built around a custom MCTS [51]. Of course, it could be argued that tree search algorithms like MCTS will not lead to "real" AI as they do not model human cognition, but their empirical success is undisputed.
- ○ **Search and planning → AI in commercial games:** Orkin's introduction of Goal-Oriented Action Planning (GOAP) in the FPS *F.E.A.R.* was widely acclaimed for leading to more interesting tactical gameplay, and led to the adoption of similar techniques in other games, such as the RPG *Fallout 3* [52]. Several planning algorithms have been tested in commercial games [53] offering new perspectives and solutions to pathfinding.

C. Player modeling

In player modeling [6], [54], computational models are created for detecting how the player perceives and reacts to gameplay. Such models are often built using machine learning methods where data consisting of some aspect of the game or player-game interaction is associated with labels derived from some assessment of player experience or affect, gathered for example from physiological measurements or questionnaires [55]. However, the area of player modeling is also concerned with structuring observed player behavior even when no correlates to experience are available, e.g. by identifying player types. Player (experience) modeling is considered to be one of the four core non-traditional uses of AI in games [10] and affects research in AI-assisted game design, believable agents, computational narrative and procedural content generation, and it provides new methods for and uses of AI in commercial games (see Fig. 5).

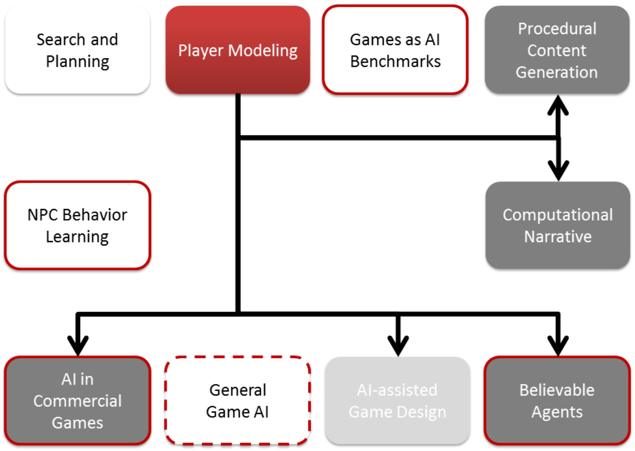

Fig. 5. Player modeling: influence on (and from) other game AI research areas.

- ○ **Player modeling → Procedural content generation:** There is an obvious link between computational models of players and PCG as player models can drive the generation of new personalized content for the player. The experience-driven PCG framework [4] views game content as an indirect building block of a player's affect, cognitive and behavioral state and proposes adaptive mechanisms for synthesizing personalized game experiences. The "core loop" of an experience-driven PCG solution involves learning a model that can predict player experience, and then using this model as part of an evaluation function for evolving (or otherwise optimising) game content; game content is evaluated based on how well it elicits a particular player experience, according to the model. Examples of PCG that is driven by player models include the generation of game rules [56], camera profiles [57], [58] and platform game levels [59]. Most work that goes under the label "game adaptation" can be said to implement the experience-driven architecture; this includes work on adapting the game content to the player using reinforcement learning [60] or semantic constraint solving [61] rather than evolution.
- ○ **Player modeling → Computational narrative:** Player models may inform the generation of computational narrative. Predictive models of playing behavior, cognition and affect can drive the generation of individualized scenarios in a game. Examples of the coupling between player modeling and computational narrative include the affect-driven narrative systems met in *Façade* [62] and *FearNot!* [63], the emotion-driven narrative building system in *Storybricks* (Namaste Entertainment, 2012), and the affect-centered game narratives such as the one of *Final Fantasy VII* (Square, 1997).

○ **Player modeling → Believable agents:** Human player models can inform and update believable agent architectures. Models of behavioral, affective and cognitive aspects of human gameplay can improve the human-likeness and believability of any agent controller — whether it is ad-hoc designed or built on data derived from gameplay. While the link between player modeling and believable agent design is obvious and direct, research efforts towards this integration within games are still sparse. However, the few efforts made on the imitation of human game playing for the construction of believable architectures have resulted in successful outcomes. Human behavior imitation in platform [27] and racing games [64], [65] have provided human-like and believable agents while similar approaches for developing *Unreal Tournament 2004* (Atari, 2004) bots (e.g. in [66]) recently managed to pass the Turing test in the 2k BotPrize competition.

⋆ **Player modeling → AI-assisted game design:** User models can enhance authoring tools that, in turn, can assist the design process. The research area that bridges user modeling and AI-assisted design is in its infancy and only a few example studies can be identified. Indicatively, designer models [13] have been employed to personalize mixed-initiative design processes [67], [37]. Such models drive the procedural generation of designer-tailored content.

○ **Player modeling → AI in commercial games:** Research and development in player modeling can inform attempts for player experience in commercial-standard games. Developed experience detection methods and algorithms can advance the study of user experience in commercial games. In addition, the appropriateness of sensor technology, the intrusiveness and technical plausibility of biofeedback sensors, and the suitability of variant modalities of human input tested *in vitro* can inform industrial developments. Player modeling provides a multifaceted improvement to game development, as it does not only advance the study of human play and the enhancement of human experience. Quantitative testing via game metrics — varying from behavioral data mining to in-depth low scale studies — is improved as it complements existing practices [10], [68], [69]. Quite a few academic papers have been published that use datasets from commercial games to induce models of players that could inform further development of the game, see for example the experiments in clustering players of *Tomb Raider: Underworld* (Square Enix 2008) into archetypes [70] and predicting their late-game performance based on early-game behavior [71]. Examples of high-profile commercial games that incorporate human player modeling experience as a part of gameplay include the arousal-driven appearance of NPCs in *Left 4 Dead 2* (Valve Corporation, 2009), the fearful combat skills of the opponent NPCs in *F.E.A.R.* (Monolith, 2005), and the avatars' emotion expression in the *Sims* series (Maxis, 2000) and *Black and White* (Lionhead Studios, 2001).

D. Games as AI benchmarks

Game-based benchmarks are games, or parts of games (e.g. individual levels or tasks from games) that offer interfaces to external AI systems and a way to evaluate the performance of those systems on the task(s) associated with the benchmark. In many cases, benchmarks are associated with competitions built around those benchmarks, held annually at conferences or continuously on the Internet. In many ways, having good benchmarks is essential for the successful development of a field, as it allows fair comparison of competing methods. However, there is also a risk that focusing on a narrow set of benchmarks can be detrimental to a research because large efforts are concentrated on solving a particular narrow subproblem.

Both the benefits and the risks can be demonstrated by looking at other areas of AI and CI research. In continuous optimization there exists an extensive set of benchmark functions (problems), that have been in development at least since the eighties, and which exemplify a large variety of different optimization problems. New benchmark functions are regularly added to the list because they have interestingly different properties from existing benchmarks, or because they represent improvements over existing benchmarks. Many of these functions are very hard, and naive optimization algorithms fare worse than random search. Occasionally, competitions are held based on a set of these benchmarks functions, and these then become the "canon". For some years, the function set used for the recurring competition at the IEEE Congress on Evolutionary Computation (CEC) was the golden standard against which to measure your algorithm if you wanted to get your paper accepted in prestigious journals [72]. More recently, the Comparing Continuous Optimisers (COCO) platform has been used in recurring Black-Box Optimisation Benchmarking competitions and has assumed the role of standard benchmark for continuous optimisation [73].

This situation can be contrasted with that in the reinforcement learning community, where the lack of modern, challenging, open-source problems make comparison of algorithms difficult. In some cases, considerable efforts go into tuning and redesigning algorithms specifically to perform even better on very simple benchmark problems, such as the double pole balancing problem [74]. The Reinforcement Learning Competition has recently made some efforts to rectify this situation but has not yet produced a universally agreed benchmark problem set[3].

