UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F209 : PROJET D'INFORMATIQUE II

WIZARD POKER

Software Requirement Document

Auteurs :

Yasin ARSLAN
Youcef BOUHARAOUA
Jalal NAH
Sacha MEDAER
Youssef SITIL
Miguel TEROL ESPINO
Jacky TRINH

Titulaires :
Joël GOOSSENS
Christian HERNALSTEEN
Assistants :
Keno MERCKX
François GÉRARD
Jacopo DE STEFANI

18 décembre 2015

Table des matières

Chapitre 1

Introduction

1.1 But du projet

Ce document a pour but de présenter, dans le cadre d'une deuxième année de Bachelier en Sciences informatiques, une description détaillée de notre projet multidisciplinaire. Le but, les caractéristiques, l'interface du système ainsi que les contraintes auxquelles le programme est soumis sont expliquées. Ce présent rapport est destiné aussi bien au client qu'au développeur soucieux de comprendre l'architecture du programme.

1.2 Cadre du projet

Ce programme, nommé *Wizard Poker* est un jeux de cartes fantastiques dans lequel des joueurs s'affrontent dans des duels un contre un. Ces joueurs portent le nom de *player*. Chaque player possède une collection de cartes à jouer lui permettant de former des paquets appelés *decks*. Lors de la création d'un nouveau player dans le système, il reçoit un certain de nombre de cartes lui permettant de créer son premier deck. Il gagnera des cartes supplémentaires en récompense de duels victorieux.
Il existe deux types de cartes. Le premier type est composé de créatures appelées *minion*, carte ayant un coût en énergie, une valeur d'attaque et un montant de points de vie. Le deuxième type est constitué de sorts appelés *spell*, cartes ayant un coût en énergie et un effet spécial.
Chaque deck est composé de 20 cartes sans restrictions sur la proportion de créatures et de spell. Le player peut mettre deux fois maximum la même carte dans un deck mais il faut qu'il possède alors deux exemplaires de la carte dans sa collection
Le déroulement d'un duel s'effectue comme suit. Les deux players choisissent un deck avec lequel ils veulent jouer avant le début du duel. Ensuite, chacun d'eux commence la partie avec un total de 20 points de vie et 5 cartes en main. Un des deux players choisi aléatoirement joue le premier tour. À chaque début de tour, le player pioche une carte et gagne un point d'énergie supplémentaire, avec un maximum de 10. Il commence à 1 point d'énergie au premier tour. Durant son tour, le player peut attaquer avec ses minions et jouer des cartes de sa main en dépensant un montant d'énergie égal au coût de la carte. Si la carte jouée est un minion, elle est placée sur le plateau de jeu. Si la carte jouée est un spell, son effet spécial est réalisé et la carte défaussée. Quand il le désire, le player peut mettre fin à son tour. Il perd alors les points d'énergie qui lui

reste et le tour de son adversaire. Les tours des deux players sont donc asynchrones. Le player dont le total de points de vie est réduit à 0 perd la partie. Si un player n'a plus de cartes à piocher dans son deck, il perd 5 points de vie à chaque début de tour.
Les minions présents sur le plateau peuvent être utilisées un tour après leur mise en jeu. Si un minion attaque le player adverse, les points de vie de ce dernier sont réduits d'un montant égal à la valeur d'attaque du minion. Si un minion attaque un minion adverse, les points de vie de chacune des deux sont réduits de la valeur d'attaque de l'autre. Quand un minion meurt, elle est défaussée.
Le jeu permet à tout utilisateur de créer un compte associé à un pseudonyme, gérer une liste d'amis afin de pouvoir discuter avec ceux-ci et consulter un classement de tous les players . Il permet également de consulter sa collection de cartes, créer des decks à partir de cette collection et enfin lancer une partie.

1.3 Glossaire

Mot	**Définition**
Duel	Combat singulier entres 2 adversaires
Game	Duel lancé entre 2 joueurs
Player	Joueur lors d'un duel
Deck	Les cartes choisi par le joueur pour le duel
Collection	Toutes les cartes accessibles d'un joueur
Spell	Une carte de type sortilège
Minion	Une carte de type créature
Guest	Un invité n'ayant pas de compte associé au jeu
Ranking	Le classement des joueurs
Matchmaking	La mise en relation des joueurs pour le duel
Chat	Système de conversations entre joueurs
Health Points (hp)	Points de vie
Energy	Point d'énergie
Character	Une cible potentielle
Graveyard	Cimetière pour créature
Contre-attaque	Offensive lancée en réponse à une attaque
Database	Zone de stockage
IA	Intelligence artificielle
ActionManager	Système gérant les interactions lors d'une partie

1.4 Historique du document

Date	**Version**	**Auteurs**	**Comment**
16/12/15	1.0	groupe 8	–

1.5 Vue d'ensemble

Cette section décrit les différentes parties du document. La première, reprend les services que le système doit fournir à l'utilisateur ainsi que les contraintes de fonctionnement du système. Ceci est réalisé notamment grâce à des diagrammes use case. la deuxième partie, quant à elle, décrit en détail les fonctionnalités du système. Des diagrammes use case ainsi que des diagrammes UML sont utilisés.

