A Taxonomy of Video Games and AI

E. A. A. Gunn and **B. G. W. Craenen** and **E. Hart**[1]

Abstract. Game developers have always striven to create better and more complex games. The use of artificial intelligence (AI) in games is increasing in game development to this end. However there appears to be a mismatch of implications and expectations between game developers and AI researchers of what AI can bring to games, mostly due to differences of interpretation of the problems involved. To overcome this we present a taxonomy to aid knowledge transfer between the two groups, including elements from environments, game theory, and information theory. We also identify other concepts of games and AI that could be important to both sides.

1 INTRODUCTION

Game developers have always striven to create better and more complex games. Despite advances in technology for rendering engines, physics engines, dynamic lighting and so on to this end, one area that is currently lacking is that of believable interactions with non-player-characters (NPCs) within games. Currently these NPC interactions can be described as static and deterministic, without the player getting a sense of character about a particular NPC, or indeed the game itself. Static and deterministic game-play means that game-play can degenerate into reacting to the behaviour of the NPCs rather than navigating through the game or its story line. This often leads to a "shallow and unfulfilling" game experience [10].

Increasingly, game developers are looking toward artificial intelligence (AI) to improve NPC behaviour. However, Fairclough et al [10] states there is little implementation of AI in computer games and identified a number of reasons why this might be the case: a lack of computer resources; suspicion by game developers as to the non-determinism of AI methods; a lack of development time; and a lack of understanding of the scope of AI.

On the other hand, academic research into AI within games is increasing to allow more dynamic, and more realistic games. However existing work is primarily focussed on the specific implementation of AI methodologies in specific problem areas. For example, the use of neuro-evolution to train behaviour in *NERO* [7]. With greater analysis of the problems faced in implementing AI methods in computer games, more accurate and efficient methodologies can be developed to create more realistic behaviour of artificial characters within games.

Overall, there seems to be a miscommunication between game-developers on the one hand and AI researchers on the other. The game-developers perhaps lack precise knowledge about AI methodologies and are consequently wary of implementing them into their games. AI researchers for their part perhaps lack a broader awareness of the requirements of games and where various AI methodologies could provide a solution.

[1] Napier University, Edinburgh, email: e.gunn@napier.ac.uk

We therefore suggest this taxonomy to facilitate communication between game developers on the one hand and AI researchers on the other, thus bridging the knowledge gap between the two. As far as we are aware, there is currently no such taxonomy, however we have found previous general classifications that provided useful background information for the taxonomy, and confirmed some of our observations. Konzack [16] proposes a critical analysis methodology for games, and goes through a specific example for his methodology. However, the areas classified are high-level and the discussion too general for direct use within our taxonomy. Aarseth [1] specifically points out the disparity between the classifications proposed by Konzack, and looks at the classification problem from an aesthetic point of view, more suitable to a social science methodology. Continuing in the theme of general classifications of games, and unlike our focus on using AI in games, Lindley [18] proposes a taxonomy of video games in general.

Our taxonomy represents a more technical view on the involvement of AI in computer games. We believe that further research to create an overall ontology to allow free discussion using a common language will help cement any interactions between game developers and AI researchers. Although we go some way in doing this by using existing concepts with complete definitions, a broader language would facilitate communication even further.

Our taxonomy (illustrated in figure 1 for reference throughout the paper), provides a bridge between game developers and AI. It can, however, also serve another purpose: Laird and van Lent [17] discuss the development of game AI as an aid to one of the original purposes of AI, that of creating human-level intelligence in an artificial system. Through use of this taxonomy, methods developed as an aid to game intelligence can be used in other spheres of research within AI. With better understanding of *why* certain AIs work in certain game worlds and types, we can thus specify *how* specific concepts within games can be built upon and generalised outside of games. Kleiner [15] reinforces this idea and applies a more methodological view of the situation, concluding that the vast sums of money involved in the game industry can benefit and accelerate general AI research. A recent article in *IEEE Spectrum* [25] discusses the possibilities.

Furthermore, AI researchers can analyse their AI methodologies, match them to the theoretic concepts in our taxonomy, and thus identify game types that would describe how they could create their own game, or use an existing game, as a core laboratory for their research.

This paper begins with a discussion on game genres in section 1.1 as initial background; analyses the environmental and player elements of game types in section 2; applies game theoretic concepts to the game type elements extracted in section 3; and finally maps AI methodologies to the game theoretic concepts in section 4. We follow with a number of useage scenarios to illustrate how our taxonomy can be used in section 5 before providing a conclusion in section 6.