Games are in many ways ideal for basing benchmarks on. Most games include scoring functions of various kinds, games can be standalone software packages, and maybe most importantly games that are made for people to play include challenges suitable for humans and therefore typically also challenging for AI methods. On the other hand, there are

[3] http://rl-competition.org

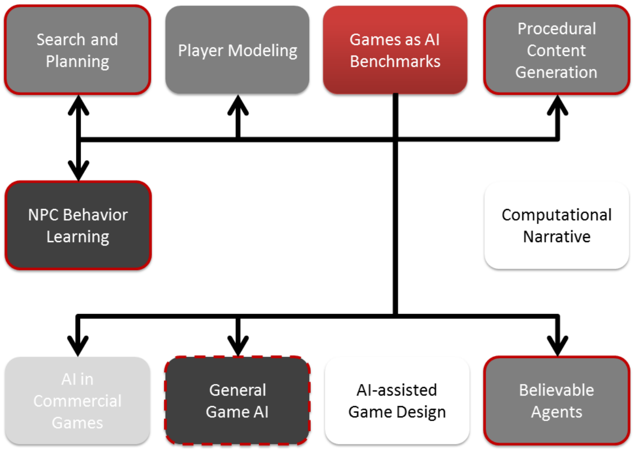

Fig. 6. Games as AI benchmarks: influence on (and from) other game AI research areas.

significant challenges in that games are sometimes tied to particular hardware platforms, rarely come with open source code making the development of a benchmarking API difficult, and that it is not clear how to measure properties such as aesthetic qualities or believability.

Game AI benchmarks impact research on seven other key game AI areas making it the most influential area in our taxonomy (see Fig. 6). Those effects are discussed below.

- **Games as AI benchmarks → NPC behavior learning:** The development of algorithms for learning to play games has benefited greatly from the availability of good benchmarks during the last decade. A number of competitions have been held at conferences such as the annual Computational Intelligence and Games conference based on well-known games. These include the Simulated Car Racing Competition [32], the Ms. Pac-Man competition [29], the 2k BotPrize (based on *Unreal Tournament*) [75], the *StarCraft* (Blizzard Entertainment, 1998) AI competition [76], [77], [30] and the *Mario* AI Championship [78]. In most cases, the objective function is the game score or some other equally simple and transparent measure, such as lap time or progress made before dying. In some cases, the competitions are based on custom interfaces to the original game code, in other cases on open-source re-implementations of the game in a language such as Java — the latter approach has several advantages, including being portability of source code, lack of dependence on software licenses, and being able to control the running speed of the game. Other competitions are based on games that were developed specifically for the competition, such as Cellz [79] or the Physical Traveling Salesperson Problem (PTSP) [80]. Most of these competitions have run for several years, with results published both in multi-authored papers and papers by competitors describing their contributions, indicating that the competitions have spurned much good research.
  - ○ **Games as AI benchmarks → Search and planning:** For search algorithms, the classic board game-based competitions — the Computer Chess [81], Checkers [82] and Go [83] tournaments have been very valuable. While the planning community develops their own planning benchmarks and run their own planning competitions, no benchmarks or competitions focused specifically on planning exist within the games research community. This being said, approaches based on various forms of informed search have done well in both the Mario AI competition and the Ms. Pac-Man competition, even though these competitions were not designed to advantage this kind of technique. For some complex domains, a part of the overall problem might be seen as a planning problem in its own right; for example, build order planning is one part of successful StarCraft playing [84].
  - ○ **Games as AI benchmarks → Player modeling:** As part of the 2010 Mario AI Championship, a “level generation track” was arranged, where the objective was not to play the game well, but rather to generate personalized levels [85]. Competitors submitted level generators, that were given information on how a player had played a set test level, and were tasked with generating a level for that player. Thus, the content generation task included an implicit player modeling task. We do not know of any direct player modeling benchmark within the games community, though there are multimodal data corpora within the neighboring affective computing community with annotated data about user experience, including game [86] and non-game applications [87].
  - ○ **Games as AI benchmarks → Procedural content generation:** The above mentioned level generation track of the Mario AI Championship [85] is to the best of our knowledge the only extant procedural content generation competition. As there is no known way of automatically judging the quality of a generated level (beyond simple concerns of reachability etc.), the scoring of submitted generators is based on crowdsourced preference (ranked) expressions between generators by human play-testers. The best methodology for doing this is still an open research question, and there is a need for competitions addressing the generation of different types of game content as well. Based on the results of such competitions, large scale numerical evaluation can be done to investigate the relative characteristics of the content generators [88].
  - ○ **Games as AI benchmarks → Believable agents:** As already discussed, two recent competitions have focused on believable rather than well-performing game-playing behavior. Such competitions might help advance the study of believability in games, at least in particular and narrow domains. However, it is not clear how well observations on what micro- or macro-patterns seem

believable in a particular game generalize; to do this, similar benchmarks will need to be constructed based on a variety of different games.

- **Games as AI benchmarks → General game AI:** A meaningful benchmark for artificial general intelligence would not be based on a single game, as an AI player built and tuned specifically for a single game is not likely to generalize very well. One benchmark that is constructed explicitly to test the performance of agents on a number of unseen games is the *General Game Playing Competition* [89]. However, the game description language used for that competition restricts the possible games to a discrete-time, discrete-state perfect information games. Therefore, a new initiative is currently working on developing a video game description language, and using that for a competition focused on general game playing in arcade-like games with graphical logics [90], [91], [92].

⋆ **Games as AI benchmarks → AI in commercial games:** While we are not aware of any case where AI developed to participate in a competition or beat a benchmark has been adopted in a commercial game, there are potential huge benefits for commercial game developers that open up APIs for their games to allow academics and others to compete in building the best solution to their open AI problems. As an example, the *StarCraft* AI competition [76], [77], [30], [84] has already produced players that by far outperform the built-in AI opponents in the game (but not good human players); unfortunately, this could not happen before an open API was developed for the game, more than 10 years after its release.

E. Procedural content generation

Procedural content generation (PCG) refers to the automatic or semi-automatic creation of game content such as levels, maps, items, quests and textures. While PCG has been included in limited roles in some commercial games since the early eighties, recent years has seen an expansion of research on more controllable PCG for multiple types of game content [10], using techniques such as evolutionary search [5] and constraint solving [93]. The influence of PCG research beyond games is already evident in areas such as computational creativity [94] and interaction design (among others). There are several surveys of PCG available, including a recent book [95] and vision paper [96], as well as surveys of sub-areas of PCG [97], [98] and of the use of specific methods [5], [93]. Figure 7 depicts the six (and eight) areas that are influenced by (and influence) PCG.

○ **Procedural content generation → NPC behavior learning:** If an agent is trained to perform well in only a single game environment, it is easy to overspecialize the training and arrive at a policy/behavior that will not generalize to other levels. Therefore, it is important to have a large number of environments available for training. PCG can help with this, potentially providing an infinite supply of test environments. For example, when training players for the Mario AI Championship it is common practice to test each agent on a large set of freshly generated levels, to avoid overtraining [28]. There has also been research on adapting NPC behavior specifically to generated content [99].

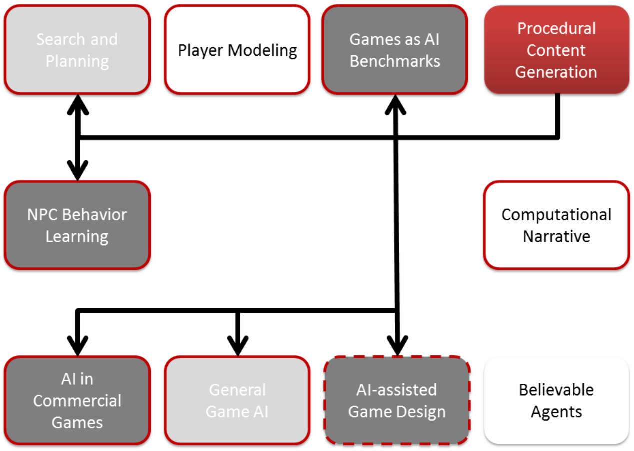

Fig. 7. Procedural content generation: influence on (and from) other game AI research areas.

⋆ **Procedural content generation → Search and Planning:** As with NPC behavior learning, planning algorithms need benchmarks. Planning problems in games are defined by game content, typically level topology or story structure. Therefore, PCG could be used to develop relevant benchmark problems for planning algorithms. For example, evolution of maps has been used to create challenging problems for path planning [100].

○ **Procedural content generation → Games as AI benchmarks:** As discussed above, the generation of new levels/environments is very important for NPC behavior learning, and this extends to benchmarks that measure some aspect of NPC behavior. Apart from the Mario AI Championship, competitions such as the Simulated Car Racing Championship uses freshly generated tracks, unseen by the participants, to test submitted controllers [101]. But there is also scope for benchmarks and competitions focused on measuring the capabilities of PCG systems themselves, such as the Level Generation track of the Mario AI Championship [85].

○ **Procedural content generation → AI-assisted game design:** As content design is a central part of game design, many AI-assisted design tools incorporate some form of assisted content design. Examples include *Tanagra*, which helps designers create platform game levels by using constraint satisfaction to complete levels and ensure playability [45] and *SketchaWorld* [102]. Another example is *Sentient Sketchbook*, which assists humans in designing strategy game levels through giving immediate feedback on properties of levels and autonomously suggesting modifications [37].