Chapitre 2

Besoins de l'utilisateur

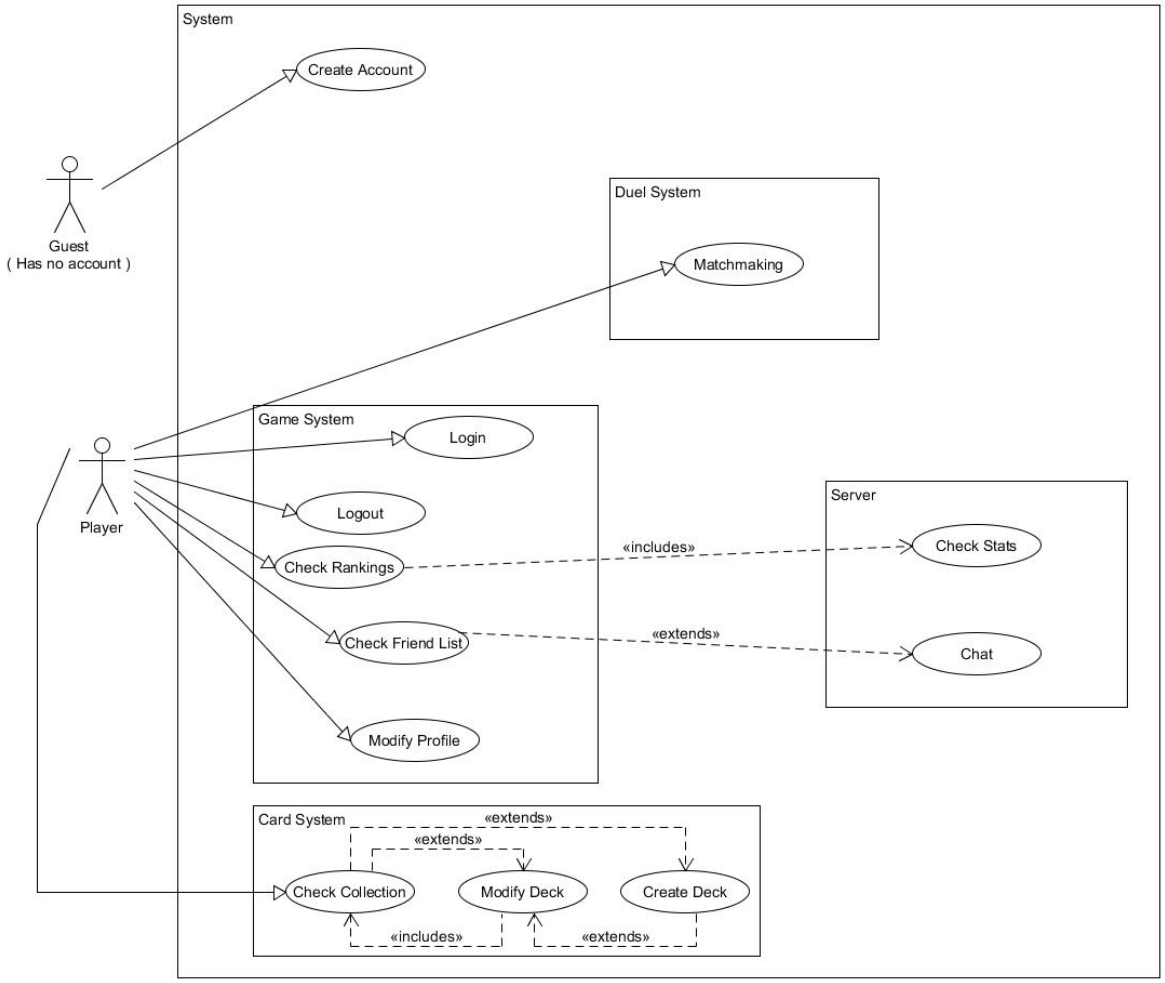


FIGURE 2.1 – Diagramme Use case du système

2.1 Exigences fonctionnelles

2.1.1 Gestion des utilisateurs

— Cas général : L'utilisateur doit pouvoir s'inscrire (s'enregistrer) sur le serveur avec un nom d'utilisateur unique et un mot de passe.

— Pré Condition : L'utilisateur ne possède pas déjà un compte.

— Post Condition : L'utilisateur aura fait une demande d'inscription.

— Cas Exceptionnel : L'inscription échouera et l'utilisateur sera invité à recommencer si le nom de compte est déjà enregistré sur le serveur.

2.1.2 Connexion

— Cas général : L'utilisateur doit pouvoir se connecter auprès du serveur avec son identifiant et son mot de passe.

— Pré Condition : L'utilisateur ne s'est pas déjà connecté.

— Post Condition : L'utilisateur aura fait une demande de connexion.

— Cas Exceptionnel : L'authentification échouera et l'utilisateur sera invité à recommencer si le mot de passe et le pseudonyme ne correspondent pas ou que celui-ci n'est pas reconnu sur le serveur.

2.1.3 Jouer un tour

— Cas général : L'utilisateur doit pouvoir jouer un tour lors d'un duel, c'est à dire placer ses cartes et/ou effectuer des actions.

— Pré Condition : L'utilisateur doit être dans un duel.

— Post Condition : L'utilisateur aura joué ses cartes et actions.

— Cas Exceptionnel : Si la connexion est perdu par un des utilisateurs, celui-ci est déclaré comme perdant.

2.1.4 Déclarer forfait

— Cas général : L'utilisateur peut déclarer forfait durant un duel.

— Pré Condition : L'utilisateur doit être dans un duel.

— Post Condition : L'utilisateur sera déclaré comme perdant.

