

## Puissance 4

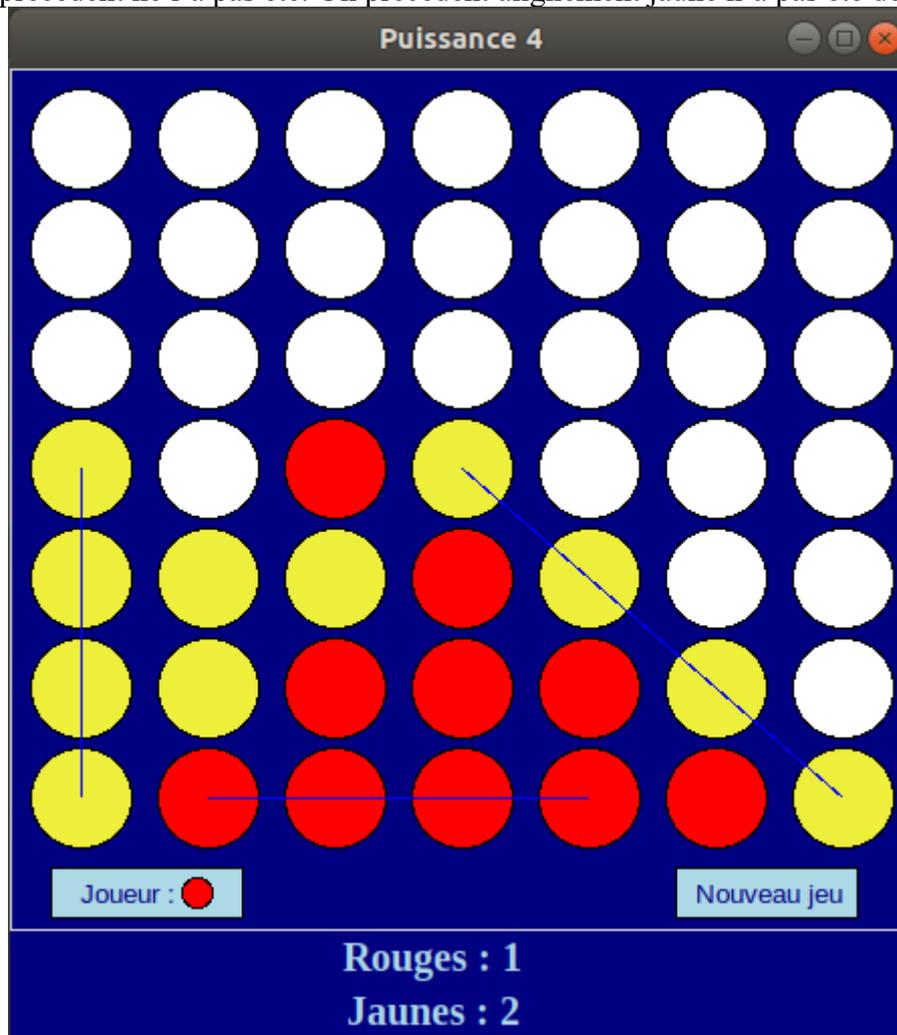
Bonne réalisation : le programme fonctionne, on l'utilise simplement avec la souris, l'indication du joueur en cours est claire avec la couleur.

La possibilité de jouer après un 1<sup>er</sup> alignement est intéressante.

Il reste néanmoins quelques problèmes :

1) Détection des alignements incertaine.

Sur l'image ci-dessous, 2 alignements n'ont pas été détectés : le jaune oblique l'a été alors que le rouge oblique précédent ne l'a pas été. Un précédent alignement jaune n'a pas été détecté.



Il faut chercher la cause dans le programme, et aussi signaler le problème comme non résolu.

```
for x in (1,-6,8):  
    if n in self.listerouge: # en s'assurant qu'elles ne sont trop près des bords (pour ne pas arriver de l'autre côté du plateau)  
        if n % 7 != 6 and n+x in self.listerouge:  
            if n % 7 != 5 and n+ 2*x in self.listerouge:  
                if n % 7 != 4 and n + 3*x in self.listerouge:  
                    liste = [n, n+x, n+2*x, n+3*x]  
                    if self.gagnantes(liste) : self.win("rouges", liste[0],liste[3])  
                    return  
            if n%7 > 0 and (n-x) in self.listerouge:  
                liste = [n-x,n, n+x, n+2*x]  
                if self.gagnantes(liste) : self.win("rouges", liste[0],liste[3])  
                return  
        if n%7 > 1 and (n-x) in self.listerouge:  
            if n%7 > 2 and n-(2*x) in self.listerouge:  
                liste = [n-2*x, n-x,n, n+x]  
                if self.gagnantes(liste) : self.win("rouges", liste[0],liste[3])  
                return
```

Cette méthode de détection des alignements obliques est trop compliquée.

Pourquoi cette boucle : `for x in (-1,6,-8)` ? Je crois comprendre qu'en ajoutant 6 au numéro de la case jouée, on examine la case située en dessous et à gauche de la case jouée ; en ajoutant -8, on examine la case située au-dessus et à gauche de la case jouée. Avec -1 on examine la case de gauche... La boucle suivante `for x in (1,-6,8)` examine les alignements obliques inverses. Dans ces boucles `for` écrite, on examine une imbrication de cas trop complexe qui, je pense, ne prend pas tous les cas d'alignement en compte. Il faut absolument simplifier cela. Mais cela demande réflexion : vous avez une liste de cases remplies et vous cherchez si l'adjonction d'une nouvelle case ajoute un alignement.

D'une façon générale, j'ai l'impression que la structure du programme a été copiée et partiellement améliorée/modifiée/simplifiée. Mais comme elle était trop complexe, elle fut mal comprise et les corrections nécessaires n'ont pas pu être effectuées.

L'adjonction du nom du joueur en cours, ce qui ne semble pas une amélioration difficile à réaliser, n'a pas pu être réalisée pour cette raison. L'image incluse dans le compte-rendu montre que sur la console le joueur affiché est Yannick, soit le joueur2, ce qui correspond à  $j=1$ . Mais sur la fenêtre du jeu, le rectangle dit que le joueur est Rayanne... Il faut chercher où s'affiche le nom du joueur dans la fenêtre. Malheureusement, vous avez supprimé votre création du rectangle où s'affichait (mal) le joueur. Je ne peux vous aider maintenant sur ce point mais c'était une question facile.

L'instruction `self.joueur = [0,1][[0,1].index(self.joueur)-1]` qui permet d'alterner dans la variable `self.joueur` le numéro du joueur est compliquée : il suffit de faire `self.joueur = (self.joueur + 1) % 2`

Vous auriez pu vous inspirer de ce qui est fait pour les couleurs :

```
self.color = ["red", "#EDEF3A"] dans l'initialisation
```

et créer une liste :

```
self.nom = [joueur1, joueur2] où joueur1 et joueur2 seraient des arguments passés à la création de l'objet lecan = Can(nom_1, nom_2)... Pour cela, il aurait fallu vous renseigner sur « comment créer une classe » et « comment passer des arguments », mais cela n'a pas été fait.
```

Projet de note : 12/20

Détail de la note :

- préprojet : 1.5/2
- compte-rendu : 3.5/4 (relativement honnête, illustré)
- Programme : 7/14 (fonctionnel mais alternance des joueurs souhaitée mais pas implémentée, pas très original (sans doute plagia), reconnaissance des alignements compliquée avec erreur, utilisation d'une classe et de `tkinter`)