

AI in Video Games: Toward a More Intelligent Game - Science in the News

by Harbing Lou
figures by Shannon McArdel

AI, the foundation of all video games

If you have ever played a video game, you have interacted with artificial intelligence (AI). Regardless of whether you prefer race-car games like *Need for Speed*, strategy games like *Civilization*, or shooting games like *Counter Strike*, you will always find elements controlled by AI. AIs are often behind the characters you typically don't pay much attention to, such as enemy creeps, neutral merchants, or even animals. But how does AI found in gaming relate to the AI that tech giants talk about every day?

Playing against an AI

Recently [Elon Musk](#) has warned the world that the fast development of AI with learning capability by Google and Facebook would put humanity in danger. Such argument has drawn a lot of public attention to the topic of AI. The flashy vision AI described by these tech giants seems to be a program that can teach itself and get stronger and stronger upon being fed more data. This is true to some extent for AI like [AlphaGo](#), which is famous for beating the best human Go players. AlphaGo was trained by observing millions of historical Go matches and is still learning from playing with human players online. However, the term "AI" in video game context is not limited to this self-teaching AI.

Rather than learn how best to beat human players, AI in video games is designed to enhance human players' gaming experience. The most common role for AI in video games is controlling non-player characters (NPCs). Designers often use tricks to make these NPCs look intelligent. One of the most widely used tricks, called the [Finite State Machine](#) (FSM) algorithm, was introduced to video game design in the 1990s. In a FSM, a designer generalizes all possible situations that an AI could encounter, and then programs a specific reaction for each situation. Basically, a FSM AI would promptly react to the human player's action with its pre-programmed behavior. For example, in a shooting game, AI would attack when human player shows up and then retreat when its own health level is too low. A simplified flow chart of an FSM is shown in the following image (Figure 1). In this FSM-oriented game, a given character can perform four basic actions in response to possible situations: aid, evade, wander and attack. Many famous games, such as [Battle Field](#), [Call of Duty](#), and [Tomb Raider](#), incorporate successful examples of FSM AI design. Even the turtles in [Super Mario](#) have a rudimentary FSM design.