2.1.5 Créer un deck

— Cas général : Création d'un nouveau deck via sa collection.

— Pré Condition : L'utilisateur est connecté au serveur et est dans le menu adéquat.

— Post Condition : Le deck contient 20 cartes et respecte les conditions d'un deck.

2.1.6 Matchmaking

— Cas général : L'utilisateur peut initialiser une recherche avec le Matchmaking.

— Pré Condition : L'utilisateur ainsi que son adversaire sont connectés et ne sont pas dans un duel.

— Post Condition : L'adversaire est connecté aussi en Matchmaking.

2.2 Exigences non fonctionnelles

2.2.1 Règles du jeu

— Il existe deux types de cartes :

- Les spells sont des cartes ayant un coût en énergie et un effet spécial. Leurs effets varient en fonction de la carte et sont les suivants :
  * Infliger des dégâts
  * Soigner une ou plusieurs cibles
  * Faire apparaître une minion aléatoire
  * ...
- Les minions sont des cartes ayant un coût en énergie, une valeur d'attaque et un montant de points de vie, également appelé Hp. Elles peuvent éventuellement avoir des effets spéciaux tels que :
  * Faire piocher une carte
  * Augmenter les statistiques d'une autre créature
  * Invoquer un autre minion
  * ...

— Un deck contient exactement 20 cartes. Les doublons sont acceptés mais le joueur ne peut avoir plus de 2 fois la même carte dans un deck.

— Les points d'énergie peuvent être cumulés. Chaque player commence la partie avec 0 point d'énergie. Au début d'un tour, un player voit ses points d'énergie reinitialisés à un. Chaque player peut cumuler jusqu'à 10 points.

— A chaque début de tour, le player pioche une carte. Si il en possède déjà 10 en main, cette carte est detruite.

— Les players commencent la partie avec 5 cartes en main.

— Un système va décider aléatoirement quel player à la main au premier tour.

— Pseudo code d'un début de tour :

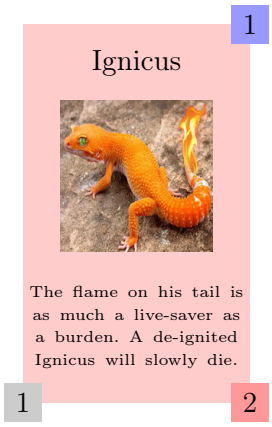
* $maxPossibleEnergy{+}{+}$ **if** $maxPossibleenergy < 10$ **else** do nothing
* $currentPlayer.drawCard()$ **if** $currentPlayer.handSize < 10$ **else** $discardCardToDraw()$
* $currentPlayer.currentEnergy = currentPlayer.maxPossibleEnergy$

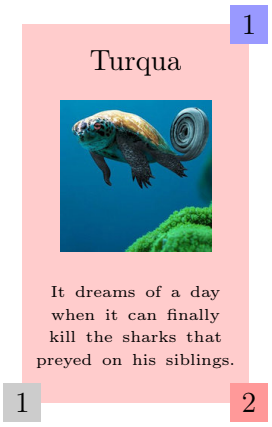
2.2.2 Choix Personnel

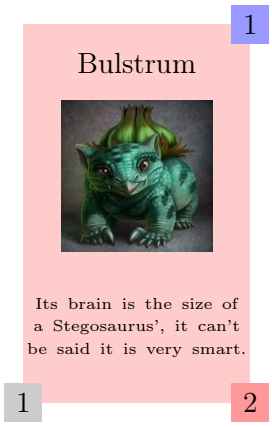
Afin de rendre le gameplay plus intéressant, l'équipe a décidée d'instaurer un système de *style*. Un style défini un gameplay, il existe dans l'univers du Wizard Poker plusieurs styles. Les différents styles auront leurs propres cartes et partagera tous ensembles des cartes neutre.
De la sorte, le Player se voit offrir la possibilité de choisir le gameplay qu'il aime. L'équilibre des cartes et du jeu seront administrés par les soins de l'équipe. Voici deux exemples de styles ainsi que des prototypes de cartes.

2.2.2.1 Style SinuMonstra

Ce style se base sur le populaire jeu Pokémon. Les Minions de cette classe ont une spécialité. Elles peuvent évolué. Cette évolution restaure l'entièreté des Hps mais la rend également plus fort. Les cartes de ce style sont principalement des Minions. Mais les quelques spells de ce style ne sont pas à négliger.

