

Intelligence Artificielle :

Les jeux de plateau à deux joueurs

Reversi

1 Présentation

Le but de ce projet est de développer un joueur de Reversi (version spéciale de Othello). Le joueur est une IA capable de jouer contre une autre IA, contre un joueur humain, etc.

Règles du jeu

Vous pouvez trouver les règles du jeu sur internet (wikipedia / Reversi). Par contre, nous allons jouer sur un plateau 10×10 pour pimenter un peu le jeu. La position de départ est celle de l' image ci-dessous. Noir commence. On ne peut poser une pièce que si on prend au moins une pièce adverse lors de la pose. Si on ne peut poser de pièce, il faut passer son tour. Si les deux joueurs passent leur tour à la suite, le jeu est terminé. Celui qui a le plus de pièces de sa couleur remporte la partie. En cas de doute, vous devez vous en tenir aux règles de jeu implémentées dans la classe `Board` du script `Reversi.py`.

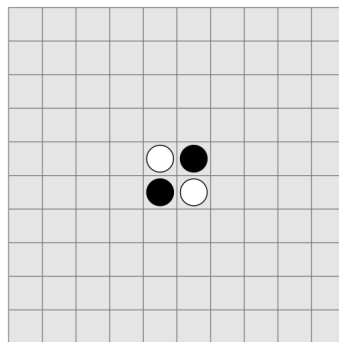


FIGURE 1 – Position des pièces au départ sur notre damier 10×10

2 Modalités

Les points seront donnés suivant le code que vous aurez rendu (expliquez vos méthodes dans le code!) et les techniques que vous aurez employées. Il faudra rendre, dans l'archive un fichier `README.txt` qui contiendra une description des points forts de votre joueur (faites court, listez les techniques, décrivez l'heuristique codée¹, précisez les structures de données, ...), Il indiquera également le nom de votre meilleure IA.

Dans un second temps, vous devez prendre en compte la contrainte de temps. Votre IA ne devra pas "réfléchir" plus que 10s. Il s'agit donc d'exploiter au maximum le temps imparti pour chercher à aller le plus loin possible dans l'arbre de jeux (pensez à l'`iterative deepening` vu en cours).

Vote code devra donc contenir une fonction `IAIterativeDeepening` paramétrée entre autre par le temps `time_limit` initialisé par défaut à 10.

Le projet sera à rendre via le Moodle qui sera associé à ce cours. Soyez donc vigilant aux instructions données via Moodle.

1. Notez que la classe `Board` du fichier `Riversi.py` contient déjà une heuristique très simple : elle retourne juste la différence entre le nombre de pièces d'ami et d'ennemi.