

---

**ALGO2 – Algorithmique et Programmation en Python 2**

---

**Fiche de TP numéro 3****Le jeu Puissance Quatre**

Voici encore un jeu de plateau : le jeu Puissance 4. Deux joueurs s'affrontent sur un plateau de jeu vertical : ils glissent à tour de rôle un pion, marqué d'un rond ou d'une croix selon le joueur. Lorsqu'un joueur pose un pion dans une colonne, le plateau étant vertical, le pion tombe jusqu'à toucher le fond de la colonne ou un autre pion déjà présent dans la colonne.

L'objectif pour chaque joueur est d'aligner (horizontalement, verticalement ou en diagonale) quatre pions lui appartenant.

Nous allons reprendre la structure du jeu du taquin que nous avons vu en cours et qui est disponible sur Moodle pour l'adapter au jeu de puissance 4. Récupérez ce fichier si vous ne l'avez pas déjà fait, et renommez le `puissance4.py`. Nous allons réécrire la classe `Taquin` en une nouvelle classe `PuissanceQuatre`.

Le programme principal est maintenant :

```
### script principal
if __name__ == '__main__':
    p4 = PuissanceQuatre()
    joueur = random.randint(0,1)
    while not p4.partie_finie():
        print(p4)
        print("Aux",p4.carac(joueur),"de jouer")
        choix = int(input("Dans quelle colonne voulez-vous jouer ? "))
        res = p4.pose_colonne(choix,joueur)
        if not res :
            print("Non, ce n'est pas possible.")
        else :
            joueur = (joueur+1)%2
    print(p4)
    print("Partie finie. Bravo")
```

On remarque que la variable `joueur` qui représente un joueur varie entre 0 et 1. On observe également que la classe `PuissanceQuatre` possède au moins les méthodes suivantes :

- un constructeur
- une méthode `partie_finie` : détecte la fin de partie
- une méthode `carac(joueur)` : renvoie la marque sur le pion du joueur (un O ou un X)
- une méthode `__str__` : pour afficher le plateau de jeu
- une méthode `pose_colonne(choix, joueur)` : pour faire *tomber* le pion de joueur dans la colonne `choix`. Cette méthode renvoie faux si `choix` ne correspond pas à un numéro de colonne ou si la colonne est déjà pleine.

Nous allons en ajouter quelques unes (toutes très simples).

### Exercice 1 : Le constructeur

Un `PuissanceQuatre` est caractérisé par

- un nombre de lignes (par défaut, 6)
- et un nombre de colonnes (par défaut, 7)
- un plateau de jeu `self._plateau`, dont les cases contiennent soit `-1` si la case est vide, `0` si elle contient un pion du joueur 0, `1` si elle contient un pion du joueur 1.
- une liste `self._carac` qui contient juste les deux caractères `'O'` et `'X'`. Cette liste n'est utile que pour rendre l'affichage du jeu agréable aux joueurs.
- les coordonnées de la dernière case remplie `self._dernier_coup`. Elles sont initialisées à `(-1, -1)`.

Modifiez le constructeur du `Taquin` pour qu'il devienne un constructeur de `PuissanceQuatre`.

### Exercice 2 : Méthode `carac`

Spécifiez puis écrivez la méthode `carac(joueur)` qui renvoie la marque sur le pion du joueur (un `O` ou un `X`). *Quel attribut cette méthode utilise-t-elle ?*

### Exercice 3 : Affichage du plateau de jeu

Inspirez-vous du dialogue avec le joueur mis en annexe pour faire modifier l'affichage du `Taquin` en un affichage de `PuissanceQuatre`. Vous remarquerez que pour aider les joueurs, le numéro des colonnes est indiqué sur la première ligne d'affichage.

### Exercice 4 : Colonne valide

Spécifiez puis écrivez la méthode `colonne_valide` qui vérifie que le numéro de colonne donné en paramètre est valide : il correspond bien à un indice de colonne, et il y a au moins une case libre dans cette colonne.

### Exercice 5 : Poser un pion

Spécifiez puis écrivez la méthode `pose_colonne(choix, joueur)` qui fait *tomber* le pion de joueur dans la colonne `choix`. Cette méthode renvoie `faux` si `choix` n'est pas valide. N'oubliez pas de mémoriser dans `self._dernier_coup` les coordonnées de la case dans laquelle le pion arrive.

Dans un premier temps, votre méthode `partie_finie` peut renvoyer `False` systématiquement. Vous devez pouvoir tester votre jeu !

### Exercice 6 : Coordonnées valides

Spécifiez puis écrivez la méthode `coords_valide` qui vérifie que le numéro de ligne et de colonne donnés en paramètre sont des coordonnées valides : ils correspondent bien à un indice de ligne et un indice de colonne.

### Exercice 7 : Compter les pions

Écrivez la méthode `compte_valeur` :

```
def compte_valeur(self, lig, col, val, inc_lig, inc_col) :
    '''PuissanceQuatre, int, int, int, int, int -> int
    retourne le nombre de cases de valeur val à partir de (lig,col)
    dans la direction (inc_lig, inc_col)
    '''
```

La case (lig, col) ne doit pas être comptée. Ainsi, un appel de la forme :

```
self.compte_valeur(lig, col, 0, 1, -1)
```

comptera le nombre de pions de valeur 0 présents dans le tableau de manière consécutive dans la demi-diagonale qui part de (lig, col) et prend une direction "sud-ouest" (vers le coin bas gauche).

**Exercice 8 : Partie finie !** Il ne reste plus qu'à spécifier et écrire la méthode `partie_finie` à l'aide des deux méthodes précédentes.

---

## Annexe

Voici un exemple d'interaction avec l'utilisateur qui montre l'affichage du plateau à différentes étapes.

```
moi@gavotte:~/enseign/algo2/$ python3 puissance4.py
```

```
 0 1 2 3 4 5 6
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
```

Aux X de jouer

Dans quelle colonne voulez-vous jouer ? 2

```
 0 1 2 3 4 5 6
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
+-+--+--+--+--+
| | | | | | |
```

```

+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |X|  |  |
+--+--+--+--+--+

```

Aux O de jouer

Dans quelle colonne voulez-vous jouer ? 3

```

0 1 2 3 4 5 6
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |X|O|  |
+--+--+--+--+--+

```

Aux X de jouer

Dans quelle colonne voulez-vous jouer ? 3

```

0 1 2 3 4 5 6
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |  |  |
+--+--+--+--+--+
|  |  |  |X|  |
+--+--+--+--+--+
|  |  |X|O|  |
+--+--+--+--+--+

```

Aux O de jouer

Dans quelle colonne voulez-vous jouer ? ...