

Conseil 7

Joël Collinet

Félix Erb

Célia Kocher

Vincent Moitry

Adam Rimelé

Carlo Spiga

Maxime Zimmer

## 1.A.1 Dossier d'analyse métier

---

**Projet Acrobatt 2014**

**La Clémence d'Auguste**

**Tuteur : Pierre Kraemer**

# SOMMAIRE

## Introduction

### I. Modélisation Métier

a) Diagramme de Cas d'Utilisation

b) Modèle Métier

i. Diagramme de Classes d'Analyse

c) Diagramme de Séquence

### II. Modèle de la Base de Données

a) Modèle Conceptuel de Données

b) Dictionnaire de Données

## Conclusion

## Introduction

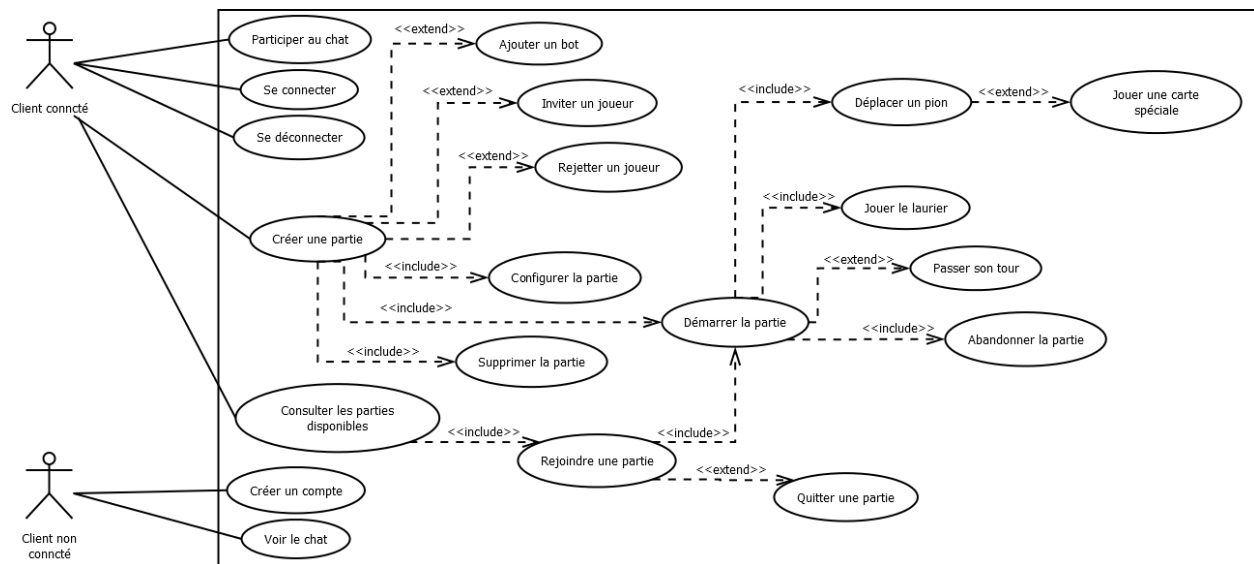
Dans le cadre du projet, il est nécessaire d'évaluer avec précision le fonctionnement de l'ensemble de l'application. Il s'agit d'un jeu de plateau dans lequel il est susceptible de trouver bon nombre de classes (pour définir les pions, les déplacements, par exemple).

Une phase d'analyse est fortement recommandée dans laquelle divers diagrammes sont conçus. Ces diagrammes permettent de schématiser aisément les divers liens établis entre les classes définies dans le code.

## I. Modélisation métier

## a) Diagramme de cas d'utilisations

Dans un premier temps, on évalue les diverses interactions pouvant se produire entre l'utilisateur et l'application. Ainsi, on peut former une liste de cas d'utilisation (lancer une partie, jouer une carte, récupérer le laurier, ...).



Un client non connecté n'a qu'un accès très limité au site. L'inscription permet toutefois d'accéder à la majorité des fonctionnalités disponibles sur le site.

