

## P\_NSInf3 – Le puissance 4

Le *Puissance 4* est un jeu bien connu à deux joueurs qui se joue sur une grille verticale de six lignes et sept colonnes. A tour de rôle, chaque joueur fait tomber un pion de sa couleur dans une colonne de son choix non encore pleine. Le premier joueur qui aligne quatre pions de sa couleur, horizontalement, verticalement ou en diagonal, gagne la partie. La partie est nulle si la grille est totalement remplie sans qu'aucun joueur ne gagne.

Pour représenter une grille de ce jeu, on utilise un tableau à deux dimensions de taille 6x7, la première dimension représentant les lignes et la seconde les colonnes. Une ligne est notée  $l$  et prend une valeur entre 0 et 5, la ligne 0 étant située en bas ; une colonne est notée  $c$  et prend une valeur entre 0 et 6, la colonne 0 étant située à gauche. Un joueur est noté  $j$  et prend la valeur 1 ou 2. Dans une grille, notée  $g$ , la valeur 0 représente une case vide et la valeur 1 ou 2 représente un pion du joueur correspondant.

1. Ecrire une fonction `grille_vide()` qui renvoie une grille vide.
2. Ecrire une fonction `affiche(g)` qui affiche une grille, avec le caractère `.` pour une case vide, le caractère `X` pour le joueur 1 et le caractère `O` pour le joueur 2. On prendra soin de bien afficher les lignes de haut en bas.
3. Ecrire une fonction `coup_possible(g,c)` qui renvoie un booléen indiquant s'il est possible de jouer dans la colonne  $c$ .
4. Ecrire une fonction `jouer(g,j,c)` qui joue un coup du joueur  $j$  dans la colonne  $c$ , en supposant que la colonne  $c$  n'est pas pleine.
5. Ecrire trois fonctions `horiz(g,j,l,c)`, `vert(g,j,l,c)` et `diag(g,j,l,c)` qui déterminent respectivement s'il y a un alignement de quatre pions du joueur  $j$  à partir de la case  $(l,c)$
6. Ecrire une fonction `victoire(g,j)` qui renvoie un booléen indiquant si le joueur  $j$  a gagné.
7. Ecrire une fonction `match_nul(g)` qui renvoie un booléen indiquant s'il y a match nul, c'est-à-dire si la grille est totalement remplie. Indication : on peut se contenter d'examiner la ligne du haut.
8. Ecrire une fonction `coup_aleatoire(g,j)` qui joue un coup aléatoire légal pour le joueur  $j$ , en supposant que la grille n'est pas pleine. On prendra soin de ne pas favoriser de colonne particulière.
9. Ecrire un programme qui fait jouer deux adversaires aléatoirement à tour de rôle, en affichant la grille après chaque coup, et qui s'arrête dès qu'un joueur gagne ou que la partie est nulle.
10. Ecrire enfin un programme qui permet à l'utilisateur de jouer contre un adversaire qui joue aléatoirement, en affichant la grille après chaque coup et le résultat en fin de partie.