

## TP 5 - Othello

### A lire attentivement!

Ce travail est noté. Toute aide externe (telle que poser des questions sur des forums ou utiliser des assistants génératifs de type Copilot) est interdite.

Vous développerez vous-même votre code ! La copie d'une (petite) portion de code préexistant est tolérable si :

- Elle reste occasionnelle et largement minoritaire ;
- Elle est clairement signalée par un commentaire adéquat. Le non-respect de ces règles sera considéré comme de la tricherie et pourra occasionner des sanctions.

### Contexte

Pour ce mini-projet du cours d'IA, nous allons développer une IA pour le jeu Othello.

Notre version du jeu Othello se jouera à deux joueurs sur un damier 7 x 9, avec des pions bicolores (à face blanche ou noire). La configuration initiale du damier sera la suivante :

Pour le reste, les règles normales du jeu Othello s'appliquent :

- Les coups admissibles pour un joueur sont ceux qui permettent de prendre « en sandwich » une ou plusieurs séries (ligne, colonne ou diagonale) d'au moins un pion de la couleur adverse. Tous les pions ainsi pris en sandwich sont retournés et changent donc de couleur.
- Quand un joueur ne peut pas jouer, il passe son tour.
- Le jeu se termine quand plus aucun des deux joueurs ne peut poser de pion, le gagnant est alors le joueur qui a le plus de pions de sa couleur sur le damier (les éventuelles cases vides comptant pour le gagnant).

(Pour plus de détails sur le jeu ou la stratégie, vous pouvez consulter par exemple <http://www.ffothello.org/othello/regles-du-jeu>.)

### Le projet

Le but du projet est d'intégrer au package Python `app_othello` un module IA selon l'algorithme alpha-beta, qui implémente une stratégie de jeu pour Othello. Ce module sera utilisé pour organiser des tournois entre les différentes IA.

Pour cela, il faut développer les points suivants :

- Une fonction d'évaluation d'un état: c'est principalement sur cette fonction que va dépendre la qualité de jeu de votre implémentation.
- L'algorithme alpha-beta: celui-ci devra être paramétrable par la profondeur maximale de recherche.

