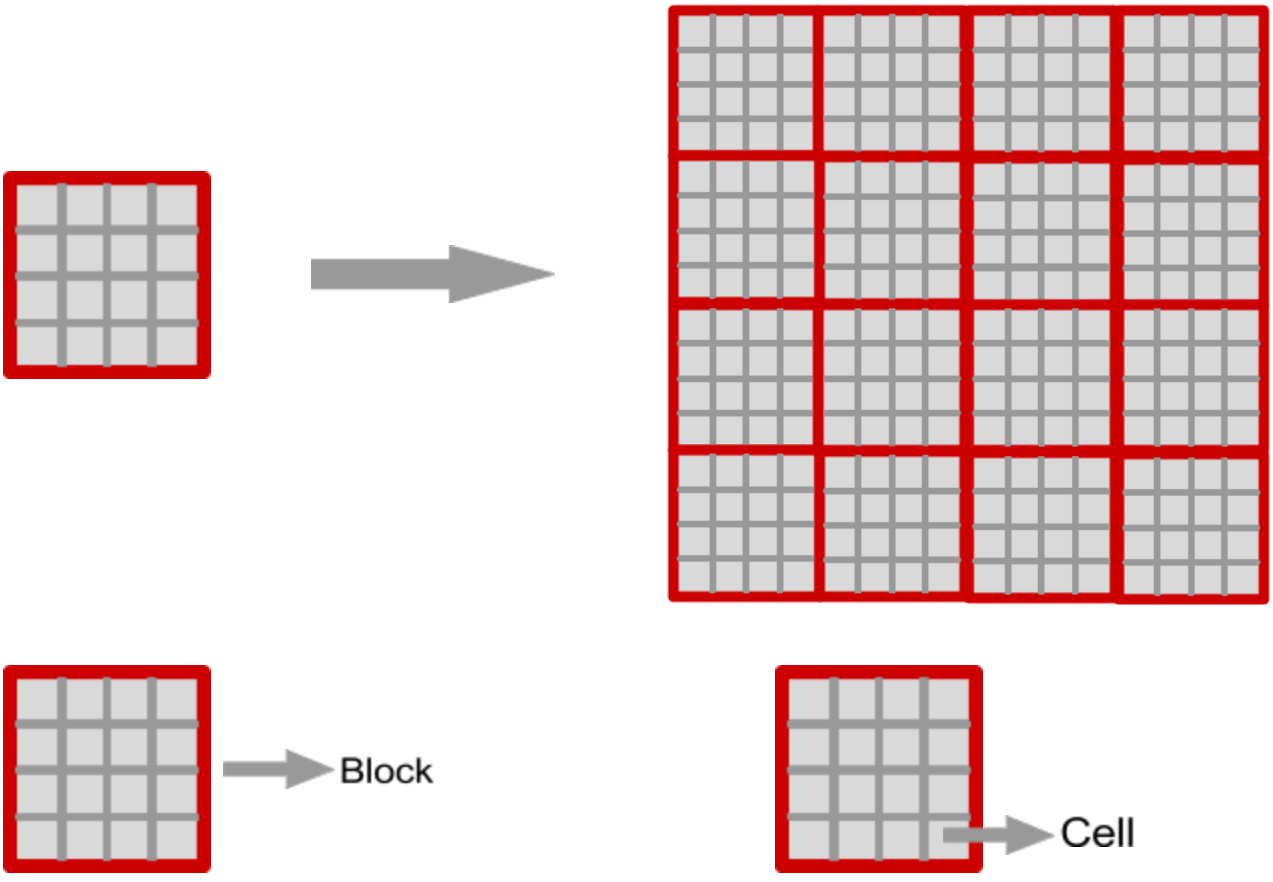
Artificial Intelligence Mini Project-1

Ultimate TicTacToe Tournament

Objective

To implement an 'Ultimate TicTacToe' game playing agent for a 4x4x4x4 TicTacToe Game. This project can be done in teams, comprising of at most 2 students each. The teams need to register here by [**February 4th, 11:59pm**]: (https://docs.google.com/forms/d/e/1FAIpQLSf_dyC4XLAmfLDKgwrw6q5lq8IyjqOnQbtucIfebbomMOUqqg/viewform?c=0&w=1)

The Game



Artificial Intelligence Mini Project-1

- Ultimate TicTacToe is an extension of the 4x4 TicTacToe(which is an extension of 3x3 TicTacToe), where there are 16 blocks each having 4x4 cells.
- Each game is between two teams.
- At the beginning, a coin is flipped to decide the team which will move first (First player).
- The marker for the first player is ‘x’ and for the second player is ‘o’
- The objective of the game is to win the board by making a legitimate pattern of the blocks.
- 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/)

The Rules:

1. [**FIRST MOVE**] The very first move of the game is an open move, i.e. Any cell on the entire board is valid.

