EFREI 2019-2020 L1 Int.

Project 2 : Black & White

Contents

1 Game Description

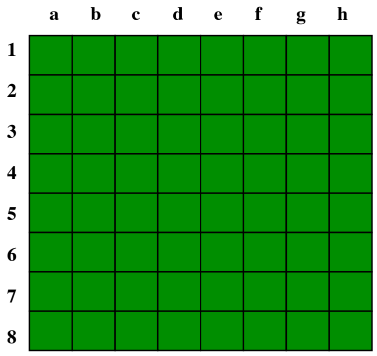
The game **Black & White** is a game that is played with two players, one plays with the camp of black pawns, the other with the camp of white pawns. The pawns are all double-sided, one white and one black. The goal of the game is to count, at the end of the game, more pawns of its color than the opponent by returning them according to precise rules. In the rest of the statement, the player playing with the black pawns will be called *Black*, the player playing with the white pawns will be called *White*.

The game board is 8x8 boxes (64 boxes in total), all of which are of the same color. Players have a set of 64 pieces that have a black face and a white face. The game is played alternately: *White* places a pawn and returns some black pawns, then *Black* places a pawn and returns some white pawns, and so on until the end of the game. The board lines are numbered 1 to 8 from top to bottom, and the columns from A to H, from left to right.

2 Placing of pawns and capturing

In his turn of play, the player must place a pawn of his color on an empty box, adjacent to an opposing pawn. He must also, by placing his pawn, enclose one or several opposing pawns between the pawn he places and a pawn of his color, already placed on the game board. He then returns the pawn(s) he has just enclosed (which become of his color). The pawns are neither removed from the game board nor moved from one box to another.

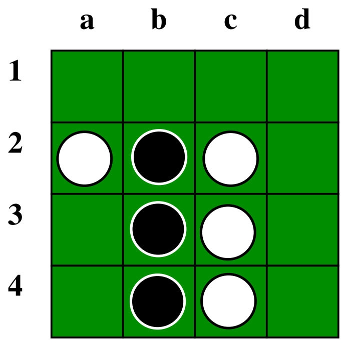
The enclosure is done according to the lines, the columns and the diagonals.



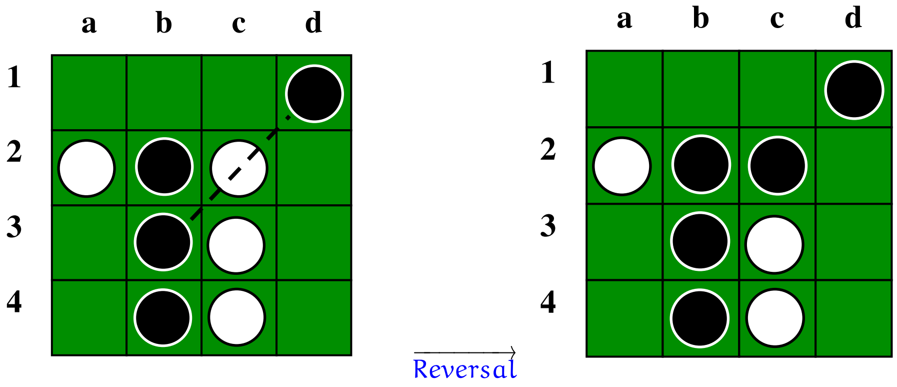
When a pawn is placed, the player returns all the opposing pawns enclosed by this pawn and his pawns already placed. In the rest of the statement, all examples will be shown on a 4x4 game board, but can be transposed on the original 8x8 board.

2.1 Illustration of the placing of a pawn

If it is *Black*'s turn to play: it can enclose white pawns only if it plays in d_1, d_2, d_3 or d_4.

