14

A Survey on the Need and Use of AI in Game Agents

Sule Yildirim and Sindre Berg Stene
Hedmark University College
Norway

1. Introduction

This chapter firstly seeks an answer to the question of whether Non Player Characters (NPCs) in Computer Games can also be viewed as game agents where reactivity, autonomy, being temporally continuous and having goal-oriented behavior are taken as the features of being a game agent. Within this scope, we will try to assess whether naming NPCs as agents point to a desire that one day they will fulfill the requirements of being an agent or whether NPCs actually already fulfill these requirements. Secondly, the chapter looks into the AI needs of video games. We present the AI methodologies that are either being used or under research for use in Game AI. The same methodologies are also likely to contribute to the increasing levels of autonomy and goal-directed behavior of NPCs and help them become more agent-like.

2. Background

In game development and game AI research, terms such as game agents and NPCs are often used interchangeably. However, considering the many years of agent research, are NPCs really game agents? In order to answer this question, we look at different definitions of being an agent from agent research and point out the features that would possibly separate an ordinary NPC from a game agent. Autonomy and goal-directed behavior are features that lie at the heart of being an agent and for that reason we look into the details of what it means to be autonomous and goal-oriented for an agent.

In agent research and applications, having its own agenda and being able to select its own actions are essential parts of being autonomous (Franklin & Graesser 1997). For that reason, the more the NPCs are able to carry out own agenda the more agent-like they are. If an NPC responds to certain conditions with predefined actions scripted in a certain programming language, it would have much less autonomy compared to an agent which can plan its future actions. Furthermore, the actions that NPCs execute cause changes in the environment. NPCs can then select further actions based on the consequences of an applied action, or they can ignore the consequences of their actions. An NPC which can consider the consequences of its actions can plan.

On the other hand, we look into the purpose of game AI and present the capabilities (reasoning, decision making, learning, prediction, goal-oriented behavior) that are expected from NPCs endowed with game AI. We also present a list of AI methodologies (e.g. path finding, Finite State Machines) which are in use or are proposed for use to achieve those capabilities. We summarize different kinds of game genres, the AI status and challenges of these games in order to be able to be more specific about the behaviors (e.g. create an army, form alliances, and so on) which would be expected from NPCs.
Another relevant definition is the definition of "intelligence". Intelligence is the ability to acquire and apply knowledge. The way in which the NPCs acquire knowledge and reason upon it will provide agents with different levels of intelligence.
Then, section 3 investigates the aspects of being a game agent. Section 4 gives the purpose and methods of game AI and the capabilities that NPCs become equipped with through game AI. It also explains what is meant by game AI for different game genres. Section 5 gives an evaluation of games for their agent characteristics. Section 6 gives the conclusions.

3. Agent Perspective: Being an NPC vs. Being a Game Agent

A definition of agents is as follows:

An autonomous agent is a system situated within and part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to reflect what it senses in the future (Franklin & Graesser, 1997).

Another definition of agent is given as follows:

Agents can be defined to be autonomous, problem solving computational entities capable of solving effective operations in dynamic and open environments (Luck et al., 2003).

On the other hand, an agent is stated to be autonomous if its behavior is determined by its own experience (with ability to learn and adapt) (Russell & Norvig, 2003).
In robotics, autonomy means independence of control. This characterization implies that autonomy is a property of the relation between the designer and the autonomous robot.
Self-sufficiency, situatedness, learning or development, and evolution increase an agent's degree of autonomy (Pfeifer & Scheier, 1999).
A list of properties is listed at the end of page 5 in (Franklin & Graesser, 1997). The authors state that the first four properties should exist in a program to name it as an agent and these four properties are as follows:

Reactive: Most of the NPCs in existing games have reactive responses, e.g. First Person Shooter NPCs.

Autonomous: Are there any NPCs that have their own agenda and that exercise control over its actions in any of the existing games? Such an NPC would not need to be told what to do but would be able to decide which action(s) to take under what conditions.

