

INFO-F-109 : Projet d'Informatique 2
Software Requirement Document
Pawn Hub

Huwart Maxence, Boonen Jacques
Pham Hong Phuc, Duc Nguyen, Caroline Forest
Antunes Andre, Romain Mardulyn

2 Mars 2019

Table des Matières

1	Introduction	4
1.1	Description du Projet	4
1.2	Fonctionnement du Jeu d'Échec	4
1.2.1	Règles du Jeu	4
1.2.2	Les Pièces	4
1.3	Les Modes de Jeu	5
1.3.1	Classique	5
1.3.2	Alice Chess	5
1.3.3	Dark Chess	5
1.3.4	Horde Chess	5
1.4	Fonctionnalités pour Utilisateur Enregistré et Utilisateur Non-Enregistré	5
1.5	Fonctionnalités pour Utilisateur Enregistré	5
1.6	Glossaire	5
1.7	Historique des Modifications	7
2	Besoins Utilisateurs : Fonctionnels	8
2.1	Connexion	8
2.1.1	S'identifier	8
2.1.2	Créer un Compte	8
2.2	Menu Principal	9
2.2.1	Jouer	9
2.2.2	Voir les Règles	9
2.2.3	Voir les Statistiques Personnelles	10
2.2.4	Voir le Classement	10
2.2.5	Voir sa Liste d'Amis	10
2.3	Durant une Partie	10
2.3.1	Avancer une Pièce	10
2.3.2	Discuter dans le Chat	10
2.3.3	Déclarer Forfait	11
3	Besoins Utilisateurs : Non-Fonctionnels	11
4	Besoins Systèmes : Fonctionnels	11
4.1	Connexion au Serveur	11
4.2	Enregistrement d'un Nouveau Compte	11
4.3	Création d'une Partie	11
4.4	Voir les Statistiques Personnelles	12
4.5	Voir le Classement	12
4.6	Voir sa Liste d'Amis	12
4.7	Envoi de Messages	12
4.8	Gestion d'une Partie	13
4.8.1	Premier à Jouer	13
4.8.2	Déplacer une Pièce	13
4.8.3	Promotion	13
4.8.4	Roque	13
4.8.5	Déroulement de la Partie	13
4.8.6	Fin de la Partie	15

5	Besoins Systèmes : Non-Fonctionnels	15
5.1	Système d'Exploitation	15
5.2	Réseau	15
5.3	Mise à Jour des Différentes Fonctionnalités	15
6	Design du Système	15
6.1	Les Différentes Classes	15
6.2	Connexion du Joueur	16
6.3	Les Actions du Joueur	16
6.4	Client-Serveur	16
6.5	Déroulement d'une partie	18
6.6	Menu	18
7	Annexe	18

1 Introduction

1.1 Description du Projet

Ce projet aura pour but de recréer un grand classique des jeux de plateau: les échecs. Le jeu sera jouable en réseau via un mode multijoueurs. Le joueur pourra jouer au mode classique ou à une de ses variantes telles que *AliceChess*, *DarkChess* et *HordeChess*, décrites plus loin. De plus, il aura également la possibilité de créer un compte à partir duquel il pourra effectuer plusieurs actions qui seront détaillées ci-dessous. Néanmoins, le joueur pourra lancer une partie en tant que *visiteur* mais ne bénéficiera pas des avantages liés à la possession d'un compte.

1.2 Fonctionnement du Jeu d'Échec

1.2.1 Règles du Jeu

Chaque joueur commence le jeu avec 16 pions: un roi, une dame, deux tours, deux fous, deux cavaliers et huit pions. Le but du jeu est d'infliger à son adversaire un *échec et mat*, une situation dans laquelle le roi d'un joueur est en prise sans qu'il soit possible d'y remédier¹.

1.2.2 Les Pièces

Le Pion : Depuis sa position initiale, le pion peut avancer d'une ou deux cases en avant, ensuite d'une case en avant uniquement. Cependant, le pion prend une autre pièce en diagonale. Également, lorsque le pion arrive à la dernière rangée, il effectue une promotion et est remplacé par n'importe quelle autre pièce, à part le roi, au choix du joueur. De plus, il peut également effectuer la prise en passant qui est expliquée dans le glossaire.