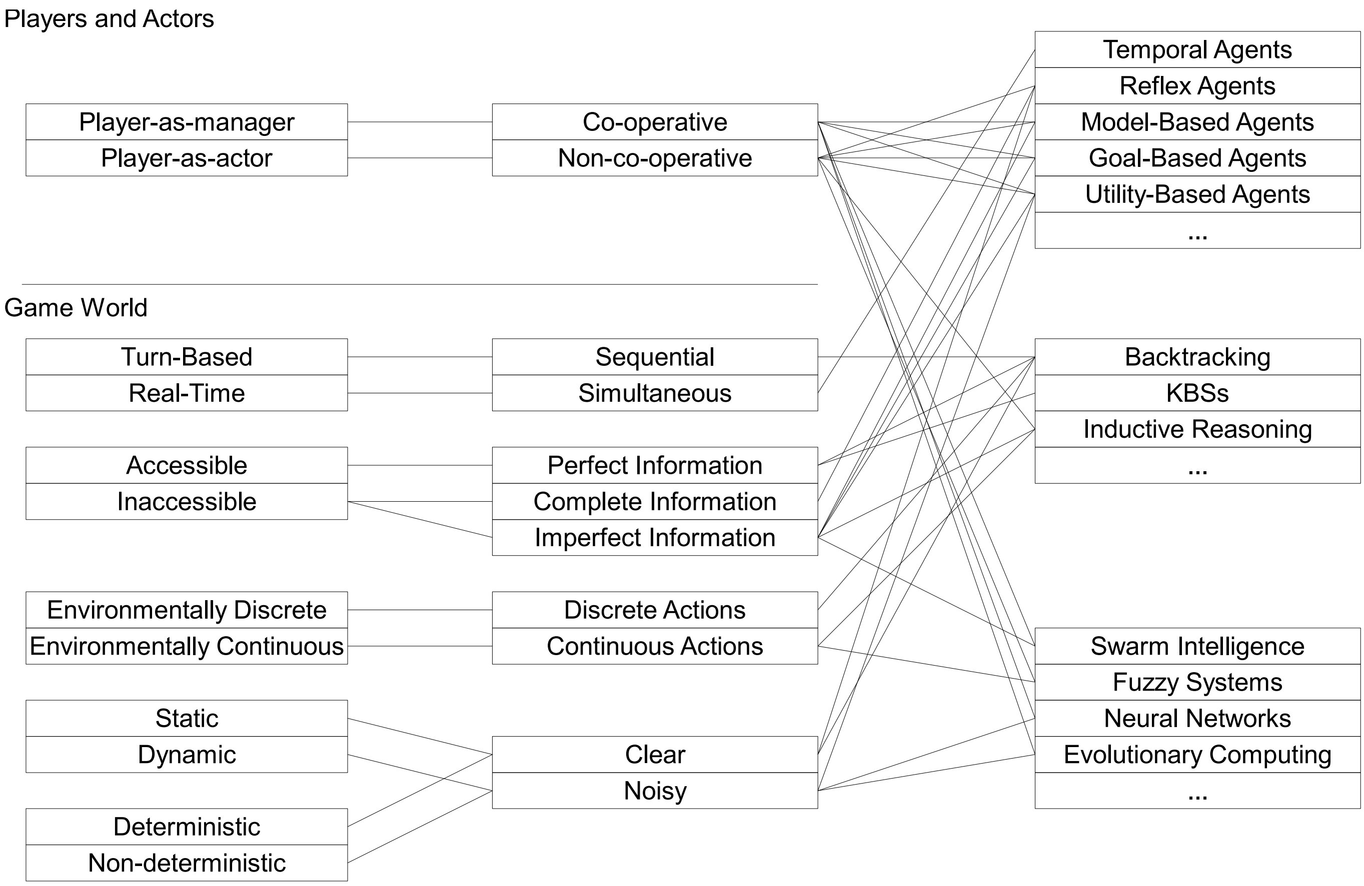

Figure 1. The complete taxonomy

1.1 Game Genres

The most logical place to start with a taxonomy of a subject is to use already existing classification schemes: within games, that would be the genres that are currently used. A comprehensive study and classification of genres has already been conducted [27]. However, Apperley [2] makes the case that current genres are merely representational: existing genres are broken down in part due to their visual aesthetics, and in part by the subject and content of the games in consideration. What is missing is any form of delineation based on interactivity.

He goes on to say that the primary problem with current genres of computer games is what is described by Bolter and Grusin [3] as a "logic of remediation", the "formal logic by which new media refashion prior media forms": placing games into categories that already exist for media types such as novels and films. Frasca [11] creates this delineation by referring to two aspects of categorisation of games: *narratological* categorisation, based on the subject, content and representation of the game (current genre analysis); and *ludological* categorisation, which is concerned with the rules, mechanisms, rewards and structure of games.

This taxonomy is designed for game developers and AI academics, both of which are more interested in the ludological categorisation of games. We therefore rejected the use of the narratological categorisation within the taxonomy. Instead we focus on the components of the games and delineate them using the ludological categorisation. We provide a short excerpt of the narratological categorisation here purely as background for the taxonomy.

1.1.1 Action

Action games are fast-paced, requiring quick judgement and snap decisions. These primarily involve interaction with computer-controlled game actors who help or hinder the player through the course of the game. The player usually controls just one character. Generally, the goals within these games are simple, however the challenge is in accomplishing these goals by navigating through a level consisting of hostile enemy NPCs.

1.1.2 Adventure

Games where there is a rigid structure to the game - for example sequence based movement throughout the game, where the player is presented with only a picture of the surroundings - fall under the category of adventure games. These games generally have interesting, long and complex story-lines, and the ultimate goal of the game is discovered through the course of the story.

1.1.3 Role-playing

Modern computer role-playing games (CRPGs) allow the player to take on the "role" of a character, directly controlling that character in the game world. One of the key elements of RPGs is the idea of character advancement, where throughout the course of the game the activities of the character are rewarded by allowing the player to advance their character to become better at certain activities. The goals of CRPGs vary wildly depending on the setting, history and type of character the player plays. Most modern RPGs also provide an environment for enthusiastic players to create their own settings. This module system is being adopted by most game genres, and can provide an ideal opportunity for testing game intelligences.

1.1.4 Vehicle Simulation

This genre is subtly different to the preceding three - instead of controlling a character, the player controls a vehicle. This vehicle exist within a game world that includes abstractions of real-world physics to ensure that the player experiences as close to a "real" experience as possible. Vehicle simulations can be purely simulation based, or may have goals that need to be achieved depending on the type of vehicle available.

1.1.5 Strategy

Strategy games take the player from a character-based perspective to a more high-level perspective. In almost all cases, escalation of force and the creation of units is tied directly into resources which are collected by specialist units or structures. These resources are then used in the creation of other units and structures. The goal in most of these games is to complete an escalating campaign of several levels, the conclusion of which is the completion of the game. Individual maps can have differing objectives but the key component is the strategic distribution and use of units with varying abilities under the player's control.

1.1.6 Management

Like strategy games, management games (also called "god" games) involve a higher-level view of the game world, and involve resources with which the player can alter the game world, or change it in some way for the population of actors within the game world. Some of the games within this genre have no specific goals in mind - they are primarily simulation games - but others offer specific objectives.

1.1.7 4X

4X is a term used to represent games that require *eXploring, eXpanding, eXploiting, and eXtermination*, and the key difference between this and strategy games (of which 4X is sometimes classed as a subgenre) is the level of abstraction of the player: in strategy games the player is usually cast in the light of a theatre commander, sent there by their superiors, whereas in 4X games, the player is playing the role of a world leader. Some 4X games allow the player to also participate as a theatre commander in individual combat scenarios.

1.1.8 Life Simulation

Life simulation games are a relatively small genre of games, but perhaps of most interest to artificial intelligence researchers. Most AI researchers are familiar with *The Game of Life*, and life simulation games are much in the same vein. There is little in the way of goals or objectives, the challenge is to create or adapt a game world or configuration of actor to allow the successful survival of actors within the game world. Broadly speaking the simulations can be broken down into biologically inspired simulations and socially inspired simulations. Specific mention should be made of the large market for console games within Japan where a large component of the game is the normal day-to-day running of a life, and the popular "dating game" market, both of which involve a high level of social interaction with NPCs.

