Artificial Intelligence techniques used in First-Person Shooter and Real-Time Strategy games.

Human Media Interaction seminar 2010/2011: Designing Entertainment Interaction

Mark Oude Veldhuis (m.oudeveldhuis@student.utwente.nl)

April 17, 2011

University of Twente

Faculty of Electrical Engineering, Mathematics and Computer Science

Department of Human Media Interaction

ABSTRACT

This paper discusses artificial intelligence as used in specific real-time strategy (RTS) and first-person shooter (FPS) game titles. Most of the time the artificial intelligence in games is scripted and pre-defined, making it predictable and less replayable. Common techniques that are used include (hierarchical) finite state machines, scripts describing a list of sequential actions, A* search algorithms or rule-based expert systems. But there also are new developments aimed at improving the adaptation of the computer player to their human opponent and decreasing predictability.

1 Introduction

This paper is a study about the use of artificial intelligence in FPS and RTS games, written for the Human Media Interaction seminar course Designing Entertainment Interaction. I chose the subject because of personal motivation. When I play computer games, most of the time they are either RTS or FPS games.

Medal of Honor: Allied Assault is a particular first-person shooter I've played for quite a while, mostly online multiplayer. Other titles include Unreal Tournament, Call of Duty and Brothers in Arms.

The Command and Conquer series and Warcraft 1, 2 and 3 were particular real-time strategy games I used to play. Because of my personal interest in these games, I thought it would be interesting to investigate the use of Artificial Intelligence in these games.

It is to be noted that there is a difference between how academic researchers and game developers use the term "Game AI". For academic researchers, game AI is the intelligent behavior of bots/non-player characters (NPCs) whereas for game developers, game AI also includes algorithms such as path finding (Spronck, Ponsen, Sprinkhuizen-Kuyper, & Postma, 2005).

1.1 What this paper is about

As also mentioned in the introduction, this paper is a study on the AI that is used in FPS and RTS games. Hence, we will look for scientific publications that describe the use of artificial intelligence in current computer games, within those genres. We limit ourselves to the artificial intelligence in FPS and RTS games for feasibility.

1.2 What this paper is not about

Artificial intelligence is a broad concept and can be difficult to define. We will not be describing AI techniques in detail, and also this paper will not describe all AI techniques that may also be available. Examples include Fuzzy Logic or Speech or Face Recognition.

2 Research Question

Taking my personal interest into account and the scope of the paper for this particular course, I have defined the research question to be:

> "What are currently used techniques of Artificial Intelligence in specific Real-Time Strategy and First-Person shooter game titles?"

The use of the phrase "specific game titles" is of course a bit vague. Fortunately this vagueness can be explained. I expect that exact implementation details of AI in games, explained by the developers of the game will not be easy to find. On the other hand, I do know that there are some companies, such as Valve or DICE, who sometimes do publish detailed implementation reports.

3 Methodology

Scopus was used as the primary resource to find relevant articles for this paper. As a secondary method of finding research papers that included particular game titles, Google Scholar was also used. We quickly describe how the papers were obtained and filtered for both the RTS and FPS genres.

3.1 Real-Time Strategy games

The search query on Scopus was "artificial intelligence rts", which yielded 39 results. Filtering on the title of the papers, the result set was brought down to 15 papers. After reading their abstracts, another 5 papers were excluded, bringing down the total to 10 papers. Combined with additional results that were found using Google Scholar while searching on some popular RTS titles (Warcraft, Starcraft, Command and Conquer), we started with 14 total research papers in the RTS scene.

3.2 First-Person Shooters

On Scopus, the search query was "artificial intelligence fps games", yielding 13 results. Filtering both on title and abstract of the papers, the result set was brought down to 6 total research papers. Additionally Valve, DICE and Guerilla Games had publications available about artificial intelligence in their games.

4 Artificial Intelligence in games

How Non-Player Characters behave in games is most of time fully designed by the developers of the computer game, hence they are not capable of properly responding to situations that were not foreseen by the developer (Hartley & Mehdi, 2009). They are often based on or implemented using rule-based expert systems, finite state machines, or other pre-programmed scripts (Hartley & Mehdi, 2009) (Cole, Sushil, & Miles, 2004). Although NPCs most of the time are not intelligent enough to adapt to unforeseen situations, they are still being referred to as Artificial Intelligence.

From here we will first investigate AI in first-person shooter games, after which we will have a look at how AI is applied in real-time strategy games.