- ⋆ **Procedural content generation → General game AI:** One approach to artificial general intelligence is to train agents to be good at playing games in general, and test them on a large variety of games drawn from some genre or distribution. To avoid overfitting, this requires games to be generated automatically, a form of PCG [103].
- ○ **Procedural content generation → AI in commercial games:** PCG is one of the areas of recent academic research on AI in games which bears most promise for incorporation into commercial games. A number of recent games have been based heavily on PCG, including independent ("indie'') game production successes such as *Spelunky* (Mossmouth, 2009) and *Minecraft* (Mojang, 2001), and mainstream AAA games such as *Diablo III* (Blizzard Entertainment, 2012) and *Civilization V* (2K Games, 2010). Some games heavily based on PCG and developed by academic researchers have been released as commercial games on platforms such as Steam and Facebook; two good examples of this are *Petalz* [104] and *Galactic Arms Race* [105]. The game industry is very open to academic innovation in PCG, though some technical hurdles remain, including the structure of most game engines being somewhat unsuitable for runtime content generation.

F. *Computational narrative*

Whether games can tell stories [106] or games are instead a form of narrative [107] is still an open research question in game studies. The study of computational narrative focuses on the representational and generational aspects of stories as those can be told via a game. Stories can play an essential part in creating the aesthetics of a game which, in turn, can impact affective and cognitive aspects of the playing experience. As seen in Fig. 8, computational narrative informs research on procedural content generation, influences the design of believable agents and impacts the use of AI in commercial games. On the other hand, it is influenced by search and planning (strong influence) player modeling (weak influence), believable agents (weak influence) and AI-assisted game design (potential influence).

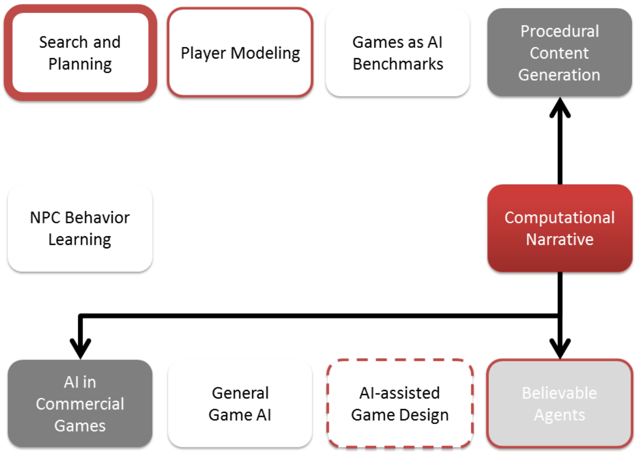

Fig. 8. Computational narrative: influence on (and from) other game AI research areas.

- ○ **Computational narrative → Procedural content generation:** By breaking the game narrative into subareas of game content we can find core game content elements such as the game's plotline [7], [108], but also the ways a story is represented in the game environment [109]. The coupling of a game's representation and the story of the game is of vital importance for player experience. Stories and plots are taking place in an environment and are usually told via a virtual camera lens. The behavior of the virtual camera — viewed as a parameterized element of computational narrative — can drastically influence the player's experience. That can be achieved via an affect-based cinematographic representation of multiple cameras as those used in *Heavy Rain* (Quantic Dream, 2010) or through an affect-based automatic camera controller as that used in the Maze-Ball game [57]. Choosing the best camera angle to highlight an aspect of a story can be seen as a multi-level optimisation problem, and approached with combinations of optimisation algorithms [58]. Games such as *World of Warcraft* (Blizzard Entertainment, 2004) use cut scenes to raise the story's climax and lead the player to particular player experience states.
  The inverse relationship between computational narrative and PCG exists if camera profiles are viewed as game content [4]: worlds as generated through the camera lens contribute to the realization of a narrative. The creation or semi-automatic generation of stories and narratives belong to the area of interactive storytelling (IS) which defines a form of story-based PCG. The story can adjust according to the actions of the player targeting personalized story generation (e.g. see [110], [111] among others). Ultimately, game worlds and plot point story representations can be co-generated as demonstrated in a few recent studies (e.g. see [112]).
- ⋆ **Computational narrative → Believable agents:** Research on interactive narrative benefits from and influences the use of believable agents that interact with the player and are interwoven in the story plot. The narrative can yield more (or less) believability to agents and thus the relationship between agents and story is strong. In that sense, the computational narrative of a game may define the arena for believable agent design.
- ○ **Computational narrative → AI in commercial games:** Research on computational narrative has influenced game development at large. Starting from popular independent attempts like *Façade* [62] and *Prom Week* [113] to the commercial success of *The Elder Scrolls V: Skyrim* (Bethesda Softworks, 2011) narrative has traditionally been amongst the key factors of player experience and immersion; particularly in narrative-heavy games as the ones aforementioned. Examples of

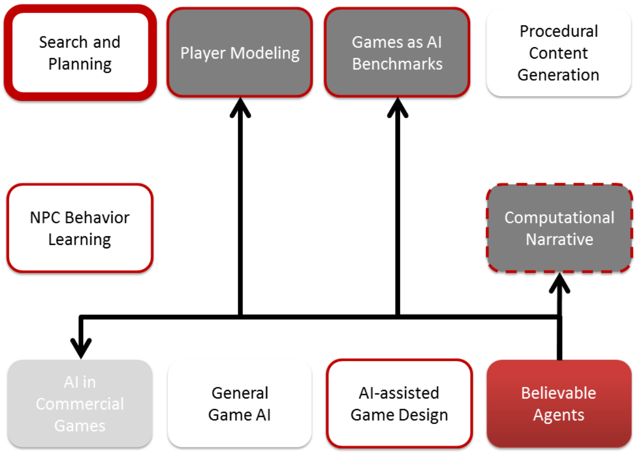

Fig. 9. Believable agents: influence on (and from) other game AI research areas.

sophisticated computational narrative techniques crossing over from academia to commercial games include the celebrated storytelling game *Versu* [114].

G. Believable agents

Research on believable agents is central to games incorporating any form of NPCs. It involves the study of believability in games and the investigations of mechanisms for the construction of agent architectures that appear to have believable or human-like characteristics. The approaches for developing such architectures can be either top-down (such as the FAtiMA model used in *My Dream Theatre* [115] and the Mind Module model [116] used in *The Pataphysic Institute*) or bottom-up attempting to imitate believable gameplay from human players such as the early work of Thurau et al. in *Quake II* (Activision, 1997) bots [117], the human imitation attempts in Super Mario Bros [27], the *UT 2004* believable bots of Schrum et al. [36] and the crowdsourcing studies of the *Restaurant game* [118]. A noteworthy sustained effort is that of Phil Hingston, who has been running the 2k BotPrize for several years; these efforts are chronicled in [119]. The study of believable agents affects research on four other game AI areas as illustrated in Fig. 9, whereas it is affected by six other game AI areas.

- ○ **Believable agents → Player modeling:** There is a direct link between player modeling and believable agents as research carried for the modeling of human, human-like, and supposedly believable, playing behavior can inform the construction of more appropriate models for players. Examples include the imitation of human play styles in Super Mario Bros [27] and Quake [117].
- ○ **Believable agents → Games as AI benchmarks:** Over the last few years there has been a growing academic and commercial interest in the establishment of competitions that can be used as assessment tools for agent believability [35]. A number of game Turing competitions have been introduced to the benefit of agent believability research including the 2k BotPrize on the *Unreal Tournament 2004* (Atari, 2004) [120] game and the Mario AI Championship: Turing Test Track [121] on the *Super Mario Bros* (Nintendo, 1985) game. Recently, the community saw AI agents that passed the Turing test in the 2k BotPrize [36]. Agent believability research has provided input and given substance to those game benchmarks.
- ○ **Believable agents → Computational narrative:** Believable characters contribute to more believable stories and game representations [122]. A typical example of the integration of characters in the narrative and the drive of the latter based on the first includes the FAtiMa agents in *FearNot!* [123] and *My Dream Theater* [115].
- ⋆ **Believable agents → AI in commercial games:** Commercial games have for long benefited from agent believability research. Examples of this include popular games such as the *Sims* (Electronic Arts, 2000) series. The industry puts a catholic focus on the design of believability in games as this contributes to more immersive game environments. The funding of believability research through game AI competitions such as the 2kBot prize is one of the many clear indicators of the commercial value of agent believability.