1.1.9 Puzzle

Puzzle games are generally small games where there are no other actors in the game - the game is completed by using logic and deduction to complete the goals.

1.2 Definitions

Throughout the paper the following definitions are used to differentiate between a number of key concepts that do occur together in the taxonomy.

- Game World
  A *game world* is a definition of the environment that the game takes place in, consisting of the rules and rewards of the game.
- Actor
  We shall define an actor as an entity that exists within a game world and that can interact with the game world, and/or other actors, and are under the control of one or more players as the main method of playing the game. These players can be human or artificial. However, this does not preclude the use of intelligence, as in many games the actors are semi-autonomous.
- Player
  We shall define a player as a controlling intelligence, either human or artificial, that plays the game. Where specific differentiation is needed between human player and artificial players, this has been explicitly mentioned.
- Participants
  We shall group actors and players together where necessary and call them *participants* within a game. This allows us to generalise concepts over multiple tiers of intelligence that may present themselves within games.

2 GAME TYPES

Genres mainly group the content of the games and the style of play, and have been mentioned as a background and to provide some observations about various types of games. Key factors in a game that affect our taxonomy have been extracted, and this provides the first layer of our taxonomy.

It is clear that there are two distinct areas that need to be considered in any game - that of the players and how they play the game; and the game itself - its rules and rewards (what we call the game world). This is a segregation that we will use throughout this and the next layer.

2.1 Players and Actors

Analysing how the players interact with a game is vital, as any artificial player designed to compete with a human player must be able to do so within the same rule-set as the human players themselves.

There is an interaction with and control of actors in the game world by a player, which can range from no actors to any number of actors (a one-to-zero-or-many link, see figure 2). There are two special cases that need to be considered: that of no actors, and that of one actor. The first case describes games where the player is competing against the world itself - manipulating it directly "from on high", such as in puzzle games and some social simulation games. We shall refer to these as *player-versus-environment* games. Player-versus-environment games generally have no intelligence as there is no artificial player to compete against (for example in puzzle games).

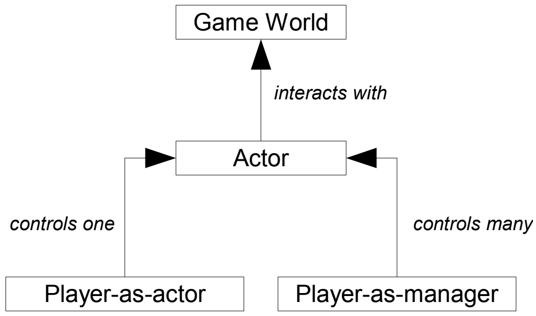

Figure 2. The player-actor-game-world interactions. Note that the player does not directly interact with the game world.

As such it is mentioned here for completeness. This does not affect the taxonomy.

The second special case is where one actor is considered the proxy for the player in the game world, and only through controlling and manipulating that character directly can the player achieve the objectives of the game. We shall call this *player-as-actor* interaction.

For numbers of actors greater than one, the game usually becomes more of a management game, where the player uses all actors under their control to complete the objectives of the game, and as such we shall refer to them as *player-as-manager* games.

2.2 Game World

We can use concepts from agent-based systems to help us categorise various types of game world by using the metaphor *a game world is an environment*. Russell and Norvig [24] provide us with a summary of environment types, which we can adapt for our purposes in the taxonomy. We need to view the environments, and thus the game worlds, from the actor level, no-matter how that actor is controlled. This is because of the way the player interacts with the game world: via an actor or actors. This is enables us to create an accurate taxonomy, as it is the actors that navigate and exist within the game world. We present these concepts in the following subsections.

2.2.1 Accessible vs. Inaccessible

The accessibility of a game world relates to the information of the game world that is available to each actor within the game world. If an actor has knowledge of every aspect of the game world and knows of everything that is going on within that game world, then the game world is *accessible* to that actor. If, however, there are limits to what an actor may know about the game world (for example using the concept of fog-of-war), then the game world is *inaccessible*.

2.2.2 Environmentally Discrete vs. Environmentally Continuous

Actors within the game world may make a number of possible actions at any point, determined by the range of potential actions within a game world. If there is a finite set of actions that an actor can take (for example only being able to move one square in any one of the cardinal directions on a grid), then the game world is *environmentally discrete*. Where there is a continuum of possible actions, such as allowing an actor to turn to any direction, then the game world is *environmentally continuous*.

2.2.3 Static vs. Dynamic

Artificial participants need time to consider their moves just as human players do - albeit they make conclusions significantly faster than any human player could achieve. If the game world alters whilst an participant is "thinking", then the game world is *dynamic*. If, however, the game world remains the same until an participant has made a move, then the game world is *static*. A third variety is *semi-dynamic*, where although the current state of the game might not alter whilst a participant is deliberating, the measure of performance of the participant reduces until a move is made. The example given to us by Russel and Norvig is that of time-measured chess games, and there are video games where a similar technique is used.

2.2.4 Deterministic vs. Non-deterministic

A game world is *deterministic* if the next state can be explicitly concluded from the present state of the game world and the actions carried out by the actors. If there is an element of uncertainty, or if the game world changes despite actions by the actors, then the game world is *non-deterministic*. There is a clear distinction in the case of an inaccessible game world, such that the deterministic nature of the game world should best be considered from the actor's perspective. Actions that are carried out without the actors knowledge may influence what that actor sees in the next state. So, despite the fact that the overall game world state is *deterministic*, from the actor's perspective, the game world is *non-deterministic*. Random elements within a game world also create a *non-deterministic* world.

A further definition of environment is provided by Russel and Norvig, that of episodic vs. non-episodic. If an actor can take an action, the results of which have no relation on future actions, the environment is episodic. If however the consequence of one action relates directly or indirectly to the available information or set of actions at a future point, the environment will be considered non-episodic. Such a definition is too strict to use in this taxonomy, as all actions within the course of a game have consequence.

2.2.5 Turn-Based vs. Real-time

Turn-based games place the players in a game-playing sequence. Whilst this type of game could be, theoretically, applied to any game, there are only a small number of genres where this mechanic is used, primarily 4X, strategy, some role-playing, and some life-simulation games. These games can require a great deal of strategic thinking, and as such having the time to analyse a situation and make decisions based on that is almost necessary.