La Tour : La tour peut se déplacer d'un nombre quelconque de cases sur les rangées et les colonnes.

Le Fou : Le fou peut se déplacer uniquement en diagonale d'un nombre quelconque de cases.

La Reine : La reine peut se déplacer dans toutes les directions d'un nombre quelconque de cases.

Le Roi : Le roi peut se déplacer dans toutes les directions d'une case uniquement. Il peut également effectuer un roque qui est expliqué dans le glossaire.

Le Cavalier : Le cavalier se déplace en "L", voir image ci-dessous.

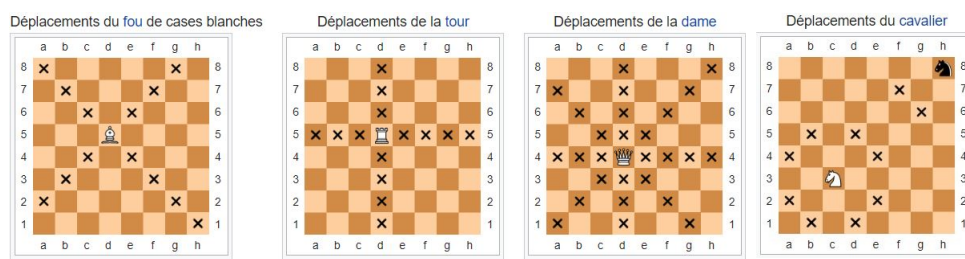


Figure 1: Image provenant de Wikipedia

¹Wikipedia

1.3 Les Modes de Jeu

1.3.1 Classique

Dans une partie de jeu d'échecs classique, le but est de mettre le roi adverse en échec sans qu'il ait la possibilité au tour suivant de ne plus l'être.

1.3.2 Alice Chess

Le jeu d'échecs d'Alice est une variante du jeu d'échecs jouée en utilisant 2 plateaux. Après chaque coup, la pièce déplacée est téléportée sur la case où elle a été placée mais de l'autre plateau. Un coup n'est valide que si la case où doit se téléporter la pièce ne comporte aucune pièce de sa faction. Mis à part ce changement d'affichage, le jeu se déroule selon les règles habituelles du jeu d'échecs.

1.3.3 Dark Chess

Le jeu d'échecs sombre est une variante du jeu d'échecs où le joueur ne voit que ses propres pièces et les cases où il peut légalement les déplacer. Le but des joueurs est de prendre le roi adverse. Contrairement aux règles classiques, le roi d'un joueur peut être en échec à la fin de son tour. De plus, il n'y a pas de prises en passant possibles. À part cela, le jeu se déroule selon les règles habituelles du jeu d'échecs.

1.3.4 Horde Chess

Le jeu d'échecs de horde est une variante du jeu d'échecs où le joueur noir possède 32 pions et le joueur blanc la collection standard du jeu d'échecs classique. Le but du joueur noir est le but du jeu classique mais le but du joueur blanc est de prendre tous les pions du joueur noir. À part cela, le jeu se déroule selon les règles habituelles du jeu d'échecs.

1.4 Fonctionnalités pour Utilisateur Enregistré et Utilisateur Non-Enregistré

Depuis le menu principal, les joueurs pourront accéder à l'option *View Rules* à travers lequel ils pourront lire en détail les règles du jeu d'échecs et de ses variantes. Les joueurs lanceront une partie via l'option *Game*. Lors d'une partie, ils pourront s'envoyer des messages via un chat dédié.

1.5 Fonctionnalités pour Utilisateur Enregistré

Les joueurs pourront communiquer via un chat global. Dans le système de matchmaking, ils auront le choix entre affronter un ami ou un joueur aléatoire parmi ceux qui sont connectés et libres². À travers le menu principal, les utilisateurs accéderont à plusieurs sous-menus tels que *View Ranking*, *View Rules*, *View Statistics*, *Friends List*.

1.6 Glossaire

Par souci de clarté et de lisibilité, toutes les informations relatives aux règles et au fonctionnement officiels des échecs et de ses variantes ne seront pas parcourues ci-dessous.

