

AI - Connect Four

Vous savez comment y jouer, donc on va faire ça vite. Tour à tour vous mettez une pièce dans une colonne, elle tombe en bas et vous devez connecter 4 pièces pour gagner.

Suivant le code qui vous est fourni, vous devez créer une classe héritant de la classe *Player* nommée *PlayerMatricule*, par exemple *Player1666666*, située dans un fichier homonyme *PlayerMatricule.py* (ex. : *Player1666666.py*). Dans cette classe, vous devrez redéfinir la méthode *play* qui doit retourner un entier entre 0 et 6 **inclusivement**. Ce nombre représente la colonne dans laquelle vous voulez insérer votre pièce. Si vous retournez autre chose ou que vous tentez d'insérer dans une colonne qui est déjà pleine, vous êtes **disqualifiés** (ça veut dire 0 points pour cette question). Le tableau de jeu est passé en paramètre à la fonction, donc cela vous permet de prendre votre décision. **Attention : en aucun cas vous ne pouvez modifier le contenu du tableau.** Vous devez utiliser le tableau en lecture seule. Un exemple vous montre déjà ce que vous devez faire. N'oubliez pas d'ajuster le script *runme.py* afin qu'il utilise votre classe comme un des deux joueurs.

Si vous voulez tester votre AI, le code par défaut de la classe *Player* retourne une valeur au hasard : vous pouvez jouer contre-elle. Sinon, vous pouvez faire jouer votre AI contre elle-même et analyser ses décisions en regardant l'impression du déroulement de la partie à l'écran en faisant rouler le script *runme.py*.

Afin de vous évaluer, votre AI sera mise à l'épreuve contre ceux qui auront répondu à la question. Vous devrez aussi faire vos preuves contre quelques AI de test, l'AI aléatoire implémentée dans la classe de base *Player*.

Pour la remise, ne remettez que le fichier contenant la classe que vous avez créée.

C'est pas mal tout. Sauf que comme je trouvais qu'il question d'examen en moins d'une page c'est un peu plate, voici quelques de *memes* pour vous rendre heureux et vous mettre dans l'ambiance.

Bon succès!