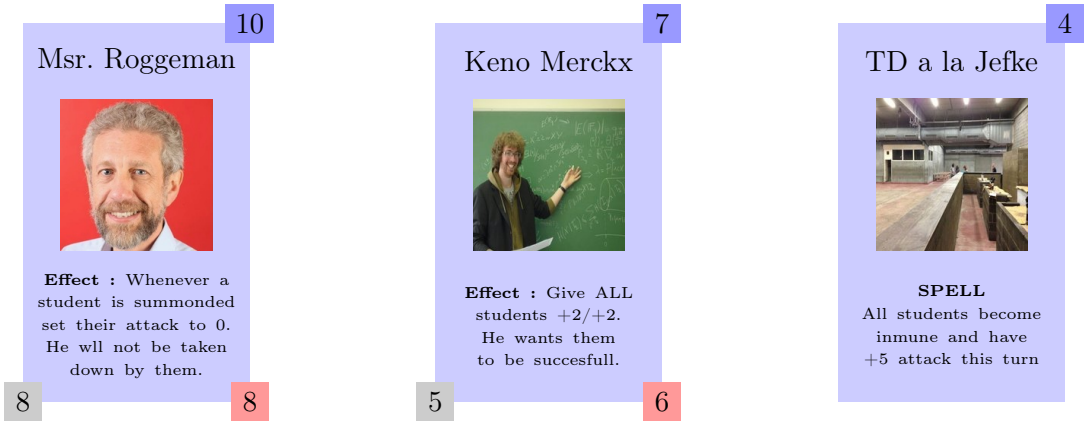






2.2.2.2 Style ULB

Ce style se base sur l'Université Libre de Bruxelles. Elle comporte des Minions surpuissants qui sont très efficaces contre des Minions de type Étudiant, mais elles peuvent également être très puissante au coté d'étudiants.



2.3 Exigences du domaine

Le domaine de ce projet étant le domaine du jeu-video, il n'y a pas de contraintes communes et absolues à tous les jeux-vidéos. Ceci nous laisse donc une grande liberté de prise de décision sur les contraintes du domaine de Wizard Poker.

Chapitre 3

Besoins du système

3.1 Exigences fonctionnelles

3.1.1 Use case view

3.1.1.1 Gestion des utilisateurs

L'utilisateur, désireux de jouer, devra passer par un système d'inscription auprès du serveur. L'inscription demandera uniquement un identifiant ainsi qu'un mot de passe. Une fois ces informations données, il devra vérifier si cette identifiant n'est pas déjà utilisé. S'il ne l'est pas, le compte sera créé instantanément et l'utilisateur pourra s'authentifier. Les Accounts seront stockés dans la Database du système. Celle-ci n'a pas encore été réfléchi.

3.1.1.2 Connexion

L'utilisateur se connectera au jeu grâce à son identifiant et son mot de passe. Cette connexion se fera auprès du serveur. Elle se chargera de valider la connexion. Pour ce faire, l'identifiant et le mot de passe doivent correspondre à un Account déjà créé et reconnu par le système. Une fois validé, il pourra accéder aux divers fonctionnalités du système.

3.1.1.3 Jouer un tour

Le Player dont c'est le tour aura plusieurs options pour jouer son tour. Les options se limitant aux cartes qu'il a dans sa main et à ses points d'énergie. Les divers options sont illustrés dans la figure suivante.

3.1.1.4 Déclarer forfait

Tout le monde s'est déjà rendu dans une situation où la victoire n'est plus du tout envisageable. Afin de gagner du temps, l'utilisateur aura la possibilité de déclarer forfait. Ceci mettra donc un terme à la partie en cours.

3.1.1.5 Créer un deck

Comme déjà expliqué plus haut, l'utilisateur a le droit de créer son propre deck suivant ses désirs. La création d'un deck se fera à partir de la collection de l'utilisateur et sera associé à son account.

Il pourra enregistrer un certain nombre de deck et en choisir un actif.