A number of games also use *semi-turn-based* mechanics, where the player has the opportunity to pause the game to make decisions or queue up actions, and then return to normal *real-time* playing afterwards; or where certain sections of the game are turn-based, and the rest is real-time.

Non-turn-based games are called *real-time* games.

2.3 Other Game Type Considerations

The concepts identified provide clear definitions to allow us to categorise games for the taxonomy. However, there are other important concepts that relate to all games. We have identified the following.

2.3.1 Layered game-play

Not all games stick strictly to one method of game-play throughout the entire game. For example a turn-based mechanic during combat, and a fast-travel mechanic that gives an overview of the entire world, in which the player can move anywhere - despite the normal game-play being on a hexagonally divided game board in real-time. These parts of the game are separate, but connected. It is therefore necessary to identify each discrete game-play types within a layered game, and apply the taxonomy to each area to better apply research methods to each area as necessary.

2.3.2 Artificial Participants

Generally, games do not pitch the human player against a multitude of completely individual and separate actors. Usually there is at least one faction that the player is playing against. The impression is that all the artificial actors that the human player faces are given directives from the faction, and as such work together. This then creates the situation where we have *human-player-as-actor* against *artificial-player-as-manager*. Such cases need to be taken into consideration to allow the correct methodologies to be used in the correct places.

2.3.3 Hierarchical Intelligence

Player-as-manager games provide us with a potential hierarchy of intelligences that would be required: the artificial player and the artificial player's actors. Different AI methods would be required in this case, as the artificial player would require high-level strategic decision making. On the other hand, individual actors might only require reflexive behaviour (*i.e.* "I am being shot, I shall move away.") Currently in these types of games (especially strategy games) there is little intelligence at the artificial player level, merely consisting of such static tactics as "build up a force of x units, and send them along y path". Observation of such tactics in has shown that there is a reliance on some form of state analysis. By considering the hierarchical nature of the player and the actors under that player's control, suitable mechanisms can be introduced. First to provide adequate high-level strategic planning for the artificial player. Secondly to provide low-level tactical planning for the artificial player's actors.

2.3.4 Single Player vs. Multi-player

Single player games were the original type of video games - it was one human player at a time playing the game - and the "single" aspect refers to that single human player (as opposed to a single player overall, including artificial players). This is the key type of game that still exists today where good NPC AI benefits the immersion into the game of the player, and subsequently their enjoyment of that game. These games rely on involving, long story-lines, usually with a large number of optional sub-quests to lengthen the time of play. This is the type of game where unintelligent NPCs are most noticeable.

As with single-player games, multi-player games refer to the amount of *human* players in the game. Despite the fact that these games are primarily designed to be played with other human competitors, almost all allow the use of *bots* for off-line playing, where the human player combats artificial players. These bots are artificial actors; and intelligent, human-level AI in these bots would allow off-line players to still hone their skills when human opponents are not available.

Lastly, there is an increasingly popular type of game called "massively multi-player on-line" games (MMOs), where there can be tens of thousands of players all playing on one server cluster. This type of game provides a rich and ever changing set of data which could be used to train intelligences for that game.

Multi-player games present an ideal laboratory for testing the effectiveness of any artificial player being developed. We can create a Turing test where we can put an artificial player into the game without the knowledge of the human players. If the human players can see no difference between a human player and the artificial player under test, the test could be considered a success.

3 THEORETIC CONCEPTS

The game types defined in the first layer can be used to specify game theoretic concepts in the second layer of the taxonomy. These concepts are based on game theory and information theory and provide a bridge between games and game types to AI and AI methodologies.

3.1 Players and Actors

Only one concept from game theory applies to players and actors of relevance in this level, and that is the concept of co-operation. *Co-operative* games are those where the participants can form binding agreements on strategies, and there is a mechanism in place to enforce such behaviour [21]. *Non-co-operative* games are where every participant is out to maximise their own pay-off. Some games may have elements of both co-operative and non-co-operative behaviour: coalitions of participants enforce co-operative behaviour, but it is still possible for members of the coalition to perform better, or receive better rewards than the others if working alone. These are *hybrid* games.

In this taxonomy, if there is a mechanism strictly in place to prevent co-operating participants from breaking away and conducting their own behaviour then we can class that as a *co-operative* game.

Player-as-manager games naturally fall into the *co-operative* category, as the player is managing a team of actors with a shared goal, and with restrictions in place as to acting in a detrimental fashion to fellow team members. Conversely, *player-as-actor* games naturally fall into the *non-co-operative* category, as most games of this type pitch the player against all other actors.

3.2 Game World

3.2.1 Discrete vs. Continuous Actions

Discrete action games within game theory consist of a finite number of participants, turns, or outcomes, resulting in a finite set of strategies which can be plotted in a matrix format for evaluation. *Continuous action* games, however, can have participants joining and leaving the game, or the stakes changing between actions, resulting in a continuous set of strategies. This represents a subset of the potential actions that the game world allows.

Within our taxonomy, this relates directly to *environmentally discrete* and *environmentally continuous* game worlds.

3.2.2 Simultaneous vs. Sequential

In direct relation to *turn-based* versus *real-time* games, *sequential* games have all the players within a game make their moves in sequence, and one at a time [12]. *Simultaneous* games are those where any or all players may make their moves at the same time. Classically, sequential games are also called dynamic games [26]: however this would cause confusion in our taxonomy. *Sequential* games allow the construction of the *extensive form* of the game - essentially a hybrid decision tree of all players and all possible moves with their rewards.

3.2.3 Information Visibility

It is not necessary for all participants within a game to have access to all information about the state of the game at any point. The available information can be *perfect*, where all participants have access to the current state of the game, all possible strategies from the current state, and all past moves made by all other participants. The latter implies that all games that impart perfect information to the participants are by their nature also sequential games [14, 19]. *Imperfect* games impart partial information about the game to at least one participant. A special case of imperfect information visibility is *complete* information where all participants are aware of all possible strategies and the current state of the game, however the previous moves by other participants are hidden.

In this taxonomy, this relates to *accessible* and *inaccessible* game worlds. However, a common observation in games is that artificial participants have access to *perfect information* of the game world, whereas the human player only has *imperfect information*. This breaks the immersion of the game.