H. AI-assisted game design

AI-assisted game design refers to the development of AI-powered tools that support the game design and development process. This is perhaps the AI research area which is most promising for the development of better games [10]. In particular, AI can assist in the creation of game content varying from levels and maps to game mechanics and narratives. The impact of AI-enabled authoring and AI-assistive tools on design and development influences the use of AI in commercial games, the study of believable agents, research on computational narrative and attempts in procedural content generation (see Fig. 10). The range of AI-assisted game design tools rule from those designed to assist with generation on complete game rulesets such as MetaGame [124] or RuLearn [125] to those focused on more specific domains such as strategy game levels [37], platform game levels [126] or physics-based puzzles [127].

- ⋆ **AI-assisted game design → Computational narrative:** Tools can assist the authoring of narrative in games. In particular drama management tools have long been investigated within the game AI community. An academic example is ABL which has allowed the authoring of narrative in *Façade* [62] but is not publicly available yet. Among the few available and well-functional authoring tools the most notable is the text adventure development system (TADS) and *Inform 7* [128] for interactive fiction.
- ⋆ **AI-assisted game design → Believable agents:** Authoring tools in forms of open-world sandboxes could potentially be used for the creation of more believable behaviors. While this is largely still an unexplored area

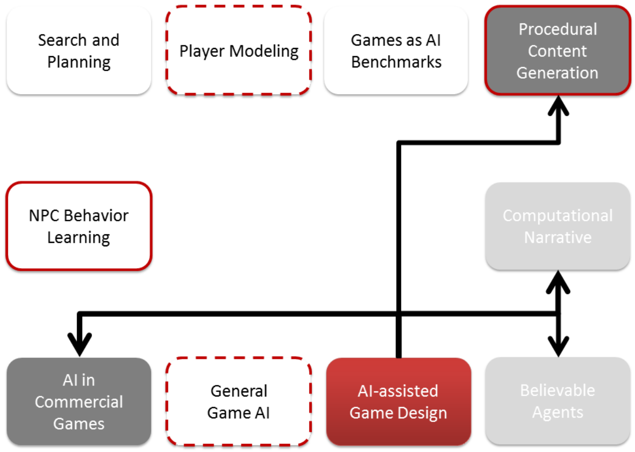

Fig. 10. AI-assisted game design: influence on (and from) other game AI research areas.

of research and development, notable attempts include the NERO game AI platform where players can train game agents for efficient and believable first-person shooter bot behaviors [129]. An open version of this platform focused on crowd-sourcing behaviours has recently been released [130].

- **AI-assisted game design → Procedural content generation:** Research on methods of mixed-initiative co-creation and design can feed input to and spur discussion on central topics in procedural content generation. Given the importance of content design in the development process as a whole, any form of mixed-initiative AI assistance in the generation process can support and augment procedural content generation. Notable examples of mixed-initiative PCG include the *Tanagra* platform game level design AI assistant [45], and the *SketchaWorld* [102], the *Sentient World* [131] and the *Sentient Sketchbook* [37] systems which generate game maps and worlds in a mixed-initiative design fashion following different approaches and levels of human computation.
- **AI-assisted game design → AI in commercial games:** AI tools have been used extensively for supporting design and game development. Examples such as the *SpeedTree* (Interactive Data Visualization Inc., 2013) generator for trees and other plants [132] have seen uses in several game productions. The mixed-initiative PCG tools mentioned above have a great potential in the near future as most of these are already tested on commercial games or developed with game industrial partners. Furthermore, there are tools designed for interactive modelling and analysis of game rules and mechanics, which are not focused on generating complete games but on prototyping and understanding aspects of complex games; such systems could be applied to existing commercial games [133].

I. General game AI

In recent years, researchers disappointed with the fragmentation of modern artificial intelligence and perceived lack of interest in the "grand questions" among their many constituent sub-communities have defined the research area of *artificial general intelligence* (AGI), with its own conference and journal. Within this area, roadmaps, concepts and (occasionally) concrete methods aimed at eventually achieving human-level domain-general artificial intelligence are proposed and discussed [134]. One influential framework for thinking about AGI is that of Legg and Hutter, who define an agent's intelligence as its performance over all possible environments, weighed by their simplicity [135]. Transformed to the language of games, this would mean that an intelligent game-playing agent would be able to competently play a large variety of games, perhaps after some initial learning process [103].

The development of agents capable of playing more than one game is studied within the area of general game playing. This area was originally formed around the annual General Game Playing Competition, where agents are submitted and tested on several unseen games described in a special-purpose game description language [89]. In recent years, the winners of this competition (such as CadiaPlayer [51]) are all based on Monte-Carlo Tree Search (MCTS) [3]. The game description language used in this competition limits the games to have discrete state and action space and perfect information and the verbosity of the language practically limits the games to the complexity of simple board games. In reaction to these restrictions, some researchers have very recently proposed ways of testing general game playing agents on video games, for example the Arcade Learning Environment (ALE) [136] based on emulation of classic Atari 2600 games, and the Video Game Description Language [91], [92] which aims to be a game description language for simple 2D graphical games. Recently, neuroevolution has showed promising results on the ALE [137].

Figure 11 depicts the impact of general game AI to six game AI areas. It is notable that general game AI is influenced strongly by three areas: NPC behavior learning, search and planning, and games as AI benchmarks.

- ⋆ **General game AI → NPC behavior learning:** Currently, much of the work that goes into a high-performing NPC controller goes into feature engineering, tuning and other domain-specific technicalities, and the result is often a controller which generalizes very badly to games other than the one it was developed for, or even variations of the same game [90], [137].
  While entries based only on general learning algorithms have done well in early iterations of several game AI competitions, the champions of later iterations tend to exhibit what in machine learning language would be called "overfitting" [32], [78]. Adopting a habit of testing controllers on unseen games, or even unseen variations of the same games, would go far towards countering this trend [103]. Even in cases where this is

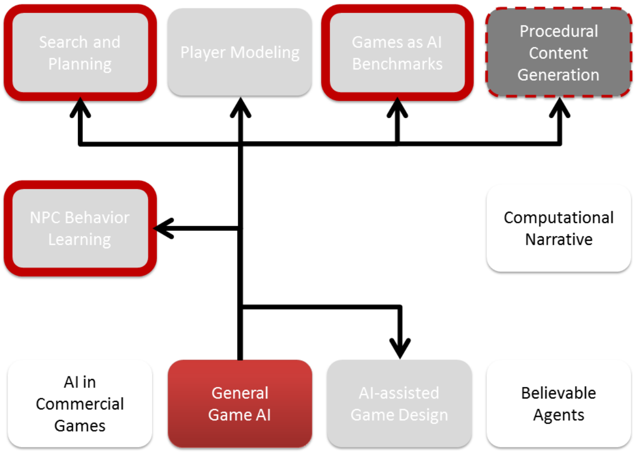

Fig. 11. General game AI: influence on (and from) other game AI research areas.

not possible, one could attempt to use only techniques that have been verified to work in a general game playing setting.

⋆ **General game AI → Search and planning:** Similarly to the case for NPC behavior learning, research in search and planning has plenty to gain from testing algorithms on multiple games in order to avoid overfitting to a single problem.

⋆ **General game AI → Player modeling:** General AI approaches for data-driven player modeling are of utmost importance for the identification of general playing patterns that map to player styles and experiences in a context-free fashion. A notable example of such an attempt is the work of Martinez et al. [138] in which detected physiological manifestations of player experience are cross-validated in both prey-predator and racing games yielding generic sets of player modeling attributes across those game genres.

⋆ **General game AI → Games as AI benchmarks:** Developers of game-based benchmarks and competitions have much to gain from adopting some of the thinking of the AGI community, in particular by adding meaningful variation to the benchmarks so as to privilege generic solutions rather than domain-specific engineering. One could argue that many current game-based AI benchmarks are too specific to test AI well, but there are many ways in which games or generic game engines could be parameterized so as to make challenges broader [103].

◦ **General game AI → Procedural content generation:** When generating game content for a particular game, simulation-based testing can be done using an NPC controller tuned for that particular game (at least if its behavior is indicative of that of a human player). However, when generating completely new games — i.e., when generating the rules and dynamics of a game — it is obviously not possible to have a controller specifically developed for that game. In other words, it is necessary to have relatively domain-general AI to do simulation-based testing. In the literature on generating complete games, this problem has been tackled in different ways. One approach is to train a controller for each particular game that is being tested via reinforcement learning, either by evolving a direct policy controller [56] or by evolving an evaluation function for MiniMax tree search [34]. Another approach is to play the game through deriving the best moves using a constraint solver [139], or to use a statistical technique such as MCTS [140], [141].