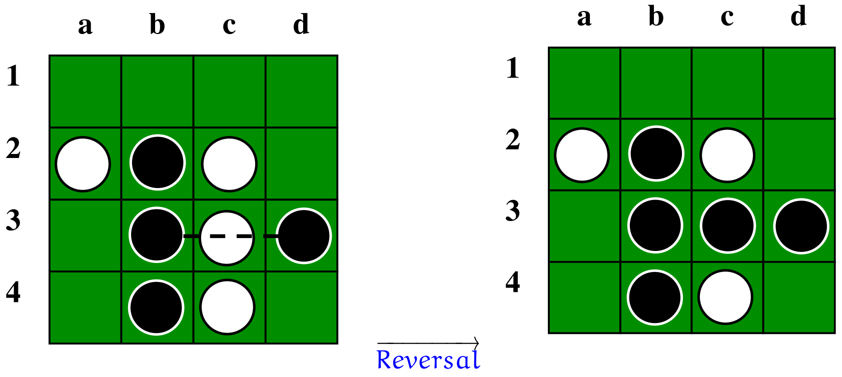


Assume he plays in d_1: there will be enclosure of the white pawn c_2 with the black pawn b_3 (already placed) and the pawn played in d_1.

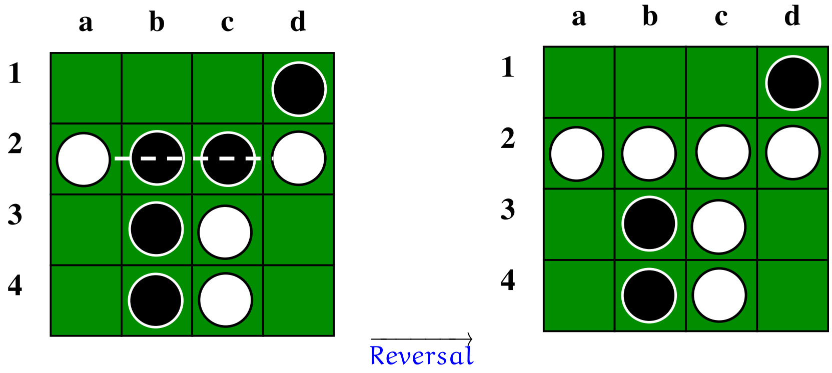


Assume he plays in d_3: there will be enclosure of the white pawn c_3 with the black pawn b_3 already placed and the black pawn played in d_3.

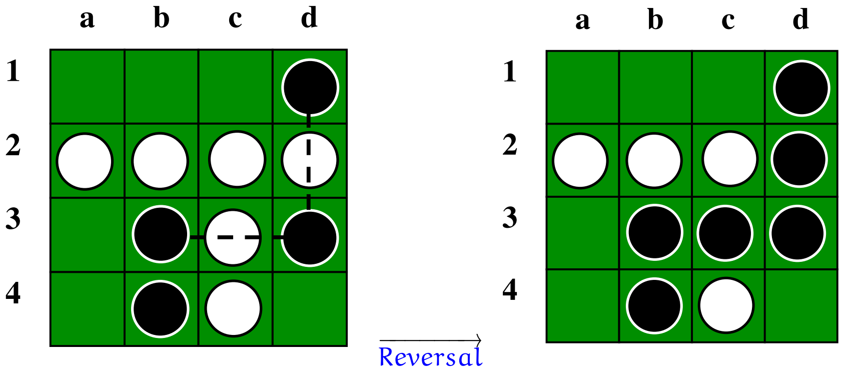
From the position: *Black* placed in d_1, the turn of *White*:



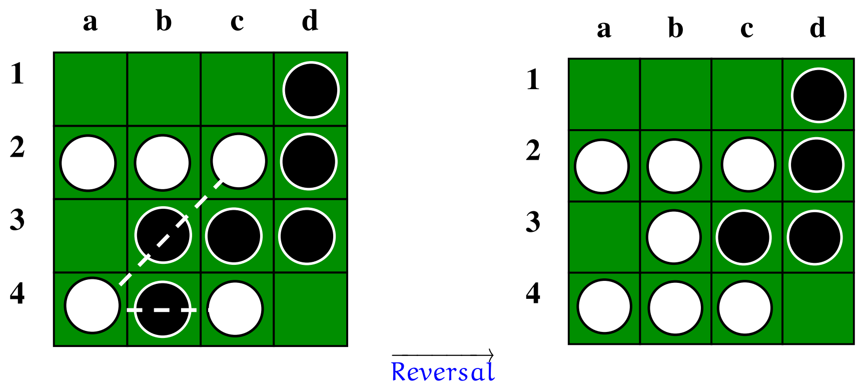
White can play in: a_1, a_3, a_4, c_1 or d_2. To illustrate, let's admit that he plays in d_2 (questionable tactical choice, do you see why?). The two enclosed pawns are turned over.



Black replicates in d_3, and the two enclosed pawns are returned (even if they are enclosed by two distinct alignments).



White finally replies by a_4: it returns b_3 and b_4.



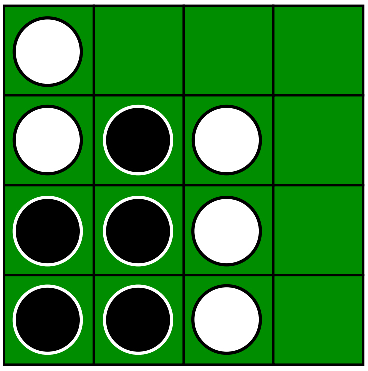
2.2 Starting position

At the beginning of the game, two black pawns are placed at e_4 and d_5 and two white pawns are placed in d_4 and e_5. *Black* plays first.

2.3 To skip the turn

When a player places a pawn, there are two possibilities:

- He can place a pawn in at least one box: the player is then obliged to place a pawn;
- He has no chance to play (this case can happen), and then he must pass his turn (but the game is not over yet).

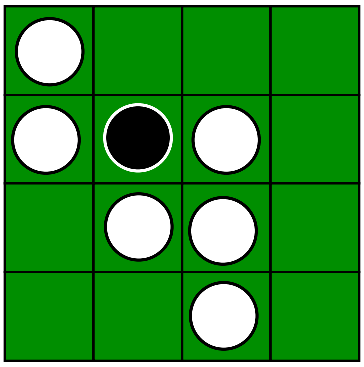


If it is *White*'s turn to play, he can not put a pawn, because no capture is possible. He must therefore pass his turn. It's up to *Black* to play. If he plays well, he can still block *White*!

2.4 Game over

1. **By elimination**

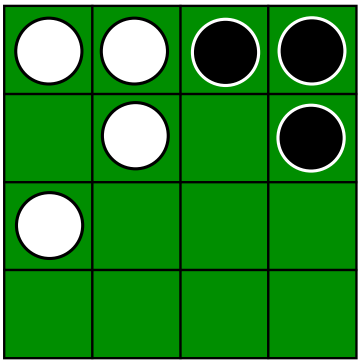
   If a player has no pawns on the game board, he will not be able to put a pawn or return any opposing pawns, so he has lost the game.



It is *White*'s turn to play: he will give the "coup de grace" and eliminate *Black*.

2. **By mutual blocking**

   It is possible that neither player can play (each player is obliged to pass his turn because he can no longer take any pawns), although there are still some free boxes on the board: in this case the game is over, and the pawns of each side are counted: the player who obtains the highest total won the game. There may be a draw.



   Here, we have a mutual blocking: *White* wins by 4 to 3

3. **By filling the game board**

   The most common game ends when the 64 boxes on the board are occupied. The counting of the pieces of each side is done: the player who obtains the highest total won the game. There may be a draw.

