Interactive visualization system of AI adversarial search algorithms

Ma, Xueran/Sherry

2021

1 Abstract

AI games usually have too many algorithms behind them, and its "thinking" process is complex to get a concrete idea for players. Although there are many useful adversarial search algorithms right now, it is still too abstract to connect those complex algorithms with the games we play together. Also, it is necessary to clarify how the game tree expand while keeping the environment invariant. This work will design an AI game tree visualization system that enables flexible visual adversarial search algorithm exploration by allowing the user to modify AI algorithm parameters, playing against the AI, and seeing the real-time animation. During the playing procedures, for each predicted AI step, the system will visualize the game tree and the animation of its expanding and corresponding AI algorithm's corresponding properties. The system will deploy a simple game engine for a simple game, connect-4, and synchronize another interface for the game tree algorithm interactive animation. With the system, the user can explore to find better properties (for example, iteration) of the AI algorithm and find a better combination of optimization methods, heuristic function, evaluation function, and all other components in adversarial search algorithms. Users could also see the performance comparison between algorithms, such as the graph of their win rate when they fight each other. On the other hand, the system could work as a visualization tool to quickly comprehend complex AI algorithms and be used as a teaching tool. This paper will introduce how to implement the system in detail, including three parts: game engine implementation, game tree algorithms, interactive and real-time interactive animation, and visualization.

2 Related Work

There are many categories and classic algorithms right now in terms of adversarial search algorithms or game tree search algorithms, such as MiniMax, Alpha-Beta pruning, depth-limit search, iterative deepening, and Monte Carlo Tree Search. Although there are many academic types of research and theories on the adversarial algorithms [3] , applications to visualize or animate these theories are still valuable, and their practical significance is overlooked. Many game AI systems mimic dynamic complex worlds and presented the results of the AI got such as scores and winning rating [1] (`https://www.researchgate.net/publication/275272459_Ame_An_Environment_to_Learn_and_Analyze_Adversarial_Search_Algorithms_Using_Stochastic_Card_Games`)and explore a broad spectrum of AI methods [3]. Some focus on implementations of certain adversarial game algorithms [2] [5] or discuss the algorithms in a different game type like in multi-player game [8]. In terms of getting insight into AI algorithms, the book [4] provides comprehensive knowledge, projects, and materials to learn. Besides, Robocode [6] (`https://www.researchgate.net/publication/234800758_Robocode_using_games_to_teach_artificial_intelligence`)and [1] Ame are frameworks that implement battles between robots and provide an event-driven programming environment that could be used to learn AI algorithms.

Figure 1: visualization of decision tree

On the other hand, one paper (`https://www.hindawi.com/journals/ijcgt/2010/578784/`) visualized decision trees in a puzzle game [7], as Figure 1 presents the game – puzzle, the complete breadth-first search tree, and a decision tree recording player's decisions. This paper provides exemplary implementations of tree diagrams in games, but it does not in-

volve any AI algorithms in its game. Other related visualization resources include D3.js, networkx, Graphviz, or others. What to use depends on language and preferences. Some specific tutorials are 1. non-interactive game trees of basic adversarial algorithms (`http://cs.brown.edu/courses/cs141/assignments/adversarial_search.pdf`) 2. Visualizing Game Trees with D3.js (`https://www.davidrobles.net/blog/2015/01/25/visualizing-game-trees-with-`
3. dtreeviz: a python library for decision tree visualization and model interpretation (`https://github.com/parrt/dtreeviz` and `https://www.kdnuggets.com/2020/04/visualizing-decision-trees-python.html`) 4. networkx: a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks (`https://networkx.org/documentation/stable//index.html`) These tutorials are not directly related to my system, but they have some similarities to learn.

However, none of the above combines the game tree's interactive visualization or animation with adversarial search algorithms in games to support AI game adversarial search algorithms' learning process as I proposed in the system. They either only implement the AI in games or visualize the tree structures in another non-related context for other purposes.

References

[1] Leonardo Preuss Angélica Ogasawara João Quadros Eduardo Bezerra Uéverton Souza Eduardo Ogasawara Ana Beatriz Cruz, Sabrina Serique. Amê: An environment to learn and analyze adversarial search algorithms using stochastic card games.

[2] R. Moore D. Knuth. *An Analysis of Alpha-Beta Priming*. 2002.

[3] Mostafa Eldeeb Elnaggar, Gadallah. A comparative study of game tree searching methods. 2014.

[4] Gary Pollice Stanley Selkow George T. Heineman. Algorithms in a nutshell, 2008.

[5] Istvan Szita Guillaume Chaslot, Sander Bakkes and Pieter Spronck. Monte-carlo tree search: A new framework for game ai.

[6] Ken Hartness. Robocode: using games to teach artificial intelligence.

[7] Robert Haworth, Sousan Sheida Tagh Bostani, and Kamran Sedig. Visualizing decision trees in games to support children's analytic reasoning: Any negative effects on gameplay?

[8] U.Lorenz and T.Tscheuschner. Player modeling, search algorithms and strategies in multi-player games, 2006.