

# Assignment-2

## Adversarial Search Game Playing Agents (17 Marks)

Implement a generic game playing agent in python. The agent should make the decision of the next move using the different variants of minmax algorithm listed below:

- (i) Basic Minimax with no augmentation (5 Marks)
- (ii) Minmax with alpha beta pruning (3 Marks)
- (iii) Minmax with a depth limit (Which requires you to come up with a heuristic evaluation function for the different games below) (3 Marks)
- (iv) Minmax with both depth limit and Alpha- beta pruning (3 Marks)
- (v) Experimental Minmax variant. You can use all the improvements listed above and in addition you can try the following (3 Marks)
  - a. You can improve the minmax by experimenting with better heuristic eval functions
  - b. Use your creativity to see if you can do more better
  - c. You should prove that your technique works better by comparing it with the remaining four, over a reasonable number of randomly generated game instances.

Except for the heuristic-eval function the remaining code should be common for all the games listed below. Hence, the term '**Generic**' game agent. We should be able to configure the variant of minmax used via a simple flag.

## The Game Playing Experience (3 Marks)

There are two ways to do the assignment.

- (i) You can do it as a simple python code in a ipython notebook. The current game state is stored in a global. You call *nextmove()* function, which makes the agent to make the next move and the result is printed with simple-text/basic visualization. (0 points)
- (ii) You can write a full fledge game using pygame (or) your favorite game library. The link: <https://realpython.com/pygame-a-primer/> shows a simple tutorial for pygame. The board games listed below are even more simpler to implement. (3 points)

Students who chose to give us a full game experience (where we can choose the algorithm/strategy used by the game-agent using a dropdown) will be rewarded this 3 points.

## The Games

1. Tic Tac Toe
2. Three men's Morris game
  - a. <http://www.cynningstan.com/game/106/three-mens-morris>
  - b. <https://www.youtube.com/watch?v=sCZFjiQCOWw>
3. Open field tic tac toe (tic tac toe played on a larger board)
  - a. You can configure the board size and the number of connecting pieces required for a win
  - b. <https://cims.nyu.edu/drecco2016/games/OpenFieldTicTacToe/winter.html>
4. Nine men's Morris
  - a. [https://en.wikipedia.org/wiki/Nine\\_men%27s\\_morris](https://en.wikipedia.org/wiki/Nine_men%27s_morris)
  - b. <https://www.youtube.com/watch?v=zvBIKOHlkRE>

## What to Submit

A IPython notebook with clear **markup** explaining the code. (Or) a zip file containing the following.

1. Game files and sprites, etc.
2. A readme which explains how to start (if necessary, install) the games
3. A 2 min video recording of playing the games (Its ok if this is a YouTube link to your channel)