⋆ **General game AI → AI-assisted game design:** Given the breadth of challenges in AI-assisted game design, AI developed for such tools would need to assume multiple roles. One wide-ranging task for AI in such tools is modelling the goals, style and preferences of the designer [13]. Further, simulation-based testing is used in many AI-powered game design tools [127], [37], so general game playing research could be useful here just as it is for PCG.

J. AI in commercial games

Academic research on AI in games and commercial game AI development efforts are rather disjoint and there is a wide divergence in methods and results. In the extreme cases, it is not unheard of for a commercial game AI developer to refer to academic efforts as “useless” and for academic researchers to refer to AI that goes into commercial game as “trivial” or even “cheating”. (Such opinions are typically expressed privately rather than in print.) This is largely because academic research and commercial game development are focused on rather different problems: academic researchers want general solutions to generic and preferably deep problems, whereas commercial developers want something that works well enough and looks good to the player in a particular context. This, in turn, makes it potentially acceptable to give AI players extra information, or simply teleport characters or invent units out of nothing, in a commercial game context.

On the other hand, the AI methods adopted in commercial game development have often provided input and inspiration to game AI research, and occasionally technologies from academia have crossed the border into commercial game AI as well. The game industry faces particular problems that have been solved with in-house game AI solutions that allowed academic game AI to flourish in several areas including NPC behavior learning, search planning, player modeling and PCG (see Fig. 12).

◦ **AI in commercial games → NPC behavior learning:** There is a long academic debate whether sophisticated NPC AI techniques have ever been developed by game studios. Regardless of their level of sophistication and complexity, AI techniques such as behavior trees have influenced the AI design of many games after their success in the *Halo* (Microsoft Studios, 2001) series [142]. In addition to commercial products, game AI researchers viewed behavior trees as a vehicle for attempting to solve commercial game AI problems and leverage

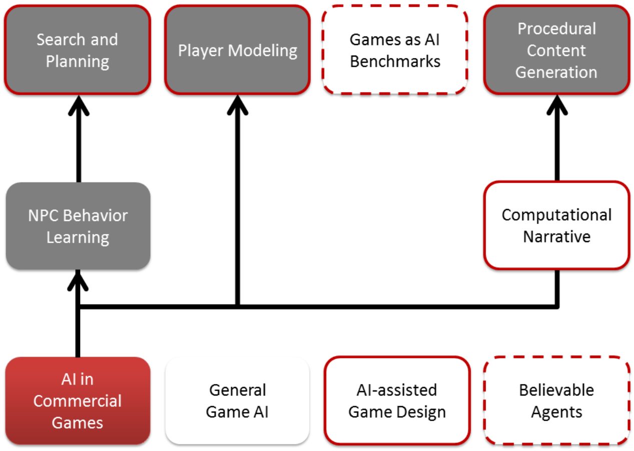

Fig. 12. AI in commercial games: influence on (and from) other game AI research areas.

their sophisticated AI techniques for controlling NPCs. The literature reports a few but promising attempts in integrating learning (such as evolutionary computation) for designing behavior trees (see [143], [144] among others).

- ○ **AI in commercial games → Search and planning:** The use of planning for NPC behavior control (aside pathfinding and the multiple versions and enhancements of the popular A* algorithm) has seen great interest in commercial game development. Modifications of well known algorithms such as the use of STRIPS planners in *F.E.A.R.* (Monolith Productions. 2005) [52] have resulted in impressive NPC behaviors within games and realized the use of computationally expensive algorithms in demanding software applications. Such successful commercial game AI stories have spurred discussion and inspired work in academic approaches to planning in games (e.g. as in [145]). Furthermore the availability of popular game benchmarks such as *Dragon Age* (Electronic Arts, 2009) have spurred research on planning and pathfinding [53].
- ○ **AI in commercial games → Player modeling:** Research on player modeling — and user modeling in general — has been boosted through recent advancements in game software, hardware and game sensor technology [10]. The *PrimeSense* sensor of *Kinect* and the *Wii* sensor series allowed game AI researchers, for the first time, to investigate body motion patterns linked to player experience in great detail. Robust physiological sensors such as the IOM hardware of the Wild Divine Game[4], and the *Empatica*[5] bracelet provide access to physiological manifestations of player affective patterns in unobtrusive ways [57]. Advances in database technology have made the storage of massive game data sets possible. The very existence of these data sets established the *game data mining* [69], [68] research area in which game data is mined to infer spatio-temporal behavioral patterns of the players. Most importantly, advanced game sensor technology has opened new research pathways in multimodal input processing and natural interaction at large.
- ○ **AI in commercial games → Procedural content generation:** The early *Rogue* (1980) dungeon and *Diablo* (Blizzard Entertainment, 1996) world generators, and the *SpeedTree* (Interactive Data Visualization Inc., 2013) vegetation generator (among many other commercial software and games featuring PCG) have provided input and inspiration to PCG research over the last few years [10]. The recent revisit of commercial-standard techniques for PCG yielded entirely new approaches to PCG such as the *search-based* [5] and the *experience-driven* PCG [4] approach. The interrelationship between PCG approaches in industry and academia is very strong at the moment, yielding discussions for the benefit of both ends.

IV. Discussion and Conclusions

This paper has initially identified the currently ten most active areas within game AI research and placed them on three holistic frameworks: an AI method mapping, a game stakeholder (end user) taxonomy and the player-game interaction loop. This analysis revealed dominant AI algorithms within particular areas as well as room for exploration of new methods within areas. In addition, it revealed the dissimilar impact of different areas on different end users such as the AI researcher and the designer and, finally, outlined the influence of the different game AI areas on the player, the game and their interaction.

From the high level analysis of the game AI field the paper moved onto the detailed analysis of the game AI areas that compose it and thoroughly surveyed the meaningful interconnections between those ten different research areas. It would be impossible to provide a complete survey of these areas within the space of a single paper, and that has not been the objective of the current paper. This means that the bibliography is indicative rather than exhaustive. The result of the analysis is a total of 52 (out of the 90 possible) interconnections identified between the ten areas (see Table II). By observing the summary of interconnections between the areas we can derive highly influential areas such as game AI benchmarks, NPC behavior learning, PCG and general game AI. Viewing Table II as columns, (or the incoming influences of all figures in Section III) instead, the most influenced research areas are clearly PCG and commercial game AI.

The total number of existing and strong influences is rather small (6 in total) compared to the existing and weak (33) and potentially strong (13) influences which clearly signals the research capacity of the game AI field for further explorations. We can distinguish a number of connections which are currently very active, in the meaning that much

[4] http://www.wilddivine.com/

[5] https://www.empatica.com/

TABLE II

A unified summary of all existing and strong (●), existing and weak (○) and potentially strong (⋆) interconnections between the game AI areas. Rows inform columns. NPC is NPC behavior learning, S&P is Search and Planning, PM is Player Modeling, AI Bench. is AI as game benchmarks, CN is Computational Narrative, BA is Believable Agents, AI-Ass. is AI-Assisted Game Design, GGAI is General Game AI and Com. AI is AI in Commercial Games.

	NPC	S&P	PM	AI Bench.	PCG	CN	BA	AI-Ass.	GGAI	Com. AI
NPC	—		○	○	○		○	○	●	
S&P	○	—		○	○	●	●		●	○
PM			—		○	○	○	⋆		○
AI Bench.	●	○	○	—	○		○		●	⋆
PCG	○	⋆		○	—			○	⋆	○
CN					○	—	⋆			○
BA			○	○		○	—			⋆
AI-Ass.					⋆	⋆	○	—		○
GGAI	⋆	⋆	⋆	⋆	○			⋆	—	
Com. AI	○	○	○		○					—

work currently goes on in one area that draws on work in another area. Here we see for example the three-way connection between NPC behavior learning, search and planning and general game AI: the MCTS algorithm was invented in the context of board game-playing, proved to be really useful in the general game playing competition, and is currently being investigated for use in games as different as *StarCraft* (Blizzard Entertainment, 1998) and *Super Mario Bros* (Nintendo, 1985). Improvements and modifications to the algorithm has been flowing back and forth between the various areas. Another indicative connection that is alive and strong is between player modeling and procedural content generation, where it is now common for newly devised PCG algorithms and experimental studies to include player or player experience models.

