CHINESE CHECKERS

THE MAKERS:

Kandarp S. Khandwala (110050005)
Rohan Prinja (110050011)
Pratyush Nalam (110050045)

THE PROBLEM STATEMENT:

Chinese Checkers is the game to be implemented. In our project, two, four or six players can play with each other, with at most two players being handled by the AI.

THE GAME:

It is a turn-based game played on a “star"-shaped board and can be played by two, three, four or six people, playing individually or in teams. The board is a large set of slots or spaces in which the “pieces" can fit. Each player starts with ten pieces of his chosen colour and places it at a corner of the star.

The objective of the game is to be the first to race one's pieces into their “home" – the corner of the star opposite the starting corner. The game involves strategizing one's moves and each game is different and deterministically unpredictable.

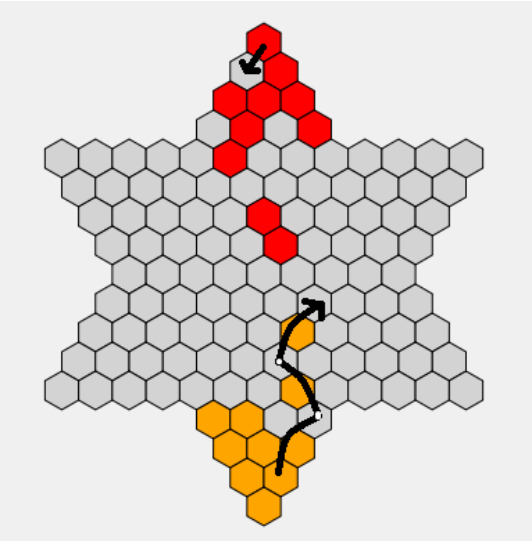
Players take turns moving a single piece, either by moving one step to an adjacent unoccupied space in any direction, or by jumping in one or any number of available consecutive hops over other single (friendly or enemy) pieces. Hopping cannot be combined with single step moves – a move is either of the two. Moreover, it is not mandatory to make the maximum number of hops possible. There is no capturing involved. The left board show a red piece exhibiting a “slide" and the orange piece making a long “hop".

THE ALGORITHMS:

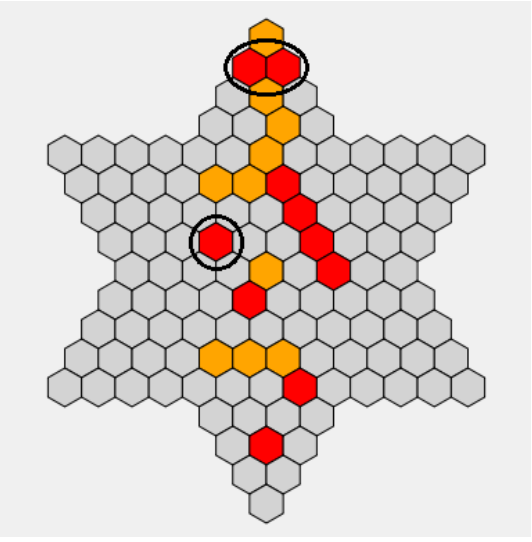
The algorithms we used were “minimax" and “alpha-beta pruning". Minimax is an algorithm that takes into account all possible moves that can be made by the opponent and itself, generate them into a tree and then chooses the best move that can be made based on an evaluation function.

Alpha-beta pruning is an optimisation wherein it removes the tree of all unnecessary moves so that time is not spent in generation of the irrelevant moves and evaluation of consequent board positions.

At the core of these algorithms is an evaluation function, which “sees" the board status and returns a numerical estimate of the “goodness" of the board. In the right board, the circled pieces are isolated and hence their contribution towards the evaluation will be negative. However, since they have advanced considerably from their initial positions, there will be significant positive contribution also.



