Soft Computing for Games Part 2

2.2.1 Task 1 - MCTS and Combat AI

Monte Carlo Tree Search

The Monte Carlo Tree Search (MCTS) algorithm is a best-search general approach method used to solve several game problems. One notable use was in Google's AlphaZero, where AlphaZero defeated the world Go champion in 2016 and later in 2017 also defeated the number one Go player. This algorithm was first used by Remi Coulom in his program Crazy Stone, where instead of using the deterministic search algorithms usually utilized for chess-playing programs, MCTS was used [1]. MCTS has become a popular method for intelligent play in games. When compared to other methods of intelligent play, the MCTS is more successful, as other methods fail without an accurate admissible evaluation function [2].

The algorithm has one goal, which is to figure out the most promising move depending on the current state of the game. It has four main steps, which are repeated until time runs out. The first step is a selection strategy, which is used to traverse the tree from the root to a leaf. The second step is a simulation strategy, which is used to finish the game from the leaf node of the search tree. The third step is the expansion strategy, which is used to determine the number of children and which children are stored as "promising leaf nodes" in the search tree. The last step is a back-propagation strategy, where the result of this evaluation is propagated backwards to the root [2].

Combat AI

Artificial Intelligence (AI) has been used in video games since the 1970s, some significant uses of AI include games such as River-Raid, Donkey-Kong, and Boulder-Dash. Combat AI is used in first-person shooters (FFP) to handle tactical reasoning and the required behaviour of the AI agents. In these types of games, path-finding systems are utilized to find the best possible path that the AI agent should take to achieve its goal, while avoiding obstacles in the way. When the AI system decides which behaviour is most appropriate according to the context, it will then need to decide how to best accomplish that task. For example, in a fight, the system will determine whether the AI agent should run toward the player attacking, hide somewhere and wait for the player, or maybe try to sneak up on the player [3]. One example of a game that utilizes combat AI is Killzone. In this game, the AI enemies used the terrain and other objects for cover, while simultaneously attempting to deny the player that same ability [4].

References

[1] M. C. Fu, "Monte Carlo Tree Search: A Tutorial," *Winter Simulation Conference (WSC),* pp. 222-236, 2018.

[2] M. H. M. W. H. J. v. d. H. G. M. J. -B. C. &. J. W. H. M. U. Maarten P. D. Schadd, "Single-Player Monte-Carlo Tree Search," in *Computers and Games*, Berlin, Heidelberg, Springer, 2008, p. 1–12.

[3] J. Grzyb, "Artificial Intelligence in Games," *Software Developer's Journal,* no. June, 2005.

[4] M. Hachman, "Extreme Tech - How AI Works In FPS Games," 6 September 2005. [Online]. Available: https://www.extremetech.com/computing/75786-how-ai-works-in-fps-games/3. [Accessed 10 Janaury 2023].