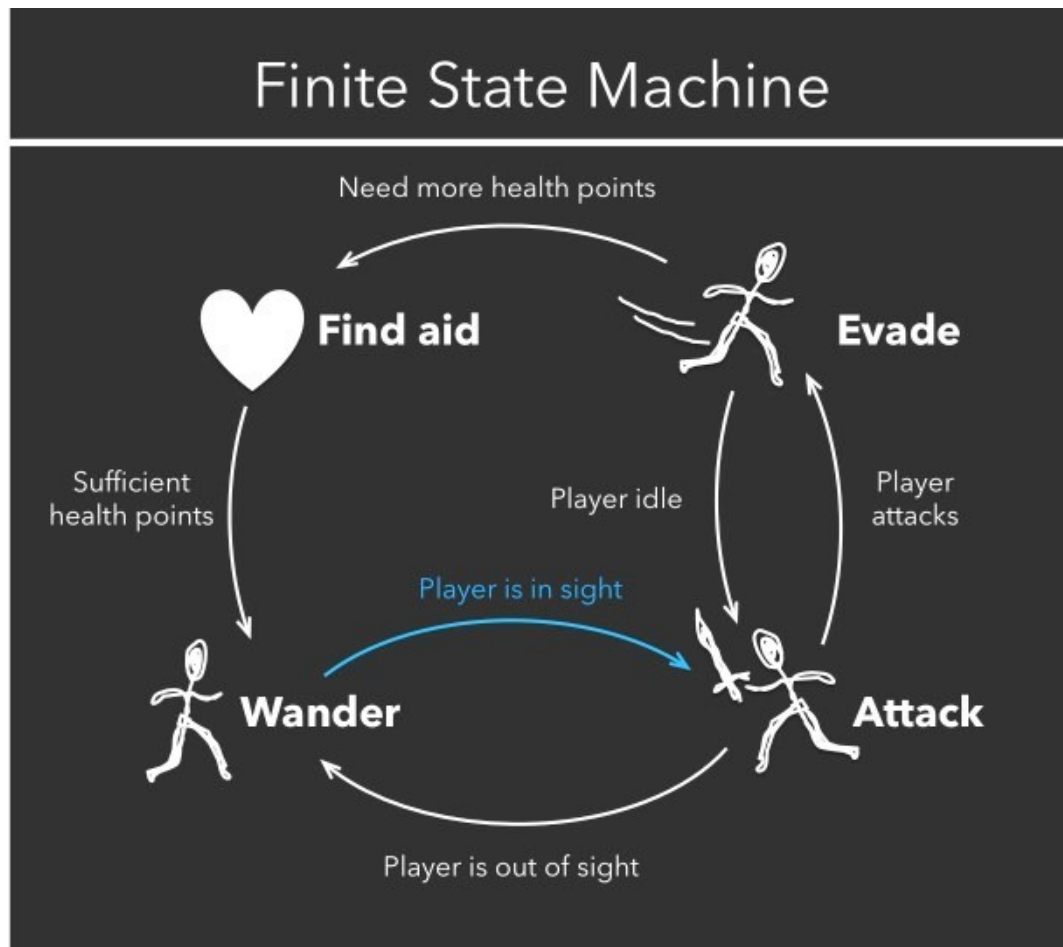


Figure 1: A simplified flow chart of how the Finite State Machine algorithm works in a shooting game. In this game, an NPC would begin with ‘wander’ status, and then engage in ‘attack’ if a human player is near (orange arrow). If the player is out of sight, the NPC goes back to ‘wander.’ In other words, NPCs are always ‘wandering’ when you cannot see them. If a player is attacking back, NPCs can ‘evade.’ If NPCs’ healthpoints are low, they can go ‘find aid’ and then ‘wander’ again.

An obvious drawback of FSM design is its predictability. All NPCs’ behaviors are pre-programmed, so after playing an FSM-based game a few times, a player may lose interest.

A more advanced method used to enhance the personalized gaming experience is the [Monte Carlo Search Tree](#) (MCST) algorithm. MCST embodies the strategy of using random trials to solve a problem. This is the AI strategy used in Deep Blue, the first computer program to defeat a human chess champion in 1997. For each point in the game, Deep Blue would use the MCST to first consider all the possible moves it could make, then consider all the possible human player moves in response, then consider all its possible responding moves, and so on. You can imagine all of the possible moves expanding like the branches grow from a stem—that is why we call it “search tree”. After repeating this process multiple times, the AI would calculate the payback and then decide the best branch to follow. After taking a real move, the AI would repeat the search tree again based on the outcomes that are still possible. In video games, an AI with MCST design can calculate thousands of possible moves and choose the ones with the best payback (such as more gold).

A similar algorithm has also been applied in many strategy games. However, since the possible moves are much more than in chess, it is impossible to consider all of them. Instead, in these games the MCST would randomly choose some of the possible moves to start with. Therefore, outcomes become much more uncertain to human players. For example, in [Civilization](#), a game in which players compete to develop a city in competition with an AI who is doing the same thing, it is impossible to pre-program every move for the AI. Instead of taking action only based on current status as with FSM, a MCST AI evaluates some of the possible next moves, such as developing ‘technology’, attacking a human player, defending a fortress, and so on. The AI then performs the MCST to calculate the overall payback of each of these moves and chooses whichever is the most

valuable.

A simplified flow chart of the way MCST can be used in such a game is shown in the following figure (Figure 2). Complicated open-world games like *Civilization* employ MCST to provide different AI behaviors in each round. In these games, the evolution of a situation is never predetermined, providing a fresh gaming experience for human players every time.