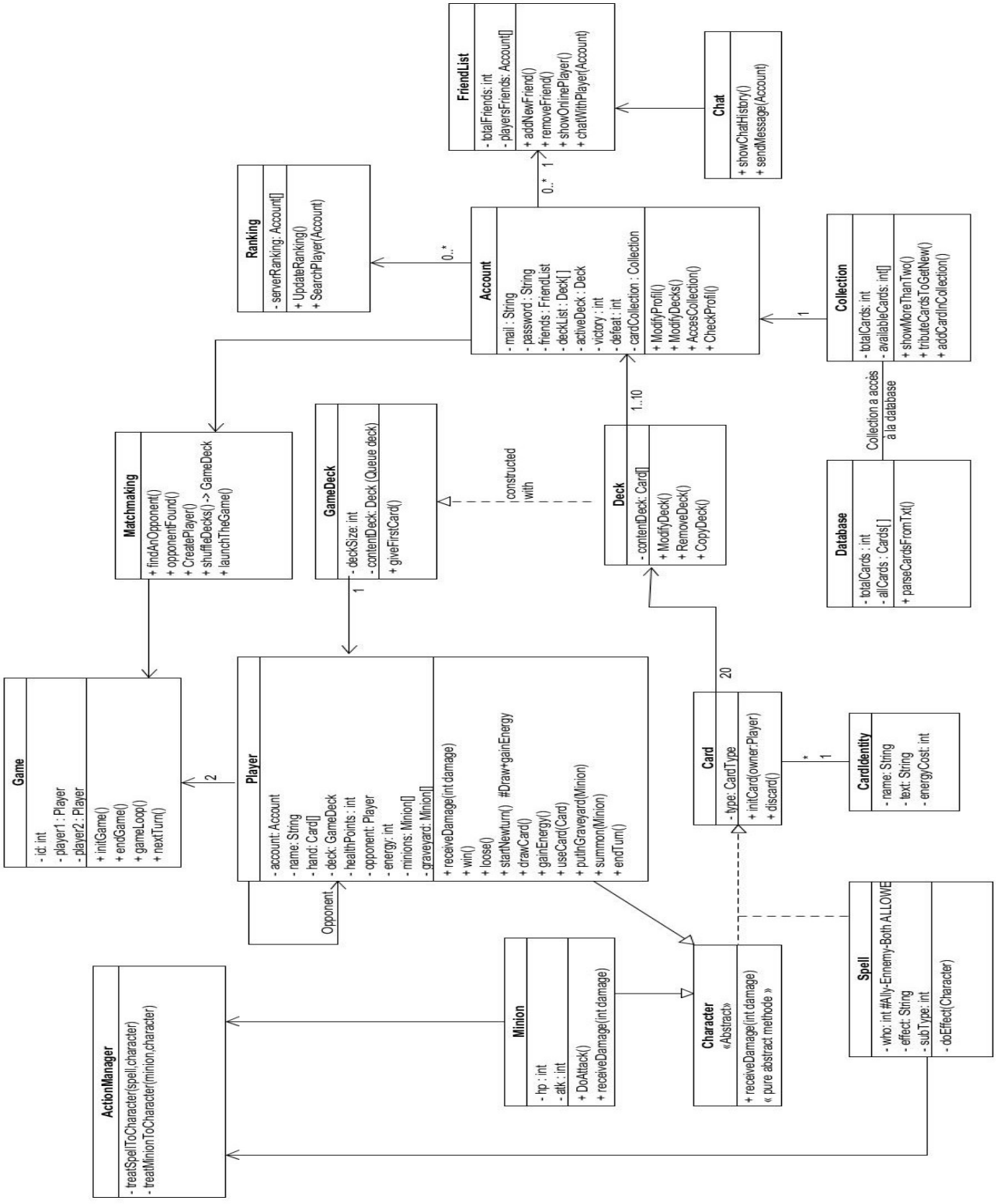
3.1.1.6 Matchmaking

Le Matchmaking sera un outil utilisé par l'utilisateur afin de pouvoir trouver un partenaire de jeu. Lorsqu'une paire d'utilisateurs a été trouvée, le Matchmaking construira les Players à l'aide des accounts des utilisateurs et instanciera l'objet Game qui gérera la partie entre ces deux Players.

3.2 Exigences non fonctionnelles

3.3 Design et fonctionnement du système

3.3.1 Diagramme de Classes



3.3.1.1 Analyse globale

Dans ce diagramme se trouve un aperçu globale de l'application. Le projet devant être codé principalement en orienté-objet, nous avons divisé le problème en un maximum de classe. Les méthodes et attributs ont des signatures adaptés pour permettre au client de comprendre la tache effectué par chaque classe. Le diagramme est sensé être claire, mais il y a certains points qui méritent plus de précisions.

3.3.1.2 ActionManager

C'est dans cette classe que va se dérouler les actions. Une action peut être de 2 types. La première est provoqué par le minion de l'un des player qui va attaquer un character adverse. La deuxième sera provoqué lors de l'utilisation d'un spell qui va cibler un character adverse.
Ces deux actions ont un effet différents. Dans le premier, il y a le système de contre-attaque qui implique une modification du character qui se fait attaqué mais le minion qui attaque également, si le character en question a une attaque plus grande que 0. Tandis que dans le cas d'un spell, c'est seulement le character visé qui sera modifié, le spell quant à lui sera défaussé de la main.

3.3.1.3 Database

Pour conserver les cartes, nous avons décider d'enregistrer toutes les cartes du jeu dans un fichier texte avec une syntaxe précise. Celle-ci sera donc chargé lors de l'ouverture du jeu. Lorsqu'un joueur veut accéder à sa collection, celle-ci sera trouvé depuis la Database. Nous voyons la Database comme la liste de toutes les cartes disponibles, de la sorte chaque joueur aura sa propre vision sur la disponibilité d'une carte. De la sorte, il ne faudra pas copier chaque carte dans la collection de chaque joueur, mais seulement déterminer pour chaque joueur, si une carte est disponible ou non.

3.3.2 Tour de jeu

Lors d'un tour de jeu, le joueur à un certains nombre d'actions possible durant son tour ou celui de l'adversaire. Ces différentes possibilités sont représentées dans le diagramme ci-dessous.

