

# Construction d'un jeu de morpion

vincent.thomas@loria.fr

June 25, 2013

## 1 Création du jeu de Morpion

Un jeu de morpion sera représenté par une liste contenant des caractères. Chaque position dans la liste désignera une case du plateau de jeu (cf figure 1(a)).

Les caractères utilisés seront '.' pour une case vide, 'X' pour une case occupée par le joueur 1 et 'O' pour une case occupée par le joueur 2.

Par exemple la liste ['.', 'X', '.', 'O', '.', '.', '.', '.', 'X'] désigne le jeu présenté figure 1(b).

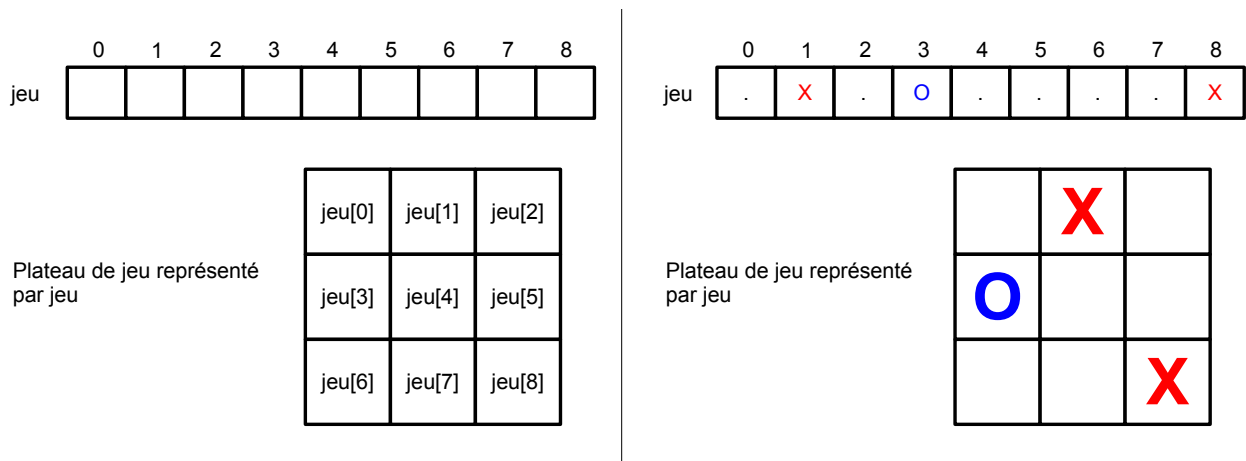


Figure 1: Représentation d'un plateau de jeu (a) plateau vide (b) plateau avec des pièces

### 1.1 Fonction nouveau jeu

Ecrire une fonction `nouveau()` qui retourne un jeu vide, c'est à dire une liste de 8 cases contenant '.' dans chaque case.

Par exemple

```
1 >>> coup=nouveauJeu()
2 >>> coup
3 ('.', '.', '.', '.', '.', '.', '.', '.')
4 >>> affiche(coup)
5 . . .
6 . . .
```

```
7 . . .
```

## 1.2 Fonction affiche

On vous donne la fonction `affiche()` capable d'afficher un jeu passé en paramètre.

```
1 #permet d'afficher un jeu
2 def affiche(jeu):
3     print(jeu[0], jeu[1], jeu[2])
4     print(jeu[3], jeu[4], jeu[5])
5     print(jeu[6], jeu[7], jeu[8])
6
7 print("affiche jeu vide")
8 affiche(nouveauJeu())
```

## 1.3 Détection joueur

Ecrire une fonction `joueur(jeu)` qui permet de savoir qui doit jouer à partir d'un jeu passé en paramètre. Il suffit simplement de compter les 'X' et 'O' et de retourner 'X' ou 'O' en fonction de ces nombres.

Comme le joueur 'X' commence, si le nombre de 'X' est égal au nombre de 'O' c'est à lui de jouer. Si par contre, le nombre de 'X' est supérieur de 1 au nombre de 'O', c'est que c'est au second joueur de placer un pion.

