

Rapport projet de réseau

DELAR Emmanoe, RAKOTOARIJAONA Camille

21 avril 2017

Table des matières

1	Introduction	3
2	Fonctionnement du jeu	3
2.1	Présentation du jeu	3
2.2	Jouer contre la machine	3
3	Structure du code	3
3.1	Partie serveur	3
3.2	Partie client	3
3.3	Protocole de communication	4
4	Extensions	4

1 Introduction

Lors de ce projet, réalisé en binômes, nous avons pour objectif de développer un jeu de bataille navale en réseau. Pour cela, nous avons utilisé le langage de programmation objet Python. On est parti du code source fourni par notre enseignant. Ce code nous permettait de jouer contre la machine uniquement. Dès lors, nous avons dû l'améliorer afin de pouvoir jouer à 1 contre 1 sur le réseau.

2 Fonctionnement du jeu

2.1 Présentation du jeu

La bataille navale est un jeu de société dans lequel deux joueurs doivent placer des navires sur une grille tenue secrète et tenter de toucher les navires adverses. Le gagnant est celui qui parvient torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient..

2.2 Jouer contre la machine

Lors ce qu'on lance le programme, le jeu commence directement. On choisi la colonne et la ligne à viser. Quand c'est au tour de l'ordinateur, une fonction choisie des coordonnées aléatoirement puis joue. Et ainsi de suite jusqu'à la fin de la partie.

3 Structure du code

3.1 Partie serveur

Contrairement à la partie contre l'ordinateur, en mode réseau, les coordonnées choisis par le client sont envoyés au serveur qui les retransmet au deuxième joueur connecté. À l'aide du constructeur socket, nous avons alors créé un serveur de famille ipv6 pour une connexion TCP. Nous l'avons rattaché au port 7777 (choisi arbitrairement) puis nous avons lancé notre serveur en mode écoute.

3.2 Partie client

On lance notre programme en mode client avec l'adresse du server. Le client a ensuite deux choix, soit il joue contre l'ordinateur ou alors en réseau. Si il décide de jouer contre l'ordinateur, le code source initial va s'exécuter. Sinon, on crée une socket de même type que celle du serveur. On récupère l'adresse du serveur puis on s'y connecte en TCP.

3.3 Protocole de communication

Une fois la connexion établie, la communication serveur/client est très simple. Chaque information nécessaire à l'initialisation d'une partie est envoyé par le serveur bit à bit et ainsi récupéré du côté client. Il faut noter la nécessité à convertir les bytes recus en entier (int).

4 Extensions