One can also study the currently strong areas by trying to cluster the trending topics in recent iterations of the IEEE CIG and AIIDE conferences. Such studies always include some form of selection bias, as papers can usually be counted into more than one area (e.g. depending on if you group by method or domain), but if you start from the session groupings made by the program chairs of each conference you achieve at least some inter-subjective validity. According to such a clustering, the most active topics over the last few years have been player (or emotion) modeling, MCTS, real-time strategy game playing (especially StarCraft), PCG (in general) and narrative generation. Another perspective of the trend in game AI research is the varying percentage of studies on NPC (or game agent) behavior learning over other uses of AI in games at the two key conferences in the field (IEEE CIG and AIIDE). Our preliminary calculations suggest that while, initially, AI was mainly applied for NPC control and for playing board/card games well — more than 75% of CIG and AIIDE papers link to NPC behavior and agent game playing in 2005 — that trend has drastically changed as entirely new (non-traditional) uses of AI became more common over the years — e.g. roughly 52% of the papers in CIG and AIIDE in 2011 do not involve game agents and NPC behavior. These facts indicate a shift in the use of AI in and for games towards multiple non-traditional applications — which tend to be traditional by now — for the development of better games [10].

But it is maybe even more interesting to look at all those connections that are unexploited or underexploited or potentially strong. For example, player modeling is potentially very important in the development of believable agents, but this has not been explored in enough depth yet; the same holds for the application for user (or else, designer) modeling principles towards the personalization of AI-assisted game design. Believable agents have, in turn, not been used enough in PCG and AI-assisted game design. The construction of game-based AI benchmarks have much to learn from progress in general game AI and commercial game AI would have much to learn from looking at academic progress in areas such as player modeling, AI-assisted game design and computational narrative. There is a persisting idea among some commercial game developers that academic game AI research is (still) only about NPC behavior learning.

We hope that with this paper, we have been able to give the readers a sense of how this by now rather large and multifaceted research field hangs together, and what could be done to integrate it further. We realize that this is only our view of its dynamics and interconnections, and that there are or could be many competing views. We look forward to seeing those in upcoming studies in the field.

Acknowledgments

Thanks to the participants of Dagstuhl seminar 12191, and our various collaborators and students for seeding many of the ideas that went into this paper. Special thanks to Mirjam P. Eladhari for discussions, and to the anonymous reviewers for their insightful comments which helped us make this paper better. This research was supported, in part, by the FP7 ICT project C2Learn (project no: 318480).

References

[1] S. M. Lucas, M. Mateas, M. Preuss, P. Spronck, and J. Togelius, "Artificial and computational intelligence in games (dagstuhl seminar 12191)." *Dagstuhl Reports*, vol. 2, no. 5, pp. 43–70, 2012.

[2] ——, *Artificial and Computational Intelligence in Games*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH, 2013.

[3] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, no. 1, pp. 1–43, 2012.

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

[5] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, "Search-based procedural content generation: A taxonomy and survey," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 172–186, 2011.

[6] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. Andre, "Player modeling," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[7] M. Riedl and V. Bulitko, "Interactive narrative: A novel application of artificial intelligence for computer games." in *AAAI*, 2012.

[8] M. O. Riedl and A. Zook, "AI for game production," in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–8.

[9] P. Hingston, C. B. Congdon, and G. Kendall, "Mobile games with intelligence: A killer application?" in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–7.

[10] G. N. Yannakakis, "Game AI revisited," in *Proceedings of the 9th conference on Computing Frontiers*, ser. CF '12. New York, NY, USA: ACM, 2012, pp. 285–292.

[11] S. M. Lucas and G. Kendall, "Evolutionary computation and games," *Computational Intelligence Magazine, IEEE*, vol. 1, no. 1, pp. 10–18, 2006.

[12] S. M. Lucas, "Computational intelligence and games: Challenges and opportunities," *International Journal of Automation and Computing*, vol. 5, no. 1, pp. 45–57, 2008.

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

[14] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[15] J. Togelius, T. Schaul, D. Wierstra, C. Igel, F. Gomez, and J. Schmidhuber, "Ontogenetic and phylogenetic reinforcement learning," *Künstliche Intelligenz*, vol. 23, no. 3, pp. 30–33, 2009.

[16] G. Tesauro, "Temporal difference learning and td-gammon," *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[17] P. Avery, S. Louis, and B. Avery, "Evolving coordinated spatial tactics for autonomous entities using influence maps," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*. IEEE, 2009, pp. 341–348.

[18] J. B. Pollack, A. D. Blair, and M. Land, "Coevolution of a backgammon player," in *Artificial Life V: Proc. of the Fifth Int. Workshop on the Synthesis and Simulation of Living Systems*. Cambridge, MA: The MIT Press, 1997, pp. 92–98.

[19] S. Samothrakis, S. Lucas, T. P. Runarsson, and D. Robles, "Coevolving game-playing agents: Measuring performance and intransitivities," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 2, pp. 213–226, 2013.

[20] T. P. Runarsson and S. M. Lucas, "Coevolution versus self-play temporal difference learning for acquiring position evaluation in small-board go," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 6, pp. 628–640, 2005.

[21] J. L. Bernier, C. I. Herráiz, J. Merelo, S. Olmeda, and A. Prieto, "Solving mastermind using gas and simulated annealing: a case of dynamic constraint optimization," in *Parallel Problem Solving from NaturePPSN IV*. Springer, 1996, pp. 553–563.

[22] K.-J. Kim, H. Choi, and S.-B. Cho, "Hybrid of evolution and reinforcement learning for othello players," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*. IEEE, 2007, pp. 203–209.

[23] H. Munoz-Avila, C. Bauckhage, M. Bida, C. Bates Congdon, and G. Kendall, "Learning and game ai," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[24] M. Sipper, *Evolved to Win*. Lulu. com, 2011.

[25] R. Miikkulainen, B. D. Bryant, R. Cornelius, I. V. Karpov, K. O. Stanley, and C. H. Yong, "Computational intelligence in games," *Computational Intelligence: Principles and Practice*, pp. 155–191, 2006.

[26] N. Van Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber, "Robust player imitation using multiobjective evolution," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 652–659.

[27] J. Ortega, N. Shaker, J. Togelius, and G. N. Yannakakis, "Imitating human playing styles in super mario bros," *Entertainment Computing*, vol. 4, no. 2, pp. 93 – 104, 2013.

[28] S. Karakovskiy and J. Togelius, "The mario ai benchmark and competitions," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, no. 1, pp. 55–67, 2012.

[29] S. M. Lucas, "Ms pac-man competition," *ACM SIGEVOlution*, vol. 2, no. 4, pp. 37–38, 2007.

[30] S. Ontanon, 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.

[31] J. Togelius, S. M. Lucas, H. Duc Thang, J. M. Garibaldi, T. Nakashima, C. H. Tan, I. Elhanany, S. Berant, P. Hingston, R. M. MacCallum, T. Haferlach, A. Gowrisankar, and P. Burrow, "The 2007 ieee cec simulated car racing competition," *Genetic Programming and Evolvable Machines*, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10710-008-9063-0

[32] D. Loiacono, P. L. Lanzi, J. Togelius, E. Onieva, D. A. Pelta, M. V. Butz, T. D. Lnneker, 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.

[33] M. Kerssemakers, J. Tuxen, J. Togelius, and G. N. Yannakakis, "A procedural procedural level generator generator," in *Computational Intelligence and Games (CIG), 2012 IEEE Conference on*. IEEE, 2012, pp. 335–341.

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

[35] J. Togelius, G. N. Yannakakis, N. Shaker, and S. Karakovskiy, "Assessing believability," in *Believable Bots*, P. Hingston, Ed., 2012.

[36] J. Schrum, I. V. Karpov, and R. Miikkulainen, "Ut2: Human-like behavior via neuroevolution of combat behavior and replay of human traces," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*. IEEE, 2011, pp. 329–336.

[37] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proceedings of ACM Conference on Foundations of Digital Games*, 2013.

[38] J. Reisinger, E. Bahçeci, I. Karpov, and R. Miikkulainen, "Coevolving strategies for general game playing," in *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*. IEEE, 2007, pp. 320–327.

[39] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100–107, 1968.

[40] P. I. Cowling, M. Buro, M. Bida, A. Botea, B. Bouzy, M. V. Butz, P. Hingston, H. Munoz-Avila, D. Nau, and M. Sipper, "Search in real-time video games," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[41] A. Botea, B. Bouzy, M. Buro, C. Bauckhage, and D. Nau, "Pathfinding in games," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[42] D. B. Fogel, *Blondie24: Playing at the Edge of AI*. Morgan Kaufmann, 2001.

