

---

---

# INFO-F209 : PROJET D'ANNÉE

## WIZARD POKER

## SOFTWARE REQUIREMENT DOCUMENT

---

---

Amin Abdelkefi - 000394225  
Lucie Borremans - 000393348  
Corentin Candeur - 000411488  
Jonas Descurieux - 000382867  
Rémy Detobel - 000408013  
Maxime Dewit - 000394939  
Carlos Requena López - 000410031  
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	Historique du document . . . . .	1
<b>2</b>	<b>Glossaire</b>	<b>1</b>
<b>3</b>	<b>Besoins de l'utilisateur</b>	<b>2</b>
3.1	Exigences fonctionnelles . . . . .	2
3.2	Exigences non fonctionnelles . . . . .	3
3.3	Exigences de domaine . . . . .	4
<b>4</b>	<b>Besoins du système</b>	<b>4</b>
4.1	Exigences fonctionnelles . . . . .	4
4.2	Exigences non fonctionnelles . . . . .	4
4.3	Design et fonctionnement du système . . . . .	5
4.3.1	Diagramme de séquence . . . . .	5
4.3.2	Diagramme d'activité . . . . .	5

# 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 architecture client/serveur avec le client qui sert principalement d'interface de jeu (graphique ou console) et les calculs qui seront effectués 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 connexion ainsi qu'un service de messagerie instantané entre deux joueurs. Pour cela, nous seront amené à développer une application capable de gérer un maximum de cas de figures pour garantir la continuité et la stabilité de ces différents 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'autres 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 Historique du document

Version	Auteur	Date modification	Description des changements
NEXT			
NEXT			
v0.5	Rémy Detobel	14/12/15	Première relecture, diverses petites corrections
v0.4	Carlos Requena	13/12/15 22h00	Ajout d'un système de glossaire
v0.3	Rémy et Lucie	11/12/15 16h20	Ajout de diagrammes, ajout de "Exigences du domaine"
v0.2	Amin Abdelkefi	10/12/15 23 :45	Ajout de la partie "Exigences fonctionnelles"
v0.1	Jonas et Raphaël	10/12/15 13 :28	Ajout de la partie "But du projet"
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 Glossaire

**Collection** Ensemble qui contient toutes les cartes disponibles et connues par le jeu

**Deck** Un deck est un ensemble de 20 cartes

**example1** description1

**example2** description2

**Tchat** Messagerie instantanée permettant à 2 personnes de s'échanger des messages écrit

**Visiteur** Personne n'ayant pas encore de compte

### 3 Besoins de l'utilisateur

Le UseCaseDiagram montre les différentes interaction entre les utilisateurs et le programme. Nous pouvons voir ici deux utilisateurs différents. Le visiteur est un utilisateur n'ayant