Goal-oriented: Does not simply act in response to the environment. Are there any agents in existing games which can pursue its goals?

Temporally continuous: This feature seems to be true in all agents.

In this chapter, we use these four properties to assess whether an NPC can be named as an agent. In existing games, most NPCs are reactive and temporally continuous. However, are they also autonomous and goal-oriented? Which AI methods can be employed to help achieve these properties?

On the other hand, we classify NPCs as software agents (Franklin & Graesser, 1997). This classification allows us further to classify NPCs according to their control mechanisms. In this way, game agents can be algorithmic, rule based, planning based, or otherwise primarily oriented around fuzzy logic, neural networks, machine learning, etc. This kind of classification is also important since it forms a natural bridge between the agent terminology such as being reactive, goal-oriented, autonomous, temporally continuous and the AI methodologies that can be employed to achieve them.

The type of control mechanism to employ depends on the kind of intelligence that is required to build into an NPC and the game genre that an NPC belongs to. For that reason, one of the coming sections will also give an overview of different game types (genres). Next we will look at what is meant by goal-based agents.

3.1 Goal-based Agents

"A game is a form of art in which participants, termed players, make decisions in order to manage resources through game tokens in the pursuit of a goal."

Greg Costikyan - Game designer and science fiction writer

A goal-based agent not only considers the consequences of its actions (See Fig. 1: What it will be like if I do action A) but also considers how much those actions are in line with its goals (See Fig. 1: What action I should do now given certain goals). As an example, F.E.A.R (Yue & de-Byl, 2006; Hubbard, 2005) is a game with goal-oriented behavior.

In (Yue & de-Byl, 2006), an example is presented for goal-directed behavior which is given in Fig. 2. In the figure, blue boxes represent keys and the current and goal values of a key. A key is a state that an agent wants to get into. For example, the topmost box holds a key which is "EnemyIsDead". This key has a current value of "False" and a goal value of "True" as the AI-controlled agent wants to have its enemy dead. In order to achieve that goal value, the enemy can choose to apply the action "fire weapon". The white boxes indicate the preconditions and effects of the action that they are bound to. The precondition of the "fire weapon" action requires that the key "HealthRecovered" has a value of "True" before the action can be applied. The effect of the action is to have the key "EnemyIsDead" have a value of "True" after the action is applied. It can be seen in Fig. 2 that the current value of the key "HealthRecovered" is "False". In order to have a value of "True" for the key "HealthRecovered", the action "take cover" is selected for application before the "fire weapon" action. The "take cover" action does not have any preconditions and for that reason, the goal state is achieved by the application of the "take cover" action first and the "fire weapon" action second.

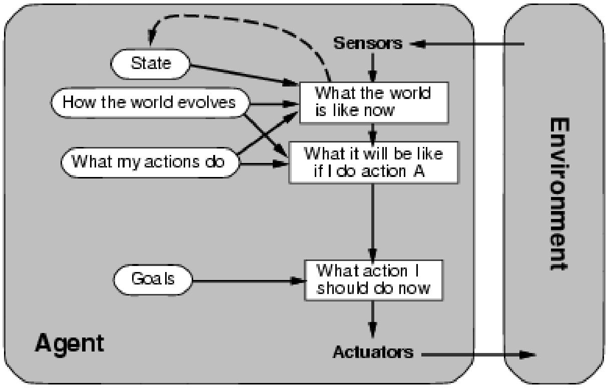
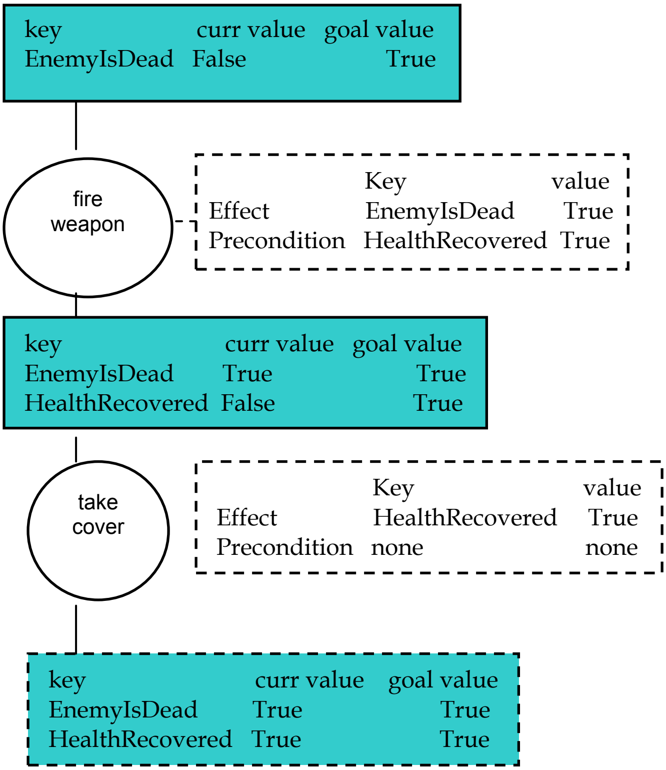

Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Fig. 1. Goal-based Agents (adapted from (Russell & Norvig, 2003)).



key	curr value	goal value
EnemyIsDead	False	True

	Key	value
Effect	EnemyIsDead	True
Precondition	HealthRecovered	True

key	curr value	goal value
EnemyIsDead	True	True
HealthRecovered	False	True

	Key	value
Effect	HealthRecovered	True
Precondition	none	none

key	curr value	goal value
EnemyIsDead	True	True
HealthRecovered	True	True

Fig. 2. Taken from (Yue & de-Byl, 2006) - F.E.A.R. © Monolith Productions 2005.

On the other hand, planning is a more complicated goal-directed behavior and more on planning will be given in Section 4. The benefits of a goal-oriented planning behavior are stated as follows in (Yue & de-Byl, 2006).

- There are no hard-coded plans
- Code maintenance is minimized
- Bugs are minimized when planning
- Code is reusable

The drawbacks of a goal-oriented planning behavior are as stated as follows in (Yue & de-Byl, 2006):

- Real-time planning requires processing time
- More complex plans require more time

Different aspects of goal-directed behavior and planning such as representation of world, architectures and applications can be found in (Orkin, 2005), (Orkin, 2004), (Narayek, 2002), (IGDA, 2005) and (Namee & Cunningham, 2001).

4. Game AI in General

One of the aims of Game AI is to increase the difficulty of the game to challenge the human players. The game AI should lead to a fulfilling game experience. On the other hand, games take place in dynamic complex worlds in which complex decisions must be made, often based on partial knowledge (Fairclough et al., 2001).

It is usual to state that Non Player Characters are AI-controlled characters that can only interact with PCs (Player Characters) through scripted events or artificial intelligence (AI). However, what is it that is called Artificial Intelligence in an NPC? For example, World of Warcraft is one of the most popular games and its NPCs are stated to be AI controlled (Blizzard, 2005). Even though it is so popular, it actually has very little AI composed of reactive agents and a great deal of scripted events. For example, a player is tasked with escorting an NPC so that he can walk around and look for his lost backpack or something similar (where you meet monsters and assassins on the way, also scripted NPCs, and everything woven into a storyline about this man's misfortune). On the other hand, even though it is "low AI", this is something which adds to the game experience in terms of AI.

What kind of intelligent behaviors are expected from NPCs? In general, an NPC is expected to maximize goal achievement by choosing the most optimal actions given the available information. An exception to the rule could be when more human-like actions may be a better choice than the most optimal action, for the sake of realism.

