
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

1	Introduction	1
1.1	But du projet	1
1.2	Historique du document	1
2	Glossaire	1
3	Besoins de l'utilisateur	2
3.1	Exigences fonctionnelles	2
3.2	Exigences non fonctionnelles	4
3.3	Exigences de domaine	4
4	Besoins du système	4
4.1	Exigences fonctionnelles	4
4.2	Exigences non fonctionnelles	5
4.3	Design et fonctionnement du système	5
4.3.1	Diagramme de séquence	5
4.3.2	Diagramme d'activité	6

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 de type HeartStone.

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'un ensemble de cartes avec lesquelles il pourra affronter ses adversaires. Ces cartes pourront être des créature qui peuvent attaquer et des cartes de type sort qui lanceront des événements spéciaux.

Le projet fournira à l'utilisateur une interface intuitive, un gestionnaire d'amis, un service de chat pour pouvoir rester en contact avec eux ainsi qu'un classement générale où chaque joueurs pourra comparer ses résultats avec les autres.

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	Changelog
NEXT			
NEXT			
v0.6	Corentin Candeur	16/12/15	Deuxième relecture, corrections orthographiques/syntaxiques et diverses petites corrections
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 et Maxime	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	Mise en place du document. Ajout des diagrammes réalisés en groupe le 25/11/15.

TABLE 1 – Changements document

2 Glossaire

Application/système Le jeu Wizard Poker à développer.

Carte asdfasdf Wizard Poker

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

Créature Carte ayant un coût en énergie, valeur d'attaque et des points de vie. Aussi (optionnellement) un effet spécial.

Deck Un deck est un ensemble de 20 cartes

Énergie Permet de jouer des cartes. Quantité, entier.

exemple1 description1

exemple2 description2

Sort Carte ayant un coût en énergie et un effet spécial.

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 Use Case Diagram montre les différentes interactions entre les utilisateurs et le programme. Nous pouvons voir ici deux utilisateurs différents. Le visiteur est un utilisateur n'ayant pas de compte et ses actions sont donc très limitées et consistent seulement en la création d'un compte. Nous avons aussi le joueur qui est l'utilisateur principal du Wizard Poker. C'est lui qui va interagir avec le jeu.

3.1 Exigences fonctionnelles

Créer un compte lié à un pseudo

Tout visiteur se aura la possibilité 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 telles 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 ou supprimer un ami de sa liste.