#### 3.1 Exigences fonctionnelles

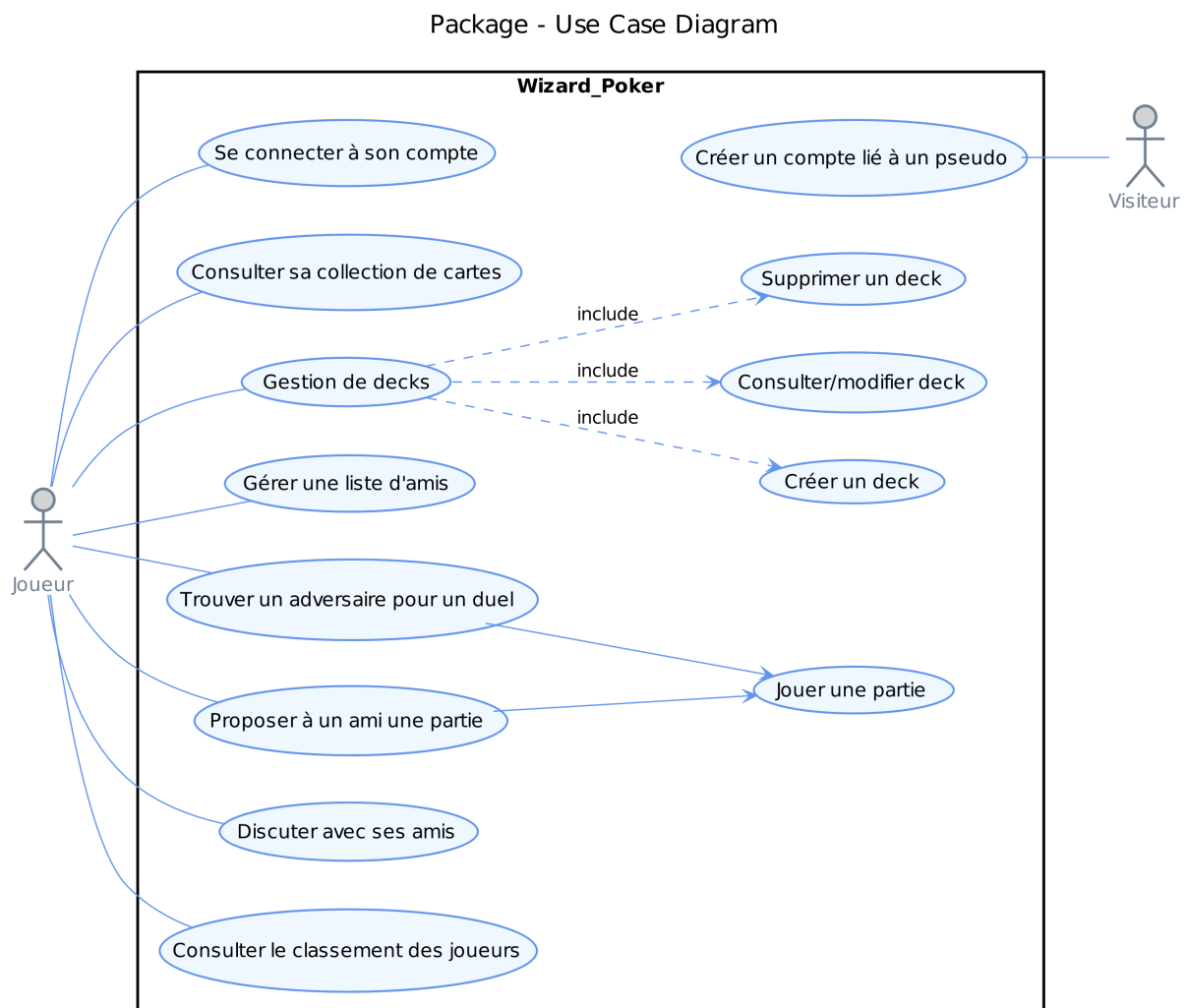


FIGURE 1 – Use Case Diagram

#### Créer un compte lié à un pseudo

Tout visiteur se verra offert l'opportunité de se créer un compte lié à un pseudo (si ce dernier est disponible) afin de pouvoir profiter pleinement du jeu.

**Se connecter à son compte**

Le joueur devra se connecter à son compte s'il souhaite effectuer les différentes actions répertoriées ci-dessous.

**Consulter sa collection de cartes**

Le joueur peut consulter l'ensemble des cartes présentes dans sa collection. Pour cela, il devra au préalable s'être connecté.

**Gestion de decks**

Un deck est un ensemble de cartes qui sera utilisé par le joueur lors d'un duel. Plusieurs actions peuvent être effectuées sur les decks telle qu'en supprimer un ou plusieurs, les consulter, les modifier ou tout simplement en créer un nouveau.

**Gérer une liste d'amis**

Chaque joueur aura une liste d'amis avec lesquels il pourra interagir s'il le souhaite. Il pourra aussi ajouter un amis à sa liste ou en supprimer.

**Discuter avec ses amis**

Le joueur pourra, à tout moment, parler avec un ou plusieurs de ses amis actuellement connectés. Pour cela, un tchat sera mis à disposition du joueur. Le joueur pourra donc communiquer avec ses amis par message à tout moment.

**Proposer une partie à un ami**

Si le joueur le souhaite, il peut proposer une partie avec un de ses amis connectés si ce dernier accepte cette demande.

