

---

---

# INFO-F209 : PROJET D'ANNÉE

## SOFTWARE REQUIREMENT DOCUMENT

---

---

Amin Abdelkefi - *matricule*  
Lucie Borremans - *matricule*  
Corentin Candeur - 411488  
Jonas Descurieux - 000382867  
Rémy Detobel - *matricule*  
Maxime Dewit - *matricule*  
Carlos Requena López - 410031  
Raphaël Slagmolen - 000377852

18 décembre 2015

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	But du projet . . . . .	1
1.2	Glossaire . . . . .	1
1.3	Historique du document . . . . .	1
<b>2</b>	<b>Besoins de l'utilisateur</b>	<b>2</b>
2.1	Exigences fonctionnelles . . . . .	2
2.2	Exigences non fonctionnelles . . . . .	2
2.3	Exigences de domaine . . . . .	2
<b>3</b>	<b>Besoins du systeme</b>	<b>2</b>
3.1	Exigences fonctionnelles . . . . .	2
3.2	Exigences non fonctionnelles . . . . .	2
3.3	Design et fonctionnement du systeme . . . . .	2
<b>A</b>	<b>Index des termes utilisés</b>	<b>2</b>

# 1 Introduction

## 1.1 But du projet

Dans le cadre du projet d'analyse et méthode et de système d'exploitation, il est demandé d'implémenter un jeu de carte en ligne basé sur une architecture client/serveur en C/C++.

Ce projet devra fournir un jeu de carte symétrique où deux joueurs pourront s'affronter et gagner des points pour un classement général. Chaque joueur disposera d'une collection de cartes à partir de laquelle il pourra construire des decks. Ces decks contiendront des cartes de type créature qui peuvent attaquer et des cartes de type sort qui lanceront des événements spéciaux.

Le projet requière la mise en place d'une architectures client/serveur avec le client qui sert principalement d'interface de jeu (graphique ou console) et les calculs qui seront effectué côté serveur. Il est donc important de développer un serveur robuste, efficace et rapide qui soit en mesure de gérer plusieurs parties simultanément tout en fournissant des services tels que le classement général, un service de compte et de login ainsi qu'un service de messagerie instantané entre deux joueurs. Pour cela, nous seront ammené à développer une application capable de gérer un maximum de cas de figure pour garantir la continuité et la stabilité de ces différent services. La gestion d'erreurs devra être telle que cette erreur ne perturbera aucune autre partie ni aucun des autres services.

Le Wizard Poker sera disponible à n'importe quel joueurs enregistré pour jouer contre d'autre joueurs et essayer de gagner des places dans le classement générale. Il permettra aussi à de nouveaux joueurs de consulter ce classement et de s'enregistrer en tant que nouveau joueur. Dans ce cas, le joueur recevra ses premières cartes et sera invité à confectionner son premier deck avant d'affronter son premier adversaire.

## 1.2 Glossaire

— **Deck** : Collection de 20 cartes qui...

## 1.3 Historique du document

Version	Auteur	Date modification	Description des changements
NEXT			
NEXT			
v0.0	Carlos Requena	26/11/15 22 :30	Mis en place. Diagrammes réalisés en groupe le 25/11/15 ajoutés

TABLE 1 – Changements document

## **2 Besoins de l'utilisateur**

### **2.1 Exigences fonctionnelles**

### **2.2 Exigences non fonctionnelles**

### **2.3 Exigences de domaine**

## **3 Besoins du systeme**

### **3.1 Exigences fonctionnelles**

Sequence diagram pour interaction ??

Le système doit interagir avec l'utilisateur, et lui proposer certaines actions, comme commencer un duel, gérer ses decks, consulter le classement des joueurs, etc.

Pour les nouveaux joueurs, l'application doit offrir la possibilité de s'enregistrer, avec un pseudonyme et un mot de passe.

### **3.2 Exigences non fonctionnelles**

Les exigences non fonctionnelles décrivent le comportement de l'application.

Le système doit être tout d'abord robuste et stable, c'est-à-dire gérer les erreurs et éviter les crashes et les déconnexions... Il doit par ailleurs être agréable d'utilisation pour l'utilisateur (ergonomie et esthétique/design) et garantir que celui-ci ne puisse pas tricher pendant une partie.