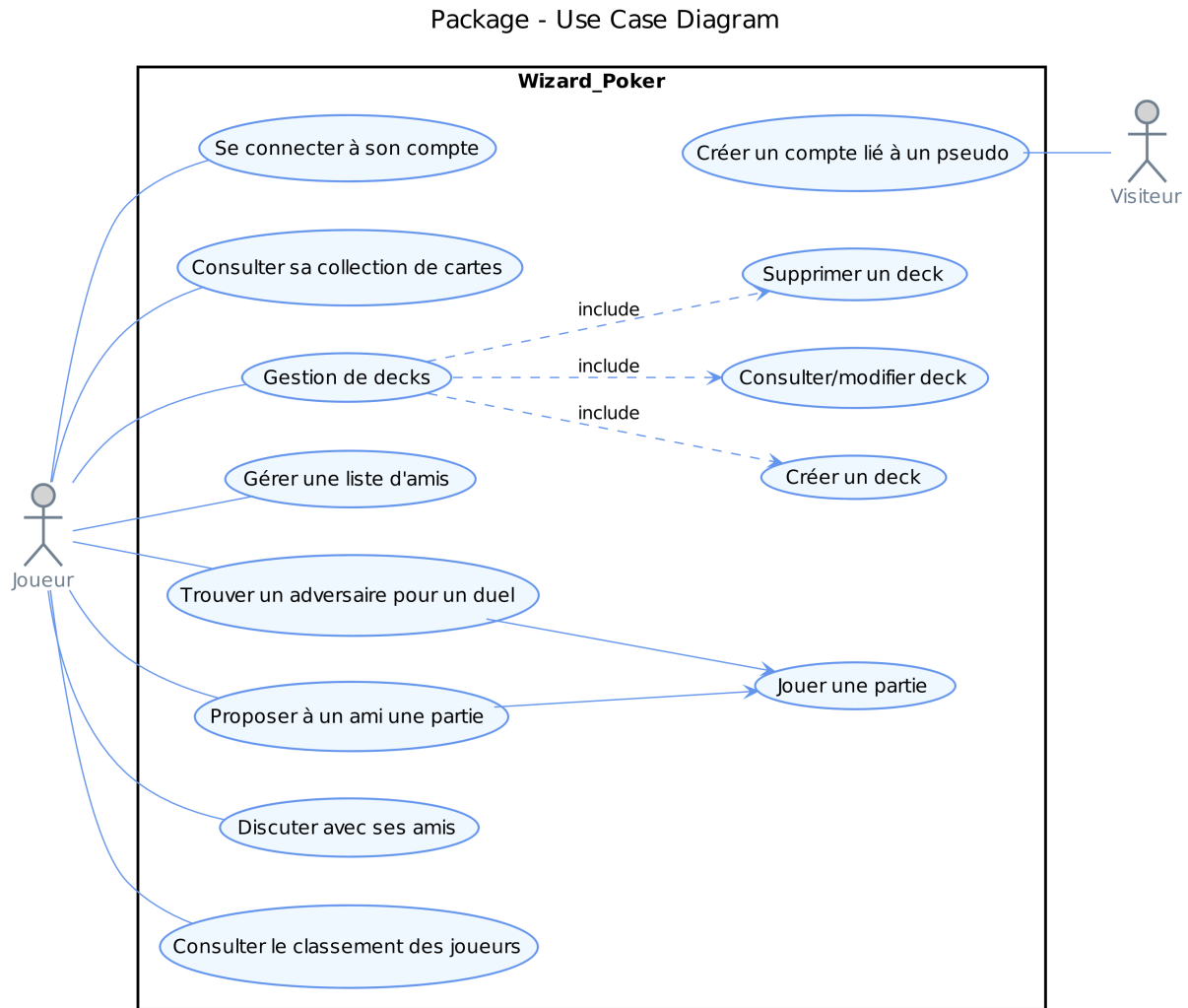


FIGURE 1 – Use Case Diagram

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. Cependant un joueur ne peut communiquer qu'avec une personne à la fois.

Proposer une partie à un ami

Si le joueur le souhaite, il peut proposer une partie à un de ses amis connectés. Ce dernier est libre d'accepter ou non cette demande. S'il l'accepte la partie est lancée entre les deux joueurs.

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 de l'adversaire est totalement aléatoire.

Consulter le classement

Le classement va permettre de classer les joueurs en fonction de leur nombre de victoires et de 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 leur utilité 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 un jeu où l'on s'amuse. Notre principal objectif sera de réaliser ce jeu et de permettre à tous les joueurs de s'amuser.

3.3 Exigences de domaine

Le domaine du jeu vidéo laisse beaucoup de liberté. En effet, chaque créateur essaye de créer son univers, de développer son concept. Cependant, tous les jeux doivent permettre à leur utilisateur de trouver du plaisir ou une certaine 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 compliquées à satisfaire complètement.

Le Wizard Poker est un jeu de cartes. Ce type de jeu 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 ait un nombre conséquent de cartes. N'oublions pas enfin le temps de jeu. En effet, une partie de cartes dure rarement longtemps pour éviter l'ennui.

Pour résumer, un utilisateur attend de l'amusement, un jeu équilibré et une stabilité acceptable (voir infaillible 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

Les exigences fonctionnelles expliquent ce que l'application doit faire pour interagir avec l'utilisateur, et lui proposer toutes les actions mentionnés ci-dessus (voir 3.1).

Dans cette partie on détaillera comment le système va se débrouiller pour réaliser ces tâches.

