

# Chapitre 12 : Tournoi

L'objectif de ce TP est d'écrire en groupe une fonction qui implémente une stratégie pour le jeu Othello. Cette stratégie participera ensuite à un tournoi avec les stratégies des autres groupes.

## 12.A Implémentation du jeu

On représente un état du jeu par une valeur de type othello :

```
type othello = int array array * int
```

tel que si (`plateau, joueur`) est un couple de type `othello` alors `plateau` est une matrice de dimensions  $8 \times 8$  telle que `plateau.(0).(0)` correspond à la case la plus en haut à gauche du plateau `joueur` au numéro du joueur (1 ou 2) qui doit jouer le prochain jeton.

Une absence de jeton est représenté par la valeur 0, un jeton noir par la valeur 1 (les jetons noirs sont joués par le joueur 1) et un jeton blanc par la valeur 2 (les jetons blancs sont joués par le joueur 2).

L'objectif de ce TP est de déterminer une stratégie pour le jeu, qui prendra la forme d'une fonction `strategie : othello -> int*int` prenant en paramètre un état du jeu `othello` et renvoyant les coordonnées de la case dans laquelle on souhaite placer un jeton.

Vous êtes libre d'écrire d'autres fonctions qui seront appelées par la fonction `strategie` ou d'utiliser vos propres structures de données.

Tout doit être calculé par votre fonction, votre fichier ne doit pas contenir de données pré-calculées.

Pour information, le programme ne sera pas compilé mais interprété.

Le fichier `othello.ml` contient de nombreuses fonctions utiles que vous pouvez intégrer dans votre programme.

## 12.B Rendu

Par équipe de trois vous devez rendre au minimum un fichier `minmaxnn.ml` où `nn` est le numéro qui sera attribué à votre équipe (sur deux chiffres donc `minmax01.ml` et pas `minmax1.ml`) utilisant l'algorithme min-max (élagage alpha-beta facultatif).

Vous pouvez également rendre un fichier `avancénn.ml` proposant des modifications/améliorations significatives de l'algorithme min-max.

## 12.C Évaluation

Votre travail sera évalué par une note sur 10 avec les critères suivants :

- **1 point** : Respect des consignes de rendu (notamment les règles du tournoi).
- **3 points** : Qualité du code (le code est lisible, bien organisé, judicieusement commenté, les variables et fonctions sont bien nommées, on donne les spécifications de chaque fonction).
- **1,5 point** : Votre programme utilisant l'algorithme min-max est capable de gagner face à un adversaire qui joue des coups aléatoires.
- **1 point** : Votre programme utilisant l'algorithme min-max est capable de gagner face à un adversaire basique.
- **2 point** : Classement lors du tournoi de vos programmes
- **1,5 points** : Évaluation de votre éventuel programme avancé

## 12.D Règles du tournoi

Pour chaque match entre deux équipes  $A$  et  $B$  du tournoi, deux parties seront jouées : une où  $A$  commence et une où  $B$  commence. Pour chaque partie, chaque équipe remportera un certain nombre de points et l'équipe qui aura remporté le plus de points après les deux parties remporte le match.

Lors d'une partie, les points sont attribués de la manière suivante :

- si la partie termine normalement, chaque équipe remporte autant de points que son nombre de jetons de sa couleur à la fin de la partie ;
- si le temps de calcul d'une équipe dépasse 2 minutes pour l'ensemble de la partie, elle remporte 0 point, l'équipe adverse remporte 55 points ;
- si la stratégie d'une équipe renvoie un coup impossible, elle remporte 0 point et l'équipe adverse 64 points ;
- si l'appel à la fonction d'une équipe plante, elle remporte 0 point, son adversaire remporte 60 points.