**Trouver un adversaire pour un duel**

Lorsqu'un joueur voudra lancer une partie, le jeu trouvera automatiquement un adversaire dans la liste des personnes connectées. Le choix est totalement aléatoire.

**Consulter le classement**

Le classement va permettre de classer les joueurs en fonction de leur nombre de victoires et de leurs défaites. Ce classement se basera donc sur le ratio victoires/défaites.

**3.2 Exigences non fonctionnelles**

Dans le cadre de ce projet nous allons d'abord implémenter les exigences fonctionnelles. Nous verrons ensuite pour ajouter les exigences non fonctionnelles.

**Ergonomie**

Créer un jeu qui sera simple d'utilisation et facilement compréhensible pour un utilisateur n'ayant jamais joué à un jeu de ce genre. Pour cela, on peut rendre l'interface intuitive à l'aide de "boutons" décrivant exactement leurs utilités ou encore à l'aide d'un rapide tutoriel après la création du compte expliquant les différentes caractéristiques du jeu.

## Design

Le design est l'apparence qu'aura l'interface graphique. Comme pour le point précédent, une belle interface va permettre à l'utilisateur de facilement voir les différentes possibilités offertes par le jeu en plus d'apprécier l'esthétique.

Le design est un "domaine" (vague) c'est-à-dire qu'il définit l'apparence du jeu mais aussi les différents sons et/ou musiques.

## S'amuser

Qu'est-ce qu'un bon jeu ?

Un bon jeu est un jeu attractif, beau, facile à comprendre mais avant tout, un bon jeu est jeu où l'on s'amuse. Notre principal objectif sera de réaliser ce jeu et de permettre à tout les joueurs de s'amuser. (????? A changer si vous le souhaitez)

### 3.3 Exigences de domaine

Le domaine du jeux vidéo laisse beaucoup de liberté. En effet, chaque créateur essaye de créé son univers, de développer son concept. Cependant, tous les jeux doivent permettre à leur utilisateur de trouver du plaisir ou une certaines forme de satisfaction. Il faut également éviter que le joueur soit interrompu durant sa partie. Les exigences du domaine ne sont donc pas nombreuses mais compliqué à satisfaire complètement.

Le Wizard Poker est un jeu de carte. Ce type de jeux impose lui aussi quelques contraintes. Il faut éviter que les cartes soient trop fortes. Pour la durée de vie du jeu il faut aussi faire en sorte qu'il y ai un nombre conséquent de cartes. N'oublions pas enfin le temps de jeu. En effet, une partie de carte dure rarement longtemps pour éviter l'ennui.

Pour résumé, un utilisateur attend de l'amusement, un jeu équilibré et une stabilité acceptable (voir infailible coté serveur). Nous nous concentrerons dans un premier temps (dans le cadre de ce projet) à ce troisième point et nous verrons pour satisfaire à 100% les "gamers" dans un second temps.

## 4 Besoins du système

### 4.1 Exigences fonctionnelles

Sequence diagram pour interaction ??

Le système doit interagir avec l'utilisateur, et lui proposer certaines actions, comme commencer un duel, gérer ses s, 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.

