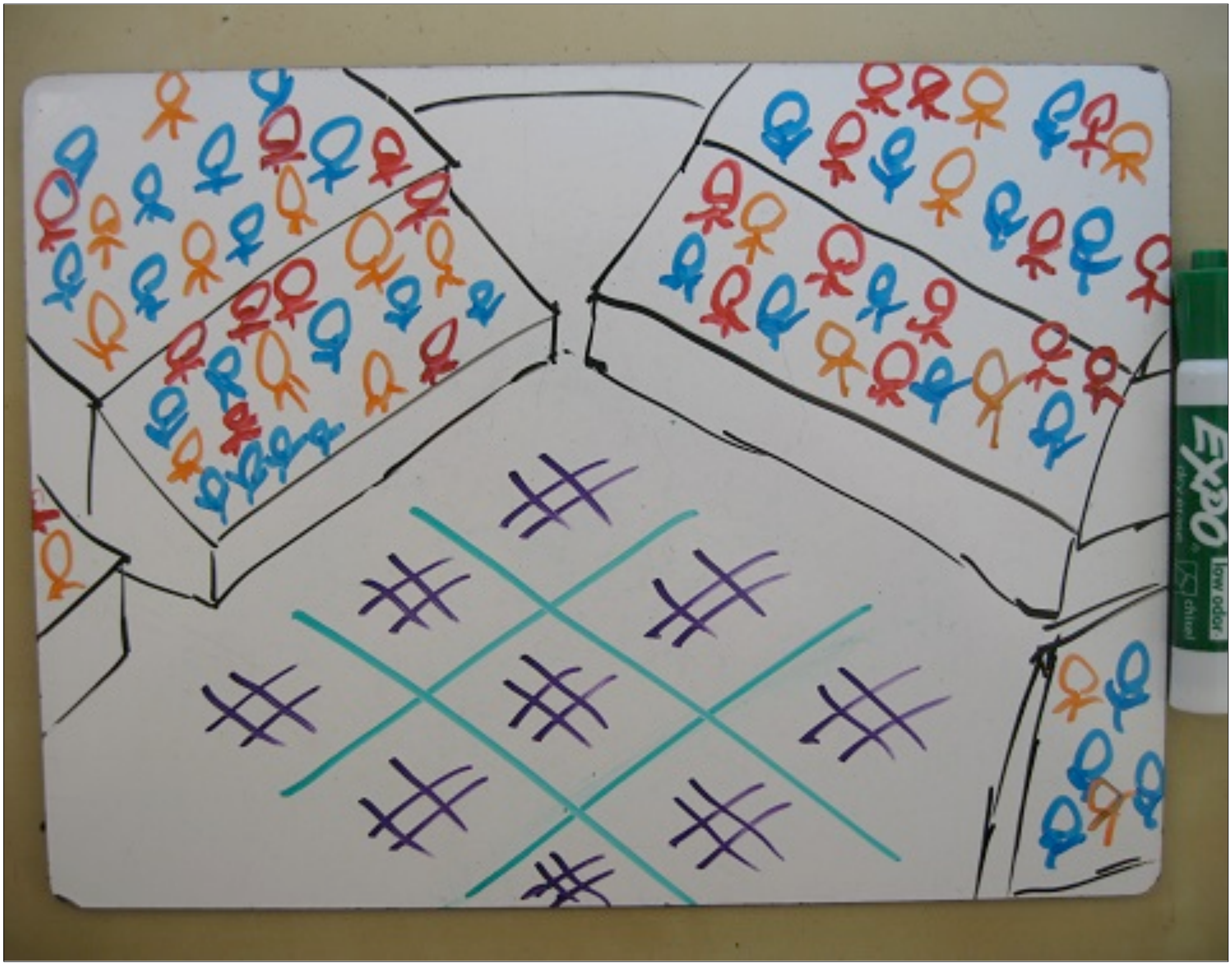
