Xtreme TicTacToe Tournament

Artificial Intelligence

February 14, 2019

1 Introduction

- The Extreme TicTacToe is an extension of 3x3 Ultimate TicTacToe which in turn is an extension of standard 3x3 TicTacToe. Extreme TicTacToe comprises of 2 boards of 3x3 board in which each cell further is a 3x3 board.
- The game is between two teams.
- Coin is flipped to choose who will move first.
- Player 1 marks ‘x’ and Player 2 marks ‘o’ on the board.
- The player who makes a legitimate pattern wins the whole board.

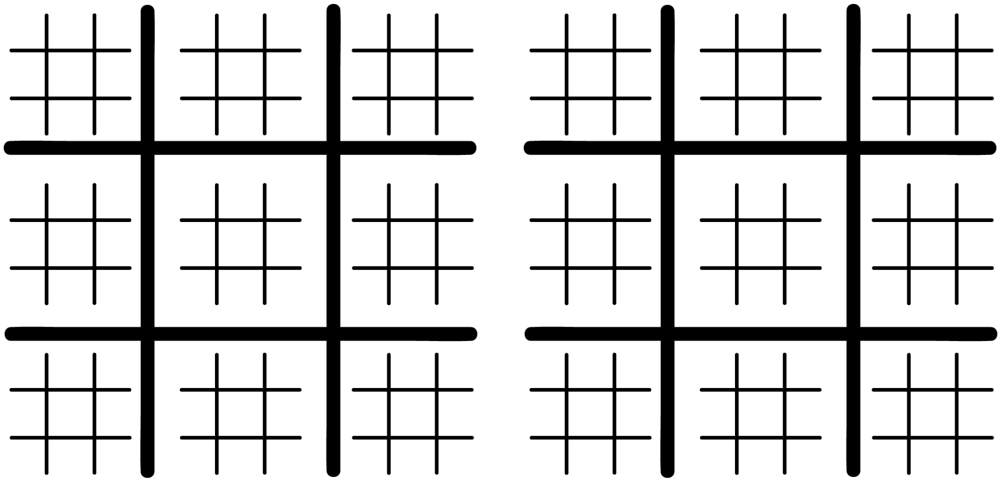

Figure 1: Extreme TicTacToe Board

- The board structure is as shown above. We will refer to the whole board as BigBoard and each small board as SmallBoard from now on. Hence, each cell of BigBoard is a SmallBoard and SmallBoard itself has 9 cells.
- As we have 2 boards we will call BigBoard1 and BigBoard2 respectively. And their small boards will be called SmallBoard1 and SmallBoard2 respectively.

- Note: This version of Ultimate TicTacToe is an extension of the original version of and is different from it.
(https://mathwithbaddrawings.com/ultimate-tic-tac-toe-original-post/)

2 Rules

- **FIRST MOVE:** The first player takes the first move and it is a free move. The player is free to move anywhere i.e. in any cell of any SmallBoard1 or SmallBoard2.
- **CORRESPONDENCE RULE:** When a player places his marker in any of the cell, the next player can only place marker in a **SmallBoard1 or SmallBoard2** corresponding to that cell. For Example: If a player places his marker in Top Right Corner cell of a SmallBoard1 then the next player can only place his marker in available cells of the Top Right Corner SmallBoard1 or Top Right Corner SmallBoard2.
- **ABANDON RULE:** Once a SmallBoard is won by some player, that SmallBoard is abandoned and it has to be considered full, i.e. no more markers can be placed in that SmallBoard.
- **OPEN MOVE:** In case all the cells of both the destined SmallBoards from “Correspondence Rule” are occupied or if both the destined SmallBoards are abandoned according to “Abandon Rule”, then the move is considered to be an open move, i.e. the player can move anywhere, on any available cell of any of the SmallBoards, given that SmallBoard is not abandoned.
- **BONUS MOVE:** If a player places his marker in a cell of a SmallBoard and this leads to the player winning that SmallBoard, then the he/she gets a bonus move, i.e., he/she gets to move again. This bonus move is limited to 2 SmallBoard wins, i.e., If you win another SmallBoard by placing the marker again in your bonus move, you don’t get another bonus move.
- **WIN RULE:** The player who places their markers on 3 cells of a SmallBoard such that they form a vertical or horizontal line or 3 cells such that they form a diagonal **wins that SmallBoard**. Similarly, player who wins 3 SmallBoards such that they form a vertical or horizontal line or win 3 SmallBoards such that they form a diagonal **wins the whole game**. The first player to make the winning pattern on any one of the BigBoards wins the game.