Valid Moves



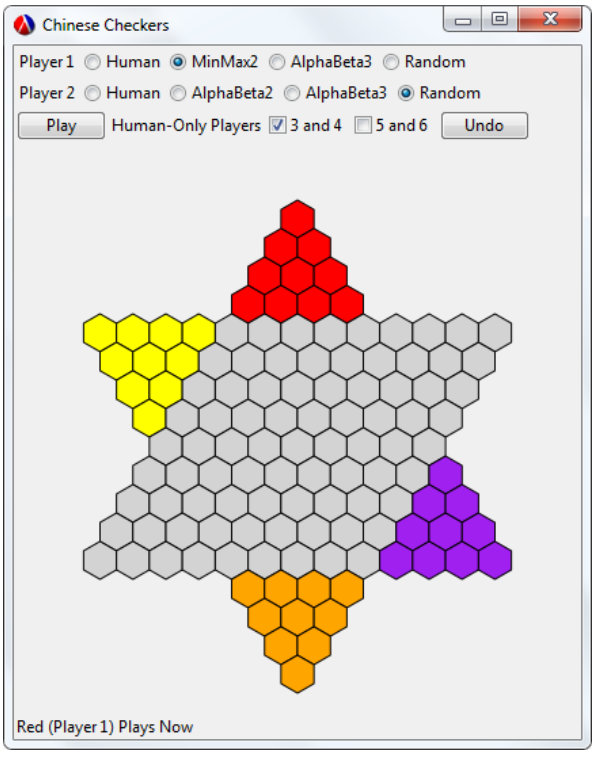
Strategy

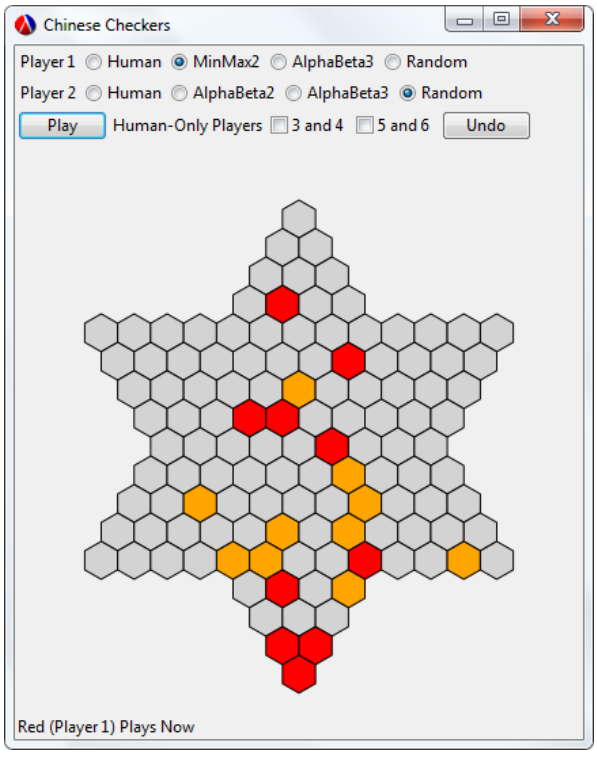
SAMPLE INPUT/OUTPUT:

For Players 1 and 2, the type of player is chosen (i.e. human or the difficulty level of AI for the computer). The other players are always human. Since it is a two, four or six-player game that we designed, “Players 3 and 4" and “Players 5 and 6" are clubbed together. Once this done, the “Play" button is clicked. The winner of the game is declared in the “status bar" at the bottom.

THE QUIRKS:

1. The minimax algorithm does not behave properly when the search depth (“ply") is three or more (plays logically wrong moves for the same evaluation function that works correctly for the alpha-beta algorithm). This arose due to the impracticality of checking search trees beyond a ply of three (as they require thousands of lines to print/display and hence cannot be humanly checked for correctness).
2. The alpha-beta algorithm at depth 2 is - in perception - slower than minimax at depth 2, though much faster at higher ply - as expected. This is likely because of the overhead involved in the implementation of alpha-beta since pruning is not quite apparent at low ply.





RESOURCES:

The racket/gui/base and racket/vector libraries are used. Mouse input is captured using the functions in the racket/gui/base module itself.

FUTURE EXTENSIONS:

1. The three player game variant is not provided as additional overhead of designing an evaluation function to handle cross-board movement for the players would have to be incurred (for the orange and red players, moves are easy to evaluate as they are forward or backward in the vector as well as the actual logical board). A human-only three-player variant is merely a triviality to implement, though.
2. Extending the minimax and alpha-beta algorithms (not merely the implementations) to a multiple-player game with multiple human(s) and/or computer(s) can be done. Currently, the algorithms (that are available and have been implemented) require enormous computation of the search tree/nodes, which is not feasible for a large number of players. As such, existing literature on Chinese Checkers does not include any such algorithms.