



UNIVERSITÉ D'AVIGNON
ET DES PAYS DE VAUCLUSE

C E N T R E
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE



Licence Informatique
spécialité Ingénierie Logiciel
UE projet de programmation

Rapport de Groupe - Moteur réseau

TORO Loïc BOUCHER Félix

19 octobre 2015

CERI - LIA
339 chemin des Meinajariès
BP 1228
84911 AVIGNON Cedex 9
France

Tél. +33 (0)4 90 84 35 00
Fax +33 (0)4 90 84 35 01
<http://ceri.univ-avignon.fr>

Table des matières

Titre	1
Table des matières	2
1 Présentation	3
2 Protocole	3
3 Classes	3
3.1 TCP/IP	3
3.2 UDP	3
4 Fonctions	3
4.1 Fonctions utilisées par la composante interface utilisateur	3
4.2 Fonctions utilisées par notre composante	4
4.3 Détails	4
5 Features envisagées	4

1 Présentation

Cette composante aura pour objectif de permettre de jouer via le réseau internet. Dans cet objectif, nous planifions d'être en interaction avec une seule autre composante. Nous ne serons en effet liés qu'à l'interface utilisateur, pour récupérer les données du moteur physique et gérer l'envoi au serveur et la réception de ces données.

Nous avons conclu qu'une interaction avec le moteur graphique est inutile dans l'envoi des fichiers au serveur. Dans le cas du moteur physique, aucune interaction directe n'est nécessaire au bon fonctionnement de notre travail. L'interface utilisateur servira donc de lien entre notre composante et les autres.

2 Protocole

Le protocole de transport retenu dans le cadre de l'application "Curve Fever" est le protocole UDP, car plus simple et plus rapide que le protocole TCP lors du déroulement d'une partie. La perte éventuelle de données n'est pas importante dans la mesure où les données seront actualisées plusieurs fois par seconde (au minimum 20).

Nous utiliserons le modèle TCP/IP pour établir la connexion entre les clients et le serveur ainsi que pour faire transiter des données plus importante comme les informations relatives aux joueurs.

3 Classes

3.1 TCP/IP

La classe Socket est l'élément fondamental de toute connexion réseau côté client. Elle utilise le modèle TCP/IP par défaut.

Le classe ServerSocket est la classe permettant d'initialiser un serveur sur lequel le client pourra se connecter.

3.2 UDP

La classe DatagramPacket représente les données qui seront envoyées via le protocole UDP : les paquets UDP s'appellent des datagrammes, d'où le nom de cette classe.

La classe DatagramSocket, elle, permet d'émettre ou de recevoir des datagrammes. Nous partagerons les structures du Moteur Physique qui auront été initialisées par l'Interface Utilisateur.

4 Fonctions

Voici les méthodes que l'on devra implémenter :

4.1 Fonctions utilisées par la composante interface utilisateur

- launchServer(); Permet de créer un serveur pour que les clients puisse s'y connecter.
- launchClient(); Permet a un client de se connecter à un serveur.
- shutdownServer(); Permet de stopper le serveur et ainsi déconnecter tous les clients.
- shutdownClient(); Permet à un client de clore la connexion avec le serveur.
- updateData(Object o); Permet au serveur d'actualiser les informations à l'aide des fonctions send et receive.

4.2 Fonctions utilisées par notre composante

- `send(Object o)` ; Permet d'envoyer des informations relatives à la partie en cours entre le serveur et les clients de celui-ci.
- `receive(Object o)` ; Permet au serveur ou aux interfaces clients de recevoir les informations concernant la partie en cours, dans l'optique de pouvoir les traiter.

4.3 Détails

Les "objets o" cités correspondront selon les situations à des informations joueur ou des informations relatives au jeu en cours, telles que les positions actuelles des snakes.

Notre module n'aura besoin d'utiliser aucune méthode implémentée par d'autres composantes. D'autres méthodes seront utilisées provenant des classes de JAVA.

5 Features envisagées

Une fois que la partie la plus importante sera développée (c'est-à-dire le jeu en réseau), il est également envisagé d'apporter des options supplémentaires.

- Un chat qui permettra aux joueurs de communiquer lors d'une partie sur internet.
- La possibilité pour un joueur de se créer un compte en ligne.
- La mise en place d'un classement global avec tous les joueurs inscrit.
- La mise en place d'un serveur rassemblant tous les serveurs créés par les joueurs, de sorte à ne plus avoir besoin de connaître l'adresse IP de la personne ayant créé le serveur pour pouvoir s'y connecter.