4.1 First-Person Shooter games

In a first-person shooter, the player takes part in a 3-dimensional combat simulation from a first-person perspective. The objective often is to progress to certain waypoints and achieve objectives set by the developer. During single player mode this involves eliminating enemies (NPCs) that try to make it more difficult for the player to achieve an objective. In multiplayer mode, a player usually plays the game simultaneously with other players, in which case the computer does not control the enemies.

First-person shooters often include NPCs that are controlled by means of a finite state machine with hard coded rules and do not include any form of learning. However, recently, behavior trees are beginning to replace finite state machines (Hoorn, Togelius, & Schmidhuber, 2009).

4.1.1 Left 4 Dead

"Left 4 Dead is a replayable, cooperative, survival-horror game where four survivors cooperate to escape environments swarming with murderously enraged "infected" (i.e. zombies)" (Booth, 2009).

In L4D a path finding algorithm is used that uses the A* algorithm, in order to move the computer controlled bots to a certain location. A path that is ultimately used by the bots can be determined using two methods (Booth, 2009). The first one is using path optimization, in which redundant nodes are collapsed. This creates a minimal and direct path, but also costs more CPU time. The second approach is called reactive path following. In this case, while moving to a certain node in the path, bots look ahead for a

node further down the path that will be the next target they navigate to. A disadvantage of this approach is that re-pathing is required while bots are moving. Fortunately re-pathing is cheap. Another advantage is that the resultant path is very close to the optimized path. In Left 4 Dead, reactive path following is used for all actors.

The behavior of bots and their decisions are managed by an intention subsystem in each actor. The intention subsystem is a Hierarchical Finite State Machine, which is different from a traditional FSM in the sense that states can be grouped so that they can be used to share state transitions. The logic is built state by state, where transitions are connected bottom up (Champandard, 2007) (Harel, 1987).

Furthermore, Left 4 Dead contains an AI director that uses structured unpredictability. This is achieved by means of four components: the navigation mesh, which represents the walkable space; the flow distance, which is the travel distance from the starting position to every other position in the navigation mash; the potential visible areas for the survivor team; and finally the active area set, which is the set of areas around the survivor team.

The AI director is responsible for spawning infected on specific locations, but also to maximize drama in the game, that is the player excitement or intensity. In Left 4 Dead the emotional intensity of each survivor is estimated. The intensity increases when a survivor is injured, becomes incapacitated, is pulled or pushed, or when nearby infected dies. The value is decreased over time. Based on the intensity value, the population of infected is modulated.

4.1.2 Killzone

Killzone is a first person shooter game for Sony's PlayStation 2, developed by Guerilla Games, located in Amsterdam, the Netherlands. In Killzone the player participates in a combat setting in the near future. In single player mode players play a little over thirty levels containing trenches, warehouses, mountain passes, et cetera. Most of the time, a team of AI buddies accompanies the player. Killzone also offers six different online multiplayer modes (Straatman, Sterren, & Beij, 2005).

From an overviewing perspective, the AI of Killzone, i.e. the bots or NPCs, tries to achieve the goal that is most desirable in a specific situation. For this, a set of corresponding actions is used. For combat situations this can be picking a destination to walk to, plan a path, move along the path (often accompanied with aiming and firing at hostiles), arrive, and start performing stationary actions. The destination and path of a bot completely rely on dynamic information, which includes the amount of cover from threats, lines of fire, danger zones, and more.

Killzone uses what Guerilla games calls dynamic procedural tactics, of which important components are positions evaluation and tactical path finding. Position evaluation functions are used to determine the attractiveness of position. An example evaluation function is based on the amount of cover it offers from the primary threat. Bots will constantly try to find a better position to move to in their direct surroundings. Position evaluation functions are also used for the A* path finding algorithm, but the challenge there is to keep the run-time costs as low as possible.

4.1.3 Commonalities and differences

While most first-person shooters seem to adapt similar techniques for the AI of their bots, such as A* for path finding and (hierarchical) finite state machine for behavior models, there of course are also differences.

Left 4 Dead uses an implementation of the A* search algorithm for path finding in combination with what is called reactive path following (Booth, 2009). This generates a smooth path that is followed by the bot. The creators of Killzone extended the A* search algorithm with their use of evaluation functions to determine the attractiveness of a specific spot a bot can move to (Straatman, Sterren, & Beij, 2005).