3.2.4 Noisy vs. Clear

Noisy game worlds are those in which there is a significant amount of information that an intelligence must analyse to either make a decision on what to do next, but where not all of that information is appropriate to the goal. Both *dynamic* and *non-deterministic* games provide levels of noise: the former due to the fact that the state of the game keeps changing even during the times when the intelligence needs to make a decision or form behaviour from learning; and the latter where there is no clear progressive state from which to base rules and analyse the game world. Although these two definitions provide the clearest example of noise, the level of information available can also create noise. Even in *perfect information* game worlds it is possible that the available information is an overly large data set for any intelligence, and thus noise is introduced.

3.3 Other Theoretic Considerations

As with our game type concepts, there are other theoretical concepts that can be used to create more concrete solutions, independent of the individual concepts above.

3.3.1 Nash Equilibria

Within game theory, a Nash equilibrium (NE) exists where an overall highest level of pay-off for all players takes place [20]. There may be many such equilibria within game strategies for a particular game, or there may be only one - in which case it is a *unique* NE.

Although the NE theorem has its problems, finding an NE for any particular state within a game is considered the accepted way of finding a strategy for game playing. The majority of NE finding algorithms are inefficient, however they could potentially be used by an intelligence to choose strategies.

3.3.2 Zero-sum

These are a type of game in game theory that are a special case of general sum games - ones where there is a fixed overall value to winning (or losing) the game [20]. The specific case where for any winning value v, there is a losing value of $0 - v$, is a zero-sum game. In other words, what one player wins, the other loses.

This does not relate directly to elements within our game world definitions above, but it does relate to the nature of games as they are played. Given there is no actual "value" in winning a computer game, even assigning arbitrary values of (1, -1) to the winning and losing values of the game will only provide a zero-sum game for two players - when subsequent players are introduced to the same game, we no longer have a zero-sum game unless we then re-assign arbitrary values.

Some team-based games implement the zero-sum concept for scoring within the game, where a winning strategy is eliminating the opposing team from the game. Where participants face a situation of i versus i team sizes for $2i$ players, the scoring of the game then becomes (i, $-i$) upon elimination of one team by the other.

Some MMOs implement a zero-sum economy, where the game world provides resources which can be extracted and then through combat or trade can move from one hand to another, but will not leave the economic system. Note that this can lead to static gameplay where the majority of the game's wealth is in the hands of a very small proportion of the players.

3.3.3 Symmetry

In game theory, a two-player game is classed as symmetrical if, for player 1 and player 2, and the matrices A and B containing the state-action utility values for the strategies for player 1 and player 2 (respectively), $A = B^T$ *i.e.* if player 1 and player 2 swap positions, they can follow the same strategies, with the same pay-offs. This can be generalised to any number of players, where no-matter which players are swapped, the game can be played as it was before. Asymmetrical games rely on the strategies of whichever player's turn it is (*i.e.* first-player-wins-game-or-draws, such as *Noughts and Crosses*).

This could be an important test for a game, as it will allow game designers to ensure that artificial participants and human players are playing the same game, so that the human player will not consider any artificial players to be "cheating". If a game is not symmetrical for all players, then depending on the suitability tests designed for the artificial players, any solution could automatically be considered a failure.

4 ARTIFICIAL INTELLIGENCE METHODOLOGIES

Using the game theoretic concepts in the second layer of the taxonomy, we can connect these to AI methodologies in the third layer of the taxonomy. This section is not segregated into a participants and game world sub-sections as many of the methodologies cross this boundary and can be applied to both. It deserves mention that the AI methodologies listed here are by no means complete. We hope however that using the general concept of the taxonomy, those methodologies not mentioned will be able to fit into the taxonomy accordingly.

4.1 Agents

Our definitions of game worlds are based around rational agents, and as such we can apply agent-based methodologies to our taxonomy, by applying the metaphor *a participant is an agent* (see [24]).

4.1.1 Reflex Agents

These agents use a conditional statement to provide the "intelligence". Currently, most actors within games follow reflex systems, to the extent that players can monitor the input-output action pairs of specific actors. Once a pattern has emerged, the human player can modify their strategy sufficiently so that the opponent artificial actor will make a significant loss whilst the human player will make a significant gain. *Reflex agents* can fall into infinite loops, as there is no concept of context within if-then statements.

It has been stated that *reflex agents* can only exist within an *accessible* environment [24], but this would be true only if the scope of the condition covered the entire environment. Smaller local conditions can still be met and actions then carried out, irrespective of knowledge of the wider environment.

Temporal agents can be considered a special sub-group of *reflex agents*, where actions are carried out after measuring the passing of time. This specific type of agent would be applicable in *dynamic* and *semi-dynamic* game worlds, where time is a factor.

Reflex agents, are useful in situations where a high level of complexity is not required by a participant. The more limited the scope of possible actions in a *discrete actions* game world, for example, the less complexity is required in decision making. In some cases *reflex agents* might present the best compromise of complexity versus believability.

4.1.2 Model-Based Agents

An agent that monitors the environment and creates a model of it on which to base decisions is called a *model-based agent*. This type of agent would be best applied to *dynamic*, *real-time* games where constant monitoring of the environment is required on which to base decisions on actions. This would also be highly beneficial in a *co-operative* game, where although the actions of other actors are independent, they are inter-related, and so a broader monitoring range covering other co-operating actors can be introduced.

4.1.3 Goal-Based Agents

Using a model of the environment, goals can be created and planning carried out to achieve those goals, even within *inaccessible* game worlds and with other participants. Although the artificially controlled participants will generally have broad goals built in to determine their over-all behaviour (such as "stop the human-player at all costs"), there is still scope within that command to create sub-goals (such as "find the player") *Goal-based agents* are also highly beneficial in *inaccessible* game worlds, as they can change their own sub-goals as the information they are made aware of changes.

4.1.4 Utility-Based Agents

A further refinement on the *model-* and *goal-based agent* methodologies is the ability to manage multiple goals at the same time, based on the current circumstances. By applying utility theory to define the relative "best" goal in any situation, we have *utility-based agents*. These would be especially useful in *player-as-manager* games, or in an *inaccessible*, *real-time* game world, where we can apply the steadily changing state of the available game world to the various goals as the information becomes available, and then alter behaviour in a gradual manner.

4.2 Computational Intelligence

4.2.1 Fuzzy Systems

Fuzzy systems utilise a qualitative approach to information, whereby the incoming data is banded into groups, and it is the membership within a group that is used as a value. For example actors within games usually have some sort of "health" determining their ability to take damage before being removed from the game. By using fuzzy systems, such terms as "low", and "high" health can be used to calculate utility values of strategies.