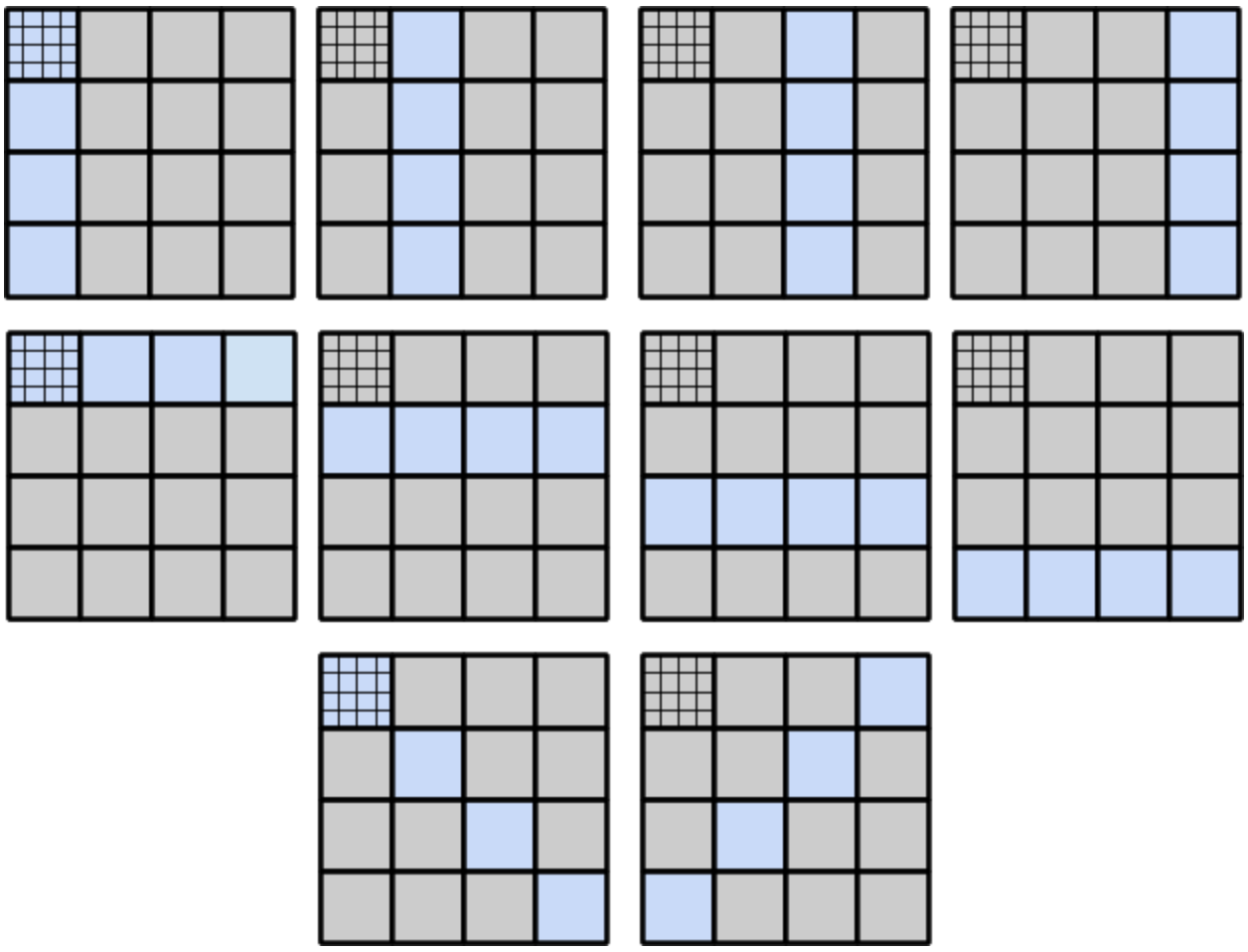
2. [**CORRESPONDENCE RULE**] If the opponent places his/her marker in any of the cells, then you need to place your marker anywhere in the block corresponding to the cell. For example, if a player places his marker in the top left cell of some block, the next player needs to move in any of the open cells in the top left block. Similarly for the right center cell, right center block is open. Please refer to the code for more clarity.

3. [**ABANDON RULE**] Once a block is won by a player, it has to be abandoned. That is, you may consider the entire block to be full and no other player may play in that block.

4. [**FREE MOVE RULE**] In case all of the cells in the destined block obtained from Rule 2 are occupied or the destined block is abandoned as per Rule 3, then the player may move in any free cell in the entire board.

5. [**WIN RULE**] The player who wins any four consecutive cells in a block which are either a row, column or diagonal wins the block and the player who wins any four consecutive blocks of the board, wins the game and the game is over. If all the cells are filled, and no pattern has been formed then the game is over.

The Winning patterns



Coding Aspects

1. [LANGUAGE AND SYNTAX CONSTRAINTS]
   a. The code needs to be written in Python.
   b. The naming of your Python class, method and file will be given once the teams are formed. You need to adhere to these conventions strictly in order for your code to be evaluated in the tournament and receive any points.

c. 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.
d. We will just be importing your class from your code and will be calling its move function as given in the code. So, please don't define anything outside the class.
e. 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]
   a. You need to return a valid move from your move function within 15 seconds. If the time exceeds 15 seconds for a particular move, then the match will be forfeited and the opponent will win by default.

Scoring

1. Winning a game, by forming a pattern as described in [WIN RULE] will give you 16 points.
2. If the player makes an invalid move, or exceeds time limit, or makes an illegal change to the board, or uses threading or makes any system call, then the opponent earns 16 points and the player earns 0 points.
3. If no player has a pattern at the end of the game, both the players get points equal to the number of blocks they have won respectively.

Evaluation

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:

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 three 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.

Manual Evaluation [50 points]:

We will consider the following metrics for manual evaluation:

1. Heuristics [10 points]
2. Search strategy (MinMax, ExpectiMax, Monte Carlo, Learning etc.) [20 points].
   You are expected to at least implement some variant of MinMax search. 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].

For the project we expect some form of mini-max in some phases of the game. A basic working mini-max will fetch you 25 marks out of 50 (in the manual evaluation).

Deadline

The deadline for this assignment is **[24th February, 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 classname and filename. The naming convention will be shared after the teams are formed. 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.

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(Apaar Garg, Rishith Reddy and Shruti Gupta).