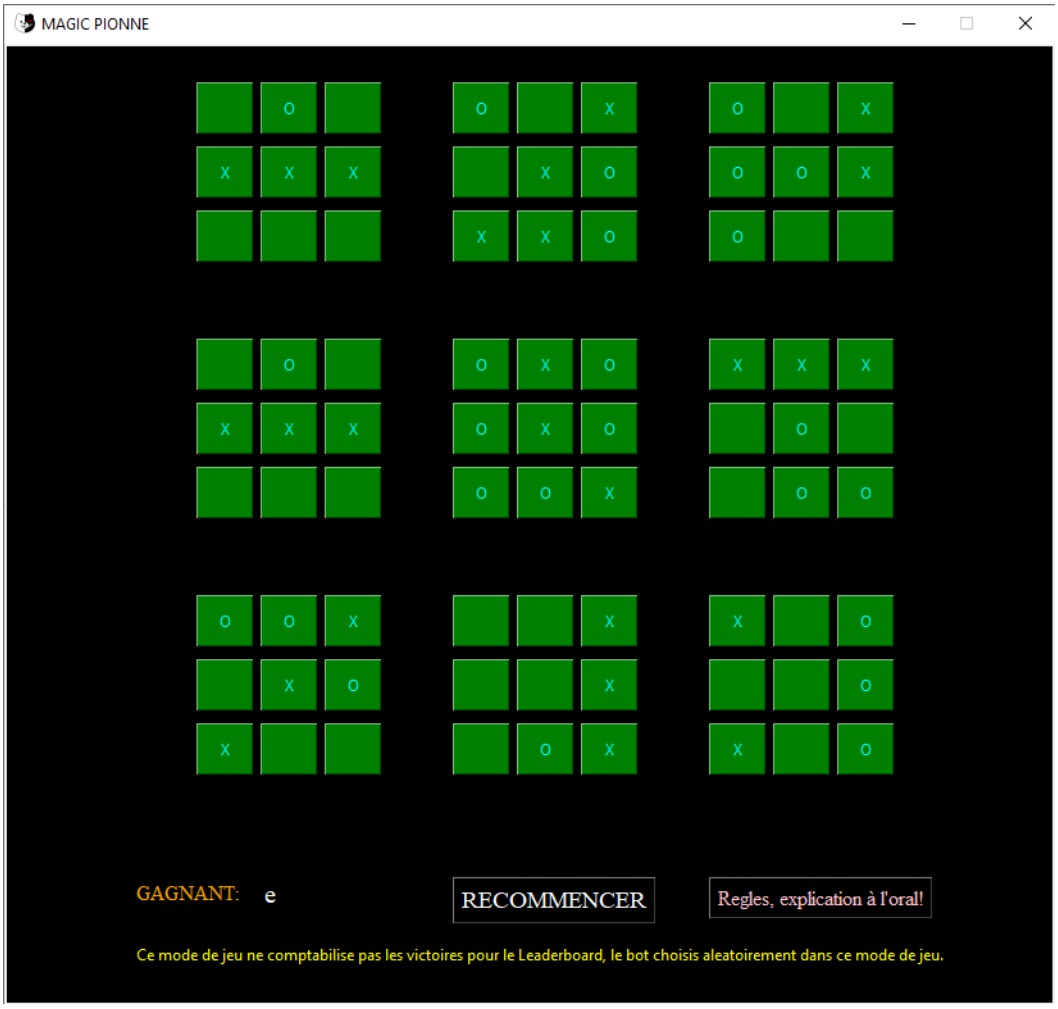
MORPIONNE

IPSA

AERO 1



Professeur référent : **Mr. Touati Lyes**

Promo: **2024**
Classe: **1 PR**
Grand projet programmation

INTRO :

Le but était de programmer un jeu du Morpion (tic tac toe) en python.

Cahier des charges :

- *Réaliser au moins 2 niveaux de difficultés différentes (1 facile => random, un un peu plus difficile).*
- *Traiter des données en Json.*

Notre groupe se compose de :

Chaouki Zakaria; Abdel-Hafez Talaat ; Pain Thomas

Structure de notre Morpion:

- 6 Niveaux de jeu
- Un mode de jeu à part entière
- Un tableau des scores OFF-LINE via Json
- Un leaderboard ON-LINE
- Un login system ON-LINE
- Deux interfaces tkinter + 4 TopLevel

I- Composition Première fenêtre :

Sur cette page on peut : sélectionner le niveau + le nombre de round que l'on souhaite joue
On peut observer nos victoires pour chaque niveau ainsi que le TOP3 des joueurs en termes de victoire.
On peut également ouvrir le mode de jeu à part entière

II- Les différents niveaux :

- **Facile :** Dans ce mode de jeu le bot répond de façon totalement aléatoire.
- **Moyen :** Dans ce mode de jeu le bot essayera au moins de récupérer le centre quand il le peut sinon les diagonales seront généralement privilégiées.
- **Difficile :** Dans ce mode de jeu, le bot adopte le même comportement que le Moyen a l'exception que si il aligne 2 cases il cherchera toujours à saisir sa chance d'aligner les 3.
- **Extrême :** Dans ce mode de jeu le bot cherche dans un premier temps à se défendre si l'utilisateur est prêt à aligner 2 cases. En termes de positionnement, le bot privilégiera le centre puis les côtés puis les diagonales, enfin si il le peux il saisira l'occasion de gagner.
  Note : Ce mode de jeu a apriori très peu de réussite puisque très peu de configurations permettent de gagner.

