

# Rapport 01 Projet IL

Félix BOUCHER et Loïc TORO

08 octobre 2015

## 1 Présentation composante Moteur Réseau

Notre groupe fait partie du développement du Moteur Réseau.  
Cette composante aura pour objectif de permettre aux utilisateurs de jouer en réseau local, mais aussi par Internet via une communication de type client/serveur.

## 2 Description des interactions

En ce qui concerne les interactions avec les autres composantes, nous serons essentiellement liés à l'Interface Utilisateur : notre tâche consistera en effet à récupérer les données du Moteur Physique et gérer leur bon acheminement vers le serveur.

Une interaction avec le Moteur Graphique est inutile dans cette gestion des fichiers entre le client et le serveur. En revanche elle le sera lors de la réception des données, pour l'affichage.

Nous n'entreprendrons aucune interaction directe avec les composantes du Moteur Physique.

## 3 Formalisation

Le protocole de transport retenu dans le cadre de l'application "Curve Fever" est le protocole UDP : celui-ci est plus simple et plus rapide que le protocole TCP/IP. La perte éventuelle de données n'est pas importante dans la mesure où les données seront actualisées plusieurs fois par seconde.

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. Il faudra cependant également utiliser le protocole TCP pour les données importantes comme l'authentification des joueurs ou un système de chat.

Nous partagerons les structures du Moteur Physique qui auront été initialisées par l'Interface Utilisateur. Voici une ébauche des méthodes que l'on devra implémenter :

- `launchServer()`; Permet d'émuler un serveur pour qu'un joueur puisse host une partie.
- `launchClient()`; Permet à un joueur de rejoindre un serveur .
- `sendDataServer()`; Permet d'envoyer les données au serveur qui les redistribuera aux différents joueurs.
- `receiveDataServer()`; Permet de récupérer les données du serveur.
- `sendChat()`; Permet d'écrire un message sur le chat et l'envoyer.
- `receiveChat()`; Permet de recevoir les messages des autres utilisateurs.
- `sendInfoPlayer()`; Permet de distribuer les informations d'un joueur (nom, nombre de points)

*D'autres méthodes seront utilisées provenant de différents packages de JAVA.*