The most commonly used AI methodologies to achieve game AI can be stated as follows (McGee, 2005):

- Decision trees: can be realized by If-Then statements..
- Finite state machines
- Command hierarchies

- Manager task assignment
- Path finding (A*)
- Terrain analysis
- Influence mapping
- Formations
- Flocking
- Emergent behaviour

Influence mapping is a technique for terrain analysis to identify boundaries of control or otherwise interesting points/areas/features of a map.
Other possibilities are:

- Artificial Neural Networks
- Genetic Algorithms
- Fuzzy logic

Scripting is the most common means of control in Game AI. Path finding is another common use for AI, widely seen in RTS (Real Time Strategy) games. Emergent AI has been explored in Black & White.

In addition to the methodologies stated above, cheating is also considered a valid technique in game AI. The position of an unforeseen object can simply be looked up in the game's scene graph (Fairclough et al., 2001). This kind of cheating can cause a feeling of unfairness on the player character's behalf and too much of it might not be appropriate for that reason.

As human beings, what is important for us when selecting an action in a certain situation is to behave in a way that we believe is the right way in the particular situation. For that reason, we have certain action formats which tell us what to do or how to act under various circumstances. Such behavior can be implemented using rules, and Finite State Machines model this by combining sets of such rules into "states" where each state also contains rules for which conditions would cause an agent to change state.

FSMs are very rigid and behave poorly when confronted by situations not dreamed of by the game designer (Fairclough et al., 2001). The behavior of the NPC is governed by a set of rules, each rule having a condition and some action(s) to execute when this condition is satisfied by the state of the world (Fairclough et al., 2001). Therefore, as a player interacts with an NPC he gets to know the behavior of the NPC and can predict what it will do next and can develop a plan to despatch the NPC (Fairclough et al., 2001). Non-deterministic FSMs are also employed to increase the unpredictability of NPCs so that the "fun" side of a game increases (Brownlee).

On the other hand, we as humans learn from our mistakes. In (Scott, 2002), it is stated that in general, game AI lacks the learning and reasoning capabilities of humans and that game AI could address the fields of machine learning, decision making based on arbitrary data input, reasoning etc. However, in several games, game AI encompasses some amount of learning and reasoning capabilities (e.g. learning capability of creatures in Black and White). An NPC which executes FSMs to mimic human behavior can learn from its mistakes and build up its own experience to be able to act smarter in time. Equipped with an FSM, an NPC should be able to find out which of its actions has been beneficial, which of them have not been beneficial and what other action types it could employ under given circumstances. Learning actions which is not in its repertoire and learning them on its own would be a real challenge for an NPC especially when it would obviously lack the observational and mimicking by imitation or natural language understanding capacities of a human player. The research in AI towards this direction can also find its application for developing NPCs.

In (Evans, 2001), learning is stated to cover the following characteristics, specifically for the game Black and White:

- Learning that (e.g. learning that there is a town nearby with plenty of food)
- Learning how (e.g. learning how to throw things, improving your skill over time)
- Learning how sensitive to be to different desires (e.g. learning how low your energy must be before you should start to feel hungry)
- Learning which types of object you should be nice to, which types of object you should eat, etc. (e.g. learning to only be nice to big creatures who know spells).
- Learning which methods to apply in which situations (e.g. if you want to attack somebody, should you use magic or a more straightforward approach?)

Learning can be initiated in a number of very different ways (Evans, 2001):

- From player feedback, stroking or slapping the creature (NPC).
- From being given a command: when the creature is told to attack a town, the creature learns that that sort of town should be attacked.
- From the creature observing others: observing the player, other creatures, or villagers.
- From the creature reflecting on his experience: after performing an action to satisfy a motive, seeing how well that motive was satisfied, and adjusting the weights representing how sensible it is to use that action in that sort of situation.

Reasoning is about finding out why something has happened while showing an awareness of goals and assessing the relative importance of different goals (Hawes, 2000). If the game AI reasons better, the NPCs would become more agent-like.

(Wintermute, 2007) presents the use of SOAR architecture to meet the demands on the AI in RTS games. A discussion of the CogAff architecture as the basis for an agent that can display goal-oriented behavior is given in (Hawes, 2000).