Fuzzy systems fit naturally into *continuous action* game worlds - given that the input of a fuzzy system is a bounded range, this works even with the example before of turning to any direction: if another goal is "not too far off" an actor's current goal course, then it might provide greater efficiency by achieving the local goal first, even if it is of a slightly lesser priority.

4.2.2 Neural Networks

Neural networks (NNs) [13] have already been used successfully in a game world, as in the case of NERO. The team behind NERO are actively researching in this direction. Uses can be seen for *NNs* in *model-* and *utility-based agents*, where the current state of the agent is monitored all the time. *NNs* are particularly good for *noisy* game worlds: through many data sets the irrelevant information can be weeded out.

4.2.3 Evolutionary Computing

Elements of *evolutionary computing* (EC) [9] have been used as an aid to finding NE [6]. This is because an NE is considered a global maximum problem - something that *evolutionary computing* is especially good at.

Given that finding an NE is considered a solution to a game for a player, finding such equilibria would allow interacting actors to maximise their game playing. If we consider actors taking actions as a game or sub-game, this would let the intelligence behind the actor make the best action based on the results of finding a NE. NE are not always easy to find, and current algorithms are mostly inefficient [20].

Specific to the taxonomy, EC methods in general are good at optimising in a noisy environment and are thus well suited to *noisy* game worlds, and their use of individuals in a population makes them well suited to model *co-operative* game play.

4.2.4 Swarm Intelligence

Swarm intelligences [8] have also been used to compute NE [22]. Ant colony modelling provides a strong methodology for actors to explore the game world to complete goals by providing path-finding around obstacles and creating search patterns to achieve their goals - something that has been seen to be lacking in games. Exploring the game world is important in *imperfect information* worlds, and such group goal finding resulting from ant colony optimisation is useful in *co-operative* game play.

4.3 Classical AI

Classic AI [23] is based around symbolic representations of knowledge to create decision trees and model knowledge-based systems.

4.3.1 Backtracking

Turn-based, discrete action, static, perfect information games are perhaps the easiest types of games to solve - *sequential* play with a fully visible game world allows the easiest construction of the extensive form of the game, which is a type of decision tree, allowing such methods as minimax and A* searching to be used.

4.3.2 Knowledge-Based Systems

Knowledge Based Systems (KBSs) provide a catalogue of information from which deductive reasoning can take place, and are primarily used as expert systems to augment the deductive capacities of certain fields, such as medical diagnosis. Certain games rely on working out the solution to a problem before an artificial participant does, and such reasoning would help in this context. Given the nature of deductive reasoning, KBSs would only be applicable in *accessible* game game worlds.

4.3.3 Rule and Induction Systems

Unlike *KBSs*, *inductive and rule based reasoning* can work in *inaccessible* game worlds, as they rely on what information is available only to reason with. These systems are however labour intensive to create, and highly volatile to error, and such would not be applicable in *real-time, dynamic* games, where there is a large range of possible elements from which to induct reason.

4.4 Other Artificial Intelligence Considerations

There are a number of other areas within AI that can relate to AI in games, although they are not specific to any one classification within the taxonomy.

4.4.1 Believable Agents

These agents are expected to behave as a human would in similar situations. Given this is one of the core purposes of developing better AI in games, any agents that are developed should fall into this category, unless (through the narrative) the actor is expected to behave differently. Even then they should be consistent in their behaviour which could be construed as providing believability in the behaviour across all actor types. Specifically, *player-as-manager* games require a great deal of believability given the large number of actors available to the player, and the semi-autonomous nature of those actors. *Player-as-actor* games will also require a high level of believability, as all the interactions with other actors in the game world must provide a sufficient level of immersion, given the player is essentially existing within an artificial social system.

Believable agents as artificial players must use the same rule set as a human player, otherwise it would be unfair to the human player. More importantly if the artificial player learns within what it considers to be its game world, then it may carry out actions that will alert the human player to the artificial nature of their opponent. This would then break the immersion and thus the believability of the agent. If we take Laird and van Lent's analysis that game intelligence will help with general intelligence, this also means that we could not generalise any intelligence that makes use of programming loopholes when it comes to other domains.

4.4.2 Machine Learning

Realistic NPCs can be made to learn, and machine learning can be used to achieve this: there is a large amount of literature on the field of machine learning as this is one of the core components of artificial intelligence, and as such there are numerous methodologies for learning in any particular situation.

Supervised learning involves giving a system a specific scenario, and then telling it what the desired outcome is. This is done repeatedly over manned training sets to allow the intelligence to learn the correlation between the two. The scenarios are carefully constructed to ensure that the correct behaviour is learned, and takes a great deal of time. In our case this would consists of creating a scenario with dumb actors who need to be specifically controlled, and then carry out the actions that the player would want in that situation, and doing this repeatedly with slightly differing scenarios to reduce the noise of the input. Neural networks are trained this way.

In the majority of cases, players do not want to spend a significant amount of time training the behaviour of what could potentially be disposable actors, thus it would be desirable for the actors to learn the optimal behaviour in any given situation without the player's direct interaction, which leads us to *unsupervised* learning. There is, however, a problem with this: such a method requires a large number of training sets that might not be available by merely monitoring each game as it is being played. A possible solution would be to record scenarios of the player playing, and then analyse these scenarios outside of the game itself.

Even this solution has problems, namely that the optimal behaviour that the learning algorithm develops might not be the desired behaviour of the player for any co-operative or subordinate actors. In which case it would be useful to implement *semi-supervised* learning, where the player can put the actors into a supervised learning mode, which would then let them monitor exactly what the player desires of them, so that they can develop behaviours based on that.

Reinforcement learning has been used to find NE within a given game context to solve strategies within multi-agent systems [4] (and thus could be used in conjunction with agent theory). The solutions are applicable to *dynamic*, *simultaneous* games: as the agents (or in this case generalised actors) learn, the strategies available may alter, and also the strategies being followed by other actors may alter as well. NERO is an example of reinforcement learning. As mentioned in the NERO paper [7], this could introduce a new type of game playing where the aim of the game is to actively train the actors within the game with the desired behaviour for the scenarios in which those actors will be placed. Operatives within NERO are given rewards and punishments for successful behaviour. It is this reinforcement of correct (or incorrect) learning that gives rise to the name of this methodology. The transition to game strategies is a simple one, as each game has a reward, and maximising the reward maximises the reinforced learning.

