

# Cahier des charges

M. FRIEDLI, A. GILLIOZ, J. GUERNE  
He-Arc Ingénierie  
2000 Neuchatel

23 avril 2017

# Chapitre 1

## Introduction

Le but de ce projet est de créer une application multi-plateforme en java. Dans ce projet nous allons créer des minis jeux, jouable sur différentes plateformes et en réseau. Les minis jeux implémenté ne seront pas très complexe, car le véritable objectif de ce projet est de faire communiquer plusieurs clients via un serveur.

Nous listons les objectifs que nous nous sommes fixé dans la seconde partie. L'architecture client/serveur que nous allons utilisé sera décrite dans la partie 3 de ce document. Et la dernière partie du document sera consacré aux fonctionnalités proposées par notre application.

# Chapitre 2

## Objectifs

Cette partie décrit les principaux objectifs que nous nous sommes fixés.

### 2.1 Objectifs

- Faire communiquer un client avec un serveur.
- Le serveur renverra ensuite les données demandé par le client.
- Créer des mini-jeux.
- Faire communiquer les clients entre eux via le serveur.
- Sérialiser les données des jeux pour permettre le jeu en ligne.
- Le serveur fera la synchronisation des clients.

### 2.2 Objectifs bonus

- Rechercher automatiquement les joueurs connectés à l'application.

## Chapitre 3

# Description du projet

Ce chapitre présente les différentes phases du programme. En expliquant l'architecture client/serveur que nous allons utiliser. La figure 3.1 montre un schéma de notre structure réseau.

### 3.1 Coté serveur

Le serveur recevra les données qui lui seront envoyées par le client. Il les traitera, puis les renverra aux différents clients qui jouent ensemble. La partie traitement que fera le serveur sera la partie « model » de notre application.

### 3.2 Coté client

Le launcher client de notre application va servir à afficher les jeux et le menu, c'est la partie « view » de notre programme. Ce launcher nous servira aussi de « controller », pour gérer les actions des joueurs.

Le joueur pourra choisir à quel mini-jeu il veut jouer, depuis sa propre plateforme (ex : Desktop, Android, Web). Une fois le jeu choisi, il ouvrira une connexion de jeu avec le serveur et un autre joueur. Une fois connecté, le joueur pourra faire des actions qui seront envoyées au serveur. Le serveur traitera les données reçues et les renverra. Quand le client reçoit des informations du serveur, il met à jour sa partie graphique.

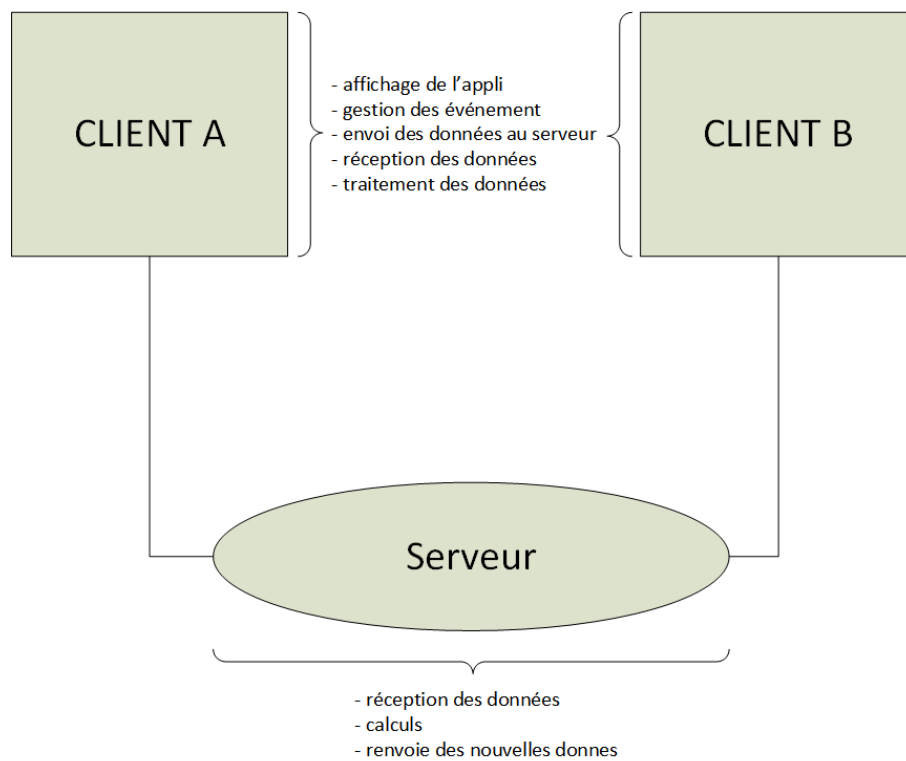


FIGURE 3.1 – Schéma de l'architecture client/serveur

## Chapitre 4

# Fonctionnalités

### 4.1 Minimum Viable Product (MVP)

#### 4.1.1 Minis-jeux

Nous implémenterons, dans l'ordre, les jeux suivants :

- Le morpion
- La bataille navale
- Jeu de dames

#### 4.1.2 Plateformes

Notre projet devra être utilisable sur, au moins, les plateformes suivantes :

- PC (Mac, Window et linux)
- Android
- IOS
- HTML5