To determine useful spots where bots can take cover from enemies, the AI in Killzone also applies evaluation functions that compute the amount of cover a certain spot has. This is partially based on the location of the primary and/or secondary threats for example (Straatman, Sterren, & Beij, 2005). The first-person shooter Battlefield: Bad Company 2, developed by DICE, uses the Frostbite AI, which is a coupling between behaviors, decisions, animations, scripting and voice overs (Hedberg, 2010). This AI system is able to automatically detect areas that supply cover.

4.2 Real-Time Strategy games

Initially the idea here was to provide a description of the use of artificial intelligence in specific game titles within the genre of real-time strategy games. Unfortunately it is very difficult to find specific implementation details about this subject. Instead, we will address the common challenges and techniques that are used in RTS games.

Real-time strategy games are a form of simulation-based games, which are accompanied by severe time constraints in which decisions must be made quickly. Examples of RTS games include the popular titles Age of Empires, Command and Conquer or Starcraft. RTS components also include resource management, decision making under uncertainty, spatial and temporal reasoning, collaboration, opponent modeling, group movement (flocking), path finding and adversarial real-time planning (Buro, 2003) (Millington, 2006). When developing AI for RTS games, it should also be capable of handling those issues.

In real-time strategy games we can distinguish two types of artificial intelligence that should be taken care of. An important aspect is of course strategic planning, i.e. the computer may 'ask' itself "what area should I focus on next", "or what opponent is best to attack first and from what flank"? But there is also AI required to control the individual units that move around, for example the melee or tank units that are part of the specific game environment (Millington, 2006).

4.2.1 Higher level strategic planning

Much of the behavior of NPCs in games is scripted, making them repetitive as well as predictable (Cheng & Thawonmas, 2004). If the artificial intelligence in games were to be adaptive, it would surely improve the entertainment value and replayability of computer games. Dynamic scripting can be used to address this problem, which in its essential is a reinforcement learning technique for scripts that define the strategy of the AI (Spronck, Ponsen, Sprinkhuizen-Kuyper, & Postma, 2005). Ponsen et al. describe the use of dynamic scripting for a real-time strategy game that automatically generates high-quality domain knowledge, i.e. the strategic or tactics, in order to deliver a strong adaptive NPC (Ponsen, Muñoz-Avila, Spronck, & Aha, 2005).

Dynamic scripting is applied in Wargus[1], an open-source RTS game based on the popular game Warcraft II by Blizzard Entertainment, and is able to real-time generate scripts that determine the strategy of the AI (Ponsen, Muñoz-Avila, Spronck, & Aha, 2005). This is realized by consulting a tactics database that was created using domain-specific

[1] http://wargus.sourceforge.net/

knowledge. Each state-specific tactic is assigned a weight. At a certain moment in the game, a specific tactic is selected based on the weight value and if the use of the tactic resulted in a positive outcome – i.e. it successfully defeated a group of opponents – its weight is increased. In case of negative outcomes, the weight is decreased.

4.2.2 Unit behavior

An important part of character behavior is of course path finding. Some form of supreme commander that determines the strategy and tactics may order a group to move from location A to location B, which means that the units in question would have to find a path from location A to B. While doing so, rocks, walls and other obstructive things should be taken into account, much like as has to be done with path finding in a first-person shooter game. In Warcraft: Orcs and Humans and Command and Conquer, path finding was the primary challenge of the artificial intelligence, as it had to cope with enormous grid-based levels (Millington, 2006). For these two games especially the available computing power was an issue, since they were published in 1994 and 1995, respectively, when we didn't have as much computing power available as we have now.

Group movement, or flocking, is also an important part of unit behavior. A number of units can often be grouped together and thus should also move together. This is often accomplished by formation motion systems, which uses pre-defined formation patterns (Millington, 2006). If a group can consist of a very large number of units, as is possible in many games, the pattern should be scalable to every number of units. In Full Spectrum Warrior, a game by Pandemic Studios and released in 2004, squad members form a line when next to a while but form a wedge when in an open field (Millington, 2006). Note that Full Spectrum Warrior is not a real-time strategy game, but for the explanation of the concept these kinds of formations could of course also be applied to RTS games.

5 Conclusion

Artificial intelligence in games is not yet at a very sophisticated level. Although many improvements have been made over time, most of the time the AI in FPS and RTS games is still very much scripted, making it highly predictable. And predictability is of course not good for replayability, as players will not be challenged enough anymore over time. The proposed technique by Ponsen et al., which uses dynamic scripting in order to improve the adaption of computer players to their human opponents, sounds promising and perhaps not too difficult to realize. It is a punishment and reward system that will ultimately figure out which tactics work best.

