

# Evolution of Artificial Intelligence

## In Video Games: A Survey

**Ken Mott**

University Of Northern Iowa

[mottk@uni.edu](mailto:mottk@uni.edu)

### Abstract

Artificial Intelligence in video games has grown greatly over the recent years. From the basic Tracking AI of Atari's Pong to the table of locations used in Sony's Killzone for the Playstation 2. The style of AI used varies greatly between genres of games. There is a large difference between how the AI in a First Person Shooter acts as opposed to the AI in a Real Time Strategy game. There is also advancements in video game AI to make characters more life-like, called A-Life. The future of AI in video games is bright. More time is being spent in game development working on making good AI than ever before. This paper will go into much more detail on all of these topics.

### 1. Introduction

The Artificial Intelligence that has been applied to video games has evolved greatly over the years. The attention being given to the Artificial Intelligence aspect of video games is growing as well. First, this survey paper gives some information about how Artificial Intelligence in video games got started, such as the basic algorithms used in Pong. Then I will go into detail about the many different methods of Artificial intelligence used in today's video games. Lastly, I will discuss the future of Artificial Intelligence in video games, and the challenges it will face.

### 2. History of game AI

One of the first video games ever created and released to the public was Atari's Pong. It was a simple game where the player and a computer controlled a paddle and used it to guard their side of the screen from the ball bouncing in the middle. The artificial intelligence implemented in this game was very simple. The computer's reaction was solely based on the position of the ball. If the ball was currently higher than the paddle was, the computer would move the paddle up. If the ball was below the paddle it would move down. This is known as Tracking AI (Charles, Fyfe, Livingstone, and McGlinchy 2008). The actual code for deciding which way to move could be written in about 10 lines. Not many people would consider this to be artificial

intelligence, but it is where artificial intelligence began in video games, with simple if else statements.

Another early form of Artificial Intelligence was seen in board games that had been made into video games to allow a computer opponent for the user. One example of this is Backgammon for the Atari 2600. It used a form of AI called Path Finding (Charles, Fyfe, Livingstone, and McGlinchy 2008). This is also commonly known as Searching. The Artificial Intelligence in this game was designed in a way that the computer would make its move based on the current state of the board. The search algorithm most commonly used was called A\*. This is more easily seen as Artificial Intelligence because this is the same way a human would play Backgammon, or any other board game.

Yet another form of Artificial Intelligence in early video games was seen in games such as Space Invaders by Atari. They implemented a form of Artificial Intelligence called Pattern AI (Charles, Fyfe, Livingstone, and McGlinchy 2008). This is when an object would follow a preset path or do something in a specific preset pattern. To make it seem more humanlike the pattern the object followed was usually somewhat random. For example, in Space Invaders when an enemy ship would drop down to try and hit the player, it would follow a preset line down the screen. To make it seem more real, it didn't just follow a straight line, or just track the player; it curved and turned as it came down the screen. A human would do this in attempts to fool the other player.

These were the three most commonly used methods in the earliest days of Artificial Intelligence. All these methods are still used to some degree in the Artificial Intelligence systems of the video games of today.

### 3. Modern Day game AI

In today's video games, the Artificial Intelligences needed for different games vary greatly. I will discuss three different genres of video games that are the most common genres Artificial Intelligence is used in;

First Person Shooters, Real Time Strategy, and Simulation.

### 3.1 First Person Shooters

Artificial Intelligence that is used in First Person Shooters is probably the most complex Artificial Intelligence being used in video games today. It has to account for many different variables when making decisions.

