

# Intelligence artificielle

## Projet : Puissance 4

---

### Objectifs du projet

Implémenter l'algorithme  $\alpha - \beta$  pour le Puissance 4.

Écrire une fonction d'évaluation permettant de donner un score à une grille non terminale (avant la fin de partie).

Comprendre que l'IA s'intègre dans des architectures applicatives, elle n'est pas une fin en soi mais un moyen d'améliorer un traitement.

Coder des fonctionnalités dans un environnement déjà défini en partie

---

### Description du TP

Un squelette d'application Puissance 4 vous est fourni dans Moodle.

Le jeu est implémenté et intégralement jouable à 2 joueurs humains.

### Consignes

Étudiez l'architecture du squelette ainsi que les fonctions et classes déjà définis mais également celle qu'il reste à compléter. Normalement, je vous ferai un rapide tour du propriétaire en début de TP.

Quand une fonction existe mais que son corps est vide (ou qu'elle ne contient qu'un `return 0`) ce n'est pas une erreur, c'est sans doute que vous aurez besoin de remplir cette fonction par la suite, soyez vigilant. Ce sont des aides pour vous.

Lisez bien l'énoncé, il contient beaucoup d'informations et chaque mot est utile!

Aidez-vous de la documentation. Lorsqu'il utilise une nouvelle librairie, un développeur passe beaucoup de temps dans la documentation pour comprendre son fonctionnement. RTFM!

Utilisez Python à 100%, Python est fourni avec un grand nombre d'outils qui facilitent votre travail, utilisez-les!

Pensez **objet**, sachez utiliser les avantages de la programmation objet.

Coder proprement avec des noms de variables et de fonctions/méthodes bien choisis. Ajoutez des commentaires utiles à votre code afin de pouvoir encore le comprendre dans quelques jours.

Respecter les coding style. Je vous conseille de suivre les consignes du PEP8<sup>1</sup> ou le Google Python Styleguide<sup>2</sup>.

---

1. PEP8 : <https://www.python.org/dev/peps/pep-0008/>

2. Google Python Styleguide : <https://google.github.io/styleguide/pyguide.html>

Simple is beautiful. N'essayez pas de coder compliqué, restez simple, cela sera plus efficace et moins sujet aux erreurs.

Réfléchissez avant de coder, prenez un peu de temps pour dessiner/écrire vos idées sur une feuille. Le temps de la réflexion que vous prendrez, vous économisera énormément de temps de debug.  
Vous pouvez travailler en binôme.

Vous avez jusqu'au vendredi 23 janvier à 18h00 pour rendre votre projet qui devra être un zip contenant votre projet, le nom du zip (et du projet) devra contenir les noms des deux étudiants du binôme.

En plus, de votre code, vous écrirez un rapport présentant la stratégie que vous avez utilisé, notamment pour évaluer les grilles de puissance 4 incomplète.

## 1 Travail

1. Le travail consistera en l'implémentation de l' $\alpha - \beta$  pour le puissance 4 (déjà fait en TP pour le Morpion).  
⚠ Le niveau de difficulté d'une IA est le nombre de coups à l'avance qu'elle prévoit (i.e. la profondeur de votre arbre de jeu)
2. Il vous faudra également remplir la méthode eval de la classe Board qui vous permet de donner un score à une grille (un état de jeu) qu'il soit terminal (fin de partie) ou en cours de partie (comment évaluer le score d'une partie non-finie). Il s'agit de la partie la plus importante du Projet car c'est cette évaluation qui va déterminer la performance réelle de votre approche, la partie 'Intelligence' de votre IA.  
⚠ **Important :** Ne recopiez pas bêtement une solution que vous aurait craché une application d'IA générative, nous le verrons ...

## 2 Bonus pour les geeks

- Actuellement la difficulté de l'IA est limité par la puissance de calcul de votre machine (même si une IA de difficulté 6 nous met déjà à l'amende), introduisez une dose de calculs parallèles pour améliorer les résultats de vos implémentations de  $\alpha - \beta$ . Les processus sont vos amis.