

Projet de réseau

**Jeu du morpion aveugle**

6 janvier 2017

MEILHAN Geoffrey

# Table des matières

1	Présentation du projet . . . . .	2
2	Fonctionnalités du programme . . . . .	2
2.1	Serveur . . . . .	2
2.2	Client . . . . .	3
3	Extensions implémentées . . . . .	3
3.1	Partie solo . . . . .	3
3.2	Premier joueur aléatoire . . . . .	3
4	Erreurs dans le programme . . . . .	3
4.1	Si le joueur en court ferme le terminal avant la fin de la partie . . . . .	3
4.2	Si le serveur est fermé avant la fin de la partie . . . . .	3

# 1 Présentation du projet

Le jeu du morpion aveugle est une variante du morpion. Dans celle-ci, les joueurs ne voient que les coups qu'ils jouent et non les coups de leur adversaire. Si un des deux joueurs essaie de marquer une case que l'adversaire a déjà marquée, il est informé et rejoue un coup. Cela peut ainsi entraîner des parties différentes du morpion classique où l'on cherche à la fois à gagner, mais aussi à chercher la position des cases de l'adversaire.

## 2 Fonctionnalités du programme

Le programme est divisé en deux parties : un serveur, et les deux clients qui s'y connectent. La suite de commandes permettant de lancer une partie est la suivante, elle doit être réalisée dans 3 terminaux différents :

python3 serveur.py dans le premier terminal

python3 client.py 8888 dans le deuxième terminal

python3 client.py 8888 dans le troisième terminal

Une fois la connexion établie, le jeu se déroule de la façon suivante :

Le premier joue son coup, le client l'envoie au serveur, et ce dernier met à jour la grille.

Le deuxième joueur joue son coup, envoie l'information au serveur, le serveur vérifie que la case n'est pas occupée, puis met à jour la grille.

Les deux joueurs alternent ensuite jusqu'à ce que le serveur détecte la fin de la partie et informe les deux joueurs.

### 2.1 Serveur

Le serveur fait une boucle infinie tant que les deux clients ne sont pas connectés. Lorsqu'un joueur se connecte au port du serveur, le serveur envoie un code de bienvenue au client et augmente la variable nb\_joueur de 1. Lorsque celle-ci vaut 2, le serveur ajoute les deux joueurs au jeu et lance la fonction main(game) contenue dans jeu.py.

Le code de bienvenue est généré par pickle.dumps dans le fichier macro.py qui permet de transcrire WELCOME en octets, celui-ci étant identifié par le client et traité par une autre fonction décrite plus bas. Tant que la condition gameOver n'est pas remplie, le serveur va effectuer des tests pour déterminer quel joueur doit jouer, puis si son coup est valide. Si le coup est valide, il l'inscrit dans la grille commune aux deux joueurs qui permet de tester la condition gameOver et fait passer le tour du joueur. Si le coup n'est pas valide, il informe le joueur et lui demande de rejouer une case. A chaque tour de boucle, il teste également la condition gameOver qui est la règle de victoire habituelle du morpion consistant à avoir 3 cases cochées alignées. Si un des deux joueurs remplit cette condition, il transmet un code de victoire à ce dernier et un code de défaite à l'autre. Il affiche également aux deux clients la grille complète.

## **2.2 Client**

Le fichier client possède une fonction permettant de transcrire les codes reçus par le serveur en messages s'affichant sur la console pour informer le joueur. Par exemple, lors de la connexion au serveur, le client va recevoir le code WELCOME qui va afficher sur la console "Bienvenue sur le jeu du morpion aveugle." Pendant la partie, le client à accès uniquement à la grille des coups qu'il à joués , ou des coups joués par son adversaire si il a joué sur la même case.

## **3 Extensions implémentées**

### **3.1 Partie solo**

Si le client lance la commande suivante :

```
python3 client.py
```

La partie se fera contre un IA qui joue des coups aléatoires. La partie se joue de la même façon que contre un autre joueur, les commandes étant les mêmes.

### **3.2 Premier joueur aléatoire**

Lors du lancement de la partie, le premier joueur est déterminé par un `random.randint(0,1)`, ce qui permet de faire varier quel joueur commence la partie en premier.

## **4 Erreurs dans le programme**

### **4.1 Si le joueur en court ferme le terminal avant la fin de la partie**

Dans le cas où le joueur jouant son tour ferme son terminal, le serveur et l'autre client crash en même temps. On pourrait mettre une sécurité coté serveur qui traite ce cas de la façon suivante :

Lorsque le client est fermé ainsi, il n'envoie rien au serveur, il faut donc intégrer au serveur un test où si il ne reçoit rien, ou NULL, il puisse envoyer un message au client encore connecté informant celui-ci de la fin imminente du jeu. Le serveur pourrait ensuite faire un appel à exit et terminer la fonction sans erreur.

### **4.2 Si le serveur est fermé avant la fin de la partie**

Dans le cas où le terminal faisant tourner le serveur, est fermé avant que la partie ne se finisse, le joueur qui n'est pas en train de jouer crash instantanément, et l'autre joueur peut encore jouer son tour. Ce dernier crashera cependant lorsqu'il donnera une case à jouer. De façon similaire au cas traité plus haut, il faudrait installer une sécurité se déclenchant lorsque la connexion est perdue et affichant un message sur la console.