One of the most common things game developers try to implement in modern First Person Shooters is the idea of finding cover when under fire. In games like Killzone for the Playstation 2 the computer enemies try to find cover from the player's current position, and trying to do it quickly. They do this through the use of waypoints placed on the map. They use these as possible position they could go to. They then evaluate the effectiveness of making this move based on things like; how much cover it gives, does it give a tactical advantage over the player, and is it going to be able to get there quickly or by a safe route. To see if the player has a line of sight on the computer, and vice versa, the developers for Killzone decided to implement a table that showed the how much line of sight each spots offers in correspondence to the other spots. While it may seem like it would take up memory space, the developers stated that it took only 64 Kb to make a lookup table for 4000 waypoints (Hachman, 2005). This decision allowed the computer to react more quickly to the player, giving more of a challenge. However there were some weaknesses with this system. One of the biggest problems they found was that the waypoints were modeled in two dimensions, while the game was in three dimensions. They were unable to take height of places into account. This could sometimes cause the computer to run into a wall forever. Another problem found with this system was that if the player moved while the computer was doing the calculations of where it should move, it would then abort the move to reevaluate the move based on the player's new location (Hachman, 2005). This makes it seem as if the computer is not responding, which is not a desirable thing to have happen.

Another area of Artificial Intelligence many game developers are working on is the area of teamwork among computer controlled units. For many years the computer units would act solely for the purpose of itself, not as a team (Schreiner, 2003). Recently however, First Person Shooters have been implementing different aspects to their computer units, allowing them to act as a group instead of a sole unit. One place this is seen is in Microsoft's game Halo 3. The enemies in this game almost always travel in groups and act as a group. When the player for example goes into hiding for a minute, most of the group will stay behind to guard their current location, while a couple of the enemies will go searching for the player. This adds a large level of realism to First Person Shooter games. Halo 3 also shows teamwork in other ways. Occasionally some of the enemies will use

suppressing fire on the player while allowing some of their allies to find a better vantage point or to take cover.

While some advancement has been made in the area of First Person Shooter Artificial Intelligence, there is still much room for improvement. According to Schreiner (2003), some main differences between humans and computer units in First Person Shooters is that humans find ways to use maps in a way that they weren't intended, such as exploiting glitch spots. Another thing humans do is seek cover, even when there is no danger present. An example of this is when humans need to reload their weapons; they tend to take cover if they are able before they reload their weapon. Another example is avoiding areas with a lot of hostiles (Schreiner, 2003). Most computer controlled units charge into an area even if there are a lot of enemies present. While some work and research is being put into making these aspects better, it hasn't been achieved yet.

### 3.2 Real Time Strategy

Real Time Strategy games differ greatly from First Person Shooter games. These types of games usually involve the players building up resources in order to purchase units or buildings. They then use these to try and complete an objective or defeat their opponent, who is trying to do the same thing. While in First Person Shooters the computer is usually able to see what the human player is doing, in Real Time Strategy they may not know what they are doing until it is too late to adjust to it.

The way many Real Time Strategy Artificial Intelligence systems are put into practice is by the use of scripts. These are lists of actions that the computer will perform one after another (Ponson, Spronck, Aha, and Avila, 2006). This method works for the most part. Usually in Real Time Strategy games there are different races or factions the player and computer can choose from, and each of these would have different scripts with them. But, over time the human player could learn the scripts and the game would become less interesting. Playing against someone that doesn't change their strategy from game to game is not very fun. That is why some research is being put into the idea of having dynamically scripting Artificial Intelligence. This would mimic the way a human plays much better. It would start the game with a script already in place for a general strategy, the way most humans play, but would also adjust the script based on learned knowledge in the game.

Another form of Dynamic Scripting is not giving the computer a script to begin with, but letting it script as it goes. Ponson, Spronck, Aha, and Avila (2006) have an algorithm for this form of scripting that in the end creates an adaptive Artificial Intelligence agent. They begin by looking at the opponents strategy and tries to formulate a counter strategy, much like a human would do. This is called the Evolutionary Algorithm. It then translates this general strategy into a series of steps to achieve the strategy, called the Knowledge Transfer step, and then puts it into practice. It periodically does a recheck of the

opponent's strategy and makes changes to its script based off of its new knowledge. When they tested this method against a static agent, one that used the same predetermined strategy every time, after each test, they found the dynamic agent improved each time. As it learned more about how the opponent acted, it could formulate better strategies to counter them.