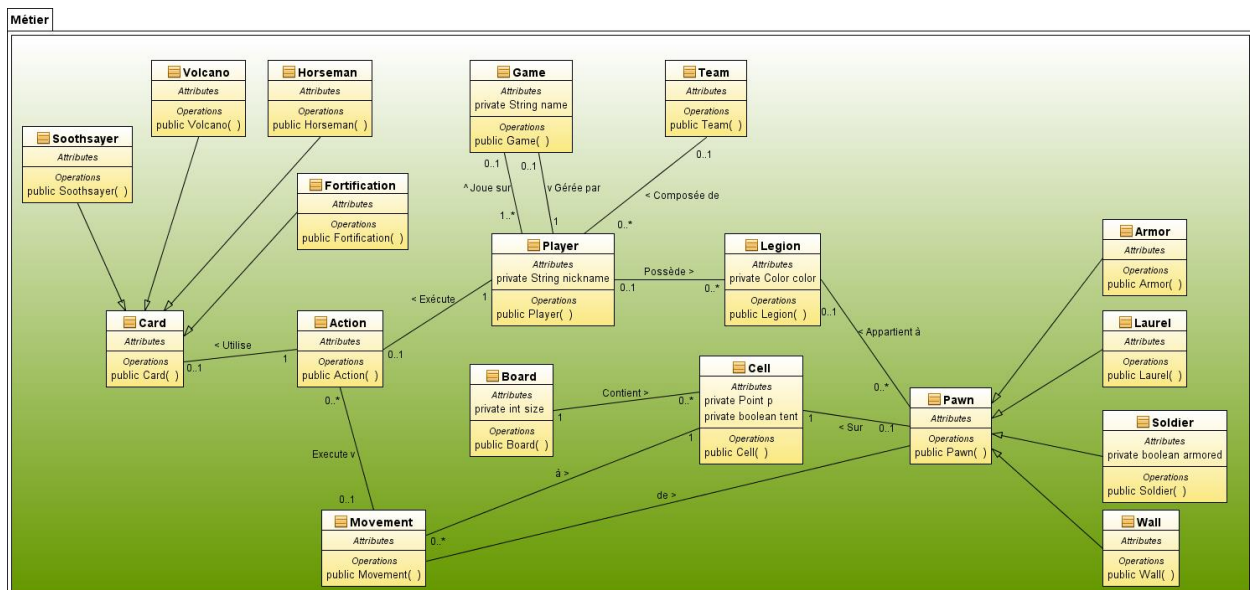
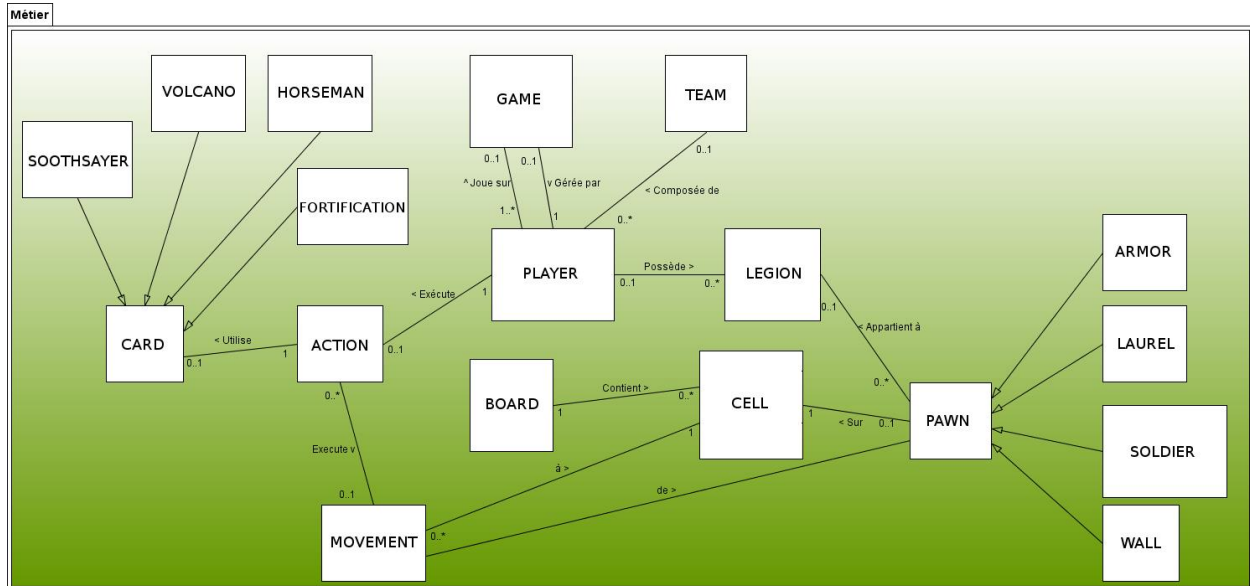
Les include représentent les fonctionnalités capitales au bon fonctionnement du jeu (démarrer la partie, rejoindre une partie, ...). Les extend matérialisent les fonctionnalités non bloquantes (jouer une carte spéciale, inviter un joueur, ...), une partie peut correctement se dérouler même si elles ne sont pas implémentées.