[43] S. Bojarski and C. Congdon, "Realm: A rule-based evolutionary computation agent that learns to play mario," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium*, 2010, pp. 83–90.

[44] J. Togelius, S. Karakovskiy, and R. Baumgarten, "The 2009 mario ai competition," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*. IEEE, 2010, pp. 1–8.

[45] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: A mixed-initiative level design tool," in *Proceedings of the Fifth International Confer-*

ence on the Foundations of Digital Games. ACM, 2010, pp. 209–216.

[46] R. Young, M. 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.

[47] M. J. Nelson and M. Mateas, "Search-based drama management in the interactive fiction anchorhead," 2005.

[48] F. Charles, M. Lozano, S. J. Mead, A. F. Bisquerra, and M. Cavazza, "Planning formalisms and authoring in interactive storytelling," in *Proceedings of TIDSE*, 2003.

[49] B. De Carolis, C. Pelachaud, I. Poggi, and M. Steedman, "Apml, a markup language for believable behavior generation," in *Life-Like Characters*. Springer, 2004, pp. 65–85.

[50] W. R. Swartout, J. Gratch, R. W. Hill Jr, E. Hovy, S. Marsella, J. Rickel, D. Traum, *et al.*, "Toward virtual humans," *AI Magazine*, vol. 27, no. 2, p. 96, 2006.

[51] Y. Björnsson and H. Finnsson, "Cadiaplayer: A simulation-based general game player." *IEEE Trans. Comput. Intellig. and AI in Games*, vol. 1, no. 1, pp. 4–15, 2009.

[52] J. Orkin, "Three states and a plan: the ai of fear," in *Game Developers Conference*, vol. 2006. Citeseer, 2006.

[53] N. R. Sturtevant, "Benchmarks for grid-based pathfinding," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, no. 2, pp. 144–148, 2012.

[54] A. M. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, "An inclusive taxonomy of player modeling," *University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13*, 2011.

[55] G. N. Yannakakis and A. Paiva, "Emotion in games," in *Handbook of Affective Computing*. Oxford University Press, 2014.

[56] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *Computational Intelligence and Games, 2008. CIG'08. IEEE Symposium On*. IEEE, 2008, pp. 111–118.

[57] G. N. Yannakakis, H. P. Martínez, and A. Jhala, "Towards affective camera control in games," *User Modeling and User-Adapted Interaction*, vol. 20, no. 4, pp. 313–340, 2010.

[58] P. Burelli and G. N. Yannakakis, "Combining Local and Global Optimisation for Virtual Camera Control," in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. Copenhagen, Denmark: IEEE, August 2010, pp. 403–401.

[59] N. Shaker, G. N. Yannakakis, and J. Togelius, "Towards automatic personalized content generation for platform games." in *AIIDE*, 2010.

[60] S. Bakkes, P. Spronck, and J. Van Den Herik, "Rapid and reliable adaptation of video game ai," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 2, pp. 93–104, 2009.

[61] R. Lopes and R. Bidarra, "Adaptivity challenges in games and simulations: a survey," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 2, pp. 85–99, 2011.

[62] M. Mateas and A. Stern, "Façade: An experiment in building a fully-realized interactive drama," in *Game Developers Conference, Game Design track*, vol. 2, 2003, p. 82.

[63] R. Aylett, S. Louchart, J. Dias, A. Paiva, and M. Vala, "Fearnot!–an experiment in emergent narrative," in *Intelligent Virtual Agents*. Springer, 2005, pp. 305–316.

[64] N. Van Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber, "Robust player imitation using multiobjective evolution," in *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*. IEEE, 2009, pp. 652–659.

[65] G. G. Jorge Munoz and A. Sanchis, "Towards imitation of human driving style in car racing games," in *Believable Bots: Can Computers Play Like People?*, P. Hingston, Ed. Springer, 2012.

[66] I. V. Karpov, J. Schrum, and R. Miikkulainen, "Believable bot navigation via playback of human traces," in *Believable Bots: Can Computers Play Like People?*, P. Hingston, Ed. Springer, 2012.

[67] A. Liapis, G. N. Yannakakis, and J. Togelius, "Adapting models of visual aesthetics for personalized content creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012.

[68] A. Drachen, C. Thurau, J. Togelius, G. N. Yannakakis, and C. Bauckhage, "Game data mining," in *Game Analytics*. Springer, 2013, pp. 205–253.

[69] M. Seif El-Nasr, A. Canossa, and A. Drachen, *Game Analytics*. Springer, 2013.

[70] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in tomb raider: Underworld," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*. IEEE, 2009, pp. 1–8.

[71] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, "Predicting player behavior in tomb raider: Underworld," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 178–185.

[72] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," *KanGAL Report*, vol. 2005005, 2005.

[73] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík, "Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009," in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*. ACM, 2010, pp. 1689–1696.

[74] F. Gomez, J. Schmidhuber, and R. Miikkulainen, "Efficient non-linear control through neuroevolution," in *Machine Learning: ECML 2006*. Springer, 2006, pp. 654–662.

[75] P. Hingston, "A Turing test for computer game bots," *IEEE Transactions on Computational Intelligence and AI In Games*, vol. 1, no. 3, 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5247069

[76] J. Hagelbäck, M. Preuss, and B. Weber, "CIG 2010 StarCraft RTS AI Competition," 2010.

[77] M. Buro and D. Churchill, "Real-time strategy game competitions." *AI Magazine*, vol. 33, no. 3, pp. 106–108, 2012.

[78] 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.

[79] S. M. Lucas, "Cellz: a simple dynamic game for testing evolutionary algorithms," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 1. IEEE, 2004, pp. 1007–1014.

[80] D. Perez, P. Rohlfshagen, and S. M. Lucas, "The physical travelling salesman problem: WCCI 2012 competition," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[81] M. Newborn, *Computer chess*. John Wiley and Sons Ltd., 2003.

[82] J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, and S. Sutphen, "Checkers is solved," *Science*, vol. 317, no. 5844, pp. 1518–1522, 2007.

[83] C.-S. Lee, M.-H. Wang, G. Chaslot, J.-B. Hoock, A. Rimmel, O. Teytaud, S.-R. Tsai, S.-C. Hsu, and T.-P. Hong, "The computational intelligence of mogo revealed in taiwan's computer go tournaments," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 1, pp. 73–89, 2009.

[84] D. Churchill and M. Buro, "Build order optimization in starcraft," in *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.

[85] 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.

[86] 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, 2011, pp. 547–556.

[87] L. Chen, R. T. Rose, Y. Qiao, I. Kimbara, F. Parrill, H. Welji, T. X. Han, J. Tu, Z. Huang, M. Harper, *et al.*, "Vace multimodal meeting corpus," in *Machine Learning for Multimodal Interaction*. Springer, 2006, pp. 40–51.

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

[89] N. Love, T. Hinrichs, D. Haley, E. Schkufza, and M. Genesereth, "General game playing: Game description language specification," 2008.

[90] J. Levine, C. B. Congdon, M. Ebner, G. Kendall, S. M. Lucas, R. Miikkulainen, T. Schaul, and T. Thompson, "General video game playing," *Dagstuhl Follow-Ups*, vol. 6, 2013.

[91] J. a. Togelius, "Towards a video game description language," *Dagstuhl Follow-Ups*, vol. 6, 2013.

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

[93] 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.

[94] A. Liapis, G. N. Yannakakis, and J. Togelius, "Computational Game Creativity," in *Proceedings of the Fifth International Conference on Computational Creativity*, 2014, pp. 285–292.

[95] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: a Textbook and Overview of Current Research*. Springer, 2014.

[96] 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.

[97] W. L. Raffe, F. Zambetta, and X. Li, "A survey of procedural terrain generation techniques using evolutionary algorithms," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–8.

[98] R. van der Linden, R. Lopes, and R. Bidarra, "Procedural generation of dungeons," 2013.

[99] M. Kemmerling and M. Preuss, "Automatic adaptation to generated content via car setup optimization in torcs," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 131–138.

[100] D. A. Ashlock, T. W. Manikas, and K. Ashenayi, "Evolving a diverse collection of robot path planning problems," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*. IEEE, 2006, pp. 1837–1844.

[101] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Interactive evolution for the procedural generation of tracks in a high-end racing game," in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 395–402.

