Rapport de Projet de Programmation Réseau

Groupe A3
Amélie RISI
Alexis PICHON

6 janvier 2017

Table des matières

Chapitre 1

Le Projet

1.1 Description

Ce projet concerne le développement en réseau d'un jeu de morpion aveugle.

1.2 Choix techniques

Afin de se conformer aux contraintes imposées pour la réalisation de ce projet, celui-ci est hébergé sur la plate-forme Savane. Le dépôt est disponible à l'adresse suivante : lien externe vers le dépôt Git sur Savane.

Chapitre 2

Conception du projet

Lors de notre conception du projet, nous nous sommes fixés pour objectif d'implémenter les fonctionnalités demandées en se conformant à un découpage modulaire du code. Ceci a pour but de permettre une relecture plus aisée du code, mais également de faciliter la compréhension de celui-ci.

2.1 Architecture du projet

Notre projet s'articule autour des fichiers qui nous ont été fournis selon le découpage suivant :

client.py : Fichier contenant les sources du module *client*

grid.py : Fichier contenant les sources du module *grid* (fichier fourni pour le projet)

main.py : Fichier contenant les sources du module *main* (fichier fourni pour le projet)

serveur.py : Fichier contenant les sources du module *serveur*

2.2 Travail réalisé

2.2.1 Grid.py

Nous avons ajouté une fonction displayStr qui nous sert à afficher la grille car la fonction display de base dans le fichier ne nous permettait pas d'afficher la grille à cause d'une conversion de type non permise.

2.2.2 Main.py

Ce fichier, nous autorise à lancer le jeu sans devoir exécuter deux différents. Pour expliquer cela, nous avons mis en place un readme pour expliquer les modalités simple de lancement.

2.2.3 Client.py

Ce module sert à la communication avec le serveur. Il n'affiche que les données qu'il recevra du serveur ainsi que la demande à l'utilisateur pour savoir quelle case il choisi pour placer son pion.

2.2.4 Serveur.py

Ce fichier implémente tout le serveur. Nous avons fait le choix d'une communication TCP afin de permettre au utilisateur d'utiliser des ordinateurs différents et ne pas jouer qu'en local. Pour cela, le serveur se charge de communiquer avec tous les clients. Les deux premiers connectés sont considérés comme les joueurs et les autres, les observateurs. Nous avons essayer de séparer chaque actions dans le jeu en plusieurs fonctions, dont la fonction turn et play. Turn nous permet de transmettre au joueur actuel un mot magique 'turn' pour lui signifier qu'il peut demander au client la case à choisir et envoyer au serveur sa réponse. Play sert à jouer le mouvement désirer par le joueur actuel et d'actualiser les vues des utilsateurs.

2.3 Travail inachevé ou non réalisé

Notre rendu est malheurement pas fonctionnel car il semblerait que nous n'arrivions pas à gérer les plusieurs utilisateurs. Probablement, il doit nous falloir des threads pour en gérer plusieurs. Donc, le problème se confronte à nous lorsque le joueur 2 veut jouer sur une case. A ce stade, le joueur peut essayer de positionner le pion sur le même endroit que l'autre joueur, la vue s'actualise mais il ne peut pas jouer une autre case après. Nous émettons les hypothèse que ceci est due au fait que les clients on peut être la même addresse donc qu'il y a un conflit avec les ronds et croix ou que le gestion simple (sans thread) de plusieurs n'est pas possible l'utilisation des threads justement. Par conséquent, nous avons donc essayer de permettre à deux joueurs de jouer l'un contre l'autre via une connexion tcp et de permettre aux clients d'être observateur sans succès malheuresement.