

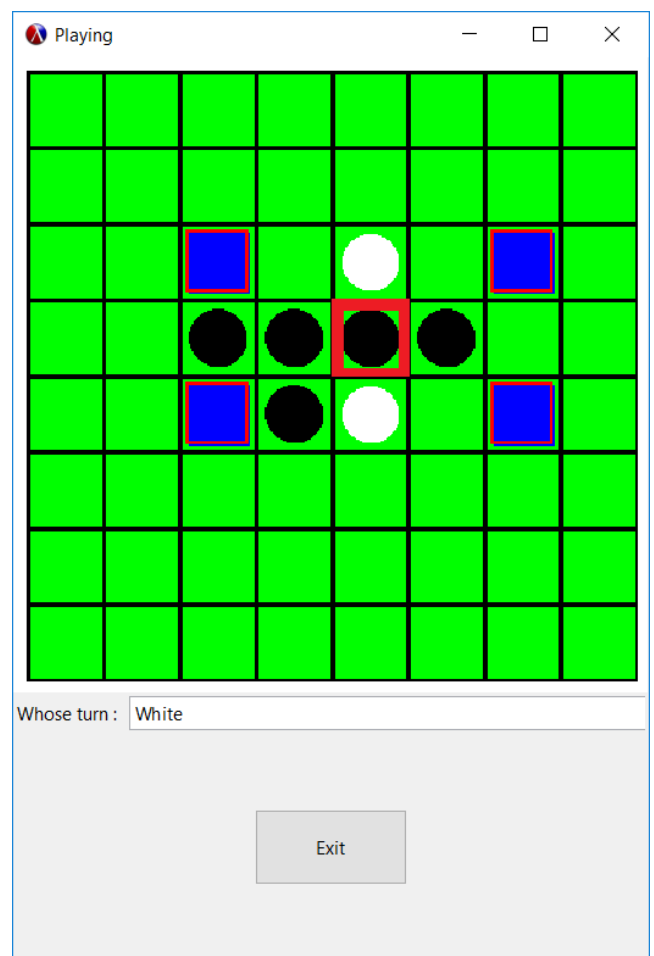
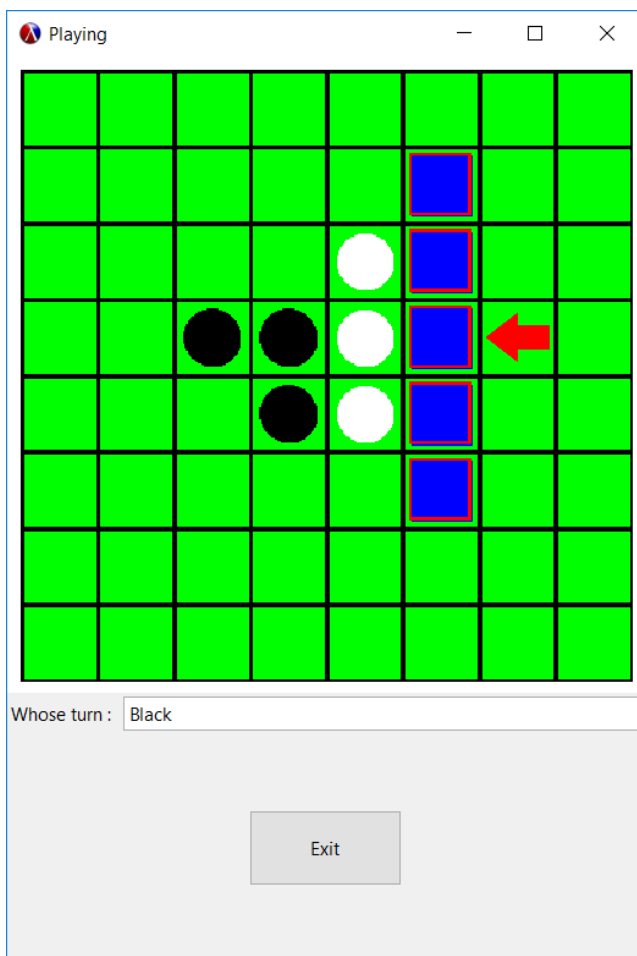
REVERSI (CS 154 Project)

Group members: - Sudhanshu Raj (160050041)
Ankit (160050044)
Sarthak Mallick (160050070)

Our project is a desktop version of the classic game of “Reversi”, a strategy board game for two players, played on square board. There are white and black coloured coins, one type for each player. During play, any coins of Player 1 that are in a straight line and bounded by the coin just placed and another coin of the Player 2 turn into Player 2’s colour.

The objective of the game is to have the majority of coins turned to display your colour when the last playable empty square is filled.

In the left image, one black coin is placed at the square marked. As a result, the white coin to its left turns black (see right image), as it has been surrounded at two ends by the black coins. Blue squares show the possible moves.



Game Design

- The board has been implemented as a list of lists, with 0 for blank, 1 for white coin, -1 for black coin and 2 to highlight possible moves. Player can select **game mode (Single or Double player)** and the **board size (6x6, 8x8, or 10x10)**.
- Upon getting the mouse input, we find out the square on which click was made, and if legal, a coin is placed on that square and changes are made to the required squares as explained earlier, along with determining the next set of possible moves.
- In Single Player mode, the AI determines and plays the next move according to the difficulty chosen **level (Easy, Medium or Expert)** which limits the depth of recursion in miniMax algorithm. Both player and computer moves are implemented simultaneously on the graphics canvas.
- The graphics has been implemented using the racket/gui package, with extensive object-oriented programming.

Bugs

- Gameplay can be stopped only by the Exit button, and not the Close button on title bar.
- Occasionally initial-board does not get updated when Play Again is clicked.

Abstractions and High-Level Design

- Use of higher order functions of lists (foldr, map), and list comprehension macros, wherever suitable.
- Single function for both double and single player, in which double player mode accepts input from mouse, while input is computer generated in latter case.
- The game can be extended to any board size, as the functions have all been written for a generalized n.

- To get optimum solution, we check each square on board whether placing a coin is allowed there and if yes, we find the number of opponent's coins it would reverse. Ties are resolved by principle that stability of corners > edges > rest.
- For Easy mode (global variable level = 1), the AI searches for optimum solution in the board-list.
- As levels increase, deeper levels of recursion (e.g. level = 2), for every valid move, it checks valid moves for opponent, and for every such move, it returns the optimum solution, and finally the best move is returned. For higher value of level, minimax function goes to deeper levels of recursion.