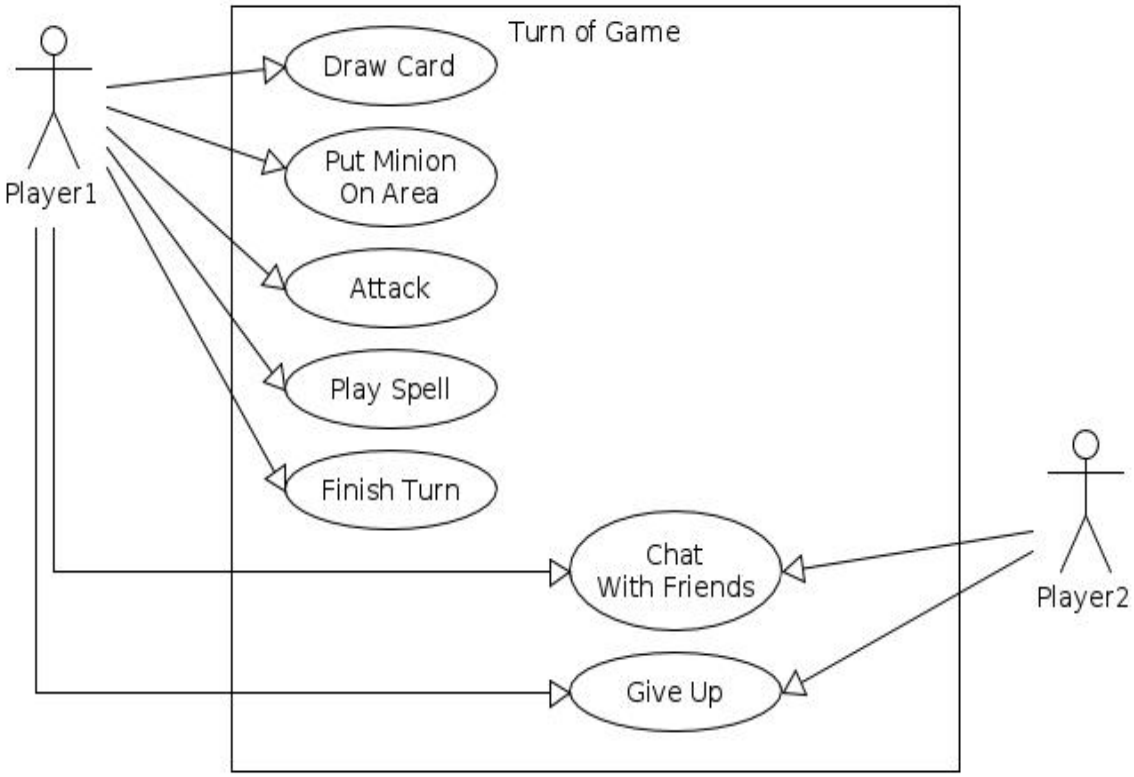


FIGURE 3.1 – Use case d'un tour

3.3.3 Diagramme de séquence

3.3.3.1 Lancement du jeu

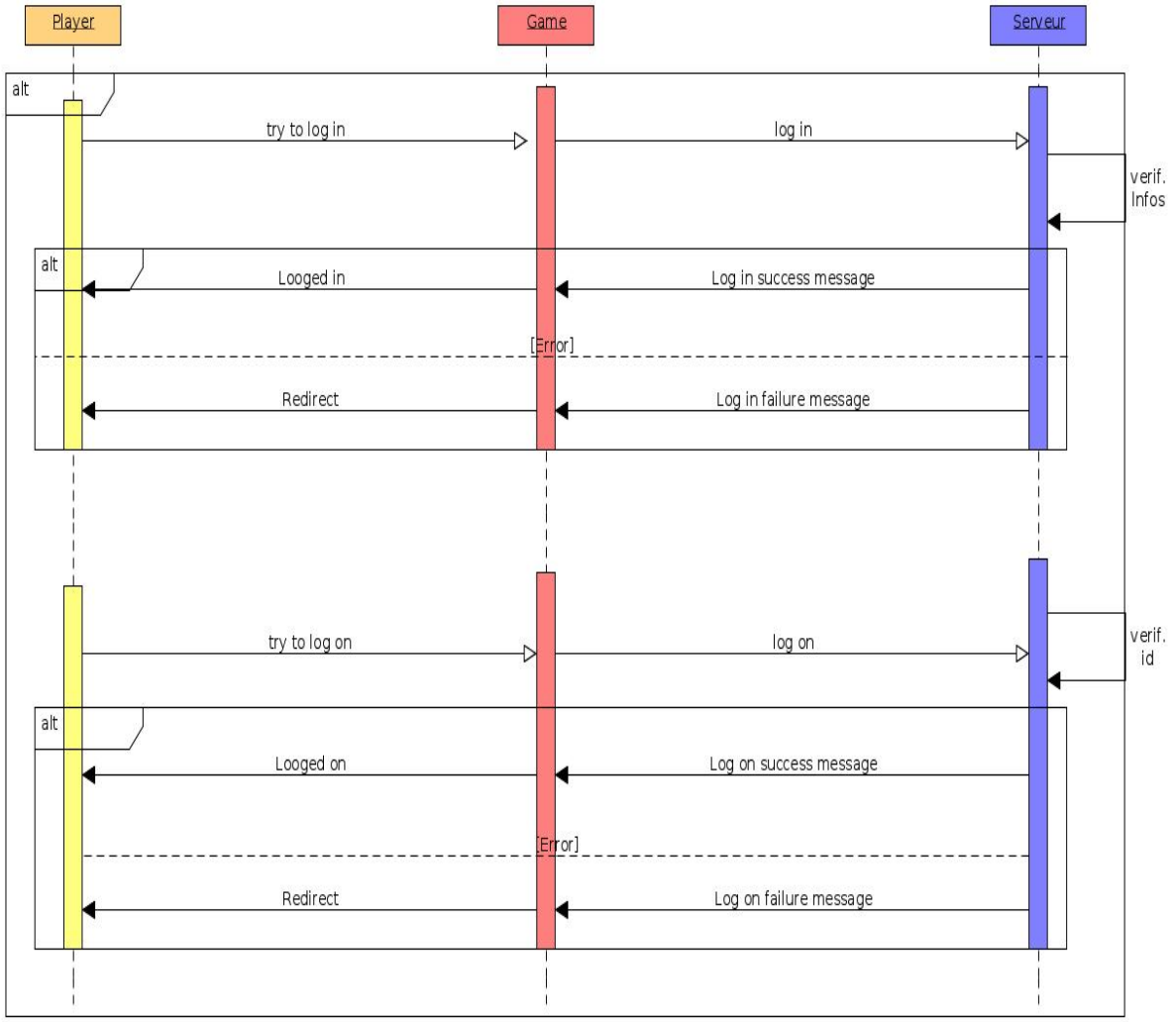


FIGURE 3.2 – Diagramme de séquence lancement du jeu

Analyse

Ce diagramme a pour but de montrer les différentes alternatives qui peuvent se dérouler lors du lancement du jeu , soit le potentiel joueur a un compte et donc il doit se connecter moyennant son identifiant et son mot de passe, et si ils sont corrects alors il sera connecté normalement, si non il sera re-dirigé , soit le joueur ne possède pas de compte et donc doit en créer un , la seule vérification faite est par rapport à l'existence de l'identifiant dans le serveur.