## b) Modèle métier

On peut ensuite définir la liste des classes qui sont utilisées dans le code, et les liens existants entre ces classes. Quel attribut peut caractériser une légion ? Combien de types de cartes existent ? Les différentes classes pouvant intervenir sont analysées,

## Conseil 7

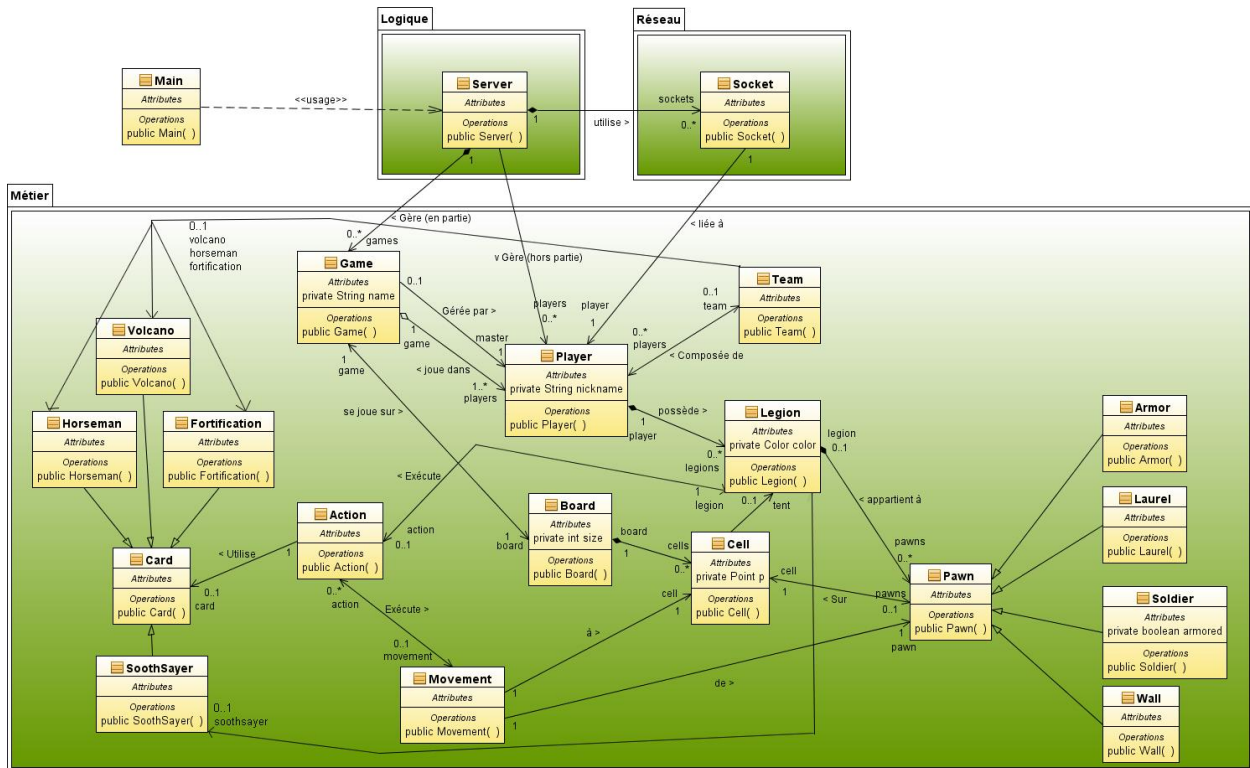
et leurs interprétations sont établies.



Card : les cartes pouvant être jouées (devin, volcan, cavalier, muraille), chaque type de carte hérite de la classe Card.

Pawn : les pions (armure, laurier, soldat, mur), chaque pion hérite de la classe Pawn.