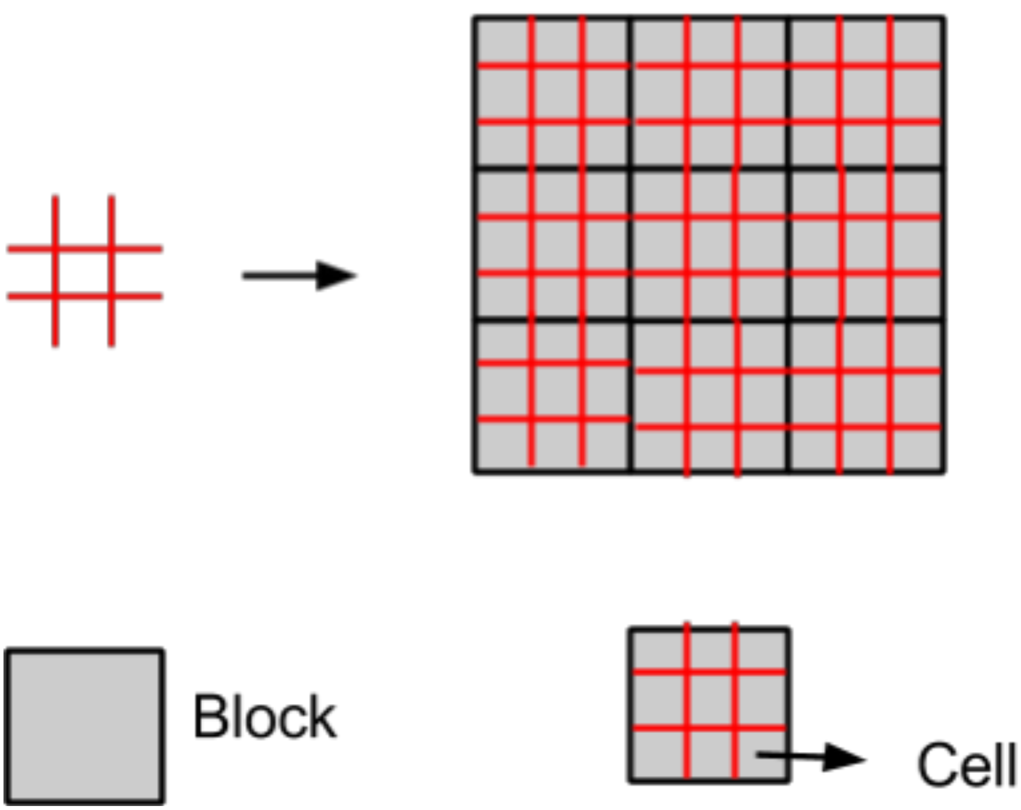
The Ultimate TicTacToe Tournament



