Projet de réseau

PROJET MORPION AVEUGLE

6 janvier 2017

Université de Bordeaux

Raphaël ANQUETIL
Brian LEBRETON

Table des matières

I- Présentation du projet

Le morpion aveugle est une variante de ce jeu dans laquelle les joueurs ne voient pas les coups joués par l'adversaire. Si un joueur essaie de marquer une case déjà marquée par l'adversaire, le joueur est informé que cette case est prise et il doit marquer une autre case. La Figure 3 représente un exemple de partie. Comme à la bataille navale, chaque joueur n'a qu'une vision partielle de l'état du jeu. Essayer de marquer une case déjà prise est un coup intéressant car il permet de savoir où l'adversaire a joué.

II- serveur.py

Le serveur est ce qu'il y avait de plus simple. Il nous fallait créer différentes listes dans lesquelles nous allions insérer les différents clients se connectant au serveur. Ensuite nous avons créé le socket du serveur avec comme port '7777' qui n'est pas utilisé.
Puis on met ce socket à l'écoute pour attendre que les clients se connectent. On initialise un compte de joueurs/clients qu'à chaque connexion nous incrémentons.
Nous avons donc différencié les joueurs des spectateurs. Les 2 premiers clients se connectant sont des joueurs et les clients suivants sont considérés comme des spectateurs.
A partir du moment où il y a 2 joueurs la partie commence en lançant le main.

III- client.py

Le client quant à lui est chargé d'indiquer au joueur, ou au spectateur l'état du jeu. Pour cela, il reçoit différent message de la part du serveur. Il effectue ainsi un traitement sur le message reçu pour afficher les informations en conséquence.
La boucle de jeu est donc une répétition de l'envoie du choix du joueur suivi d'une réception du message que lui renvoie le serveur.
Si le client indique un numéro de port en paramètre, une tentative de connexion va être effectuée entre le serveur appartenant à ce port et s'il y a un autre joueur de disponible lancer une partie. Mais si le client n'indique pas un numéro de port, le jeu se lance contre l'ordinateur, comme avec la version initiale.
On a donc comme procédure pour lancer le jeu en réseau:

- python3 serveur.py
- python3 client.py 7777
- python3 client.py 7777

Chaque commande doit être exécutée depuis un terminal différent.

IV- main.py

main.py a été modifié pour satisfaire nos besoins. Il permet au serveur de lancer le jeu du morpion à l'aveugle. Il gère les grilles des joueurs 1 et 2 et les réceptions et les envois à faire au serveur.

V- Les extensions

1. Jouer en solitaire

Les fichiers fournis contenaient le jeu du morpion aveugle contre l'ordinateur qui joue des coups aléatoirement. Nous avions déjà la plus grande partie du travail déjà faite, il nous a suffi de gérer le "comment jouer seul". Puisqu'il faut utiliser le même client que ce soit en réseau ou seul, ce qui est logique, nous avons ajouté une condition dans le client. S'il n'y a pas de paramètre alors on lance le programme qui nous a été fourni depuis "solo.py". S'il y a un paramètre alors le programme du client continue et va essayer de se connecter au serveur.

2. Le mode spectateur

Nous nous sommes penchés sur le problème après le jeu en solitaire pensant que c'était une extension assez simple. Malheureusement cela a été plus compliqué que prévu et malgré nos idées nous ne sommes pas parvenus à une solution.
Nous n'avons pas traité le reste des extensions car le jeu en réseau en lui-même, sans erreurs, a été très long à réaliser. Il ne nous restait pas assez de temps pour finir le reste des extensions.

VI- Difficultés rencontrées

Les difficultés ont été multiples. Tout d'abord nous avons eu des problèmes avec le langage python que nous n'avions que très, très peu vu moi et mon binôme malgré les 2 travaux pratiques en cours de réseau. Il a donc fallu se "remettre dans le bain". Ensuite nous avons eu des problèmes concernant quelles parties du code allait dans le client ou le serveur. Pensant au début que nous allions mettre le jeu sur le serveur et que le client ne ferait uniquement que recevoir des informations et en renvoyer au serveur. Nous n'étions pas loin de la vérité pour le client mais pour le serveur, nous avons préféré dissocier le code du jeu du serveur. C'est toujours lui qui s'en occupe et qui l'appelle mais nous avons pensé que le code du serveur serait trop lourd, d'où ce changement. Nous avons aussi eu des problèmes lors de l'envoie de données pour les messages à envoyer du main (géré par le serveur) au client. En effet, au cours du jeu le client doit être au courant de certaines informations (quelle case jouer ?, case déjà jouée, gagné, perdu,...). Pour cela nous avons utilisé le module python pickle qui nous a permis de transmettre des données plus facilement du serveur aux clients par le biais de code, que le client va interpréter et en fonction de ceux-là réagir.

VII- Conclusion

Dans l'ensemble ce projet était intéressant et nous a plus à tous les 2. Cela nous a été instructif. Nous avons appliqué, les connaissances que nous avons acquises pendant ce premier semestre. Nous pensions que nous n'aurions pas de problème à arriver au bout du projet mais la tâche n'a pas été aussi facile que prévue. De plus, ce projet était un mini-jeu ce qui le rend plus attrayant, pour nous étudiant, qu'un simple projet qui parfois n'a pas vraiment de but mis à part de nous faire faire un projet. Si nous pouvons reprocher une chose au projet, c'est la méthode de rendue. Il aurait été plus simple de pouvoir le déposer sur moodle comme nous avions fait dans les semestres précédents.