CSCI-6660 Intro to Artificial Intelligence

FINAL PROJECT

Tic-Tac-Toe Implementation using MultiAgents

Prof. Shivanjali Khare

Sunethri Seelam
Kamal Siddharth Teki

INRODUCTION

TIC-TAC-TOE is a game for two players who take their turns to mark the spaces in a 3x3 grid representation with X or O. The player who succeeds in marking and placing three of their marks in a horizontal, vertical, or diagonal row will be the winner of the game. This game is all about probability and predictability.

TIC-TAC-TOE GAME IMPLEMENTATION

- In this project, We have to perform with multiple agents like Alpha-Beta pruning, Min-Max algorithm from Adversarial search and Q-learning from Reinforcement Learning. Here every agent as in player attempts to perform an optimal move by checking the number of moves, number of nodes and the time taken to complete the game.

- So, comparison of Min-Max, Alpha-Beta algorithms and reinforcement learning agents in solving the Tic-Tac-Toe game is the primary task for our project.

OBJECTIVES

- The goal of this project is to design an adversarial search algorithm and reinforcement agents to solve the TIC-TAC-TOE game.
- Design of 3 Artificial Intelligence Agents
  - Q learning Technique
  - Min-Max Algorithm
  - Alpha-Beta Pruning
- Playing Tic Tac Toe game multiple times among them to find the efficient Algorithm via testing.
- The player who gets the three consecutive symbols in a row or column or diagonally gains a winning point in this game which is the primary goal of this game.
- Calculating the corresponding performance metrics for these three algorithms and observing which performs better, adversarial search or reinforcement learning.

APPROACH

- The first approach is Q-Learning from Reinforcement Learning.
- The second approach we have planned to use is from Adversarial Search.
  - Min-Max Algorithm
  - Alpha-Beta Pruning
- Technology Stack
  - Python 3

MIN-MAX ALGORITHM

- It is a recursive or backtracking algorithm that will participate in the decision-making and game theory.

- It gives the player the best possible move by presuming the opponent is likewise playing as good as the player.

- Recursion is used to search throughout the Game Tree in this game.

- This algorithm is commonly implemented in games like Chess, Tic-Tac-Toe and similar games where only two players are involved.

- To explore the entire game, our algorithm initially uses a Depth-First Search where it goes all the way down to the terminal node and then recurses back to the top.

ALPHA-BETA PRUNING

- Alpha-beta pruning is a search technique that always attempts to decrease the number of nodes in it’s search tree that are evaluated and given by the Min-Max algorithm.

- It is a two-player game with an adversarial search method. So, when applied to a min max tree, it always produces the same moves as in the Min-Max, but prunes out those branches that have no nodes on to the final decision.

Q-LEARNING | REINFORCEMENT LEARNING

- A popular reinforcement learning technique known as Q-Learning, which is based on the Bellman Equation, asks the agent to learn the best behaviors to adopt in order to maximize rewards. The best moves are those that have been refined throughout time.

- The agent's goal is to maximize the value of "Q," which is always used to denote the caliber of actions executed at each state.

- It is a machine learning technique that allows the agent to learn by trial and error in an interactive environment, using input from its own actions and experiences.

DELIVERABLES

- User documentation model which gives details about the Tic Tac Toe game implementation using Min-Max, Alpha-Beta and Reinforcement Learning Agents are used.
- Algorithms developed for AI agents using Python Programming Language.
- Github repository link for Python code and related files.
- Youtube video demonstrating project implementation and slides if needed with detailed information.

EVALUATION METHODOLOGY

- Project is evaluated based on the implementation of all the AI agents.
- Calculating the number of wins of the corresponding agent through which it plays multiple times one-one to find the efficient agent.
- A comparison table given for the three agents based on the Tic Tac Toe moves, corresponding score and also the win rate has been provided.
- Graphs depicting the performance comparisons are represented.
- Time and space complexity comparison for these three agents are implemented.

THANK YOU