Planning in real-time is an alternative to the more common techniques of modeling character behavior with scripts or finite state machines (FSMs). Rather than traversing a predefined graph of state transitions, a planning Non Player Character (NPC) searches for a sequence of actions to satisfy some goal. Considerations for developing an agent architecture for real-time planning in games is given in (Orkin, 2005).

One project regarding saving the NPCs from being predictable and even allowing them to make predictions about the human player is the TCD Game AI project (Fairclough et al., 2001). On the other hand, Case-Based Plan Recognition is proposed as a prediction mechanism for NPCs to predict a player's actions (Kerkez & Cox, 2001).

4.1 Game AI Specific to Different Genres: NPC Behaviors, AI Methodologies in Use, Expected AI Challenges

Games can be grouped under different types: according to their nature: Real Time Strategy (RTS), First Person Shooter (FPS), Role Playing Game (RPG), God Games etc. This section will present a mixture of the need for AI, the methods of AI and the behaviors that can be obtained by AI for different kind of games.

Role Playing Games

In RPG games, a team is built up to reach a common goal. Some fine examples of RPG's are Secret of Mana, the games in the Final Fantasy series and other many varieties. The intelligence required from a team or from individual characters (collaborating or cooperating) depends on how complex behavior is required to reach the common goal, what kind of resources the team or individuals in the team have and whether the team members have primitive or advanced collaboration strategies. There is a need for more realistic and engaging NPCs in these games (Fairclough et al., 2001).

First Person Shooter Games

First-person shooter games (FPS games), emphasize shooting and combat from the perspective of the character controlled by the player. FPS-type games usually implement the layered structure of the artificial intelligence system. Layers at the lower levels handle the most elementary tasks, such as determining the optimal path to the target or playing appropriate sequences of character animation. The higher levels are responsible for tactical reasoning and selecting the behavior which an AI agent should assume in accordance with its present strategy. The examples for tasks of higher layers are whether the agent should patrol the area, enter combat or run through the map searching for an opponent (Grzyb, 2005).

Some fine examples of FPS games would be Half-Life 1, Half-Life 2, Counter-Strike, Doom III and Quake IV.

The NPCs in the original Half-Life 1 released in 1998 had AI behaviors which were not present in previous games. For example, AI comrades and enemies had different reactions for getting shot, spotting grenades and even a realistic awareness of the actions of the human player. As a result of displaying this intelligent behavior, Half-Life 1 quickly asserted itself as having the best AI of any game at the time. After Half-Life 1, more and more games started to focus on the AI aspect of game design instead of just graphics. Today, combat AI can be seen ducking around corners or behind boxes and tossing the player's grenades back. In many cases, AI controlled NPCs are even standing in for real players in multiplayer games. Even though combat AI can dodge incoming fire and shoot like a skilled player, there are four major things that human combatants offer over AI: knowledge of their environment, efficient use of teamwork, the ability to "hunt", and survival instincts (Schreiner). However, some progress has since been made to address each of these issues.

Real Time Strategy Games

Strategy games require that the human player displays skillful and planned behavior in order to play efficiently and hence achieve victory. Real-time strategy games are strategy games which are played in a real-time environment. In the RTS game environment, the human and computer controlled players compete for resources. The most common setting for RTS games is the war-game setting, where in addition to resource management the player engages in war. The resource management in RTS games encompasses obtaining resources and utilizing those resources to gain an advantage. Resources are valuable minerals, energy, or other materials. Building an army and attacking strategic objectives, such as locations with access to resources, are aspects which make an RTS game a "war game".

Fine examples of RTS games would be Starcraft 1 and 2, Warcraft 1-3, Supreme Commander, the different games in the Command and Conquer series, and many other varieties of games as well.

At the low levels of AI, path planning is important whereas at higher levels, there are modules responsible for economy, development or, very importantly, a module to analyse the game map (Grzyb, 2005).

In Warcraft 3, users are called upon to create an army to defeat one or more computer-controlled villages with armies of their own. Computer controlled villages form alliances, scout surrounding areas and devise appropriate battle plans to do their best to be the last village standing (Wexler, 2002).

