AI in First Person Shooters

Introduction

In the modern gaming world the first person shooter (FPS) genre has become extremely popular. There are two sub-genres of the FPS that I will introduce in the paper. First is the role playing first person shooter (RP-FPS), which generally pitches the player many enemies which are controlled by the game AI, and must be challenging but fun. Second is the competitive first person shooter (C-FPS), which is the more popular of the two. This allows players to fight each other, which means that you don’t need a computer combatant, but some games have them as an option for the player.

Making an AI smart enough to compete against a skilled player (in a human like fashion) is extremely hard to do, which is why some developers just opt out of providing a computer combatant. In this paper I will cover methods that have been used to implement non-player characters (NPC's) in both RP-FPS's and C-FPS's. By establishing the pro's and con's of each approach, and their success in specific sub-problems of the field; I will show that each is solving a subproblem of the overall problem. A hybrid approach would bring more overall successful in creating a competitively balanced AI.

There has always been AI used in RP-FPS's from the very beginning. Although AI scripts were very simplistic. For example in the popular game “Doom” considered by most as the father of the FPS genre itself; Enemies paths were determined by the straight line vector to the player’s position. Also enemies projectile vectors were always aimed at the players current position, instead of ahead of them. As the genre evolved, so did the complexity of what players expected.

Game Theory Methods
In a published conference paper by Rune Rasmussen (Rasmussen, 2008), a method is proposed to improve C-FPS AI. The article uses game theory as a mechanism for improving AI complexity in FPS’s. In game theory you look at the move that is going to be the most beneficial to you, with the overall goal being to win. The approach in the article is goal driven, and focuses on movement that keeps the AI in the most beneficial position possible. For example standing in the open will increase the amount of angle’s you can be shot from making that a low score move. This makes sense, because real players do this all the time. They look at what benefits and risks they take by making a move, and when the benefits outweigh the risks, they do it.

This approach was successful in creating an AI that made decisions like a human would, but did not succeed in an overall playable AI. The decisions and knowledge of the game are all things that can be controlled or more importantly, expected. Environments, and human players are both examples of unexpected events, where this method will fail. The maneuverability of environment can be remedied by precomputing all obstacles, objectives, vantage points, and C-space of the 3D environment. Which would mean that you have to redevelop your AI for every new area, or map you want it to play on.

Another relevant paper written by Kenneth Hullett (Hullett, 2010) defines a quantifiable measure for aspects of map/level design in FPS games. It asks the question “what aspects of level design are hard, challenging, fun, balanced, and generally successful”? Game developers have found out through iteration what tends to work for giving the player a good experience. Using pattern matching, and cause-effect relationships between level design and gameplay gave a predictable results in most maps. For example, high elevated areas of the map with limited ways up it almost always turn into “sniper positions”. These positions create fun and challenging game dynamics, but can be unfair and frustrating if the sniper is given too much cover.
This research can be used to help make a heuristic to help guide an AI through an unfamiliar environment; Which would lead to smarter AI path decisions. So with the methods talked about above game mechanics, and decision making can be achieved. Although there are drawbacks in dealing with unexpected events, or playing in a human like fashion. People generally try to do unexpected things in order to get the jump on their opponent. an AI with the above methods would not be able to act in an unexpected way. The next section of my paper deals with these issues.

**Adaptation Methods**

An extremely interesting article in “Computers in Entertainment” magazine written by Thomas Hartley(Hartly, Mehdi, 2009), attempts to create a more adaptable NPC. A case-based reasoning approach was taken to achieve this. the cases were populated by recording all actions of players while playing a competitive FPS called “Unreal Tournament 3”. In order to increase the accuracy of matching cases, a dual state representation was used. Basically the unpredictability of human players were recorded in all the different cases. As new movement is encountered by a AI, the cases are updated accordingly to account for these new movements. This approach resulted in an NPC able to more accurately predict human players’ movements.

Another great source comes from a paper written by David Conroy, Peta Wyeth, and Daniel Johnson (Conroy, Wyeth, Johnson, 2011). The goal of this research was to construct a model of how a human player react’s to general scenarios in a FPS. Which can then be used to make an NPC which acts more human like. This is done by analyzing the gameplay of humans, much like a case study. Although this research differs by accounting for human perceptions/and behavior. To be specific, A player has a completely different state of mind going into a game where the goal is to capture a flag vs. get the most kills.
Another unique aspect of recorded human behavior was facial expressions, which helps determine what scenarios surprise the player, stress the player, or doesn't seem to affect the player. The following information can be used to mimic human actions, for example in a stressful or unexpected situation it is common for him/her to make mistakes. The model design is centered around goal driven decision, and resultant actions pairs.

The final relevant paper I will draw information from is written by Stephano Zanetti, and Abdennour el Rhalibi (Zanetti, Rhalibi, 2004). This work discusses the use of machine learning to achieve more human like behavior from NPC’s. A number of “sub-behaviors” are defined such as movement, prediction, and shooting. A neural network is defined for each of the listed sub-behaviors. Genetic algorithms were then used to train the neural networks. The data used to train the neural networks were recorded from analyzing expert human players, playing against each other in a 1v1 battle. This approach had a problem in learning when to shoot, since a player generally only shoots during 5 percent of a whole match.

This problem isn’t specific to this paper. All the research above deals with learning from events, and interaction with players. AI’s in all the above methods perform badly when it comes to making it’s own decisions not based on another player. Given enough time and resources, some of the above methods could probably learn what areas of the map, or what weapons are best used in specific scenarios. Although this would be remedied much easier by applying game theory and expert systems (Game Theory Methods).

Conclusion

Game theory methods showed that a quantifiable collection of cases could be constructed which resulted in AI that made extremely smart decisions. Although it lacked in adaptability, which would allow a player to take advantage of. Also it isn’t as fun when you play an opponent who always knows where you are at, since you can’t surprise them or outsmart
them. A final issue with these approaches involves executing decisions in a potentially changing environment.

All of the drawbacks of the above method can be solved using methods discussed in the adaptation section. Whereas the drawbacks of adaptation is vice-versely solved by using game theory approaches. It is obvious that a combination of the two types of methods would be beneficial.

Specifically work from (Rasmussen, 2008), and (Conroy, Wyeth, Johnson, 2011) both dealt with goal driven, case based, rule-action pairs. If the information of cases could be combined, it would provide a human like, but heuristically beneficial move. It would play based on the game theory cases, but in the case that it is losing, it would be able to adapt and avoid the situations that are getting it killed. Also when getting in a beneficial spot deemed “best” by the game theory information, the human characteristic cases would overrule. Causing it to feel “stressed” because it has been staying in the same place for too long without finding an enemy.

These are just examples about the potential benefits of using multiple methods. These ideas were not tested or built. The goal of this paper was to shine light on methods already tried, and to combine solutions of subproblems to achieve a better approach to the overall complexity of FPS AI.
References

Ordered as they appear in the paper.

Rasmussen, Rune. “A game theory approach to high-level strategic planning in first person


Hartley, Thomas, and Mehdi, Quasim. “Online action adaptation in interactive computer
games”. ACM Volume 7 Issue 2, June 2009.

Conroy, David, and Wyeth, Peta, and Johnson, Daniel. “Modeling player-like behavior for
game AI design”. ACM 2011.

Zanetti, Stephano, and Rhalibi, Abdennour, el. “Machine learning techniques for FPS in Q3”.