Tic Tac Toe

MATLAB Challenge - Summer Term 2023/24

April 10, 2024

1 Introduction

Welcome to the Tic Tac Toe Algorithm Challenge! This assignment is designed to test your skills in programming, algorithms, and strategy development using MATLAB . Tic Tac Toe, also known as Noughts and Crosses or Xs and Os, is a classic game that has been played for thousands of years. Did you know that the earliest version of Tic Tac Toe was played in the Roman Empire around the first century BC? It was called Terni Lapilli and was played on grid surfaces without the need for any equipment. Fast forward to today, and we're bringing this ancient game into the realm of modern computing for a fun and educational challenge.

In this assignment, you will develop an algorithm to play Tic Tac Toe on a 30×30 board. Your algorithm will compete against those of your classmates in a tournament to determine the ultimate Tic Tac Toe champion.

2 Assignment

Your task is to implement a MATLAB function that decides the next move in a game of Tic Tac Toe. The game board is a 30×30 matrix, where each cell can be:

0 , indicating the cell is empty,

1 , indicating the cell is occupied by player 1's marker (X),

2 , indicating the cell is occupied by player 2's marker (O).

2.1 Function Signature

Your function will have the following signature:

```
function [row, col] = playerMove(player, board, minToWin)
```

Input Parameters:

1. `player`: An integer (1 or 2) representing the current player.
2. `board`: An $N \times M$ matrix representing the current state of the game board. (In the competition 30×30)
3. `minToWin`: A minimum number of consecutive elements to achieve victory. (In the competition 5).

Output:

1. `row`, `col`: The coordinates of the move your algorithm chooses. If no move is possible, return -1 for both row and col.

3 Testing your solution

Together with the assignment description you will receive MATLAB Tic Tac Toe API, which you are encouraged to use for testing your algorithms:

```
function victoriousPlayer = ticTacToe(player1FHandle, player2FHandle)
```

Input Parameters:

1. `player1FHandle`, `player2FHandle`: function handles for players 1 and 2. The order does not matter. The starting player is selected randomly.

Out Parameters:

1. `victoriousPlayer`: integer indicating the victorious player (1 or 2), and 0 for a draw.

The function already displays the game board and animates the game procedure. In development, you can reduce the time between animations by setting parameter `timeout` or just disable the animations altogether.

To test your algorithms use function `randomMoves`, which simply makes random moves, or `simpleAtomStrategy`, which implements a very primitive strategy.

4 Requirements

- Strategy Implementation: Implement a strategy that goes beyond random moves. Your code must dynamically adapt to the board state. However, global variables are not allowed, nor are files saved or loaded except external data for the strategy. So, the strategy decision always has to depend only on the current state of the board, omitting the past sequence of players' moves.
- Competition-Ready: Ensure your algorithm runs efficiently. It will be pitted against your classmates' algorithms in a tournament setting to determine the most effective Tic Tac Toe player.
- Move Evaluation: The computation of the decision for the move can not take longer than approx. 5 s.

5 Evaluation Criteria

The winner will be selected on a tournament basis where all players will play against each other. Winning a match award you with three points, drawing with one point, and losing results in zero points. To eliminate randomness from the process, each match will consist of several rounds and players will switch the role of the starting one.

- Correctness: Your algorithm should make valid moves within the rules of Tic Tac Toe.
- Strategy: Algorithms that demonstrate a deeper understanding of the game and employ a more sophisticated strategy will score higher.
- Efficiency: Your code should be optimized for performance, especially for larger board sizes.
- Creativity: Creative approaches to problem-solving and strategy development are highly encouraged.
- Interactivity (optional hint): Create an interface for a single-player game version to allow you to beat your own algorithm.

6 Submission Guidelines

Submit your MATLAB function file (playerMove.m) by **18.5.2024**. Ensure your code is well-commented to explain your strategy and any key decisions. Attach a brief report (presentation) explaining your algorithm, its complexity, and any unique strategies or techniques you employed.

7 Disclaimer

Small changes both in the challenge assignments and in the organization of the contest are reserved.

8 Rewards

Participation in the competition guarantees you exclusive MATLAB merchandise as a reward, as depicted in Figure 1.

Additionally, the top three solutions will receive Humusoft vouchers for MATLAB courses. These courses cover advanced features of the MATLAB language, including Simulink, parallel computations, and embedded coding.

9 Fun Fact

Tic Tac Toe is solved completely (for boards 3×3), meaning we know the best possible moves from any position. However, the challenge and fun come from designing an algorithm that can discover and execute these moves on its own, especially on unconventional board sizes. Can your algorithm outsmart the competition?

Best of luck, and may the best algorithm win!

Figure 1: Demonstration of MATLAB merchandise that you can win. Received rewards can differ.