God Games

The focus of a god game tends to be control over the lives of people, anywhere from micromanaging a family to overseeing the rise of a civilization as in Spore (Maxis, 2008). Also, Black and White is a god game developed by Lionhead Studios and it was implemented through a variety of AI algorithms and techniques in 2002. What sets "Black and White" apart from any other game before it is its advanced use of AI. There are two types of intelligent agents in "Black and White". The first type is the community of villagers. The villagers have their knowledge, desires and beliefs represented in large tables and situation calculus. The other intelligent agent is the creature. AI in the creature allows the creature to learn how to satisfy its master and know how to correctly act based upon its beliefs and perceptions (Funge, 1999). Symbolic attribute-value pairs are used to represent a creature's belief about any individual object. This type of representation is used with rule-based AI to give creatures basic intelligence about objects (Wexler, 2002). Decision trees are used to represent agents' beliefs about general type of objects (Wexler, 2002). Neural networks are used to represent desires (Wexler, 2002). The creatures also learn facts about its surroundings, how to do certain tasks, how sensitive to be to its desires, how to behave to or around certain objects, and which methods to apply in certain situations. The user can help the creature to learn or it can learn to please its user by differentiating the tasks that please the user versus the tasks that do not.

More advanced AI in terms of a partial-planner with which a creature can satisfy its desires and goals is foreseen to be included in "Black and White". This way a user can create a list of goals that it wishes for the creature to complete (Bandi et al., 1999).

Action-Adventure Games

Action-adventure games focus on exploration and usually involve item gathering, simple puzzle solving, and combat, e.g. Tomb Raider (Core Design, 1996). In action games, having the player as a member of a squad or a team provides an opportunity for the further use of more complex AI.

Also there is need for more realistic and engaging NPCs in Adventure Games (Fairclough et al., 2001).

The other types of games are sport games, racing games and puzzles. There are also games which do not fit squarely in any category.

5. Evaluation of Games for Their Agent Characteristics

In Table 1 and Table 2, we use the agent definition given in (Franklin & Graesser, 1997) to evaluate the agent characteristics of individual NPCs in a diverse selection of popular games.

In reactive agents, a program is just a set of percept → action rules, commonly termed a production-rule system. In each step of the control cycle a rule is selected whose left hand side matches the agent's current percept, and the action on that rule's right hand side is then effected. The first column of Table 1 shows our evaluation of the reactive capabilities of the selected games. The value for reactivity can be High, Low or None in the table, indicating the ability of acting out useful reactive responses in rapidly changing situations. For example, an NPC is unexpectedly attacked on the way to a distant destination, and temporarily stops in order to respond to the attack.

An agent is a temporally continuous software process that senses the environment and acts on it over time, in pursuit of its own agenda and so as to effect what it senses in the future. The corresponding column in Table 1 lists the extent to which an agent's behavior is influenced over time by what it senses in the environment.

In goal-oriented agents, an agent applies actions in pursuit of a goal. A goal-oriented agent is different from an agent where the agent does not necessarily try to achieve a goal, but it applies an action just because it is right to do that action in a given situation (Table 1).

Autonomy characteristic of an agent is split further into following aspects: Self-sufficiency, Complexity of decision making, Variety of action repertoire, Learning and Situatedness (Table 2). These are the aspects of autonomy that we deemed to be relevant for game AI agents.

Self-sufficiency is a property of autonomy in agents. It has three aspects:

1. Supervision by higher level agents where there is a hierarchy of agents and each level of the hierarchy can have various amounts of complexity.

2. Visible inter-agent communication.

3. Being capable of supporting an agent's own life without help, food, shelter etc. This aspect does not apply in game environments.

	Reactivity	Temporally continuous	Goal-oriented
World of Warcraft	Low	Low	Yes
Half Life 2 (Single player mode)	High	Low	Yes
Supreme Commander	Low	High	Yes
Warcraft 3	Low	High	Yes
Black & White 2	Low	High	Yes
Tomb Raider: Anniversary	Low	Low	Yes
Unreal Tournament 2004 (bots)	High	High	Yes

Table 1. Evaluation of different games for their agent aspects related to reactivity, being temporally continuous and goal-oriented.