Par exemple

```
1 >>> jeu=nouveauJeu()
2 >>> joueur(jeu)
3 'X'
4 >>> jeu=('.', 'X', '.', '.', '.', '.', '.', '.', '.')
5 >>> joueur(jeu)
6 'O'
```

## 1.4 Coups possible

Ecrire une fonction `coupPossibles(jeu)` qui retourne les coups possibles à partir d'un jeu passé en paramètre. Il s'agit de retourner la liste des indices des cases possédant '.'.

Par exemple

```
1 >>> jeu=nouveauJeu()
2 >>> coupPossible(jeu)
3 [0, 1, 2, 3, 4, 5, 6, 7, 8]
4 >>> jeu=('.', 'X', '.', '.', 'X', '.', '.', 'O', '.')
5 >>> coupPossible(jeu)
6 [0, 2, 3, 5, 6, 8]
```

## 1.5 Gagné/Perdu

La fonction suivante `gagner(jeu)` permet de savoir qui a gagné ou perdu. Cette fonction est écrite rapidement (on pourrait avoir une fonction plus générique en utilisant des boucles - par exemple pour le puissance 4 - mais ce n'est pas l'objectif de cet atelier).

Cette fonction retourne

- 1 si le joueur 1 'X' a gagné
- -1 si le joueur 2 'O' a gagné
- 0 si aucun joueur n'a gagné (qu'il reste des coups possibles ou non)

```
1 #permet de savoir si un joueur remporte la partie
2 def gagner(jeu):
3     res='.'
4     if (jeu[0]!='.' and jeu[0]==jeu[3]==jeu[6]):
5         res=jeu[0]
6     if (jeu[1]!='.' and jeu[1]==jeu[4]==jeu[7]):
7         res=jeu[1]
8     if (jeu[2]!='.' and jeu[2]==jeu[5]==jeu[8]):
9         res=jeu[2]
10    if (jeu[0]!='.' and jeu[0]==jeu[1]==jeu[2]):
11        res=jeu[0]
12    if (jeu[4]!='.' and jeu[4]==jeu[5]==jeu[3]):
13        res=jeu[3]
14    if (jeu[7]!='.' and jeu[7]==jeu[8]==jeu[6]):
15        res=jeu[6]
16    if (jeu[0]!='.' and jeu[0]==jeu[4]==jeu[8]):
17        res=jeu[0]
18    if (jeu[2]!='.' and jeu[2]==jeu[4]==jeu[6]):
19        res=jeu[2]
20    if (res=='O'):
21        return(-1)
22    else:
23        if res=='X':
24            return(1)
25    return(0)
```

## 1.6 fonction Jouer

Ecrire la fonction `jouer(jeu,coup)` qui permet de jouer un coup et qui retourne le nouveau jeu.

Cette fonction prend en paramètre un jeu et un numéro de coup. Elle appelle la fonction `joueur` pour savoir quel est le joueur qui a joué et elle ajoute la bonne pièce à l'endroit joué sur **une copie** du jeu.

Par exemple

```

1 >>> jeu=nouveauJeu()
2 >>> jeu2=jouer(jeu,1)
3 >>> jeu
4 ('.', '.', '.', '.', '.', '.', '.', '.', '.')
5 >>> jeu2
6 ('.', 'X', '.', '.', '.', '.', '.', '.', '.')
7 >>> jeu2=jouer(jeu2,5)
8 >>> jeu2
9 ('.', 'X', '.', '.', '.', 'O', '.', '.', '.')
10 >>> jeu2=jouer(jeu2,0)
11 >>> jeu2
12 ('X', 'X', '.', '.', '.', 'O', '.', '.', '.')

```

## 1.7 Bilan

Grâce à ces fonctions, nous avons défini le jeu de morpion.

- nouveauJeu permet de créer une situation de jeu initiale
- coupPossible encode les règles du jeu qui donne les coups à une situation donnée
- jouer encode les règles du jeu qui traduisent l'évolution du jeu
- gagner encode les règles de victoire