

Séance 5: MORPION

Université Paris-Diderot

On représente une grille du jeu morpion¹ par un tableau de longueur 3 dont les éléments sont des tableaux d'entiers de longueur 3. Les cases vides de la grille sont des 0, les cases prises par le joueur "X" sont des 1 et celles prises par le joueur "O" des 2. Par exemple, la grille

○		
×		○
	×	

est représentée par $\{\{2,0,0\},\{1,0,2\},\{0,1,0\}\}$.

Remarque:

Vous rédigerez les réponses aux exercices dans le fichier fourni `Exos.java`. Il est important de **tester** au fur et à mesure votre code, à l'aide des tests fournis dans le `main`.

Exercice 1 (Tester, **)

Écrivez une fonction `isStartingGrid` qui prend un tableau de tableaux d'entiers en entrée, et renvoie `true` si ce tableau représente une grille de morpion dont toutes les cases sont vides, et `false` sinon. □

Exercice 2 (Jouer, **)

Écrivez une procédure `move` qui prend en entrée :

- une grille de morpion² `grid`,
- un entier `line` entre 0 et 2,
- un entier `column` entre 0 et 2 et
- un entier `player` entre 1 et 2,

qui vérifie que la case `grid[line][column]` de la grille est vide, et qui la remplit avec `player`. Si la case n'est pas vide, `move` affiche le message `Illegal move`. □

Exercice 3 (Victoire, ***)

Écrivez une fonction `winner` qui prend une grille de morpion et renvoie l'entier correspondant au vainqueur de la partie, si vainqueur il y a, et qui renvoie 0 sinon. □

Exercice 4 (Bonus, ***)

Écrivez une procédure `play`, sans paramètres, pour permettre de jouer une partie de morpion (il faut inviter les joueurs, chacun à son tour, à jouer et terminer dès qu'il y a un vainqueur ou que la grille est pleine). □

1. <https://fr.wikipedia.org/wiki/Tic-tac-toe>

2. C'est à dire un tableau de tableaux d'entiers ayant les dimensions requises, et rempli de manière conforme aux règles du jeu.