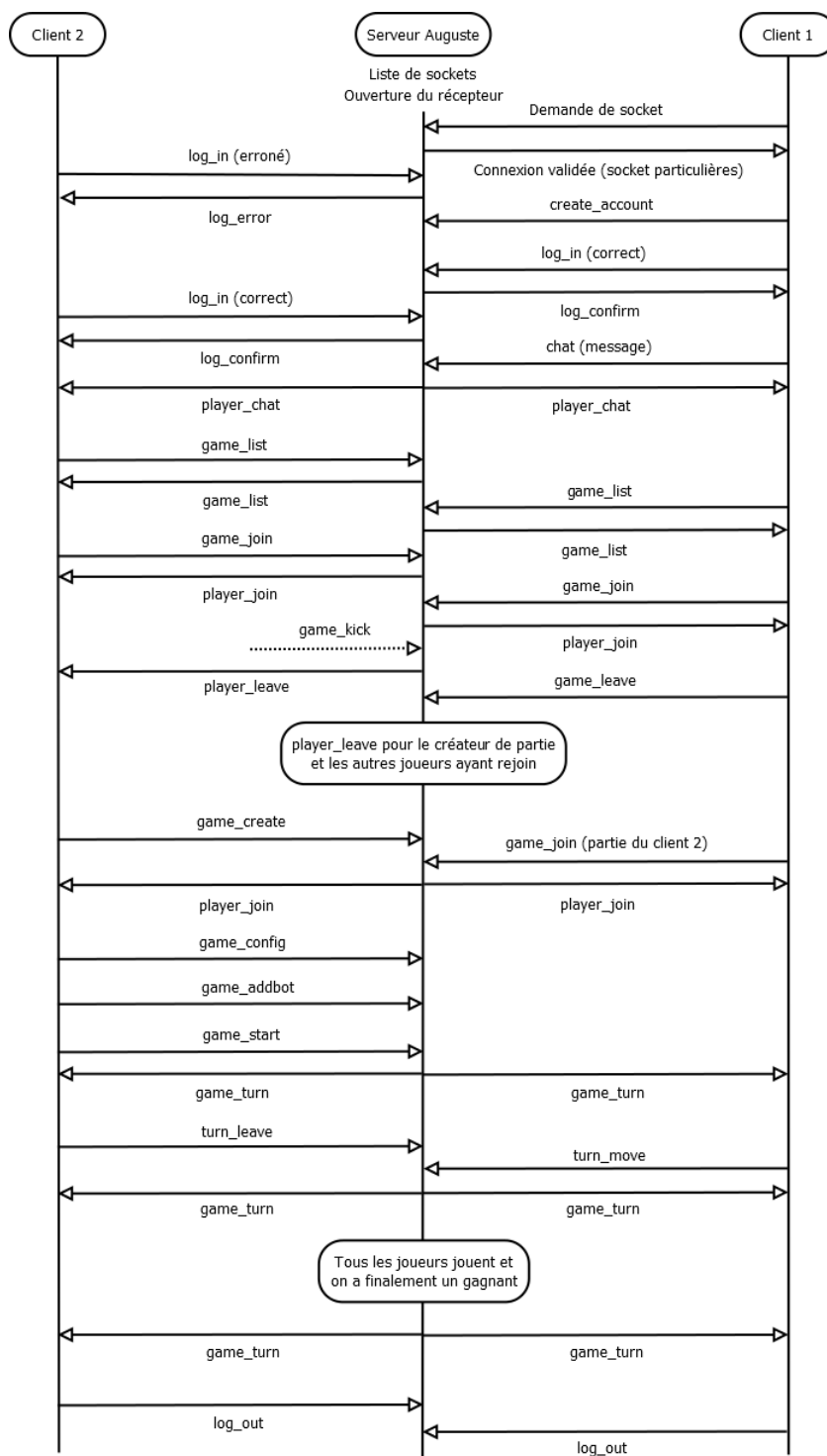
Il est ensuite nécessaire de regrouper les classes au sein de divers packages. Ici, l'interface principale n'est incluse dans aucun package. La logique serveur et les communications font chacune l'objet d'un package. Le reste des classes est regroupé dans un seul et unique package de classes métier.



## ii. Diagramme de séquence

Pour pouvoir visualiser au mieux le déroulement d'une partie, un diagramme de séquence est réalisé. Les noms des commandes JSON sont utilisés pour matérialiser au mieux chaque action (portent les noms des actions).

## Conseil 7



## II. Modèle de la base de données

### a) Modèle conceptuel de données

Il faut établir la structure de la base de données. Quelles données faut-il stocker ? Quels sont les liens à établir entre ces données ?

À l'heure actuelle, seuls les comptes des membres sont stockés en base de données. Une table player contient trois colonnes : identifiant, pseudonyme et mot de passe (haché). Le modèle de données est susceptible d'évoluer au fil du développement, pour y intégrer de nouvelles tables. Toute réflexion est encore ouverte à ce stade.

player	
<u>id</u>	INT(11)
login	VARCHAR(50)
password	VARCHAR(50)



### III. Dictionnaire de données

La base de données est nommée 'Auguste'. Le dictionnaire de données MySQL est situé dans la base de données Information\_Schema. Il est possible de s'y référer pour récupérer contraintes, procédures et déclencheurs au besoin.

L'utilisation concrète du dictionnaire dans le cadre du développement n'a pas encore été clairement définie. La réflexion est encore ouverte à ce sujet.

## Conclusion

Les différentes analyses à mener permettent à une logique de se construire au sein du projet. On prévoit les diverses utilisations de l'application, on établit un ensemble de classes au sein du code, des tables au sein de la base de données, on définit le bon déroulement d'une partie.

On construit ainsi en partie l'application avant de commencer à la coder. La structure pouvant s'avérer complexe, elle est envisagée avant d'être mise en application. On définit quoi coder avant de coder.