3 Winning Combinations

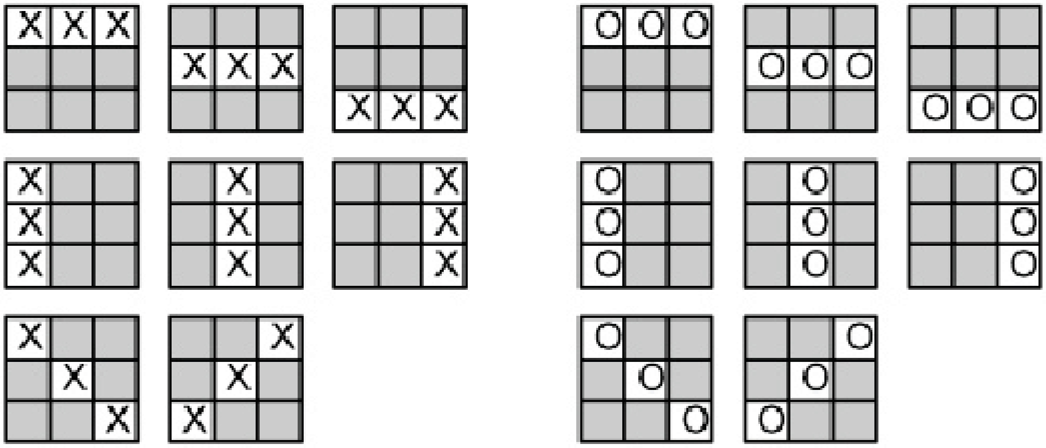

Figure 2: Winning combinations for both players in a SmallBoard.

4 Coding Aspects

1. **Language and Statistics**

   - Code needs to be written in Python2.7.
   - Use of numpy, pandas and sklearn is permitted. However, use of any other external library must be confirmed with the TAs.
   - You need to adhere to the naming conventions (specified in the Deadline section) of your Python class, method and file **strictly** in order for your code to be evaluated in the tournament and receive any points.
   - Your submission should just contain a python class (as per the naming convention), which should have an **init** and a **move** function with the same signature as given in the evaluator code. You are free to create more functions as per your requirements.
   - We will just be importing your class from your code and will be calling its move function as given in the code. So, please do not define anything outside the class.
   - You can use any of the functions directly available in the evaluator code. But you must not have any of the evaluator code in your submission to make sure you do not get flagged for plagiarism.

2. **Time Limit**

   - You need to return a valid move from your move function within **24 seconds**. If the time limit exceeds 24 seconds for a particular move, then the match will be forfeited and the opponent wins the whole match by default.
   - Use of signal library for time tracking is **prohibited**.

5 Scoring

- Winning a game, by forming a pattern as described in WIN RULE will give you **86 points** and the opponent **0 points.**
- If the player makes an invalid move, or exceeds time limit, or makes an illegal change on the board, or uses threading or makes any system call, then the opponent earns **86 points** and the player earns **0 points**.
- If no player has a pattern at the end of the game, both the players get points according to the SmallBoards they have won respectively. **Each small board has different scores** as mentioned in section below. Smallboards which are drawn do not count as points for either player.

5.1 Scoring of SmallBoards

- Corners: 4 points
- Center: 3 points
- Remaining SmallBoards: 6 points

6 Evalution

The assignment will be evaluated out of **100 points** which will map to around 12-15% of your final course grade. The evaluation will be divided into 2 parts:

6.1 The Tournament (50 points)

1. **Group stage:** The teams will be divided into **10 pools** at random. Each pool will play a league in a round robin fashion and the **top 3** bots will move to the next round.
2. **Semi-final stage:** The teams will be seeded and divided into **3 pools of 10 bots each**. Each pool will play a league in a round robin fashion and the **top 3** bots will move to the next round.
3. **Final Stage:** The **top 9** teams will play a league in round robin fashion and the **top 3** teams will be felicitated.

Note that there will be 2 games between every bot in a pool, and both the bots will get a chance to make the first move.

Points will be awarded out of 50 in the following manner:

1. **30 points** for group stage, based on relative score of top team in pool.
2. **10 points** for semi final stage, if qualified, based on relative score of top team in pool.
3. **10 points** for final stage, if qualified, based on relative score of top team.

6.2 Manual Evaluation (50 points)

We will consider the following metrics for manual evaluation:

1. Heuristic [10 points]

2. Search strategy (MiniMax, ExpectiMax, Monte Carlo, Learning etc.) [20 points]. **You are expected to at least implement some variant of MiniMax search along with Alpha-Beta pruning.** Any other advanced search technique is encouraged.

3. Performance analysis of your bot in the tournament [10 points]. We will be providing the logs of all the games of the tournament. You are expected to have a look at these logs and analyse your bot's performance in the tournament.

4. Viva based on the concepts used in the assignment [10 points].

7 Deadline

The deadline for this assignment is **February 25th, 11:55pm**. Please note that this is a hard deadline, and there will be no extensions.

Each team needs to submit only one .py file with a given class name and filename. The classname should be TeamX (eg. Team39) and filename should be teamX.py (eg. team39.py) where X is the number of that team. The .py file should **ONLY** have original code (not the shared code).

Your class should implement the **'move'** function as described in the code. All submissions need to be made on moodle.

8 Code of Conduct

1. Teams are free, and encouraged, to discuss possible strategies and play games against each other to improve their original bots.

2. Sharing of code from any resource is strictly prohibited. Plagiarism will be checked. Any flagged cases will be reported and awarded zero and further actions will be taken as appropriate.

In case of any queries, please contact the TAs for this assignment (Aanshul Sadaria, Gunjan Karamchandani, Tirth Maniar).