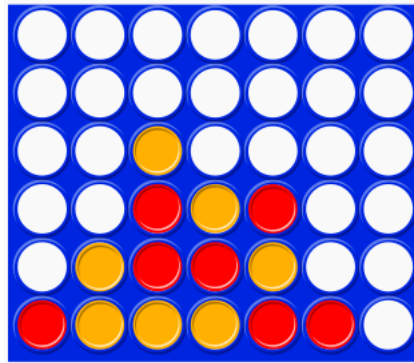

Jeu du puissance 4



Un début de partie de Puissance 4.

Règles du jeu

Le but du jeu est d'aligner une suite de 4 pions de même couleur sur une grille comptant 6 rangées et 7 colonnes.

Chaque joueur dispose de 21 pions d'une couleur (par convention, en général jaune ou rouge). Tour à tour, les deux joueurs placent un pion dans la colonne de leur choix, le pion coulisse alors jusqu'à la position la plus basse possible dans ladite colonne à la suite de quoi c'est à l'adversaire de jouer.

Le vainqueur est le joueur qui réalise le premier un alignement (horizontal, vertical ou diagonal) consécutif d'au moins quatre pions de sa couleur.

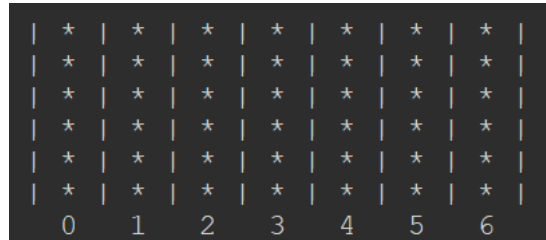
Si, alors que toutes les cases de la grille de jeu sont remplies, aucun des deux joueurs n'a réalisé un tel alignement, la partie est déclarée nulle.

Les jetons sont tous sortis de la grille et la partie continue sur de nouvelles bases jusqu'à ce qu'un des joueurs parvienne à aligner quatre pions.

Extrait de Wikipédia - https://fr.wikipedia.org/wiki/Puissance_4

Le cahier des charges

- Mettez au point le jeu du puissance 4 en utilisant le langage Python.
- Le but est de prendre en compte les règles du jeu présentées précédemment
- Le projet se fera en binôme. Donner vos noms au professeur quand le binôme est formé.
- Le rendu visuel sera un affichage type texte tels que présenté ci-dessous :



*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
*	*	*	*	*	*	*
0	1	2	3	4	5	6

- Analysez les problèmes à résoudre et décomposez-les en sous-problème jusqu'à en faire des tâches simples intégrables dans une fonction
- Une fonction doit faire une action et une seule
- Il est tout à fait possible de schématiser le programme principal par un algorithme.
- Le professeur passera voir chaque binôme régulièrement. Chaque binôme doit être dans la capacité de montrer : leur analyse du problème et de sa décomposition, l'état d'avancement de chaque tâche, présenter leurs difficultés, poser des questions ...
- Vous utiliserez (à minima) les fonctions suivantes :

```
def ajouter_jeton(grille:list, colonne:int, jeton:str)->bool:  
  
def verifier_victoire(grille:list, jeton:str)->bool:  
  
def reinitialiser_grille(grille:list)->None:  
  
def jouer()->None:
```

- Vous utiliserez les variables suivantes :

```
jetons_restants = [21, 21] # Index 0 : Joueur X, Index 1 : Joueur O  
joueur_actuel = "X"  
index_joueur = 0 # Index 0 pour "X", 1 pour "O"
```

Rendus attendus

- Le programme Python nommé avec les deux noms de familles séparés par un tiret (-)
Exemple : dupond-leblanc.py
- Un rapport (Par binôme)

Bonus

Si toutes les fonctionnalités sont implémentées et que votre rapport est terminé (et pré-contrôlé par le professeur) vous pouvez envisager d'ajouter :

- Analyse des parties précédentes

Enregistrer les parties jouées : Exporter la grille : Sauvegarde de la grille dans un fichier texte.

- Design personnalisé

Permettre aux joueurs de personnaliser le jeu :

- Choix des symboles : Remplacer les jetons "X" et "O" par des émojis ou des caractères au choix.
- Thèmes visuels : Modifier les couleurs ou l'affichage de la grille pour une expérience plus immersive.
- Mode chronométré

Ajouter une contrainte de temps pour chaque coup :

- Si un joueur dépasse la limite (par exemple 10 secondes), il perd son tour ou la partie.

Les parties (à minima) du rapport (un rapport par binôme)

Partie 1 : Décrivez la décomposition des tâches au sein de votre binôme et la méthode de travail.

Qui fait quoi ? Organisation temporelle (planning), moments d'échanges ...

Partie 2 : Présentez le projet en sous-problème. Présenter les tâches permettant de résoudre chaque sous-problèmes. L'idée est de **décrire l'algorithme général du jeu**. Pour cela, vous pourrez décrire l'algorithme en français (avec des verbes d'actions facilement traduisible et avec indentation) ou illustrer votre propos par un algorithme.

Partie 3 : Présentez l'interaction que vous avez programmé.

Expliquez quels éléments de votre programme permettent une interaction avec l'utilisateur et comment ?

Partie 4 : Quel processus de mise au point et de validation de programmes avez-vous utilisé ?

Quels outils et méthodes de tests ?

Objectifs intermédiaires (pour le mardi 17 décembre)

Vous donnerez au professeur :

- L'ensemble des tests unitaires de l'équipe
- L'algorithme ou l'algorithme décrit en français (avec des verbes d'actions facilement traduisible et avec indentation) - du programme principal

Le barème

Fonctionnalités (8 points)

- Palier 1 (0~2/8) : Le programme correspond à peu près aux spécifications demandées
- Palier 2 (2~4/8) : Le programme correspond aux spécifications demandées
- Palier 3 (4~6/8) : Le programme ne présente aucun bug et correspond aux spécifications demandées
- Palier 4 (6~8/8) : Des fonctionnalités supplémentaires ont été ajoutées

Algorithmes et code (6 points)

- Utilisation pertinente de structures conditionnelles, de boucles, de variables, etc.
- Utilisation correcte des différents types de valeurs et des fonctions de conversion quand nécessaire
- Décomposition fonctionnelle pertinente (chaque fonction est suffisamment petite, on évite la répétition de code)

Style du code (2 points)

- Utilisation de noms pertinents pour les variables et les fonctions
- Documentation pertinente des fonctions grâce à des docstrings
- Utilisation pertinente des commentaires

Rédaction du rapport (4 points)

Au travers du compte-rendu, les critères d'évaluation seront :

- L'organisation ;
- L'analyse des problèmes à résoudre ;
- La clarté de l'expression et des illustrations ;
- L'explicitation des méthodes et des choix effectués ;
- L'identification et le réinvestissement de notions vues en cours ;
- L'analyse critique des solutions.