One major problem developers of Real Time Strategy games run into is having certain strategies or units always being too powerful or too weak. For a long time, human testers were used to try and find these dominant and inferior choices. Recently however, programmers have developed an Artificial Intelligence agent to find these dominant and inferior choices. It does this through the use of Genetic Algorithms (Salge, Lipski, Mahlmann, and Mathiak, 2008). The fitness function of these Genetic Algorithms is based on how well the computer did. They have the computer try many different strategies until it finds some good ones. If the developers see the different strategies they found never use a certain unit, or if one unit is used a lot in all of them, they would realize that the units in question were either too weak, or too strong, and need to be balanced out more.

### 3.3 Simulation

For the purposes of this paper, I am defining Simulation as games such as board games like Chess, and other games such as the Sims. Simulation games have not evolved much in the way of the Artificial Intelligence they have used, but they have made some advancement to make the computer seem more lifelike.

Games like chess seem like they could easily be given an Artificial Intelligence agent that simply makes a tree of all possible moves and pick the best one. But chess is a game with many different moves at any given turn, and games can last for many turns. If you were to assume that you could make 25 moves on any given board (a good average), and could defeat your opponent in 20 moves (a very short game), the resulting tree would have  $25^{40}$  nodes in it to decide upon (Bridger, Groskopf, 2000). This makes it near impossible to create a full search tree. Instead however, different methods have been used to limit the search tree to a much more manageable state. These methods could be ones such as Depth Limited, which would look ahead only a few turns to determine its move, or a heuristic search which only expands on moves it believes would be the most advantageous (Bridger, Groskopf, 2000). Most commonly the heuristic method is used, and it uses a method known as A\*. This is where the programmer provides values, known as heuristics, to help the computer determine which path seems the best to take. These values could be things such as distance to the goal, the cost of making that move, or many other things. The heuristics don't have to be accurate (although it is better if they are) but they cannot overestimate the actual value the move would provide. The way A\* works is to not only look at the heuristics provided, but also to look at how much cost has accumulated or how much closer they

really are to the destination in order to help balance out poor heuristics. If the heuristic says the trip should bring to 20 miles closer, but only brings you 2 miles closer, the A\* method will notice this and take into account the real distance remaining to find the optimal solution. In the game of Chess, this heuristic could be giving a value to current state of the board. If the move would allow you to keep your pieces and maybe remove some of the opponents pieces, that would seem like a desirable move. Most video games that are board games use this method in their Artificial Intelligence agents.

The other type of Simulation game I will discuss is games like the Sims. The Sims is a game where the player builds a town or theme park or some other similar place. After the player builds this place, the people of the game, called the Sims, begin living a normal life in the town, or visiting the theme park. The player then gets to see what things the people would like to have added, such as another ride, or an extra police station. They also get to watch the Sims live in the town and go about their daily routine.

While this game doesn't have much in the way of what most people would think Artificial Intelligence, it does have something in it called Artificial Life. This is when programmers try to make characters seem more human-like, not just in their decisions, but also in their behavior. The game Sims is where the player creates a world or town for the people, known as Sims, to live in. After building up the town a bit the Sims will decide to do some things that make them happy. Each Sim is programmed with a set of behaviors that it can do, what has to be available for it to do the particular behavior, and something to determine its current state of "happiness" (Mata, Martinez, 2008). When the Sims are not being directly controlled by the player, they will make decisions about what they want to do and will go and do it. This lets the Sims develop patterns of behavior and do the things it likes to do. For example, one Sim may love to go skateboarding; it will decide to go skateboarding more often than it would go to the library for instance. While this isn't really a form of Artificial Intelligence, it does add a human like quality to the Sims. Humans tend to do what makes them happy, and it is interesting to see a computer controlled unit do the same thing.

## 4. Future of Video Game Artificial Intelligence

In the past the main goal of video game companies has been to make the worlds and characters they create look more lifelike. The graphics used in video games today is extraordinary, often regarded as near-photo realism. Now the demand for better Artificial Intelligence agents is growing quickly. Video game companies are now putting forward more money and time into making better Artificial Intelligence agents. According to Charles, Fyfe, Livingstone, and McGlinchey (2008) the International Game Developers Association has been working to establish a set of standards in Artificial Intelligence for future games.

