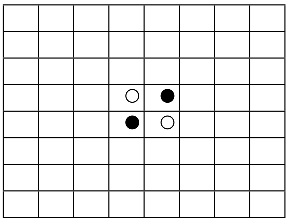
Département IMA / 3A (S5)
Programmation Structurée

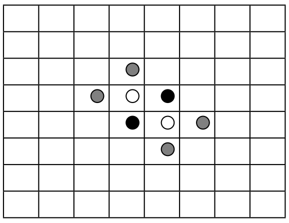
Project: the Othello board game.

1 The Game

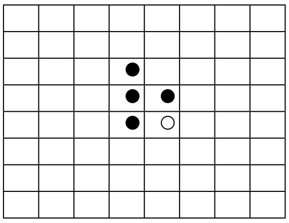
Othello is a strategy board game for two players (Dark and Light). The 8 × 8 board initially contains 4 pieces in the center, as depicted below:



Players take turns, with Dark going first. Dark must place a piece in such a position that there exists at least one line (horizontal, vertical, or diagonal) between this piece and another dark piece, with one or more contiguous light pieces between them. For instance, the figure below depicts in gray all the possible moves for Dark on the first turn.



After placing the piece, all light pieces lying on a line between the new piece and any dark pieces becomes dark (there may exist several such lines). For instance, the figure below depicts the outcome when Dark chooses to play in the topmost location.



Then, Light plays with the same rules (except that the colors of the pieces are of course reversed).

If one player **cannot** make a valid move, play passes back to the other player. It is however illegal not to play when there exists a valid move.

When neither player can move, the game ends. The winner is the player with the most pieces on the board at the end of the game.

2 Your work

1. Develop a program that implements the Othello board game for two human players, who alternate playing with the same keyboard;
2. Add the possibility to replace one player by a computer controlled player (an Artificial Intelligence). Implement the following strategies:
   - The AI plays random, but **valid** moves;
   - The AI maximizes the gain of its current move;
   - The AI follows a Minimax strategy (see `https://en.wikipedia.org/wiki/Minimax`). There exists many variants for this strategy, start simple and then improve your solution if you still have time;
   - *(Optional)* You can combine the following strategy with the previous ones: the AI favors edge moves, and even more so corner moves (can be combined with the other strategies);
3. *(Optional)* You can use the NCURSES API for cleaner display of the board: `http://www.tldp.org/HOWTO/NCURSES-Programming-HOWTO/helloworld.html`.

3 Evaluation

You must send an archive (.zip or .tgz) by email to `julien.forget@univ-lille.fr` before Thursday 19/12/2019, 19:00. The archive must contain:

- Your source code (no executable);
- A report, max 10 pages, that presents your work and in particular details:
  - Your main algorithms, including your implementation of the Minimax strategy;
  - An estimation of the time-complexities of your different AI strategies;
  - What is implemented and what is not implemented;
  - The limitations of your program.

The following criteria will be taken into account to evaluate the quality of your code:

- Time/space complexity;
- The program compiles without warnings with option `-Wall`;
- The program executes correctly, and causes no segmentation fault;
- Code is correctly indented;
- No magic numbers;
- Variable and function names are explicit;
- Difficult points are (briefly) commented;
- Code is structured in procedures;
- No code duplication.