Visible inter-agent communication is an aspect that influences an agent's autonomy level. More agent interaction might mean higher autonomy, e.g. one NPC informs another that there is a danger.

The complexity of an agent's decision-making mechanism influences the quality of decisions. For example, an agent that uses non-deterministic finite-state machines is better at considering alternative actions for a given situation compared to an agent that uses a deterministic finite state machine. An NPC with a controller of the former type is more likely to survive compared to the latter.

Having a variety of action repertoire will increase an agent's levels of autonomy, by enabling it to choose between more actions.

Being able to learn would mean an agent is likely to become informed about what to do in situations it is not currently able to take an action or take the most suitable action. From the games referred to in Table 2, learning is a feature only present in Black & White 2.

Being situated and hence being informed about the environmental state is a property that exists in all the selected games of Table 2 and is a property of being autonomous for agents.

	Supervision by higher level agents	Visible inter-agent commu-nication	Complexity of decision making	Variety of action repertoire	Learning	Situatedness
World of Warcraft	None	Low	Low	Low	No	Yes
Half Life 2 (Single player mode)	None	High	Low	High	No	Yes
Supreme Commander	High	Low	Low	High	No	Yes
Warcraft 3	High	Low	Low	High	No	Yes
Black & White 2	Low	Low	High	High	Yes	Yes
Tomb Raider: Anniversary	None	High	Low	Low	No	Yes
Unreal Tournament 2004 (bots)	None	High	High	High	No	Yes

Table 2. Evaluation of different games for their agent aspects related to autonomy

6. Conclusion

In this chapter, we investigated whether it is possible to view NPCs in games as game agents. We used the qualities of reactivity, being temporally continuous, autonomy and being goal-oriented for evaluating a representative set of popular commercial games on how much agent-like they are. Our evaluations show that some games have NPCs that achieve more agent-like behavior than others.

Also, we give a survey on the use of various AI methodologies in commercial games and in game research. We indicate that the need for AI in games is centered around maximizing goal achievement by choosing the most optimal actions given the available information. NPCs can challenge human players and add to the quality of experience if they can become better at achieving optimal behavior.

Furthermore, we postulate that further investigation is required before conclusions can be made about whether the given methodologies (especially those that are not commonly used in commercial games at the moment) are really applicable in commercial games. Regarding this point, the use of learning in games is relatively low in commercial arena as can be observed from Table 2 even though learning is an important aspect in achieving artificial intelligence. Another relevant point is that the non player characters are required to make real-time decisions and that it does not seem practical to add time consuming and complicated intelligence control mechanisms in them.

7. References

Bandi, S., Cavazza, M. et al. (1999). Situated AI in Video Games, University of Branford

Blizzard (2005). World of Warcraft, Blizzard. www.worldofwarcraft.com

Brownlee, J. Finite State Machines (FSM), Artificial Intelligence Depot http://ai-depot.com/FiniteStateMachines/FSM.html.

Core Design (1996). Tomb Raider, http://www.tombraider.com/, Eidos Interactive

Evans, R. (2001). The Future of AI in Games: A Personal View, Game Developer Magazine.

Fairclough, C.; Fagan, M. et al.(2001). Research Directions for AI in Computer Games, *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science*, pp. 333 – 344.

Franklin, S. & Graesser, A (1996). Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Proceedings of the Workshop Intelligent Agents III, Agent Theories, Architectures, and Languages*, pp. 21–35, ISBN:3-540-I-0 Springer Verlag, London.

