

Devoir maison n°6

À rendre lundi 16/12

★ ★ ★

Othello

Le jeu d'Othello ou Reversi est un jeu stratégique à deux joueurs. La situation initiale est un plateau de 8×8 cases, avec 4 pions posés sur les 4 cases du milieu, deux étant noirs (en diagonale) et les deux autres étant blancs. Les joueurs jouent tour à tour (traditionnellement en commençant par les noirs). À son tour, un joueur place un pion de sa couleur (blanc ou noir) et transforme tous les pions capturés de la couleur opposée en pions de sa couleur, les pions capturés étant ceux qui sont encadrés (horizontalement, verticalement ou diagonalement) par un pion de la couleur du joueur en cours et le pion qui vient d'être posé. Si aucun pion n'est capturé, le coup n'est pas valide. Si un joueur n'a aucun coup valide, il passe son tour. La partie termine lorsque la grille est remplie ou que les deux joueurs ont passé leur tour. Le gagnant est celui qui a le plus de pions de sa couleur.

On cherche dans ce devoir à établir une stratégie pour un jeu d'Othello, à programmer en langage OCaml.

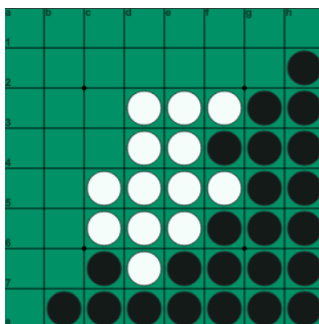
Ce qui est demandé

L'implémentation

On suppose qu'une partie d'Othello est implémentée par le type suivant :

```
type othello = {plateau : int array;  
                mutable joueur : int}
```

tel que si `oth` est un objet de type `othello` décrivant une partie, alors `oth.plateau` est un tableau unidimensionnel de taille 64 correspondant aux cases du plateau ligne à ligne et `oth.joueur` correspondant au joueur courant, valant 1 ou 2. Le tableau `oth.plateau` contiendra des 0 pour les cases sans pion, 1 pour les pions noirs et 2 pour les pions blancs. La figure 1 représente un plateau d'Othello.



```
let oth = {plateau = [
    0; 0; 0; 0; 0; 0; 0; 0;
    0; 0; 0; 0; 0; 0; 0; 1;
    0; 0; 0; 2; 2; 2; 1; 1;
    0; 0; 0; 2; 2; 1; 1; 1;
    0; 0; 2; 2; 2; 2; 1; 1;
    0; 0; 2; 2; 2; 1; 1; 1;
    0; 0; 1; 2; 1; 1; 1; 1;
    0; 1; 1; 1; 1; 1; 1; 1 |];
    joueur = 1}
```

FIGURE 1 – Un plateau d'Othello. C'est au joueur noir de jouer.

L'objectif

L'objectif de ce devoir maison est de déterminer une stratégie pour le jeu, qui prendra la forme d'une fonction :

```
strategie (oth: othello) : int
```

prenant en argument une configuration d'Othello et renvoyant un entier compris entre 0 et 63 représentant l'indice de la case dans laquelle le joueur courant doit jouer. La fonction ne devra pas modifier le jeu donné en argument.

Par ailleurs, vous êtes libres d'écrire des fonctions auxiliaires qui seront appelées par la fonction `strategie`, voire d'utiliser vos propres structures de données intermédiaires.

J'attends de vous un investissement minimum, quitte à écrire une fonction qui renvoie simplement un coup choisi arbitrairement parmi les coups possibles.

Ce qu'il faut rendre

L'évaluation sera faite automatiquement. Vous devrez rendre un fichier `.ml` dont le nom sera `othello` suivi d'un suffixe de taille 5 contenant un tiret bas, les trois premières lettres de votre nom et la première lettre de votre prénom (tout en minuscules, sans accent). Par exemple, mon (Nathaniel Carré) fichier devrait être nommé `othello_carn.ml`.

Le fichier devra contenir la fonction `strategie`. Il devra pouvoir être **interprété** (mais pas nécessairement pouvoir être compilé).

Il ne devra pas effectuer d'affichage.

Le fichier source devra m'être envoyé par mail ou via messagerie ENT.

Tout travail rendu ne respectant pas ces contraintes ne sera pas évalué.

Les modalités d'évaluation

Votre travail sera évalué lors d'un tournoi à double élimination. Les résultats du tournoi vous seront communiqués. Pour chaque match entre deux élèves A et B du tournoi, deux parties seront jouées : une où A commencera à jouer et une où B commencera à jouer. Pour chaque partie, chaque joueur remportera un certain nombre de points, et celui qui a le plus de points après les deux parties remporte le match. Si les deux joueurs remportent le même nombre de points, les deux sont considérés comme ayant remporté le match. Lors d'une partie, les nombres de points attribués le sont de la manière suivante :

- si la partie termine normalement, chaque joueur remporte autant de points que son nombre de pions de sa couleur à la fin de la partie ;
- si le temps de calcul total d'un joueur dépasse 2 minutes^{1 2} pour l'ensemble de la partie, il remporte 0 point, son adversaire remporte 65 points ;
- si la stratégie d'un joueur renvoie un coup impossible, il remporte 0 point, son adversaire remporte 66 points ;
- si l'appel à la fonction d'un joueur plante, il remporte 0 point, son adversaire remporte 67 points.

1. visez plutôt 30 secondes sur votre ordinateur si vous voulez de la marge

2. attention, il s'agit du temps d'un calcul interprété et non compilé