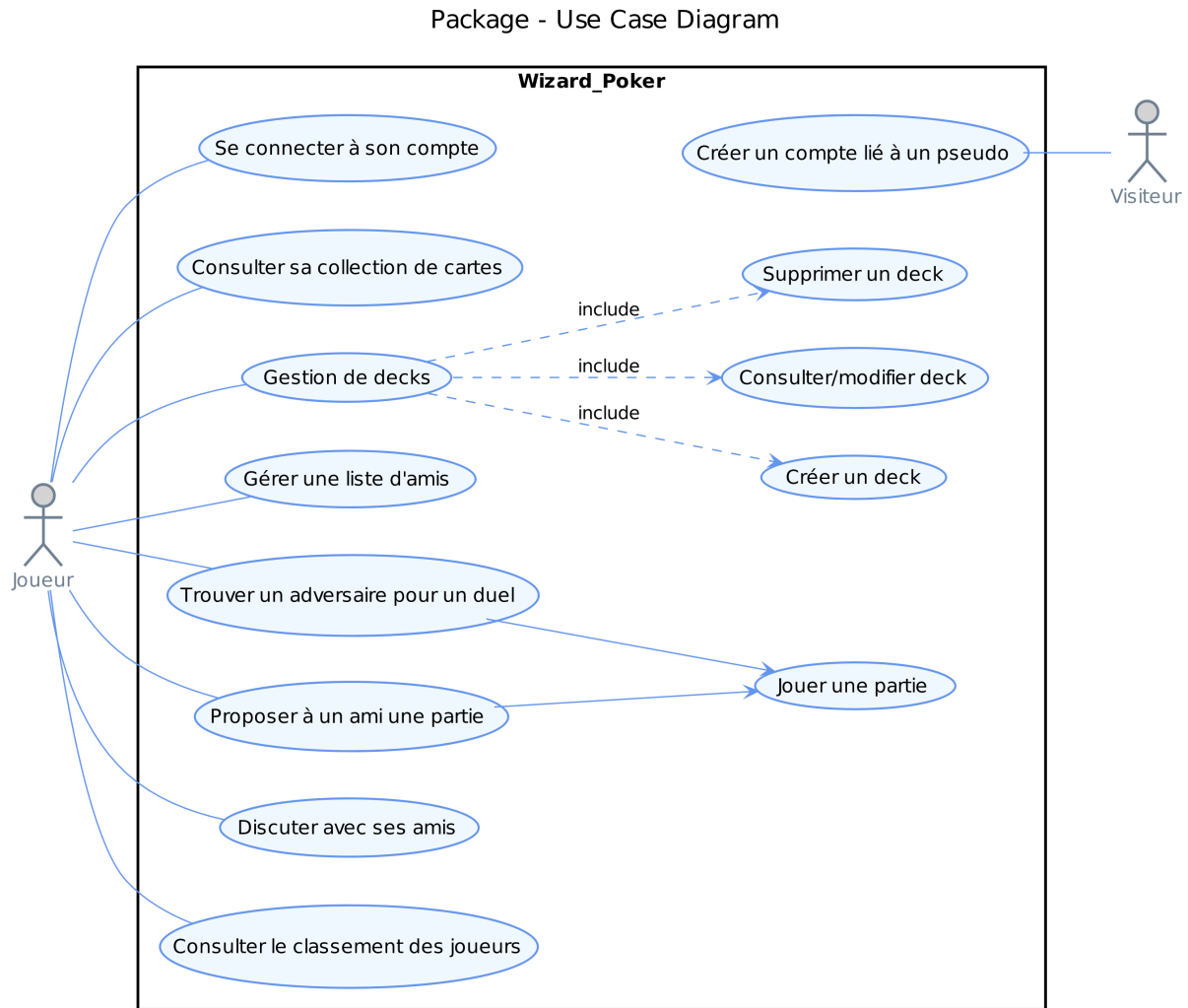


FIGURE 2 – Use Case Diagram du jeu en question

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 tous les erreurs possibles et éviter les crashes et eventuelles 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.

4.3 Design et fonctionnement du système

Afin de mieux présenter le fonctionnement du programme voici quelques diagrammes où vous pouvez voir la manière dont 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 détailler la manière dont une partie se déroule.

