

Reversi - Project 1, EDAF70

Ola Johansson (jur10ojo@student.lu.se)
Niklas Lundström (tpi13nlu@student.lu.se)

February 7, 2018

Design - Classes

GameRunner Holds the main method and handles input from the user for selecting color and amount of time granted to the agent for calculations. Contains a main loop for running the game. Prints the winner and the score by calling methods in Game. Has the following attributes:

- Game
- Agent

Game This class contains the game mechanics and methods for making moves, updating and printing the board, calculating score, and designating valid moves for both players. Has the following attributes used for representing the board and the lists of valid moves:

- GameMatrix
- List<Tuple>
- List<Tuple>

Tuple A basic integer tuple class with getters used to represent coordinates, and tuples.

Util Holds constants and methods that are used throughout the project. Handles the symbols for the game board (X, O, etc).

GameMatrix Holds a two-dimensional char array which represents the game board and markers placed on it. Methods for setting and getting values at given coordinates.

Agent Calculates the best move using an implementation of the *MiniMax* algorithm implementing *Alpha-Beta Pruning*. The Agent uses an iterative approach to maximize the time given, applying the minimax algorithm to increasing depths until the time limit is reached. When the terminal condition is fulfilled (max depth or end of game reached) a *fitness function* is evaluated for the given state.

The fitness is calculated as the difference between the agents score (number of markers) and the opponents score, with a bias towards corners (10 extra points are awarded for capturing corners).

Manual

The program *Reversi.jar* is found at path `/h/dk/x/jur10ojo/edaf70/reversi` and is run by executing the command `java -jar Reversi.jar` from the command line. Alternatively, the main method is found in the *GameRunner* class and can be invoked for example by running the command `java -cp bin/project_reversi.GameRunner` from the directory mentioned above. The program will prompt the user for selection of player and time limit for the agent. The game will then run until end of game, or the process is killed by the user. The board is represented by ASCII symbols. "." represents an empty slot. "+" represents an empty slot which is playable by the current player. "X" and "O" represent the two players respectively.