Les classes : "Serveur" et "BridgeClass" sont des éventuelles classes à rajouter dans le futur .

3.3.3.2 Gestion des événements

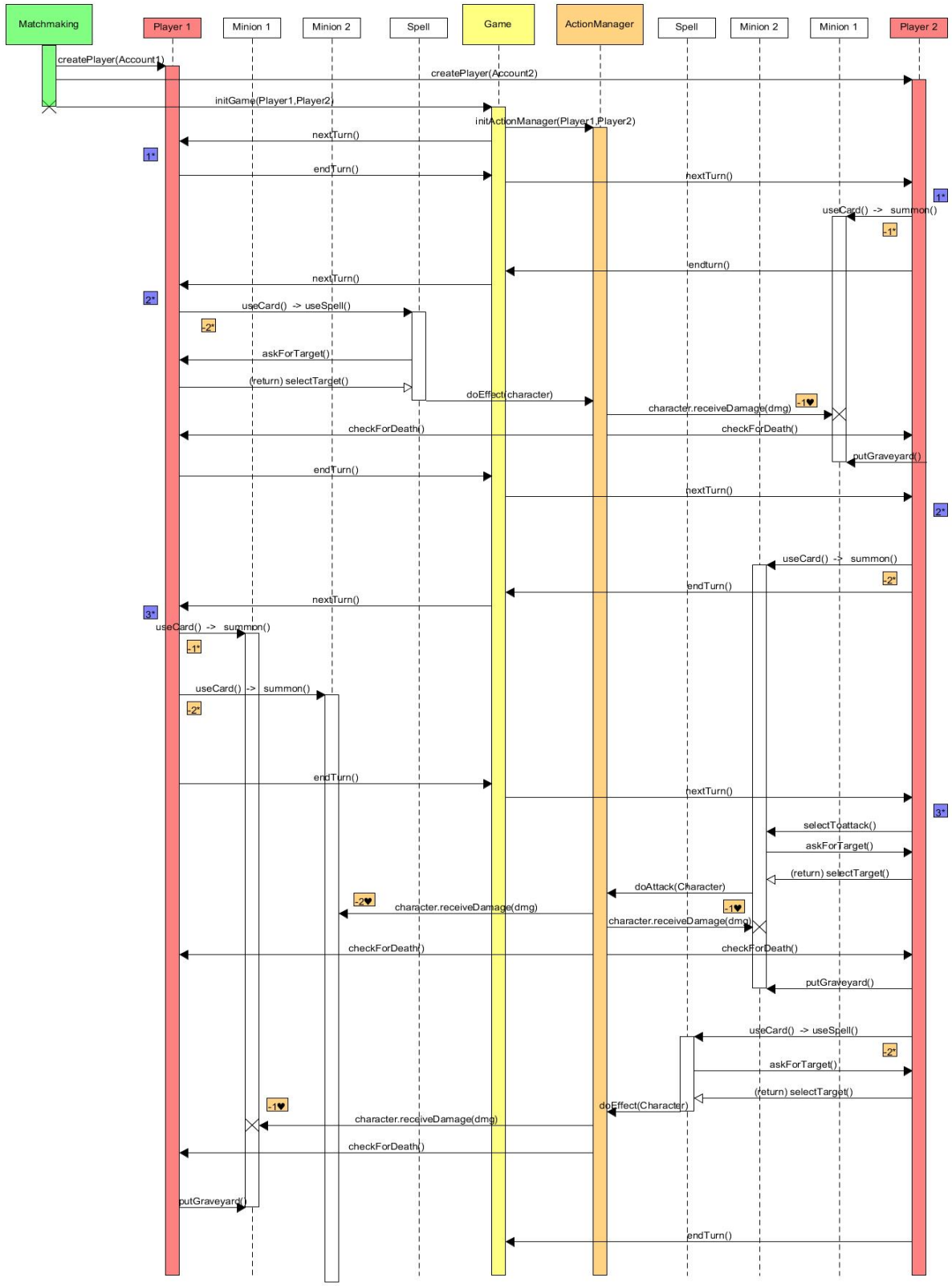


FIGURE 3.3 – Diagramme de séquence d'un début de partie

Analyse

Ce diagramme de séquence a pour but de visualiser les interactions entre certaines classes lors d'une début de partie. Lorsque le Matchmaking a trouvé deux personnes prêtes à se battre, il instancie le Player 1 et le Player 2. Avant d'être détruit, il instanciera également la partie, c'est-à-dire Game, qui, elle, instanciera à son tour l'ActionManager gérant les modifications des objets de type character. Dans cette section, nous illustrerons seulement le contexte du diagramme ci-dessus. Il ne va sans dire que le déroulement de chaque début de partie est différent.

La Game donne la main au Player 1 pour lui signaler que c'est son tour. Comme il n'a qu'un point d'energy et qu'il n'a aucune carte qu'il puisse jouer, il passe donc son tour.