Wizard - Class Diagram

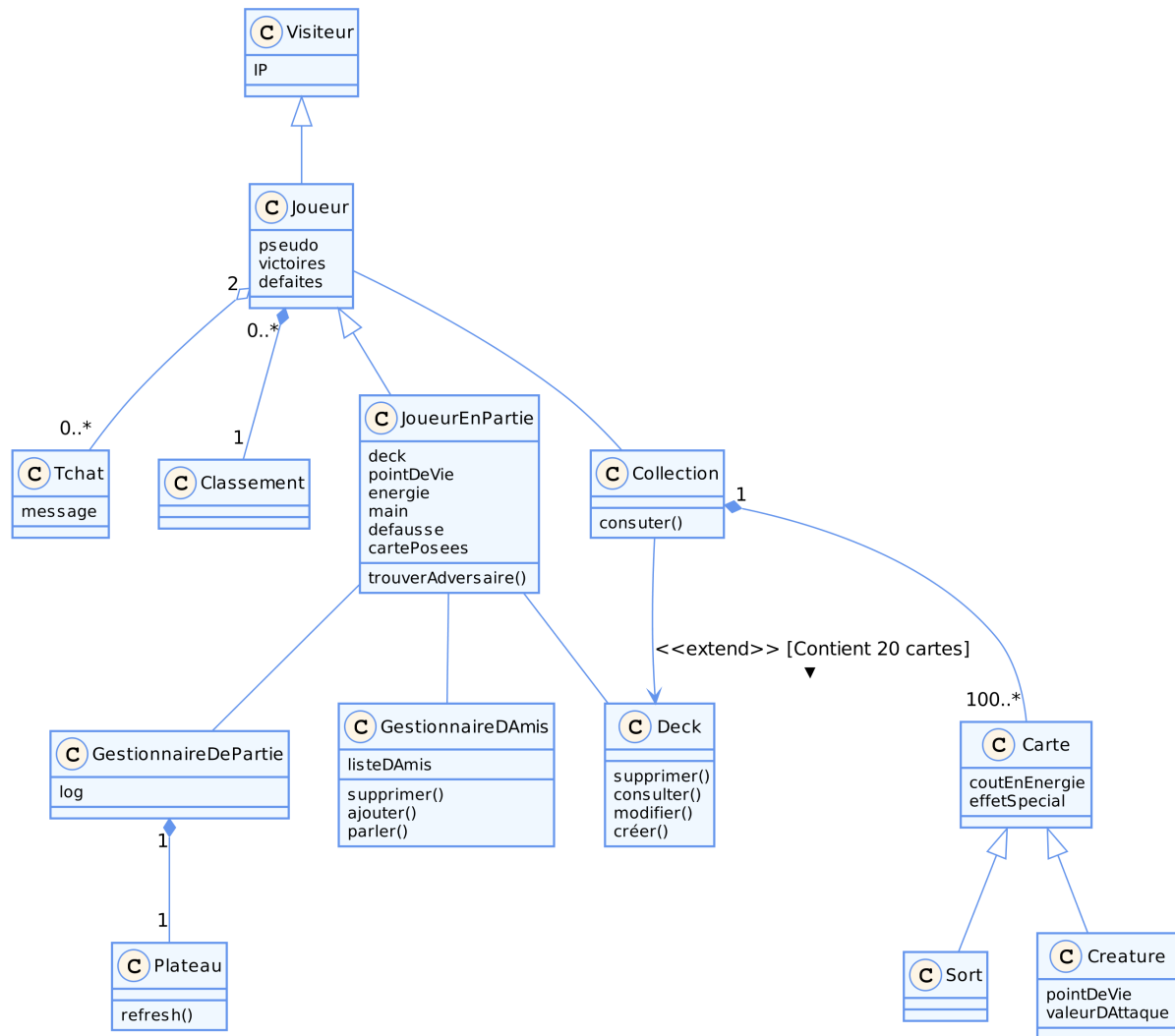


FIGURE 3 – Diagramme des classes

4.3.2 Diagramme d'activité

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