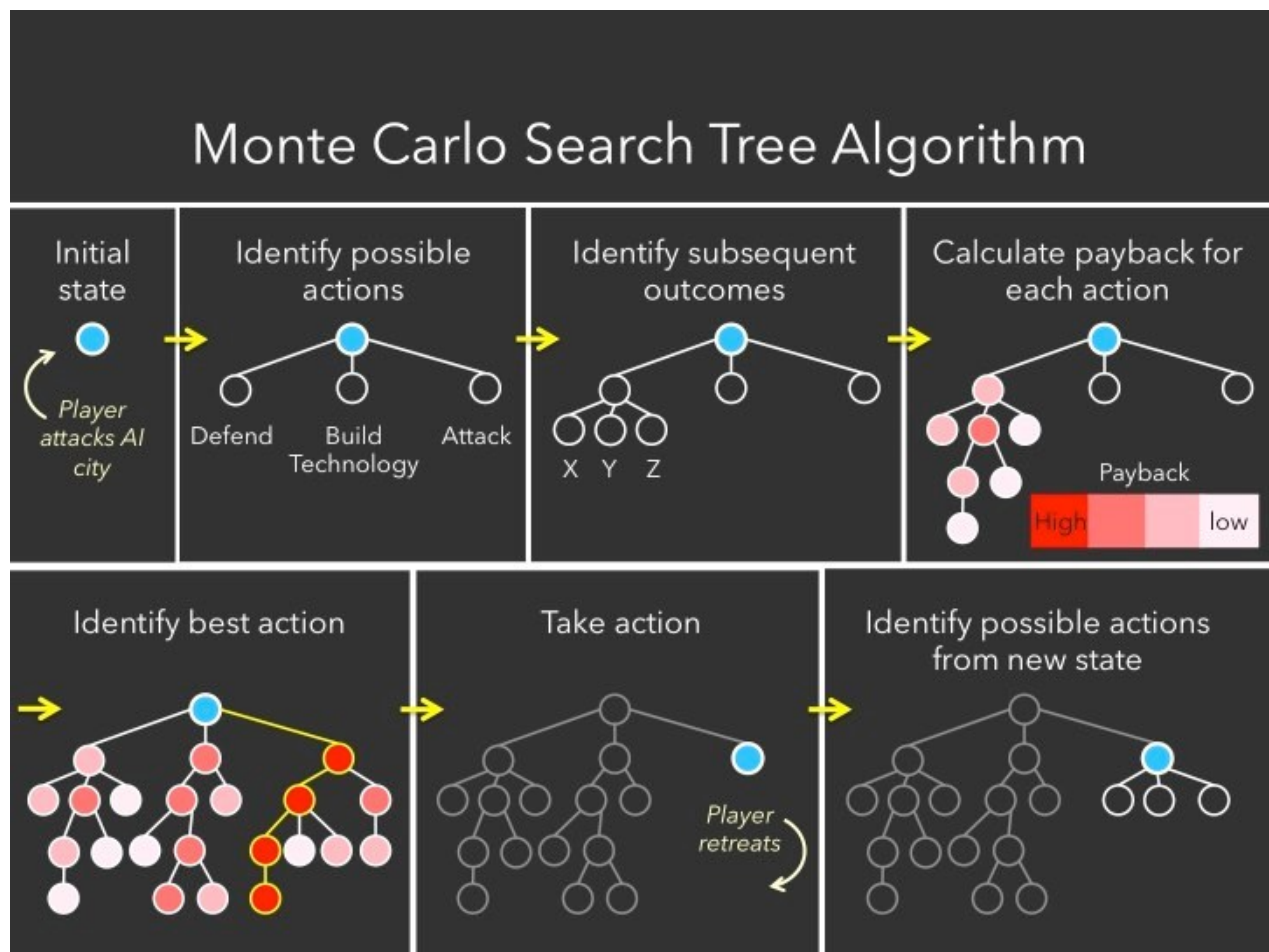


Figure 2. A simplified MCST demonstration.

Learning to become a smarter AI

Although AI designers in the 1990s worked very hard to make NPCs look intelligent, these characters lacked one very important trait: the ability to learn. In most video games, NPCs' behavior patterns are programmed and they are incapable of learning anything from players, e.g. they don't evolve based on human players' input. The reason most NPCs do not exhibit the capability to learn is not only because it is difficult to program machines to learn, but also because most designers prefer to avoid any unexpected NPC behaviors that could impair the experience of a human player.

One of the earliest video game AIs to adopt NPCs with learning capabilities was the digital pet game, [Petz](#). In this game, the player can train a digitized pet just like he or she may train a real dog or cat. Since training style varies between players, their pets' behavior also becomes personalized, resulting in a strong bond between pet and player. However, incorporating learning capability into this game means that game designers lose the ability to completely control the gaming experience, which doesn't make this strategy very popular with designers. Using shooting game as an example again, a human player can deliberately show up at same place over and over, gradually the AI would attack this place without exploring. Then the player can take advantage of AI's memory to avoid encountering or ambush the AI. Such strategy is beyond the designers' control. Until now, virtual pets games still represent the only segment of the gaming sector that consistently employs AIs with the ability to learn.

Playing against or living with an AI?

After the success of AlphaGo, some people raised the question of whether AIs could also beat human players in real-time strategy (RTS) video games such as StarCraft, War Craft, or FIFA. The short answer is yes, they can. In terms of possible moves and number of units to control, RTS games are far more complicated than more straightforward games like Go. In RTS games, an AI has important advantages over human players, such as the ability to multi-task and react with inhuman speed. In fact, in some games, AI designers have had to deliberately reduce an AI's capability to improve the human players' experience.

In the future, AI development in video games will most likely not focus on making more powerful NPCs in order to more efficiently defeat human players. Instead, development will focus on how to generate a better and more unique user experience. As [Virtual Reality](#) (VR, which provides an immersive viewing experience by means of a display) and [Augmented Reality](#) (AR, which combines a human's physical view of the world with virtual elements) technologies continue to expand, the boundary between the virtual and real world is beginning to blur. Last year's [Pokémon Go](#), the most famous AR game, demonstrated the compelling power of combining the real world with the video game world for the first time. In the future, VR- and AR-based open-world video games may provide players with a "real world" experience, perhaps similar to that imagined by the TV series "[Westworld](#)." In this series, the human players can play whatever they want with AI controlled robots and feeling exactly the same with real world. With the increasing capability of natural language processing, one day human players may not be able to tell whether an AI or another human player controls a character in video games as well.

[Andrew Wilson](#), the CEO of Electronic Arts, famously predicted that "Your life will be a video game." As AI-VR/AR technology matures and prompts us to immerse ourselves in an increasingly virtual world, his vision may actually come true after. In that case, do you think you would prefer playing with an AI or a real person? That will become an increasingly pertinent question.

Dr. Harbing Lou has recently graduated with his PhD from the Department of Chemistry and Chemical Biology at Harvard University. He is currently working for T-Mobile in Seattle. You can visit his website at <https://projects.iq.harvard.edu/harbing>.

This article is part of a [Special Edition on Artificial Intelligence](#).

For more information:

Dota2 video game bot: <https://blog.openai.com/dota-2/>

Zuckerberg and Musk on the future of AI: <https://www.theatlantic.com/technology/archive/2017/07/musk-vs-zuck/535077/>