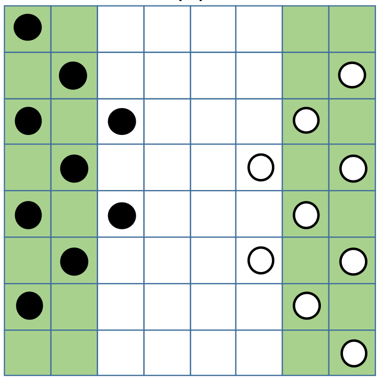
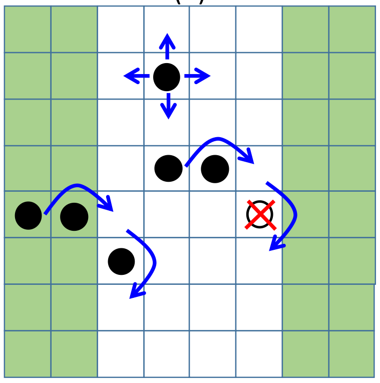
Introduction to AI **Spring 2019** **Group Game Project** **Due 5/21/2019**

The objective of this assignment is for you to design a game-playing agent. The game to be played is a variant of checker on an 8x8 square board. The rules are:

(a)



(b)



- Each player has 9 pieces initially, arranged as in image (a) to the right.
- ***Black*** moves first.
- The valid moves (see image (b) to the right) are:
  - Move one piece to an adjacent unoccupied square.
  - Have one piece hop over another piece of either player to an unoccupied square. Multiple hops by one piece can be taken within one move as long as all the hops are valid and form a sequence. However, no loop is allowed; you cannot hop to the same square twice within one move.
- When hopping over an opponent's piece, the opponent's piece is considered "captured" and removed from the board.
- The overall objective is to move as many of a player's pieces as possible to the shaded squares in the opposite side (the target region).
- A player's move is skipped if he/she makes an invalid move or has no piece remaining on the board. The other player will continue to play.
- The game ends when
  - Either player has all his/her pieces on the board in the target region, or
  - A maximum of 200 moves per player has been played.
- At the end, the score of a player is the number of pieces placed in the target region. The player with the higher score wins the game.

In the tournament, each pair of players will play two games, with each player being ***Black*** once. The "tournament points" are awarded as: two points for the winner, zero point for the loser, and one point each if the game ends with a draw. The overall ranking is based on the total tournament points of the players. The tie-breakers, in case multiple players have the same tournament points, are

(1) Number of games actually won.

(2) Total score differential (a player's total scores minus the opponents' total scores).

Your program should run as a stand-alone program, not as a function or a library. For simplicity, the communication will be through files. Once your program is started, it should go into a loop that checks a text file for the current game state, decides on the move, and writes the move to a text file. The files will also contain the index of the move, starting from one. The allowed programming languages / environments will be posted by the TAs, as well as the information on the input/output file formats. A Team ID will always be included in the filenames of files accessed by that team. The Team IDs will be assigned after the groups are formed.

The submission is to be through E3. <u>No late submission is accepted for this project</u>. You should submit your program source code (all in a single text file if possible) and a report file. The report (maximum 5 pages single-spaced) should describe how your game AI works, and your reasons.