### 4.2 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.

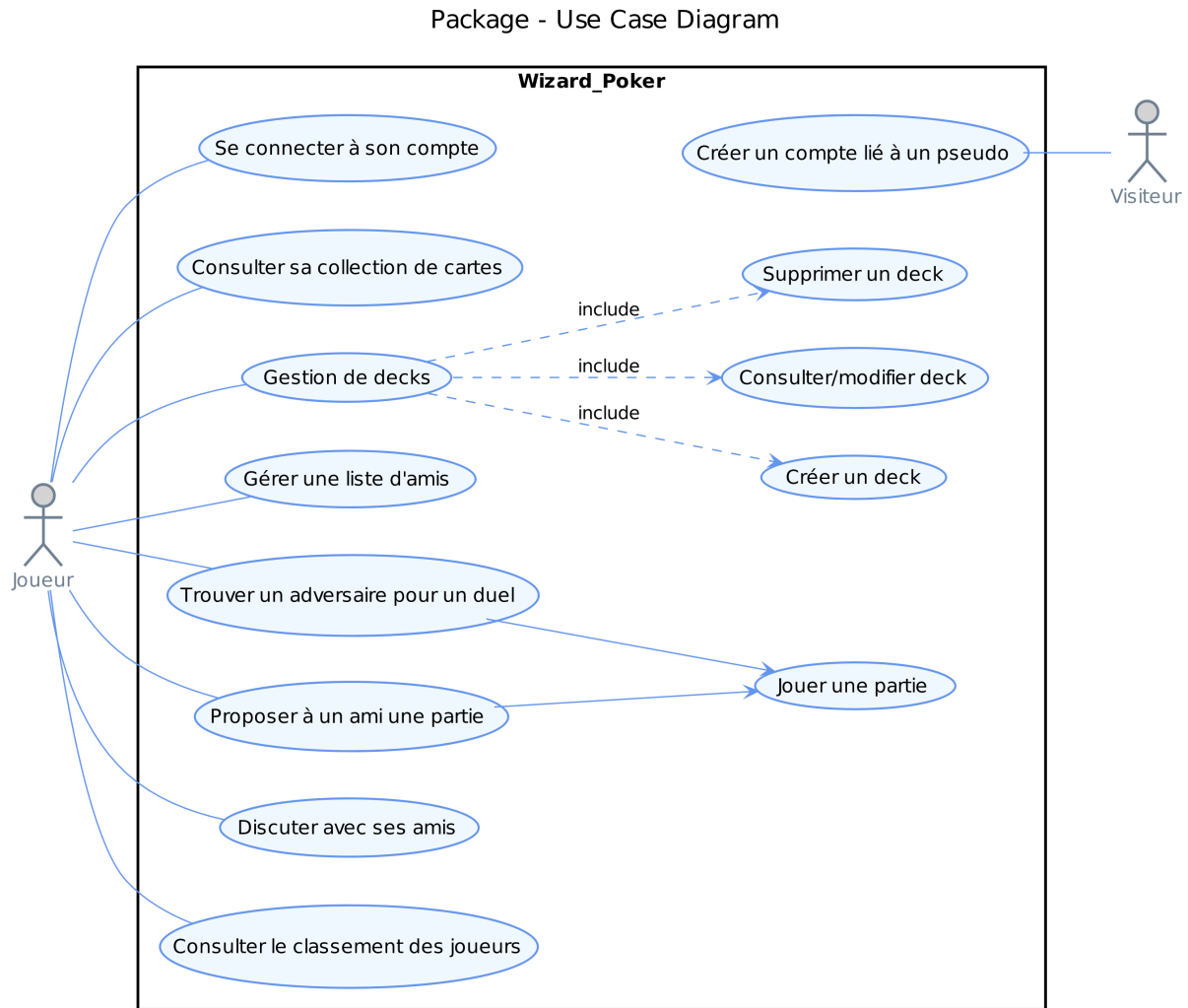


FIGURE 2 – Use Case Diagram du jeu en question

### 4.3 Design et fonctionnement du système

A fin de mieux présenter le fonction du programme voici quelques diagrammes où vous pouvez voir la manière donc nous allons essayer de développer Wizard Poker.

#### 4.3.1 Diagramme de séquence

Ce diagramme de séquence vise à montrer la manière dont un joueur va pouvoir initialiser une partie. Les diagrammes d'activités vont permettre de détaillé la manière dont une partie se déroule.

#### 4.3.2 Diagramme d'activité

Le but de ces deux diagrammes d'activité est de décrire comment va se dérouler une partie.