Furthermore there are quite a few proposals to improve the artificial intelligence that today is being used in computer games, as we will also briefly present in the next section. The publications by companies like Valve and DICE show that they are working on improvements in AI, and are willing to openly present it and discuss about it.

A reason why there isn't a very high pace in improving the artificial intelligence in games is perhaps because most people nowadays prefer playing against human opponents over the Internet. The reason for that may be because artificial intelligence in games lacks in providing a proper challenge and that the Internet now almost is a common good for most people. Either way, game AI is an area that is open for new developments.

5.1 Research and future work

Much of the research that can be found about the use of artificial intelligence in real-time strategy and first-person shooter games concerns the proposal of concepts that can be

used to improve and further develop the use of artificial intelligence in computer games. This section of the report provides brief descriptions of some of those concepts.

To help players during gameplay, Galli et al. propose a framework for first-person shooters to collect dataset from game sessions, learn a policy to automatically select the proper weapon and to deploy the learned models in the game to replace weapon-switching policies (Galli, Loiacono, & Lanzi, 2009). Policarpo et al. suggest the use of dynamic scripting in first-person shooters to improve unpredictability (Policarpo, Urbano, & Loureiro, 2010). Weber et al. present reactive planning for developing multi-scale AI (Weber, Mawhorter, Mateas, & Jhala, 2010), a system that is able to reason and perform action at different levels of granularities; something that one would need in real-time strategy games for example.

6 References

Weber, B. G., Mawhorter, P., Mateas, M., & Jhala, A. (2010). Reactive planning idioms for multi-scale game AI. *IEEE Conference on Computational Intelligence and Games*, (pp. 115-122).

Buro, M. (2003). Real-Time Strategy Gaines: A New AI Research Challenge. *IJCAI'03 Proceedings of the 18th international joint conference on Artificial intelligence* (pp. 1534-1535). Morgan Kauffman.

Booth, M. (2009). The AI Systems of Left 4 Dead. *Artificial Intelligence and Interactive Digital Entertainment*. Valve.

Champandard, A. (2007, 09 05). *The Gist of Hierarchical FSM*. Retrieved 04 12, 2011, from AI Game Dev: http://aigamedev.com/open/articles/hfsm-gist/

Cheng, D. C., & Thawonmas, R. (2004). Case-Based Plan recognition for Real-Time Strategy games. *5th Game-on International Conference*, (pp. 36-40).

Cole, N., Sushil, J., & Miles, C. (2004). *Using a Genetic Algorithm to Tune First-Person Shooter Bots*. University of Nevada, Department of Computer Science.

Galli, L., Loiacono, D., & Lanzi, P. L. (2009). Learning a Context-Aware weapon selection policy for Unreal Tournament III. *IEEE Symposium on Computational Intelligence and Games*, (pp. 310-316).

Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming , 8*, 231-274.

Hartley, T., & Mehdi, Q. (2009). Online Action Adaptation in Interactive Computer Games. *ACM Computers in Entertainment , 7* (2), 28:1-28:31.

Hedberg, M. (2010). Building the Battlefield: Bad Company 2 AI Experience. *Game AI Conference 2010*. DICE/EA Games.

Hoorn, N., Togelius, J., & Schmidhuber, J. (2009). *Hierarchical Controller Learning in a First-Person Shooter*. IEEE.

Millington, I. (2006). Real-Time Strategy. In *Artificial Intelligence for Games* (pp. 809-812). Elsevier.

Policarpo, D., Urbano, P., & Loureiro, T. (2010). Dynamic scripting applied to a First-Person Shooter. *5th Iberian Conference on Information Systems and Technologies*, (pp. 1-6).

Ponsen, M. J., Muñoz-Avila, H., Spronck, P., & Aha, D. W. (2005). Automatically acquiring domain knowledge for adaptive game AI using evolutionary learning. *The Seventeenth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence*, (pp. 1535-1540).

Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I., & Postma, E. (2005). Adaptive game AI with dynamic scripting. *Machine Learning , 63* (3), 217-248.

Straatman, R., Sterren, R., & Beij, A. (2005). *Killzone's AI: dynamic procedural combat tactics.* Guerilla Games/CFG-AI.