Funge, J. (1999). AI for *Computer Games and Animation: A Cognitive Modeling Approach*, AK Peters Ltd, ISBN: 1-56881-103.9, Natick, MA

Grzyb, J. (2005). Artificial Intelligence in Games, *The Software Developer's Journal*

Hawes, N. (2000). Real-Time Goal-Oriented Behavior for Computer Game Agents. *Proceedings of the 1st International Conference on Intelligent Games and Simulation*, London, November, 2000

Hubbard, V. (2005) F.E.A.R First Encounter Assault Recon, Monolith Productions, Vivendi Universal

IGDA (2005). The AIISC Report

Kerkez, B. & Cox M. T. (2001). Incremental Case-Based Plan Recognition Using State Indices, *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development,* pp. 291–305, ISBN: 978-3-540-42358-4, Springer-Verlag, Berlin

Luck, M. & McBurney, P. et al. (2003). Agent Technology: Enabling Next Generation Computing A Roadmap for Agent Based Computing, AgentLink 2.

Maxis (2008). Spore, Electronic Arts. http://www.spore.com/

McGee, K. (2005). Advanced Game Programming: AI Session 2: Genres, Roles, Techniques, http://www.ida.liu.se/~kevmc/courses/2005-ht02/tsbk10-session-2.pdf

Namee, B. M. & Cunningham, P. (2001). Proposal for an Agent Architecture for Proactive Persistent Non Player Characters, Proceedings of the 12th Irish Conference on Artificial Intelligence and Cognitive Science: 221 – 232

Narayek, A. (2002). Intelligent Agents for Computer Games. Computers and Games, *Proceedings of Second International Conference, CG 2000*, pp. 414-422, Springer

Orkin, J. (2004). Symbolic Representation of Game World State: Toward Real-Time Planning in Games, *Proceedings of the Workshop Challenges in Game AI*, AAAI

Orkin, J. (2005). Agent Architecture Considerations for Real-Time Planning in Games, *Proceedings of AIIDE*

Pfeifer, R. & Scheier ,C. (1999). *Understanding Intelligence*, MIT Press.

Russell, S. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, Prentice Hall

Schreiner, Artificial Intelligence in Game Design, http://ai-depot.com/GameAI/Design.html

Scott, B. (2002). The Illusion of Intelligence, In: *AI Game Programming Wisdom*, Rabin, Steve (Eds), 19–20, Charles River Media

Wexler, J. (2002). Artificial Intelligence in Games: A look at the smarts behind Lionhead Studio's "Black and White" and where it can go and will go in the future, University of Rochester.

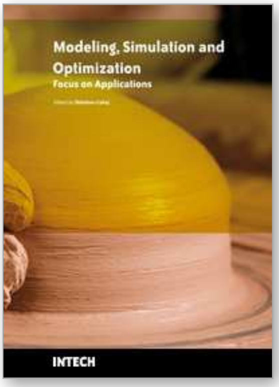
Wintermute, S., Xu, J., & Laird, J.E. (2007). SORTS: A Human-Level Approach to Real-Time Strategy AI, *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-07)*, Stanford, California.

Yue, B. & de-Byl, P. (2006). The state of the art in game AI standardisation, *Proceedings of ACM International Conference*

Acknowledgement

This chapter is a revised version of a paper by the authors published in the Proceedings of Agent Directed Simulation (ADS'08) within the Conference SpringSim'08, pp. 124-131, ISBN: 1-56555-319-5, Ottawa, Canada, April 2008, The Society for Computer Simulation International, San Diego.

We would like to thank Arild Jacobsen from Gjøvik University College for sharing his ideas about different parts of this chapter with us and his valuable help with the proof-reading of the chapter.



Modeling Simulation and Optimization - Focus on Applications
Edited by Shkelzen Cakaj

ISBN 978-953-307-055-1
Hard cover, 312 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

The book presents a collection of chapters dealing with a wide selection of topics concerning different applications of modeling. It includes modeling, simulation and optimization applications in the areas of medical care systems, genetics, business, ethics and linguistics, applying very sophisticated methods. Algorithms, 3-D modeling, virtual reality, multi objective optimization, finite element methods, multi agent model simulation, system dynamics simulation, hierarchical Petri Net model and two level formalism modeling are tools and methods employed in these papers.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821