Sanders Lauture
Advanced Writing
Public Document

Context
This is an editorial game news piece aimed towards gamers. Gamers who tend to read gaming news sites tend to be invested and more passionate about games. They however may not be very knowledgeable about the computer science aspects behind games such as AI practices and principles. This article aims to teach gamers more about the AI behind games and where AI in games can move towards in the future.

The Potential for Better AI in Games

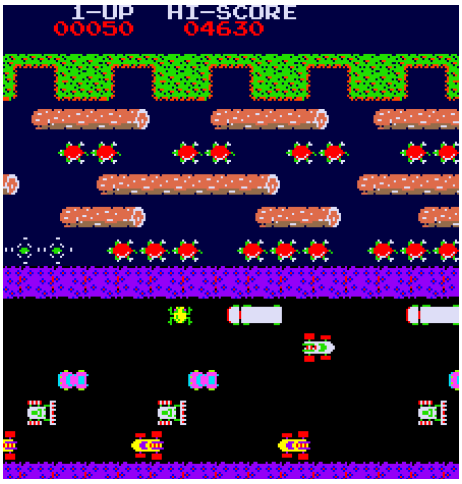

Figure 1: Frogger (https://en.wikipedia.org/wiki/File:Frogger_game_arcade.png)

If you ever have played a game with enemies in it, you have played a game that includes AI. Even Frogger, cars and trucks either move left or right down the screen. It isn't good or particularly complex AI but it's AI none the less. How many games have you played where the AI is too easy? Where the computer is no match for your superior skill. Or have you played a bunch of games where the AI seems to be cheating, gaining extra resources, being able to see through walls or the fog of war, or superhuman reflexes. One simply gets boring, the other can become frustrating to the point where you just give up. It seems like games where the AI is perfectly suited to your skill level are very rare. Fortunately, there is a hot, new, computer science buzzword that potentially will bring better AI to games, it's called machine learning (ML). ML may also bring about Skynet and the end of humanity as we know it but that's a small price to pay for good games.

So how does AI in games even work? Well let's start with a simple game like tic-tac-toe (noughts and crosses for the British folk). Typically, when building AI, a programmer needs to think, "If I was playing this game what would I do?" Well first, a player needs to know the rules, there is a 3 by 3 grid of empty spots. You want to get three of your marks in a row while trying to stop the other player from getting three marks in a row. Ok, simple enough. So, say our AI will always be the second player.

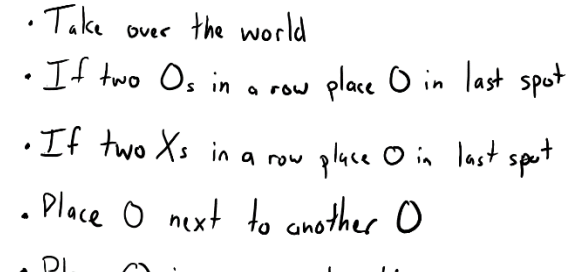

Figure 2: Priority list of AI actions

After the opponent places an X we randomly place an O somewhere. Then they place another X and we see that they have two in a row. So, we place an O to block them. This continues until the game ties. How about instead on our second turn we place an O so that we have two O's in a row and

so that the other player needs to block us, that might be better. That would be a really simple AI. But what if the other player knows great strategies and they beat our AI every time, we could program our AI to take advantage of those strategies but then the game might be too hard for some players. We could then program in a couple difficulty modes.

The flaws of our hand coded AI show up when the rules of the game change. Refer to our priority list above. What happens if the AI tries playing on a 4 by 4 board or a 5 by 5 board. The AI would be trying to take over the world after it saw three O's in a row. For every bigger board, we would need to add more rules about placing an O next to other O's. Machine learning helps us solve this problem.

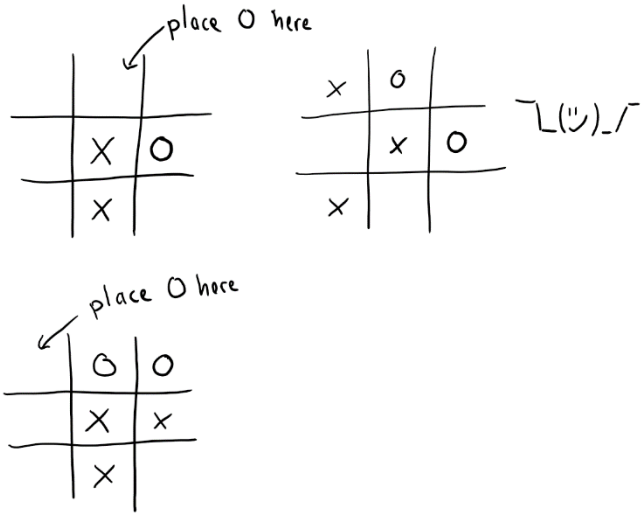

Figure 3: A better AI would be able to avoid the third situation

Machine learning allows computers to learn how to do something without being specially programmed to do that something. Kind of like how humans work. When using ML, computers go through a learning process that involves real input data. For our tic-tac-toe example that means our new AI would need to play against opponents in order to learn how to play. It's similar to learning how to play a game by watching someone play the game. You might be asking, "How will the AI know where to place a mark if we didn't program that?" In our example, placing a mark would be an input. The AI would then receive feedback on what the result of placing that mark was. Either they won (positive), lost (negative), or the game kept going (neutral). After that input the AI would then observe its opponent placing a mark. They would then get feedback again, same as before. This would keep going until somebody won. When the AI gets a positive result they will be more likely to do that action in the future, when the AI gets a negative result they will be less likely to do that action in the future. So, over time by playing opponents the AI would get as good as the opponents it played. To adjust difficulty against weaker opponents the AI would just need to make less optimal moves, moves that have less positive results.

That was a pretty basic and high level overview of how machine learning works but that's all there really is to it. One example you may or may not have seen is Sethbling's video on machine learning being used to play Super Mario World (link). It's pretty similar to the tic-tac-toe example. The AI can press buttons, those buttons cause things to happen in the game, over time the AI gets better when it knows pressing a button at a certain time will cause a very positive result. One thing that I didn't cover but that Sethbling's video goes into is the notion of species and generations. The neural networks in the video "breed" after a period of time in a certain generation to create new species. Random mutations are also introduced so that new behaviors are introduced into the AI.