- **Cauchemar :** Dans ce mode de jeu le bot ne cherchera pas toujours la victoire, ce qui rend ce niveau plus difficile est la vision limitée.
  Note : malgré le peu de visibilité ce mode de jeu est faisable

- **Virtuose :** Ce mode de jeu dispose des mêmes atouts que le mode Extrême. Ce qui le rend particulier et son mode de rotation et de visibilité très réduite. En effet a chaque fois que le joueur joue, le plateau tourne vers la droite (case 1 devient case 7 puis 9 etc.) Ensuite le bot joue et plusieurs phases de visibilité « ultra » réduite sont accordé à l'utilisateur :
  Avant rotation du plateau : 1 phase de visibilité (environs 0.07sec)
  Apres rotation du plateau : 4 phases de visibilité (environs 0.009sec puis 0.007sec puis 0.009sec puis 0.007sec
  Note : Le seul niveau de jeu ou en étant concentré il est possible de perdre !

III- Procédés algorithmiques :

Pour chacun de ces 6 niveaux nous n'avons pas fait usage de l'algorithme du minimax, sans cet algorithme il semble en effet impossible de traiter tout les cas 3^9 configurations. Cependant, en implémentant une stratégie a notre bot, des choix de priorités, il est possible de « balayer » une grande partie des cas. Enfin si l'on souhaite faire gagner notre bot ou garantir l'égalité a presque 100% il suffit alors d'implémenter des fonctions permettant le désavantage de l'utilisateur. Nous avons ainsi créé une fonction Flash() permettant de donner une visibilité réduite sinon quoi, les cases sont noires sur noir ; nous avons également créer une fonction Mozart() pour le mode virtuoso avec la rotation des cases.

IV- LeaderBoard OFF-LINE :

Pour cette partie nous avons fait appel au module Json. Nous avons procédé en 3 étapes :
La création d'un dictionnaire ayant pour clé les pseudos utilisateurs, les valeurs sont un nouveau dictionnaire prenant pour clé les différents niveaux et pour valeurs a nouveau un dictionnaire prénom en valeur : Victoire, Défaite, Egalite.
Cette manière d'organiser le fichier nous permet d'accéder très rapidement à la valeur voulu pour effectuer des actions dessus : l'afficher sur le leaderboard etc.

V- Leaderboard ON-LINE

Pour cette partie nous sommes parties sur une base de données No SQL : Py mongo.

Pourquoi ce type de base de données ? Contrairement a une data base de type SQL les données se structurent de la même façon qu'un fichier Json, cela simplifie ainsi la tache lors de transfert de données. Py Mongo permet de pallier la complexité du SQL.

Pour cette partie nous avons laissé la clé d'accès à la base de données, il est évident que normalement dans l'exécutable la clé ne soit pas trouvable.

Proposition : on pourra également joindre un fichier config dans lequel on placera le mot de passe et l'identifiant puis on pourra encrypter ces données (de manière la plus simple on pourra utiliser une clé rsa pour encrypter ces données).

Interrogations : puisqu'il est possible décompiler l'application créer et d'être ainsi en mesure de retrouver la clé d'accès à la base de données, encrypter est-ce la seule solution pour pallier la confidentialité de la base ?

VI- Login System

Pour le système de login nous procédons de la même façon que le leaderboard en ligne.

VII- Mode de jeu à part entière

Ayant eu une semaine supplémentaire pour réaliser le projet, nous avons rajouter un mode de jeu 3x3x9 : 9 Morpions, le but est d'aligner 3 morpions pour gagner.

Complexité : Ce mode a été d'une réelle complexité et a mis notre moral a épreuve puisque les règles de ce mode de jeu imposent une certaine rigueur dans la façon de procéder :

La case sélectionner indique le morpion dans lequel l'adversaire doit jouer. Si la case indique un morpion qui est déjà gagne ou perdu, le joueur peux alors joue n'importe ou sauf sur les morpions déjà gagné ou perdu.

Nous n'avons ainsi, malheureusement fait qu'un bot facile.

VIII- Pallier les bugs

Grace a une dizaine d'Ipsalien testeur et de connaissances à l'extérieur notre groupe a pu noter plusieurs problèmes, et ces problèmes devraient pour la plupart (voir tous) avoir été gérer.

Cependant, le dernier mode de jeu n'a pu être tester que par une personne en dehors des membres de notre groupe.

IX- Sonores

Nous avons ajoute des voix pour plusieurs aspect du jeu (victoire, défaite, explications règles, taquinerie) nous avons utilise le module mixer de pygame et nos fichiers audios ont été créés grâce au site :

https://notevibes.com

X- Images

Nous avons malheureusement, pour l'icone principale et le bouton du mode de jeu supplémentaire utilisées des images non libres de droit (apriori) :

iconeAPP.ico

(2 images tirées du manga : danganrompa)

Pour le fond principal il s'agit d'un fond pris sur https://wallpaperaccess.com/3840x1080

Pour l'icone du login il s'agit d'une photo libre d'usage puisqu'elle appartient à la team de Zakaria :

Ces images ont toutes été choisis dans la thématique du : malin (le rusé) et innocent.

XI- Répartition des taches :

> Responsable niveau facile/extrême => Talaat

> Responsable niveau moyen/cauchemar => Thomas

> Responsable niveau difficile/virtuose => Zakaria

> Responsable tri des données => Principalement Zakaria

> Responsable customisation => tout le monde

>Responsable mode de jeu MagicPionne (3x3x9) => tout le monde

* On s'est tous entre-aider pour chaque chose dont ont été responsable.
* Nous utilisons Git pour communiquer nos codes en temps réels sans perdre de temps.

Conclusion :

Ce projet a été très instructif pour nous, il nous a permis d'apprendre a travaille en groupe, de coordonnées nos idées et de faire preuve d'imagination. On a ainsi pu mettre en œuvre ce qu'on a appris en cours et devenir plus à l'aise en python.