Assignment 5: Reversi

Submission date: May 22^{nd}, 2015. 16:00

Objective: Java Threads.

1. Introduction

In this assignment you will first extend assignment 4 with a new game: *Reversi.* The purpose is to "appreciate" the design that we have followed so far, and understand how it allows adding a new game without modifying exiting code. The new game is also a board game, however, with different rules that we detail below:

- The board size is 8 × 8. One difference from the other games, that we have implemented so far, is that the initial board is not empty, but rather includes 4 counters (2 of each color) in the central area of the board where each color form a diagonal (see Figure 1).
- Assume that the counters are white in one side and black in the other side (because they can be flipped as we will see later). Each player is assigned a color. The first move is done by the player with the black counters, alternating the turn with the other player.
- The game ends when all cells are occupied. The winner in such case is the one with more counters placed on the board, if they are equal then it is a draw.
- How to perform a move:
  1. A move is done by placing a counter in an empty position, but it is valid only it flips one or more counters of the other color. *Flipping* occurs when a player places a counter of color A at the end of a line of consecutive counters of color B only (one or more) such that its other end is of color A. In such case all Bs **must** become As (i.e., flipped). The line can be vertical, horizontal or diagonal (see Figure 2).

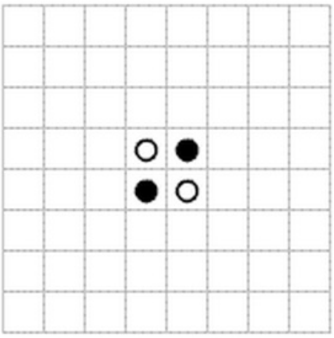


Figura 1: Reversi: initial board.

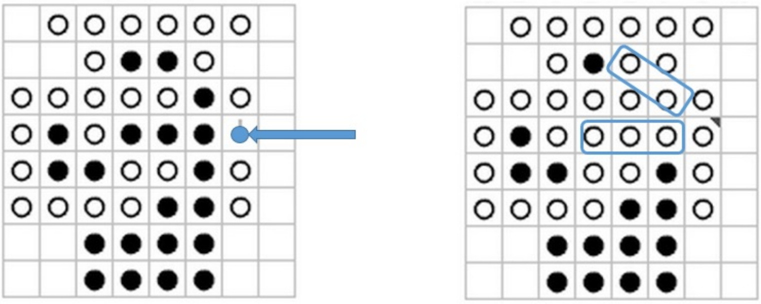


Figura 2: The state of the board after placing a white counter in the place marked by the blue arrow

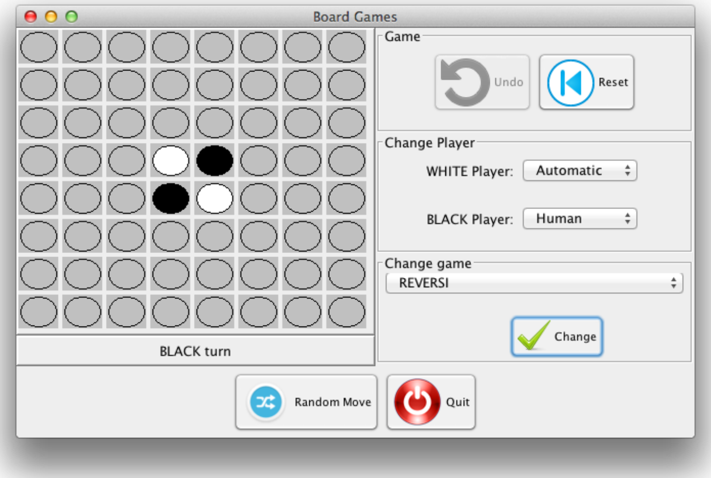
2. If placing one counter can flip several lines (at the same time), then all corresponding counters must be flipped.
3. If a player has no possibility to place a counter that flips other counters, then the turn passes to the other player (automatically).

In addition to the new game, you should extend the window mode to allow changing the type of player associated with each color – human or automatic (see Figure 3). This way it is possible to play against the automatic player, or just let both players be automatic, in which case the user just sits back and watch the game. It is important that while the automatic player is deciding which move to make (we will add some seconds of delay to make it take time), the Swing window **must remain responsive** and: (1) restart the game, changing a game, or quiting the application are allowed; and (2) *Undo*, making a move by a human player or changing the type of players must be deactivated. To keep the window responsive you should use Threads. **In class we will elaborate on this part**.

2. Application Parameters

The application parameters, and the corresponding help, should be changed to allow for the new game:

```
%> java [details omitted] tp.pr4.Main -h
```



WHITE Player: Automatic

BLACK Player: Human

Change game: REVERSI

BLACK turn

Figura 3: Reversi

```
usage: tp.pr4.Main [-g <game>] [-h] [-u <mode>] [-x <columnNumber>] [-y
       <rowNumber>]
 -g,--game <game>          Type of game (c4, co, gr, rv). By default, c4.
 -h,--help                 Displays this help.
 -u,--ui <mode>            Type of inteface (console, window).
                           By default, console.
 -x,--dimX <numColumns>    Number of columns on the board (Gravity only).
                           By default, 10.
 -y,--dimY <numRows>       Number of rows on the board (Gravity only).
                           By default, 10.
```

Remember to add the option ’rv’ also to the help that is shown in the console mode when typing the command ’help’.

3. Submission of the assignment

The assignment should be submitted via Campus Virtual before the deadline that appears at the start of the present document.

The submission takes the form of a zip file named `GroupNN.zip`, where `NN` is the group number, to be submitted by only one member of the group. This file must include the following[1]:

- The `src` directory containing the source code of all the classes of the assignment.
- A text file `students.txt` containing the names of the members of the lab group.

[1] You may also include the information files of the Eclipse project if you wish