TP 1 - Puissance 4

Ferdinand Mom

May 7, 2019

Consignes de rendu

A la fin de ce TP, vous devez rendre une archive respectant l'architecture suivante:

```
|___ toolbox/
     |___ toolbox.py
|___ board.py
|___ main.py
|___ player.py
|___ tp_1_puissance_4.pdf
```

Pour faire ce TP, veuillez créer votre environnement virtuel en utilisant **virtualenv** puis l'activer. (cf le **Survival Kit > CheatSheet > Virtualenv**).

Ensuite, lancez le script **toolbox > toolbox.py**

Vous pouvez désormais commencer le TP. Bonne chance!

1 Introduction

L'objectif de ce TP est de créer un Puissance 4. Pour ceux qui ne savent pas ce que c'est, voici à quoi cela ressemble.

Rappel rapide des règles:

- Le puissance 4 est un jeu de tour par tour (2 joueurs).
- Pour gagner une partie de puissance 4, il suffit d'être le premier à aligner 4 jetons de sa couleur horizontalement, verticalement et diagonalement.
- Si lors d'une partie, tous les jetons sont joués sans qu'il y est d'alignement de jetons, la partie est déclaré nulle.

Désormais, vous avez toutes les règles en tête, le TP peut donc commencer !

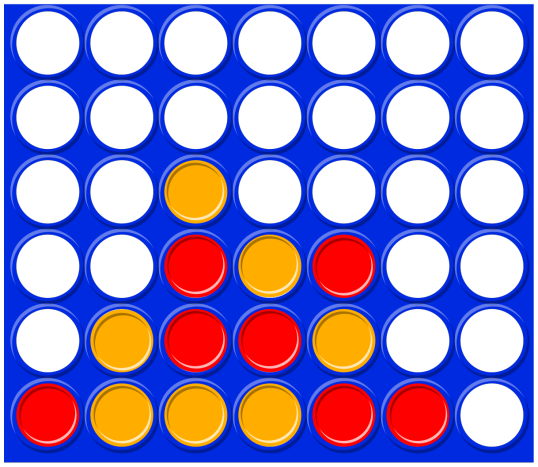


Figure 1: Partie de Puissance 4

2 Essentiel

Créez le constructeur de **Board.py**

```
__init__(self, row, col)
```

Maintenant écrivez la fonction suivante.

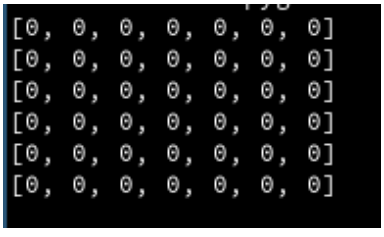
```
drawBoard(self)
```

Elle doit retourner le tableau de jeu (boardGame).

Ecrire la méthode statique suivante:

```
prettyPrint(boardGame)
```

Elle doit afficher le tableau de jeu de manière plus jolie dans la console.



3 Player.py

Créez le constructeur de **Player.py**

```
__init__(self, boardGame, coin)
```

Remplissez la fonction **putCoin()**.

```
putCoin(self, screen, index, col)
```

La fonction doit retourner True si la colonne est pleine. Sinon, vous devrez placer la pièce et retourner False.

Pour dessiner une pièce d'une certaine couleur, vous pouvez utiliser le code suivant:

```
pygame.draw.circle(screen, color, (col * 100 + 50, i  * 100 + 50),
                                  radius)
```

Les paramètres sont les suivants:

- **screen** est l'écran généré par pygame.
- **color** est soit le RED ou YELLOW en valeur RGB.
- **radius** est le rayon du cercle. Il doit être égal à 40.

4 Board.py

Remplissez la fonction **checkHorizontal()**:

```
checkHorizontal(self, boardGame, playerCoin, col)
```

Remplissez la fonction **checkVertical()**:

```
checkVertical(self, boardGame, playerCoin, col)
```

Remplissez la fonction **checkDiagonal()**:

```
checkDiagonal(self, boardGame, playerCoin, col)
```

5 Main.py

Désormais, remplissez le **main.py**.
Utilisez la fonction suivante pour mettre à jour l'écran de pygame.

```
pygame.display.update()
```

True knowledge exists in knowing that you know nothing.

- Socrates