

Ressources: nim65s/connect 4

Fichier principal game.py  
classe cellule  
joueurs A et B

Classe grille

tableau à 2 dimension  $\rightarrow$  1<sup>er</sup> dim sont les lignes  
 $\rightarrow$  2<sup>em</sup> dim sont les colonnes

Méthode place appelée lorsqu'un joueur veut jouer dans une colonne donnée

self fait référence à l'instance de la classe

si volonté de remplir une colonne pleine alors levée d'exception

Méthode win renvoie un booléen si un joueur a gagné à VRAI

# TODO Vertical

# TODO Diagonal

Méthode tie: match nul

# TODO

Class player: abstraite  $\Rightarrow$  faire une classe dérivée pour implémenter  
Méthode play prend en entrée l'état actuel de la grille  
le joueur envoie un entier



Classe Game : déclare les joueurs et la grille

L> Méthode play (à ne pas confondre avec play associé à la classe player) prend en paramètre un joueur et la grille

Lancer le programme :

Le fichier `--main--.py` crée une instance du jeu et instance un joueur A et un joueur B  
`game.main()` lance

Lancer des tests unitaires : `connect4\tests`

Le fichier `test_game.py` vérifie l'affichage de la grille  
Lancé avec la commande `py -m unittest` sans Windows

Objectif décrit dans le README

- ① Commenter le décorateur `@unittest.skip` qui passe le test de l'IA  $\Rightarrow$  maintenant il s'exécute  
 $\Rightarrow$  il faut implémenter une IA classe fille de `Player` qui joue la première case vide dans `dumb-IA.py` jusqu'à ce que le test fonctionne à nouveau
- ② Commenter le décorateur par le test de win si colonne remplie et implémenter la colonne gagnée
- ③ Commenter le décorateur par le `grid.tie` (jeu nul) et l'implémenter