Le Player 2, quant à lui, possède un minion coûtant un point d'energy et décide de l'invoquer.

Le Player 1 décide d'utiliser un spell afin de détruire le minion de son adversaire. Le Player 2, voyant son minion détruit, le place ensuite dans son graveyard. Le tour du Player 1 est terminé.

À ce stade, le Player 2, n'ayant que deux points d'energy est limité dans ses choix. Il invoque ainsi un autre minion coûtant, cette fois-ci, deux points d'energy.

Le Player 1 est en avance d'un point d'energy par rapport au Player 2 puisqu'il a commencé en premier. Il juge nécessaire d'agir comme son adversaire et de rentabiliser ses points et choisit d'invoquer deux minions lors de ce tour-ci.

Voyant un certain danger s'approcher, le Player 2 opte pour la minimisation des dégâts et décide de sacrifier son seul minion afin d'abattre le minion le plus faible de son adversaire. Les deux joueurs placent respectivement leur minion mort dans leur graveyard. Par la suite, le Player 2 lance un spell dans le but d'affaiblir le minion restant de son adversaire.

3.3.3.3 Fin de partie

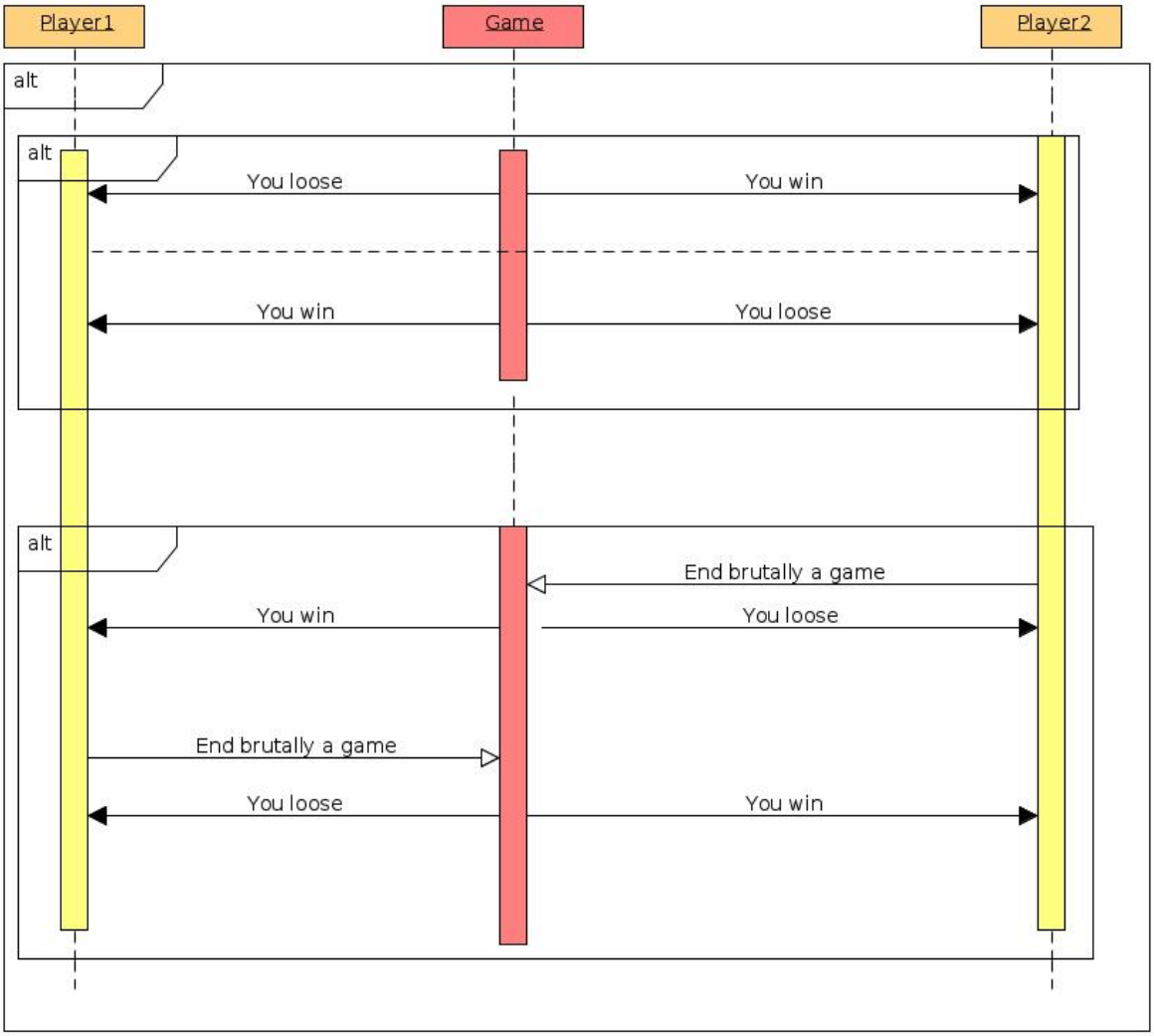


FIGURE 3.4 – Diagramme de séquence d'une fin de partie

Analyse

Ce diagramme a pour but de montrer les différentes alternatives qui peuvent se dérouler dans une fin de partie , et donc soit un des deux joueurs gagnent le match soit l'un des deux quitte la partie et donc la victoire est attribué à l'adversaire .