In past years one of the largest things slowing down the growth of Artificial Intelligence in video games is the fact that to improve the Artificial Intelligence, companies often had to sacrifice some graphical quality, which was not desired. In the past few years however, the processing power and memory space of computer chips has grown to the point that companies can now keep graphics at the level they currently are at now and still increase the capabilities and speed of their Artificial intelligence agents (Charles, Fyfe, Livingstone, McGlinchey, 2008).

## 5. Challenges for the Future of Video Game Artificial Intelligence

Different games present different sets of challenges to game developers and programmers. I will discuss the three areas of video games used in the previous sections and the challenges they will face in the future.

### 5.1 Challenges for First Person Shooter Games

In the area of First Person Shooter games, the enemy has always been in a sort of “guard state” (Schreiner, 2003). The enemy would not do anything until the player would walk into their area. This creates a lot of repetitiveness in video games, which decreases the overall enjoyment of it. Developers and programmers have been working on ways to have the enemy hunt the player in some way so the levels in the game weren’t the same every time you played through it. As of now, this has not yet been achieved.

### 5.2 Challenges for Real Time Strategy Games

In the area of Real Time Strategy games the problem has been solved partially, but not completely. As previously discussed, Ponson, Spronck, Aha, and Avila (2006) have an algorithm used to generate an adaptive Artificial Intelligence agent. This agent did improve its strategy after each test against the same strategy, but has not been successfully tested against another agent that changes its strategy. As of now it is able to find patterns the opponent has and exploit them to win. But against an opponent who doesn’t use the same strategy every time, it is near impossible to find patterns to exploit. This is the main goal of Real Time Strategy programmers, to create an adaptive Artificial Intelligence that can adjust to different strategies quickly, offering more of a challenge to players. This will also offer much more replay ability to the player, since they will be able to essentially play against someone different every time they play the video game.

### 5.3 Challenges for Simulation Games

The area of Simulation games is not growing very much at all. The Artificial Intelligence currently employed in these games is very good as is. Until we can have a large enough memory space to make a decision

tree for an entire game of chess, there are not many foreseeable advancements in the future. The only real advancement in the foreseeable future is to make characters behave in a more lifelike manner. This applies not just to Simulation games, but to all types of games.

## 6. Conclusion

The area of Artificial Intelligence in video games is finally getting some of the recognition it deserves. It is also being given more funding and time dedication than ever before. The Artificial Intelligence in video games has been evolving ever since video games started, and it will continue growing for a long time.

## References

- Bridger B., & Groskopf C. (2000). Fundamentals of Artificial Intelligence in Game Development. *Proceedings of the 38th annual on Southeast regional conference*, p. 51-55.
- Charles, Fyfe, Livingstone, & McGlinchey. (2008). *Contemporary Video Game AI*: IGI Global.
- Delgado-Mata C., Ibanez-Martinez, J. (2008). AI Opponents with Personality Traits in Uberpong. *Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment, article 1*.
- Hachman M. (Sept. 6<sup>th</sup>, 2005). How AI works in FPS Games. *ExtremeTech*. Retrieved March 30<sup>th</sup>, 2009, from <http://www.extremetech.com/article2/0,1558,1855839,00,asp?kc=ETRSS02129TX1K0000532>
- Ponson M., Munoz-Avila H., Spronck P., & Aha D. (2006). Automatically Generating Game Tactics through Evolutionary Learning. *AI Magazine, Volume 27*(issue 3), p. 75-84.
- Salge C., Lipski C., Mahlmann T., & Mathiak B. (2008). Using genetically optimized artificial intelligence to improve gameplaying fun for strategical games. *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*. p. 7-14.
- Schreiner T. (2003). Artificial Intelligence in Game Design. *AI-Depot*. Retrieved March 30<sup>th</sup>, 2009, from <http://ai-depot.com/GameAI/Design.html>