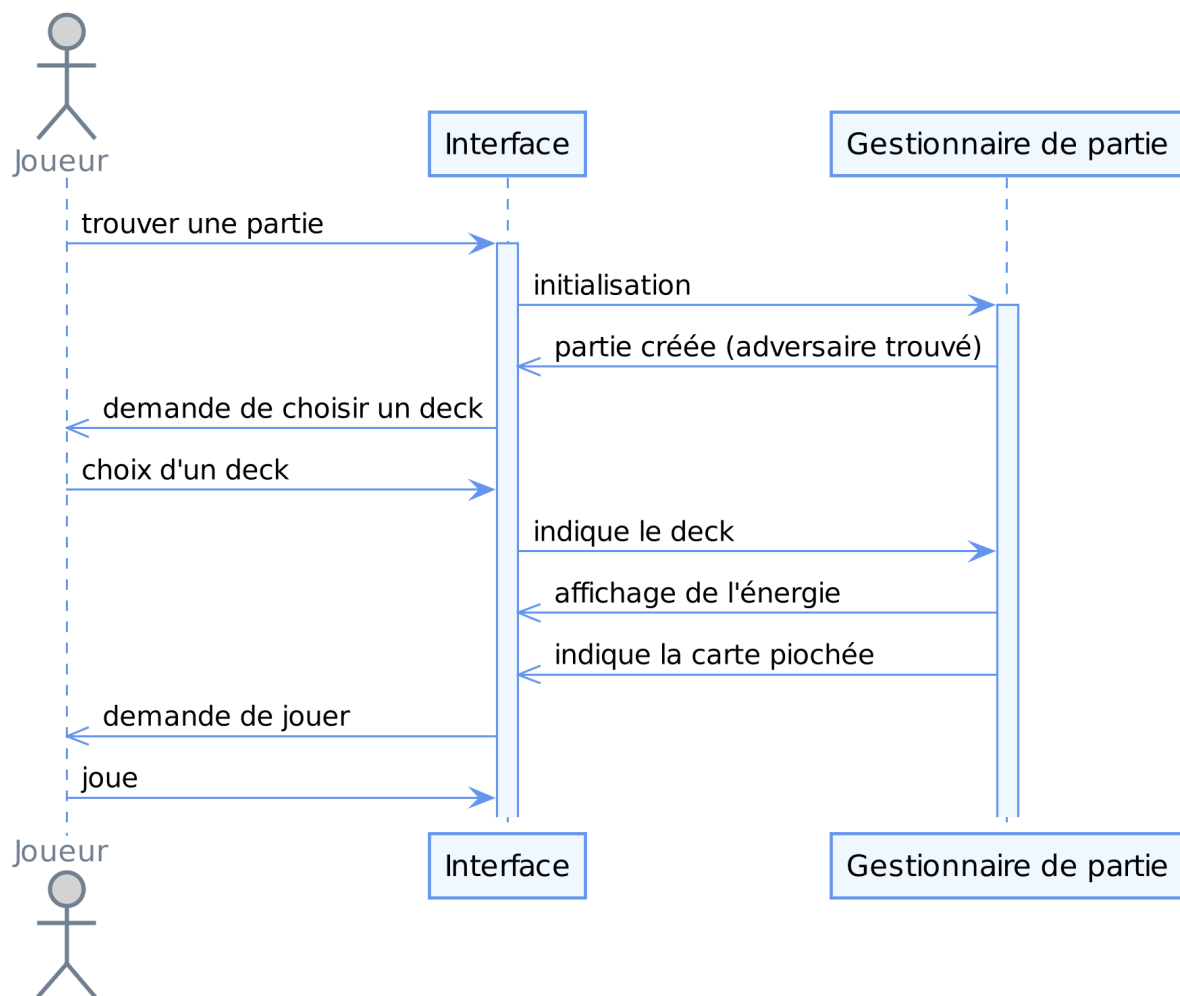
Wizard - Sequence Diagram

FIGURE 4 – Diagramme de séquence

Wizard - Activity Diagram

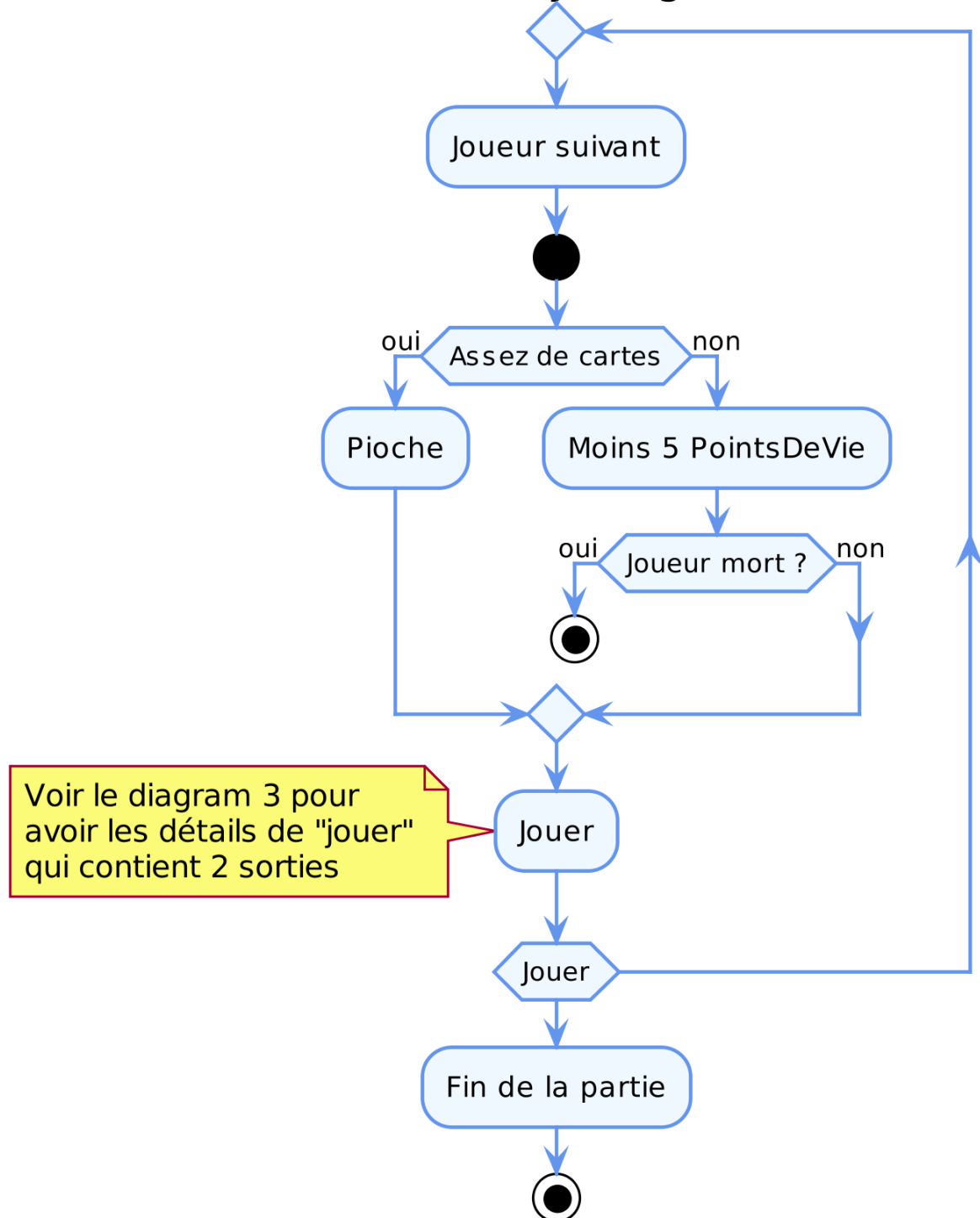


FIGURE 5 – Diagramme d'activité

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

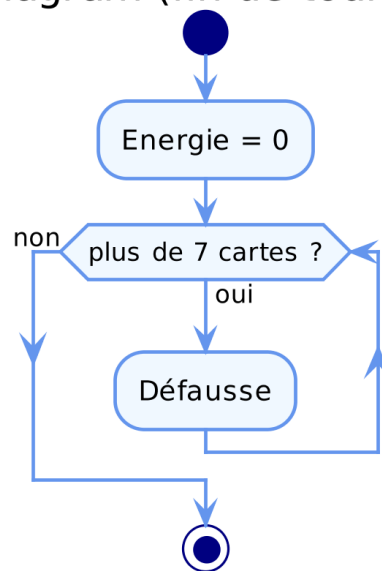


FIGURE 6 – Diagramme d'activité 2