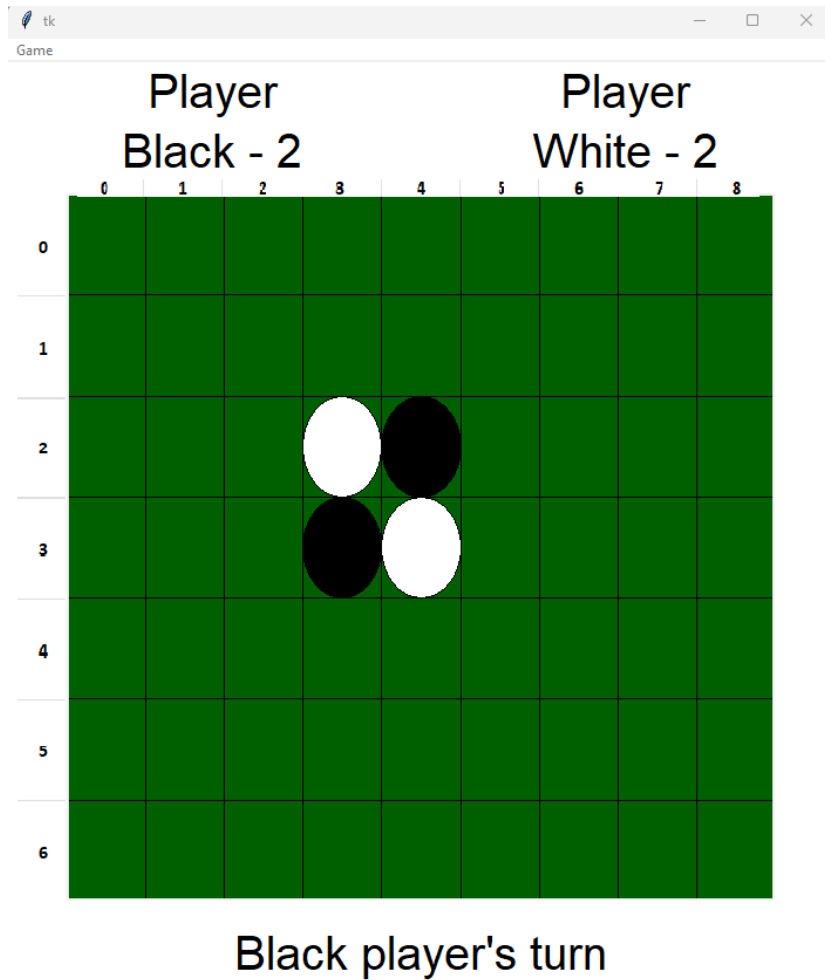


Figure 1: App Othello avec grille 7x9 (lignex x colonnes)

## Organisation

- Le projet est réalisé par groupe de deux personnes
- Les semaines de cours du 06, 13, et 20 décembre seront consacrées à ce projet
- **Le projet est à rendre pour le dimanche 8 janvier 2023 à 23:59 sur Cyberlearn**
- **Les présentations et le tournoi sont obligatoires et auront lieu pendant le cours de la semaine du 10 janvier**
- La présentation doit être courte et concise. Il faut expliquer les choix spécifiques pour votre IA, en particulier la fonction d'évaluation. Elle doit contenir 5 slides maximum, et doit durer au maximum 10 minutes, questions comprises. Chaque membre du groupe doit présenter avec une répartition uniforme.
- À rendre : `Nom1_Nom2.py`
- Ajouter un fichier `readme` avec quelques commentaires sur ce qui fonctionne, ce qui ne fonctionne pas, et les éventuels autres points notables de votre implémentation.

## Tournoi

- Le tournoi aura lieu en classe, et chaque groupe présentera son IA.
- Tous les matchs se dérouleront à la même profondeur (fixée à 5).
- Le règlement et l'organisation du tournoi seront définis ultérieurement.
- Le score d'un match est la différence des pions (les éventuelles cases vides comptant pour le vainqueur). Une erreur (coup invalide ou exception) donne une défaite 0-63 pour l'équipe fautive.

## Évaluation

Le projet donnera lieu à une note. Il sera évalué selon les critères suivants :

- Qualité de l'implémentation du jeu (bonne représentation du jeu, ia, ...) (25%)
- Qualité du code (maîtrise du langage, lisibilité, commentaires) (25%)
- Performance au tournoi (25%)
- Présentation et justifications des choix IA (25%)

De plus, il est judicieux de faire particulièrement attention au nettoyage de votre code avant de le rendre :

- Pas de code inutile ! (Retirer les import et fonctions qui ne servent plus à rien, . . .)
- Évitez les blocs de 50 lignes de code mis en commentaire “pour l’instant” !
- Est-il encore besoin de le préciser : Commentez votre code !

## Structure du code (app\_othello.zip)

Le code est fourni sous la forme d'un package Python. Le dossier `app_othello` du package contient :

- `othello_gui` : Fichier principal qui contient l'interface graphique GUI et logique du jeu. Pour lancer le programme, c'est ce fichier qu'il faut exécuter, à partir du dossier `app_othello`, avec la commande suivante:

```
python othello_gui.py
```

- `othello` : implémente la logique Othello et s'occupe de faire l'arbitre.
- `ai.Random` : version d'une IA qui joue toujours de manière aléatoire.
- `ai.Nom1_Nom2.py` : c'est à vous de créer votre fichier, dans lequel vous implémenterez votre IA. **Pour cela, il faut créer une classe avec exactement le même nom du fichier (classe `Nom1_Nom2`)** qui implémente les méthodes suivantes (voir `ai.Random` pour un exemple):

```
- __init__(self)
- next_move(self, board)
```

**Pour l'évaluation, seul ce fichier sera pris en compte. Les signatures ne doivent pas être modifiées.**

## Quelques pistes pour vous aider

- Faites déjà une version fonctionnelle de votre minimax, ensuite avec élagage alpha-beta doté d'une fonction d'évaluation simple, puis améliorez la fonction pour rendre votre IA meilleur.
- Testez votre IA contre la version Random.
- Rusez pour tromper votre adversaire sans pour autant perdre en performance.