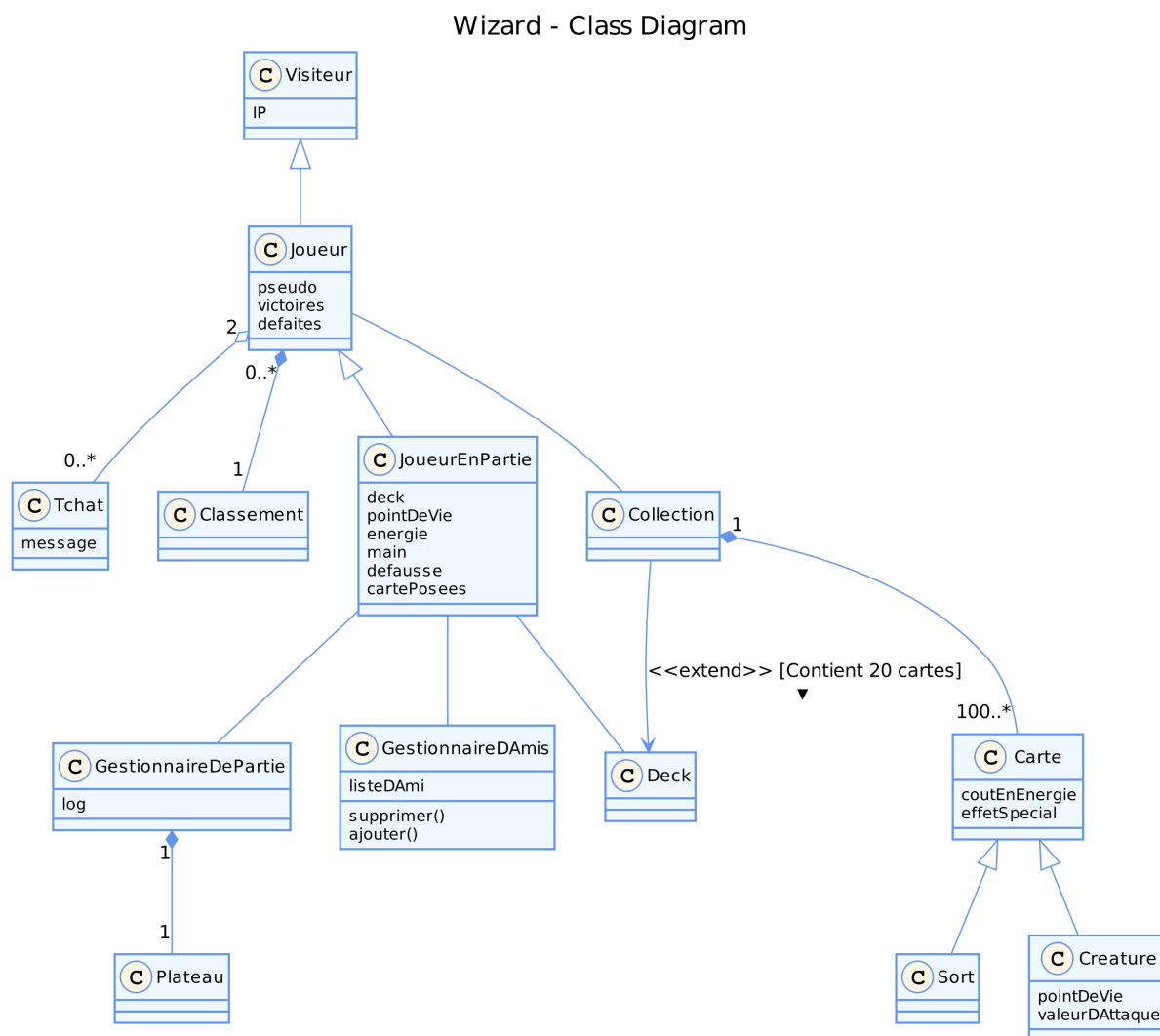


FIGURE 3 – Diagramme des classes



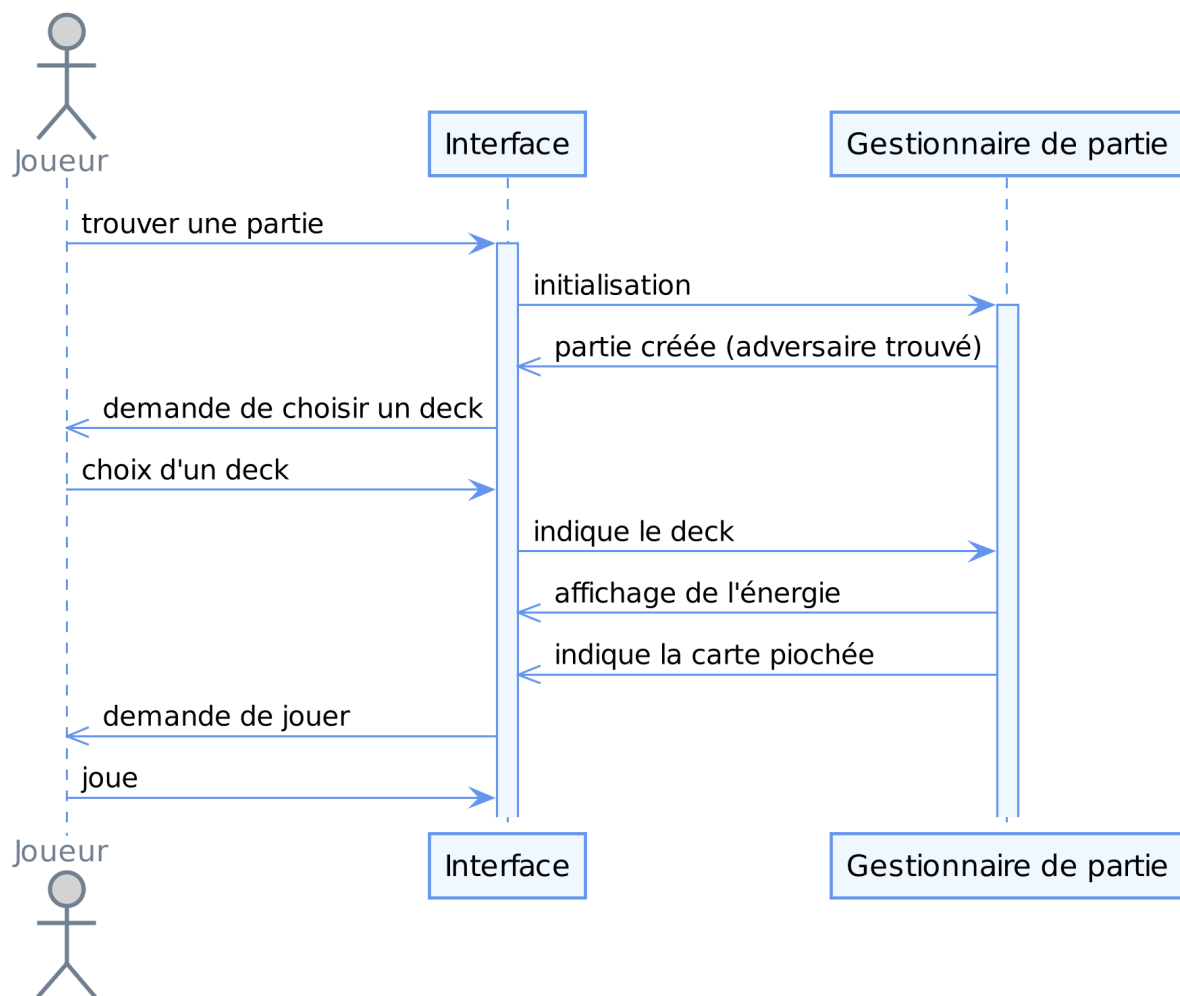
**Wizard - Sequence Diagram**