3 Specifications

3.1 Game Management

- Place the pieces at the beginning of the game;
- Manage a variable size game board (6x6 minimum);
- Refuse invalid moves;
- Propose (on request) the list of valid moves;

- Identify the fact that a player must pass his turn;
- Identify and indicate all end-of-game cases;
- Propose to stop the game at any time;
- Determine the winner;
- Provide help on keys to use;
- Enter the position of a pawn to be placed with its coordinates (you do not have to use a cursor on the screen).

3.2 Optional

- The program must be able to go back (cancel N moves, N being chosen and managed at your convenience).
- The program plays against the user (AI).

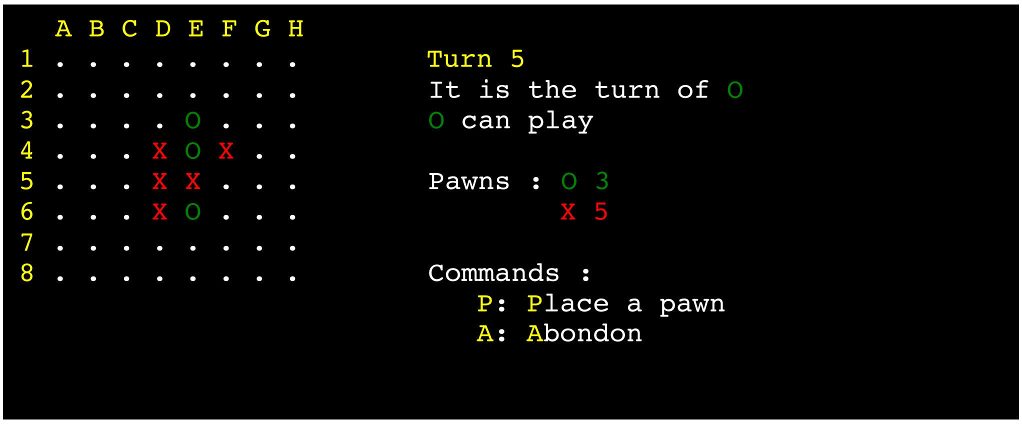
3.3 Display

Simple text interface, in which must appear:

- The game board with pawns, with a display of column names and line numbers;
- The player whose turn it is;
- The possibility for the player whose turn it is to pose a pawn or not (i.e., the program must detect if the player must pass his turn);
- The number of pawns on each side;
- The number of the game round;
- The indication of the keys or actions to be used for the players.

You can display colors using the same technique as for the first project.

3.3.1 Example:



```
  A B C D E F G H
1 . . . . . . . .      Turn 5
2 . . . . . . . .      It is the turn of O
3 . . . . O . . .      O can play
4 . . . X O X . .
5 . . . X X . . .      Pawns : O 3
6 . . . X O . . .              X 5
7 . . . . . . . .
8 . . . . . . . .      Commands :
                          P: Place a pawn
                          A: Abondon
```

Optional: You have the possibility to make an advanced interface but, apart from the elements listed in the *Specifications* Section, an advanced interface will not bring any bonus. Do this only if the basic version of the project is complete.

3.4 Technical aspects

1. To encode the game board, you will necessarily use a two-dimension array.
2. Use a function for each requirement listed in the *Specifications* Section
3. **All entries must be secured.**

3.5 For the project defense

You will have to present the game program. The teacher will ask for a demonstration of the program, in which all the points of the specifications will be verified.

Note: It would be appreciated if, for your demo, you prepare some established scenarios to illustrate your work.

3.6 For the report

The report must contain: an introduction, a simple presentation of the project, the different algorithms, an explanation of the main difficulties encountered, an analysis of the project and a conclusion.

4 Project planning

1. Publication of project subject: November 21, 2019
2. Project follow-up session:
   - INT1: December 3rd, 2019
   - Groups A and B : December 4th, 2019
   - Groups C, D, F, E, BN, R: December 12th, 2019
3. Project defense:
   - INT1: December 17th, 2019
   - Groups INT2, INT3: January 9th, 2020
4. Submission of code sources and report (on the moodle): December 20th, 2019 at 11:59 pm.

Have fun