[102] R. M. Smelik, T. Tutenel, K. J. de Kraker, and R. Bidarra, "Interactive creation of virtual worlds using procedural sketching," in *Proceedings of eurographics*, 2010.

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

[104] S. Risi, J. Lehman, D. B. D'Ambrosio, R. Hall, and K. O. Stanley, "Combining search-based procedural content generation and social gaming in the petalz video game." in *AIIDE*, 2012.

[105] E. J. Hastings, R. K. Guha, and K. O. Stanley, "Automatic content generation in the galactic arms race video game," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 1, no. 4, pp. 245–263, 2009.

[106] J. Juul, "Games telling stories," *Game studies*, vol. 1, no. 1, p. 45, 2001.

[107] E. Aarseth, "Genre trouble," *Electronic Book Review*, vol. 3, 2004.

[108] S. Giannatos, Y.-G. Cheong, M. J. Nelson, and G. N. Yannakakis, "Generating narrative action schemas for suspense," in *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2012.

[109] J. Valls-Vargas, S. Ontanón, and J. Zhu, "Towards story-based content generation: From plot-points to maps," in *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*. IEEE, 2013, pp. 1–8.

[110] D. Roberts, H. Narayanan, and C. Isbell, "Learning to influence emotional responses for interactive storytelling," in *Proceedings of the 2009 AAAI Symposium on Intelligent Narrative Technologies II*, 2009.

[111] M. Cavazza, F. Charles, and S. J. Mead, "Character-based interactive storytelling," *Intelligent Systems, IEEE*, vol. 17, no. 4, pp. 17–24, 2002.

[112] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, "Toward supporting stories with procedurally generated game worlds," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*. IEEE, 2011, pp. 297–304.

[113] J. McCoy, M. Treanor, B. Samuel, A. A. Reed, N. Wardrip-Fruin, and M. Mateas, "Prom week," in *Proceedings of the International Conference on the Foundations of Digital Games*. ACM, 2012, pp. 235–237.

[114] R. Evans and E. Short, "Versu – a simulationist storytelling system," *IEEE Transactions on Computational Intelligence and AI in Games*, 2014.

[115] H. Campos, J. Campos, J. Cabral, C. Martinho, J. H. Nielsen, and A. Paiva, "My dream theatre," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 1357–1358.

[116] M. P. Eladhari and M. Sellers, "Good moods: outlook, affect and mood in dynemotion and the mind module," in *Proceedings of the 2008 Conference on Future Play: Research, Play, Share*. ACM, 2008, pp. 1–8.

[117] C. Thurau, C. Bauckhage, and G. Sagerer, "Learning human-like movement behavior for computer games," in *Proc. Int. Conf. on the Simulation of Adaptive Behavior*, 2004, pp. 315–323.

[118] J. Orkin and D. Roy, "The restaurant game: Learning social behavior and language from thousands of players online," *Journal of Game Development*, vol. 3, no. 1, pp. 39–60, 2007.

[119] P. Hingston, *Believable Bots: Can Computers Play Like People?* Springer, 2012.

[120] B. Soni and P. Hingston, "Bots trained to play like a human are more fun," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 363–369.

[121] N. Shaker, J. Togelius, G. N. Yannakakis, L. P. K. Satish, V. S. Ethiraj, S. J. Johansson, R. Reynolds, L. Kinnaird-Heether, T. Schumann, and M. Gallagher, "The Turing Test Track of the 2012 Mario AI Championship: Entries and Evaluation," in *Proc. IEEE Computational Intelligence and Games*, 2013.

[122] M. Riedl and A. Stern, "Believable agents and intelligent story adaptation for interactive storytelling," *Technologies for Interactive Digital Storytelling and Entertainment*, pp. 1–12, 2006.

[123] A. Paiva, J. Dias, D. Sobral, R. Aylett, P. Sobreperez, S. Woods, C. Zoll, and L. Hall, "Caring for agents and agents that care: Building empathic relations with synthetic agents," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1*. IEEE Computer Society, 2004, pp. 194–201.

[124] B. Pell, "Strategy generation and evaluation for meta-game playing," Ph.D. dissertation, University of Cambridge, 1993.

[125] J. Togelius, "A procedural critique of deontological reasoning," in *Proceedings of DiGRA*, 2011.

[126] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 3, pp. 201–215, 2011.

[127] M. Shaker, N. Shaker, and J. Togelius, "Evolving playable content for cut the rope through a simulation-based approach," in *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2013.

[128] G. Nelson, "Natural language, semantic analysis, and interactive fiction," *IF Theory Reader*, p. 141, 2006.

[129] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Real-time neuroevolution in the nero video game," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 6, pp. 653–668, 2005.

[130] I. V. Karpov, L. Johnson, and R. Miikkulainen, "Evaluation methods for active human-guided neuroevolution in games," *Evaluation Methods for Active Human-Guided Neuroevolution in Games. Dept. of Computer Science, The University of Texas at Austin*, vol. 1616, 2012.

[131] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient world: Human-based procedural cartography," in *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. Springer, 2013, pp. 180–191.

[132] Interactive Data Visualization, "Speedtree," 2010. [Online]. Available: http://www.speedtree.com/

[133] A. M. Smith, M. J. Nelson, and M. Mateas, "Ludocore: A logical game engine for modeling videogames," in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 91–98.

[134] B. Goertzel and C. Pennachin, *Artificial general intelligence*. Springer, 2007.

[135] S. Legg and M. Hutter, "Universal intelligence: A definition of machine intelligence," *Minds and Machines*, vol. 17, no. 4, pp. 391–444, 2007.

[136] 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.

[137] M. Hausknecht, J. Lehman, R. Miikkulainen, and P. Stone, “A Neuroevolution Approach to General Atari Game Playing,” *IEEE Transactions on Computational Intelligence and AI in Games*, 2014, (in press).

[138] H. P. Martínez, M. Garbarino, and G. N. Yannakakis, “Generic physiological features as predictors of player experience,” in *Affective Computing and Intelligent Interaction*. Springer, 2011, pp. 267–276.

[139] A. M. Smith and M. Mateas, “Variations forever: Flexibly generating rulesets from a sculptable design space of mini-games,” in *Computational Intelligence and Games (CIG), 2010 IEEE Symposium on*. IEEE, 2010, pp. 273–280.

[140] T. Mahlmann, J. Togelius, and G. N. Yannakakis, “Modelling and evaluation of complex scenarios with the strategy game description language,” in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*. IEEE, 2011, pp. 174–181.

[141] J. M. Font, T. Mahlmann, D. Manrique, and J. Togelius, “Towards the automatic generation of card games through grammar-guided genetic programming,” in *Proceedings of the Conference on the Foundations of Digital Games*, 2013.

[142] D. Isla, “Handling complexity in the Halo 2 AI,” in *Game Developers Conference*, vol. 12, 2005.

[143] C.-U. Lim, R. Baumgarten, and S. Colton, “Evolving behaviour trees for the commercial game defcon,” in *Applications of Evolutionary Computation*. Springer, 2010, pp. 100–110.

[144] D. Perez, M. Nicolau, M. O’Neill, and A. Brabazon, “Evolving behaviour trees for the Mario AI competition using grammatical evolution,” in *Applications of Evolutionary Computation*. Springer, 2011, pp. 123–132.

[145] S. Ontañón, K. Mishra, N. Sugandh, and A. Ram, “Case-based planning and execution for real-time strategy games,” in *Case-Based Reasoning Research and Development*. Springer, 2007, pp. 164–178.

Georgios N. Yannakakis (S04–M05) is an Associate Professor at the Institute of Digital Games, University of Malta. He does research at the crossroads of AI, affective computing, advanced game technology, and computational creativity. He has published over 150 journal and international conference papers in the aforementioned fields. His research has been funded by several national and EU grants and has appeared in *Science Magazine* and *New Scientist* among other venues. He is an Associate Editor of the IEEE TRANS. ON AFFECTIVE COMPUTING and the IEEE TRANS. ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES.

Julian Togelius (S05–M07) is an Associate Professor at the Center for Computer Games Research, IT University of Copenhagen, Denmark. He works on all aspects of computational intelligence and games and on selected topics in evolutionary computation and evolutionary reinforcement learning. His current main research directions involve search-based procedural content generation in games, game adaptation through player modeling, automatic game design, and fair and relevant benchmarking of game AI through competitions. He is a past chair of the IEEE CIS Technical Committee on Games, and an associate editor of IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND GAMES. Togelius holds a BA from Lund University, an MSc from the University of Sussex, and a PhD from the University of Essex.