FIGURE 4 – Diagramme de séquence

## Wizard - Activity Diagram

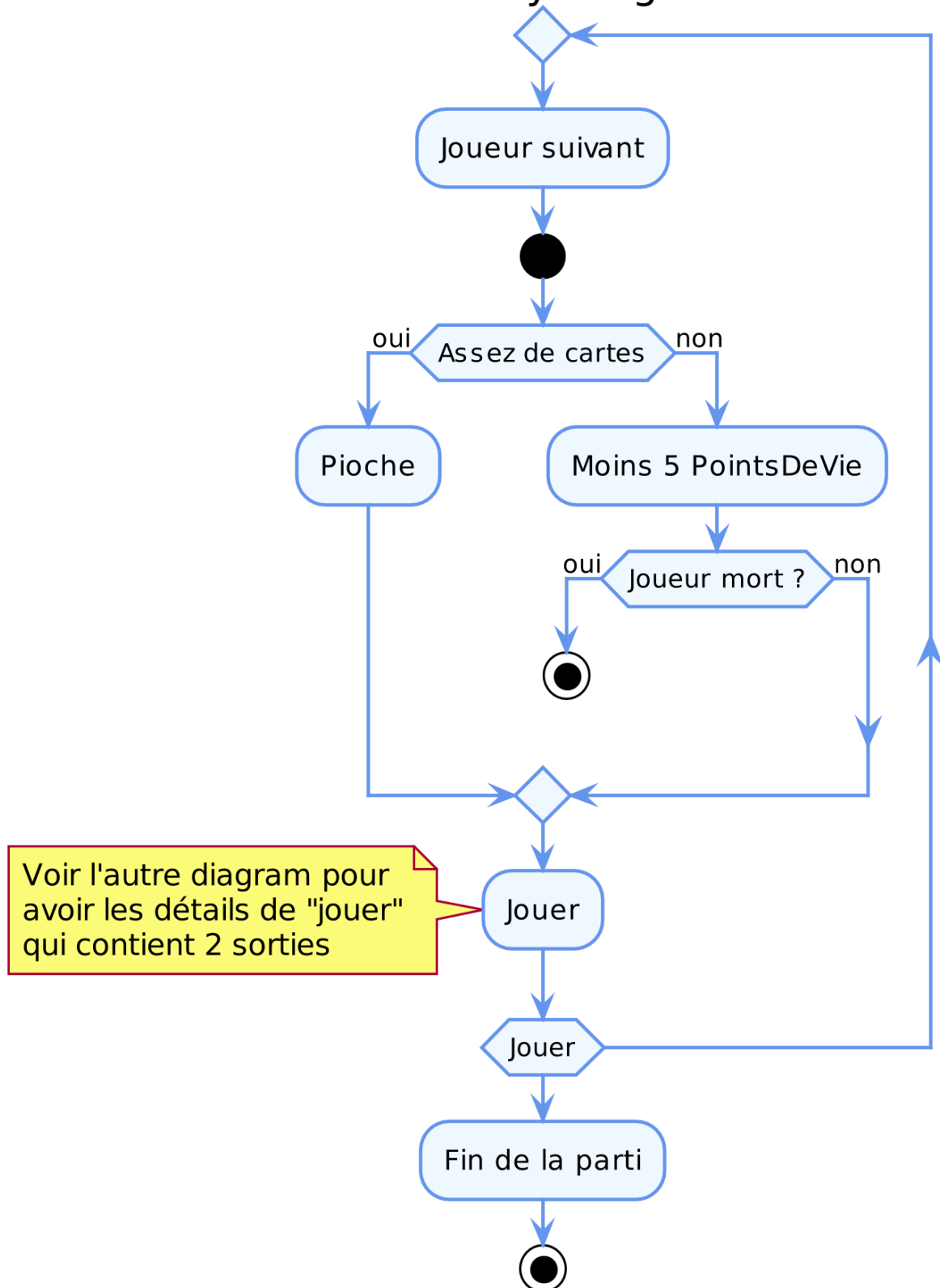


FIGURE 5 – Diagramme d'activité

## Wizard - Activity Diagram (fin de tour de jeu d'un joueur)

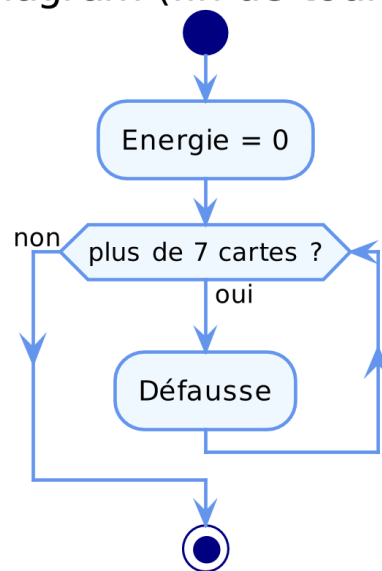


FIGURE 6 – Diagramme d'activité 2

## **Index**

deck, 4