Quite often actors will be faced with a set of problems at once. *Multi-objective* learning techniques takes this whole set and learns how to solve each problem by analysing the whole set, and extrapolating common themes between the problem which allow a greater categorisation of problems in the future [5]. This is pertinent in *dynamic, real-time* games, which - especially when used with utility-based agents - might present a set of problems or objectives to be completed.

4.4.3 Natural Language Processing

The aim of natural language processing (NLP) is to create a language interface between AIs and humans that seem clear and realistic. Most current games that implement a dialogue system between human players and actors most provide a *discrete action* game world to do it in. There are only a limited number of options available to the player to discuss with the actor. Whilst this is not normally a problem, it can break immersion if a conversation topic is available for discussion that is seemingly unrelated to the actor - something that keen observers would note and then assume that this actor had a bigger part to play than initially thought. NLP can bypass this by allowing a continuous conversational interaction with between the human player and actors in the game, such that the human player would have to put in some effort into the conversation to gain the information needed - or to discover that that actor did not know anything about what the player needed. This is would greatly increase the immersion of the game.

It is useful to note that most text-based adventure games mostly have an open way of playing the game. The human player could type entire actions into the game to be carried out, and the game would interpret these actions, albeit using keyword analysis.

5 USING THE TAXONOMY

Having now created the taxonomy, it would be useful to discuss the envisioned use of the taxonomy, taking a number of examples: first we will illustrate the overall mechanism for using the taxonomy between game developers and AI researchers; then we will illustrate the reverse mechanism, from AI researcher to game developer. We further show the use of the taxonomy from a purely research-based approach, and from a purely game development approach. Lastly, we also provide an example of a specific game, showing the identification of concepts within our taxonomy. Figure 3 shows the bi-directional traversal of the taxonomy.

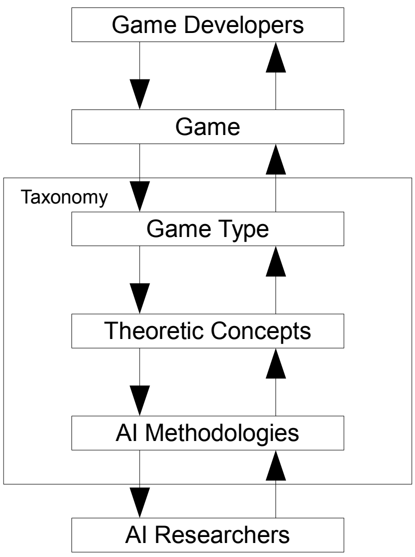

Figure 3. Bi-directional traversal of the taxonomy

5.1 Game Developers to AI Researchers

The most likely scenario for use will be where a game developer wants to incorporate sophisticated AI into their game, and wishes to develop this with the assistance of an expert in the field. The game developer will be able to apply the *game types* layer directly to their proposed game, picking out the interaction and game world concepts identified above: for example the proposed game is a *dynamic, real-time, environmentally discrete, deterministic, accessible, player-as-actor* game.

Using this information, the game developers map the identified concepts from the first layer to the second: again using our example, this gives us the theoretic concepts *non-co-operative, simultaneous, perfect information, discrete actions, noisy*.

Lastly, the game developer identifies a number of AI areas that might be beneficial from this analysis, mapping from the second layer to the final layer. Potential methods can be used, such as *neural networks* or *evolutionary computing* to cope with the *noisy* game world; *tree searching*, given the *perfect information* available and the *discrete actions*; the use of *utility based agents*, given the *noisy* environment and the *non-co-operative* nature of the game-play; or any combination thereof.

With this as a base, the game developers can then approach experts in the field focussed on the methodologies identified to collaborate on specific implementations within the game.

5.2 AI Researchers to Game Developers

The reverse of the above would see an AI researcher develop a novel approach to a particular method within AI, for example utility based agents. The researcher can then identify which of the concepts in the second layer are pertinent to their research. We can see that utility based agents are beneficial for *non-co-operative, imperfect information, noisy* games.

From this, the researcher can map from the second layer to the first layer to identify the overall type of games that their method could be applied to. We can see that this would be *player-as-actor, inaccessible, dynamic, non-deterministic* game types. This allows the researcher to approach various game developers that develop those types of games, to see if there is any scope for collaboration.

5.3 Code Library Selection

Game developers, and developers in general, find the use of various code libraries to be beneficial when creating games - by using a pre-developed set of interfaces, development time can be reduced. It is possible that certain libraries of AI methods could be implemented to provide intelligence for game actors and players, to again reduce the development time associated. As before the game developer would traverse through the taxonomy and identify the types of AI methods that could be used. They could then identify the various libraries that would be beneficial for their completed product. AI researchers may still be needed to provide their expert knowledge, as even though these libraries may provide the technicalities, they will need to be adapted for each instance - in which case the use of the taxonomy is still beneficial in providing common concepts for discussion.

5.4 Experimental Laboratory Selection

AI researchers also benefit from the taxonomy for pure research purposes. As mentioned in the introduction, games can be used as a laboratory in which to conduct experiments and research into various AI methodologies. Using our taxonomy a researcher can identify a type of game to test their ideas, and through analysis of various games available short-list a number of alternatives that will provide them with the environment they need.

5.5 Real Game Example

Fallout is a role-playing game set in an alternate-universe future. The game is single-player, focused on the actor that the player is controlling. The player may meet other actors within the game who join them in their mission and aid them, however these are uncontrolled by the player, thus the interaction is *player-as-actor*. The entire world cannot be seen at once, thus the game world is *inaccessible*. The primary actions of the character take place in *real-time*. The game world is divided into a hexagonal map through which the game actors move - an *environmentally discrete* game world. The game is *dynamic*, as other actors in the game may carry out actions whilst the player is thinking; given this, and the fact the game world is inaccessible, the game world is also *non-deterministic* - the player (or actors) cannot explicitly determine what will happen next. Normal game-play is *simultaneous*, and other actors in the game can be seen to have regular objectives depending on the time of day or other external stimuli. Despite the fact that other actors may join with the player's actor, there are no mechanisms in place to explicitly restrict non-co-operative action: *e.g.* the player may choose to engage in combat with the friendly actors; thus making the game *non-co-operative*. As the world is both *dynamic* and *non-deterministic* it is also *noisy*.

