

Intelligence artificielle

Projet : Puissance 4

Objectifs du projet

Implémenter les algorithmes de jeu minimax et alpha-beta.

Les appliquer au jeu Puissance 4.

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

Coder en Java dans un projet déjà défini en partie (le jeu avec des joueurs humains est déjà codé à 100%)

Description du TP

Un squelette d'application Puissance 4 vous est fourni sous forme d'un projet eclipse contenu dans le fichier `squelette_puissance4.zip` disponible dans Moodle.

L'interface utilisateur se fait pour l'instant par saisie clavier dans une console mais en observant le programme vous verrez que les bases sont prêtes pour une interface graphique.

Pas besoin d'implémenter le jeu en lui-même. Tout est déjà fait, et le jeu entre deux joueurs humains est fonctionnel.

Consignes

Étudiez l'architecture du squelette ainsi que les fonctions et classes déjà définis mais également celle qu'il reste à compléter.

Quand une fonction existe mais que son corps est vide (ou qu'elle ne contient qu'un `return null` ou 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.

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

Utilisez java à 100%, notamment pour les structures de données.

Utilisez les algorithmes présentés dans le cours.

Programmez proprement avec des noms de variables clairs et une indentation correcte sous peine de ne pas être noté! Je vous conseille fortement de suivre les règles de *coding style* appliquées chez Google : <https://google.github.io/styleguide/javaguide.html>.

Vous pouvez travailler en binôme.

Vous avez jusqu'au dimanche 6 mai à 23h50 pour rendre votre projet qui devra être un zip contenant votre projet eclipse, le nom du zip (et du projet eclipse) devra contenir les noms des deux étudiants du binôme.

Vous me présenterez votre projet lors de la première semaine de projet 2A, je vous donnerai un planning.

1 Travail

1. Le travail consistera surtout en l'implémentation des algos *minimax* et $\alpha - \beta$:
 - classe `Minimax.java`
 - classe `AlphaBeta.java`
- ⚠ 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 `evaluer` de la classe `Grille` 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).

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 *minimax* et $\alpha - \beta$. Les threads sont vos amis.
- Faire une interface graphique en JavaFX