Objective

To implement a 'Ultimate TicTacToe' game playing agent. The project has to be done in teams, comprising of at most 2 students each. **The teams need to register <u>here</u> by January 25th, 11:59pm.**

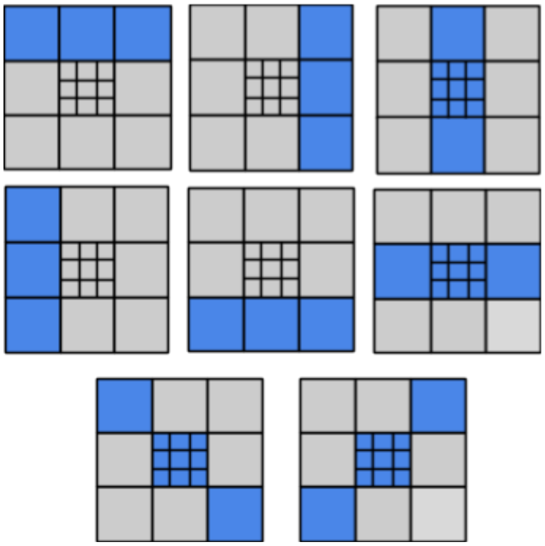
The Game:

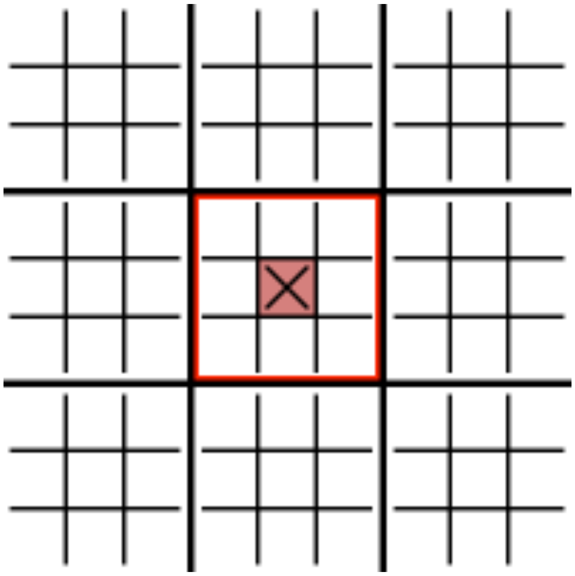


- Ultimate TicTacToe is an extension of the 3x3 TicTacToe, where there are 9 blocks each having 3x3 cells.
- Each game is between two teams.
- At the beginning, a coin is flipped to decide the team which will move first (First player).
- The marker for the first player is 'x' and for the second player is 'o'
- The objective of the game is to win the board by making a legitimate pattern of the blocks.

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, except for the center cell of a block, then you need to place your marker anywhere in the two blocks adjacent to the block corresponding to the cell. For example, for the top left cell, the next player needs to move in center left and top center. Similarly for the right center cell, top right and bottom right blocks are open. Please refer to the code for more clarity.
3. [CENTER RULE] If the opponent places his/her marker in the center cell of any block, then you need to place your marker in the center block only. Please refer to the below figure.
4. [FREE MOVE RULE] In case the all of the cells in the destined blocks obtained from Rule 2 or Rule 3 are occupied, then the player may move in any free cell in the entire board.
5. [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. If the
6. [WIN RULE] The player who wins any three blocks which are either a row, column or diagonal 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.





Coding Aspects

A. [LANGUAGE AND SYNTAX CONSTRAINTS]

1. The code needs to be written in Python.
2. 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.
3. You may NOT use any of the code given in the evaluator. You must not have any of the evaluator code in your submission to make sure you do not get flagged for plagiarism.

B. [TIME LIMIT]

1. You need to return a valid move from your move function within 12 seconds. If the time exceeds 12 seconds for a particular move, then the match will be forfeited and the opponent will win by default.

C. [SCORING]

1. Winning a game, by forming a pattern as described in [WIN RULE] will give you 3 points.
2. If no player has a pattern at the end of the game, the player who has won more number of blocks will earn 2 points and the opponent will earn 1 point.
3. If both the players have won the same number of blocks, the player who has their marker in more number of center cells will gain 2 points, and the opponent 1 point.
4. If the player makes an invalid move, or exceeds time limit, then the opponent earns 2 points and the player earns 0 points.
5. If any player makes an illegal change to the board, uses threading or makes any system call, they will be awarded 0 points and the opponent 3 points.
6. If none of the above holds and the game ends, both players will be awarded 1 point each.

Evaluation

The assignment will be evaluated out of 100 points which will constitute of *upto 15%* of your final course grade. The evaluation will be divided into 2 parts:

The Tournament [60 points]

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.
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.

Final Stage: The top 9 teams will play a league in round robin fashion and the top three teams will be felicitated.

Points will be awarded out of 60 in the following manner:

1. Out of 40 points for group stage, based on relative score of top team in pool.
2. Out of 10 points for semi final stage, if qualified, based on relative score of top team in pool.
3. Out of 10 points for final stage, if qualified, based on relative score of top team

Manual Evaluation [40 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. Any form of pruning/optimization [5 points]
4. Adapting strategy to best perform with the given [CORRESPONDENCE RULE] and [CENTER RULE] [5 points]

Deadline & Submission format

There will be two deadlines for the project:

1. Soft Deadline [**15th February, 11:59 pm**]: Each team can submit their working code. We will test the code against a random bot, a few intelligent bots and provide the feedback to each team. The aim of this deadline is to help teams confirm that their code is valid, efficient and understand how it stacks up against benchmark. This submission is optional and will NOT be evaluated, but is highly recommended.

2. Hard Deadline [**28th February, 11:59 pm**]: This is the final deadline. No further submissions will be considered.

Each team needs to submit only one .py file with a given classname and filename., The .py file should ONLY have original code (not the shared code) and the trace of your game against the random bot commented below the code. The naming convention will be shared after the teams are formed. Your class should implement the ‘move’ function with the signature as described in the code. All submissions need to be made on moodle.iiit.ac.in

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 awarded zero and further actions will be taken as appropriate.