Fallout has a number of different levels of game-play: primary game-play (as described above); combat game-play; and fast travel game-play. Combat game-play changes to be *turn-based*, and also reinforces the non-determinism of the world by using random numbers for such mechanisms as accuracy and damage of particular weapons on each hit. Fast-travel game-play changes to be *static*, as the game state does not change whilst the player chooses where to travel; and *environmentally continuous*, as the fast-travel map presented to the player allows the player to move their actor anywhere on the map.

Using the taxonomy, we can see that the most applicable AI methods to implement in *Fallout* would be *evolutionary algorithms*, *neural networks*, or to use a *utility-based agent* approach. Other methods are available, but when taken as a whole there are competing aspects of the game world that mean we must take the most widely suited: for example, although the *inaccessible* nature of the game world could point to the use of *inductive reasoning*, the *noisy* nature of the environment precludes this.

6 CONCLUSION

Given both the interest in the game industry for realistic actors within games, and the opportunity for involvement by academic researchers, we hope that the taxonomy described in this paper will provide a starting point by which to facilitate collaboration between the two sides in order to further both agendas. The taxonomy presented here is in the early stages of development. It is hoped that a refined version of the taxonomy will provide developers with a framework within which they can discuss AI techniques with relevant experts. Conversely, a fully developed taxonomy should also provide AI researchers with a formal process for utilising AI techniques in a gaming context.

Future work is aimed at further developing the taxonomy and in evaluating its utility. We aim to conduct a comprehensive study of the AI literature in order to map existing techniques to our taxonomy, refining it as necessary. Although we have highlighted the most obvious examples of existing work in the initial taxonomy, this process will no doubt also identify additional approaches utilised in AI that can be useful in a games environment. An additional feature of the taxonomy development is likely to be directed towards the construction of an *ontology*, as referred to in the introductory section, to be used alongside the taxonomy. This will facilitate dialogue between developers and researchers by formalising the concepts within the taxonomy, leading to greater understand and better communication on both sides.

An important aspect of our future work is to validate that the proposed taxonomy is both *useful* and *correct*. In order to achieve this, we are working towards identifying a series of metrics which can be used to measure the success of the taxonomy. Close involvement with industry is critical to this process. From a practical perspective, we aim to illustrate the effectiveness of the taxonomy by identifying a number of case-studies which will enable the implementation of an effective game by following the taxonomy.

The taxonomy was initially developed in order to select a relevant game environment for using as a laboratory in which to experiment with AI techniques. However, we hope that a more fully developed version will have a much wider scope, proving useful across the spectrum of game development and AI research.

REFERENCES

[1] E. Aarseth, 'Playing research: Methodological approaches to game analysis', in *spilforskning.dk Conference*, (August 2003).

[2] Thomas H. Apperley, 'Genre and game studies: Toward a critical approach to video game genres', *Simulation Gaming*, **37**(1), 6–23, (March 2006).

[3] Jay D. Bolter and Richard Grusin, *Remediation: Understanding New Media*, The MIT Press, February 2000.

[4] M. Bowling and M. Veloso, 'An analysis of stochastic game theory for multiagent reinforcement learning', Technical report, Computer Science Department, Carnegie Mellon University, (2000).

[5] Richard A. Caruana, 'Multitask learning: A knowledge-based source of inductive bias', in *Proceedings of the Tenth International Conference on Machine Learning*, pp. 41–48, (1993).

[6] *Evolutionary Computation in Economics and Finance*, ed., Shu-Heng Chen, Studies in Fuzziness and Soft Computing, Physica-Verlag Heidelberg, May 2002.

[7] T. D'Silva, R. Janik, M. Chrien, K. O. Stanley, and R. Miikkulainen, 'Retaining learned behavior during real-time neuroevolution', in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, (2005).

[8] Russell C. Eberhart, Yuhui Shi, and James Kennedy, *Swarm Intelligence*, The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann, March 2001.

[9] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Natural Computing, Springer, November 2003.

[10] C. Fairclough, M. Fagan, Mac, and P. Cunningham, 'Research directions for ai in computer games', in *Proceedings of the Twelfth Irish Conference on Artificial Intelligence*, (2001).

[11] G. Frasca, 'Simulation versus narrative: Introduction to ludology', in *The Video Game Theory Reader*, chapter 10, Routledge, (2003).

[12] Yoav Freund and Robert E. Schapire, 'Game theory, on-line prediction and boosting', in *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*, pp. 325–332, New York, NY, USA, (1996). ACM.

[13] K. Gurney, *An Introduction to Neural Networks*, Ucl Pr Ltd, 1997.

[14] J. C. Harsanyi, 'Games with incomplete information played by "bayesian" players, i-iii. part i. the basic model', *Management Science*, **14**(3), 159–182, (November 1967).

[15] A. Kleiner, 'Game ai: The possible bridge between ambient and artificial intelligence', *Emerging Communication*, **6**, 143–158, (2003).

[16] L. Konzack, 'Computer game criticism: A method for computer game analysis', in *CGDC Proceedings*, (2002).

[17] John E. Laird and Michael van Lent, 'Human-level ai's killer application: Interactive computer games', in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 1171–1178. AAAI Press / The MIT Press, (2000).
[18] Craig A. Lindley, 'Game taxonomies: A high level framework for game analysis and design', *Gamasutra.com*, (2003).
[19] Oskar Morgenstern and John Von Neumann, *Theory of Games and Economic Behavior*, Princeton University Press, May 1944.
[20] Noam Nisam, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, September 2007.
[21] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, July 1994.
[22] N. Pavlidis, K. Parsopoulos, and M. Vrahatis, 'Computing nash equilibria through computational intelligence methods', *Journal of Computational and Applied Mathematics*, **175**(1), 113–136, (March 2005).
[23] E. Rich and K. Knight, *Artificial Intelligence*, McGraw Hill Higher Education, 2 edn., March 1991.
[24] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, December 2002.
[25] J. Schaeffer, V. Bulitko, and M. Buro, 'Bots get smart', *IEEE Spectrum*, (December 2008).
[26] James N. Webb, *Game Theory: Decisions, Interaction and Evolution*, Springer Undergraduate Mathematics Series, Springer, December 2006.
[27] Mark J. P. Wolf, *The Medium of the Video Game*, University of Texas Press, June 2002.