Donc en résumé, robuste, stable, sûr, et excessivement amusant. Des heures de franche rigolade, en toute sécurité, sont assurées à tout utilisateur du jeu.

### **3.3 Design et fonctionnement du systeme**

## **A Index des termes utilisés**

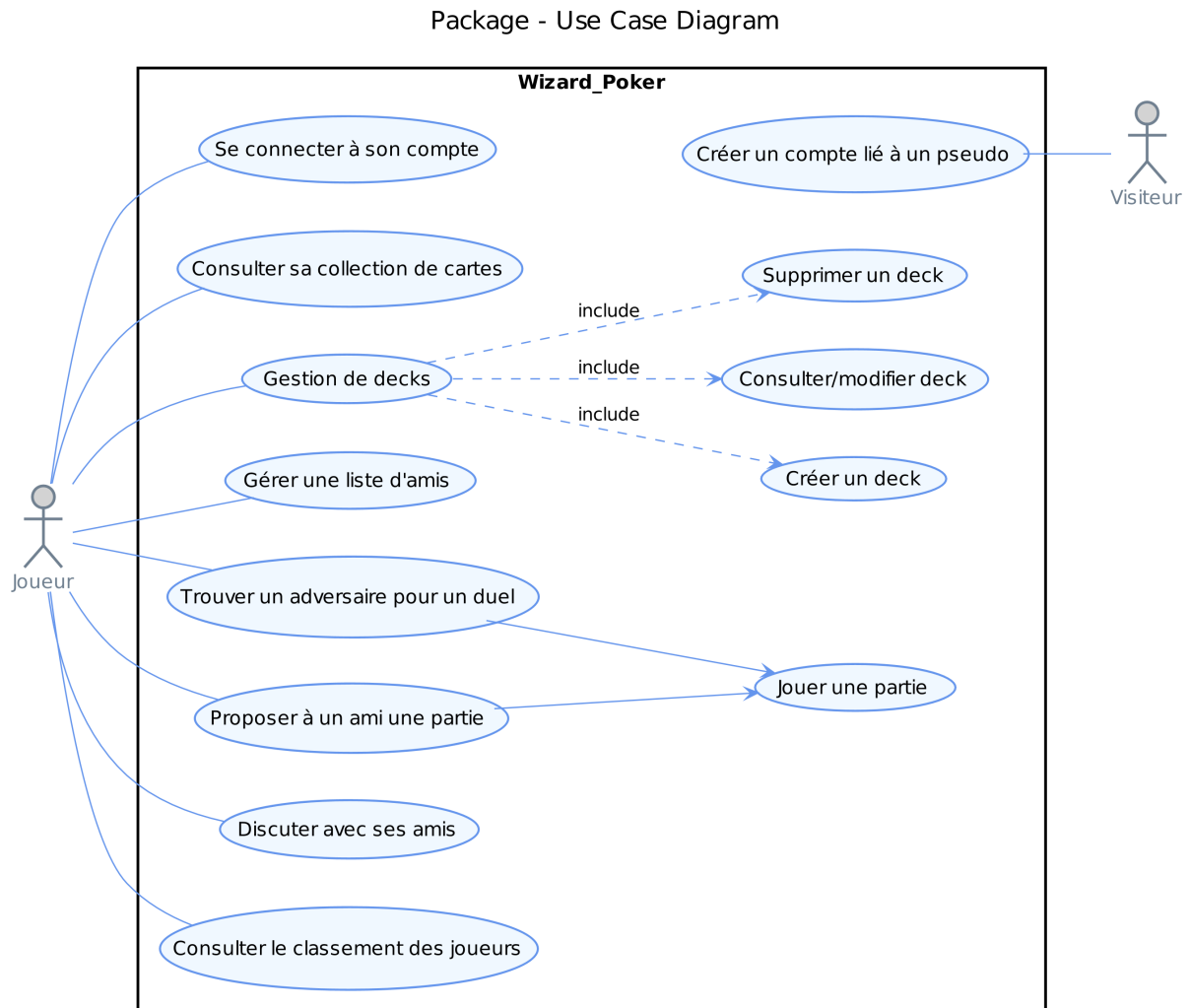


FIGURE 1 – Use Case Diagram du jeu en question

## Wizard - Class Diagram

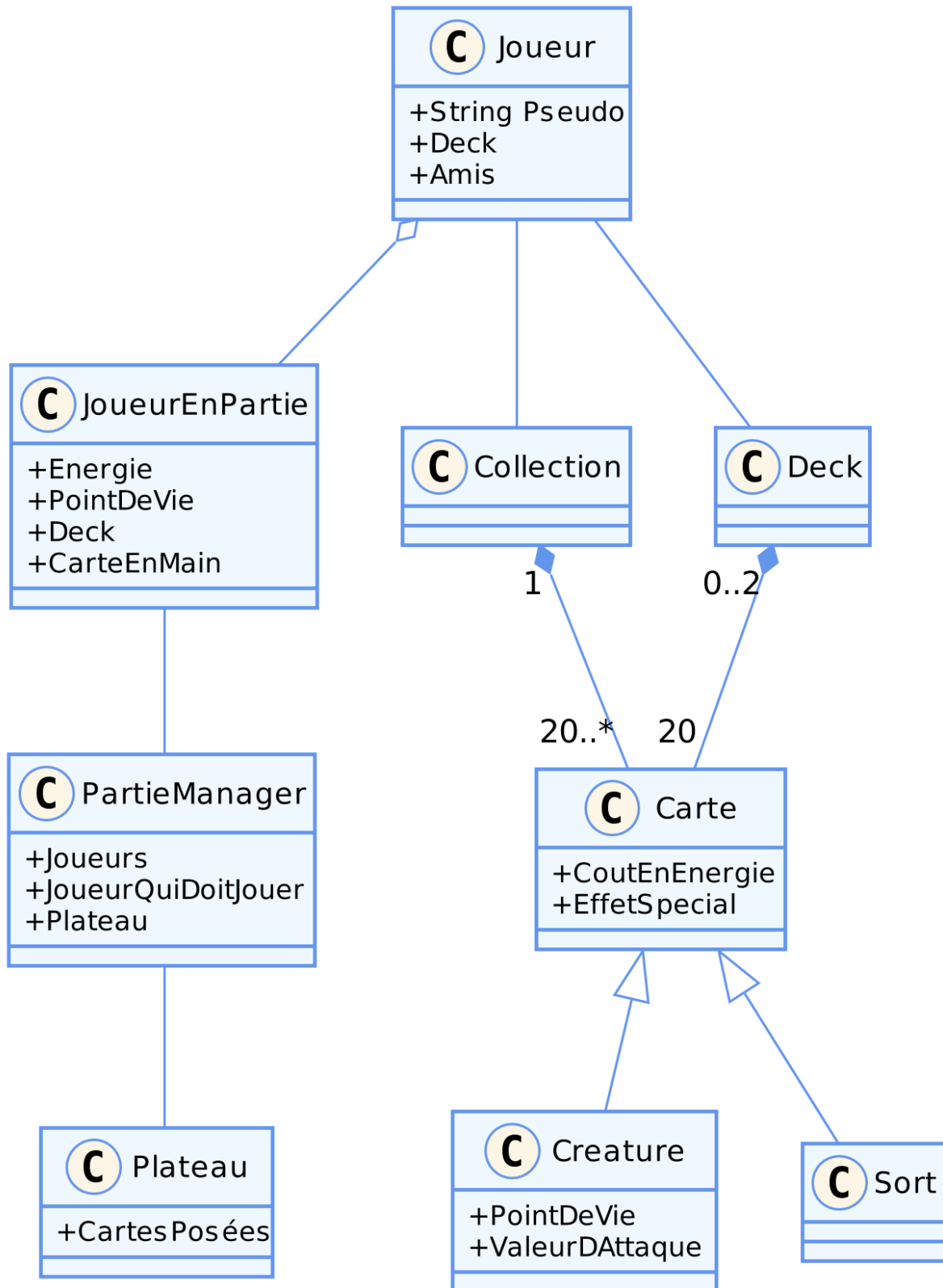


FIGURE 2 – Diagramme des classes