Un utilisateur non-enregistré : Utilisateur ne s'étant pas identifié au système via un compte.

Un utilisateur enregistré : Utilisateur s'étant identifié au système via un compte.

Joueur : Utilisateur en partie.

²ne se trouvant pas en partie.

Matchmaking : Système qui met en relation deux joueurs avant le lancement d'une partie.

Chat : Système de communication par messages instantanés entre deux joueurs.

Echec et mat : Un joueur est en échec et mat lorsque son roi est en une position telle qu'il ne peut qu'être pris au prochain tour par son adversaire.

Pat : Il y a pat lorsqu'aucun des joueurs n'est en mesure de placer l'autre en échec et mat.

Le Roque : Le roque consiste à déplacer en un seul coup le roi et l'une des tours. On déplace d'abord le roi de deux cases vers la tour puis, avec la même main, on fait passer la tour de l'autre côté, juste à côté du roi (voir le diagramme ci-dessous). Les conditions suivantes sont nécessaires pour pouvoir roquer : aucune pièce ne se trouve entre le roi et la tour concernée ; le roi et la tour concernée n'ont encore jamais joué et le roi n'est pas en échec ; la case traversée par le roi n'est contrôlée par aucune pièce adverse³.

La Prise en Passant: Au jeu d'échecs, la prise en passant est une possibilité particulière de capturer un pion. Lorsqu'un pion se trouve sur la cinquième rangée et que l'adversaire avance de deux cases un pion d'une colonne voisine (les deux pions se retrouvent alors côte-à-côte sur la même rangée), le premier pion peut prendre le second. Pour effectuer la prise en passant, le joueur avance son pion en diagonale sur la sixième rangée et la colonne du pion adverse, et ôte ce dernier de l'échiquier⁴.

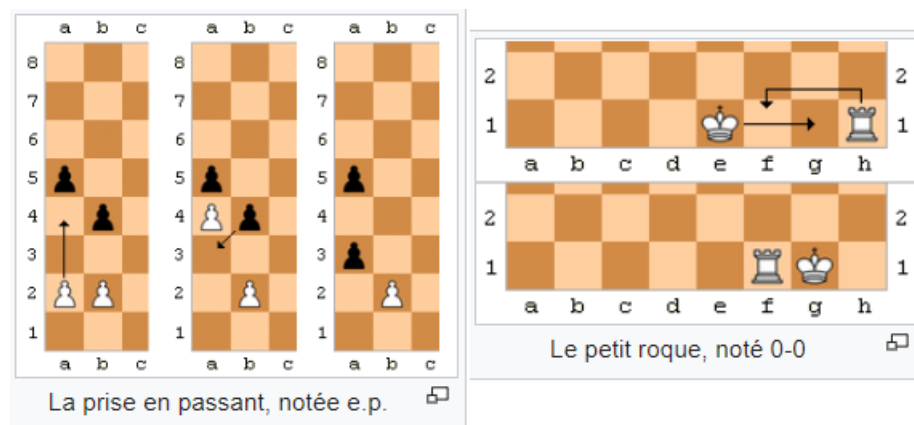


Figure 2: Image provenant de Wikipedia

³source Wikipedia

⁴source Wikipedia

1.7 Historique des Modifications

version	date	auteur	description
1	4/12	Pham Hong Phuc	Création du SRD + ajout structure
1.1	6/12	Forest Caroline	Ajout de diagrammes UML
1.2	8/12	Boonen Jacques	1.Introduction
1.3	11/12	Huwart Maxence	Besoins systèmes
1.4	12/12	Mardulyn Romain	Explication des différent modes de jeu
1.5	12/12	Antunes André	Explication des options durant une partie
1.6	12/12	Nguyen Duc	Modification des diagrammes de séquence
1.7	13/12	Pham Hong Phuc	Correction et réajustement
2	12/02	Boonen Jacques	Fonctionnement du jeu + ajout images
2.1	13/02	Boonen Jacques	Déroulement d'une partie

Historique des modifications

2 Besoins Utilisateurs : Fonctionnels

2.1 Connexion

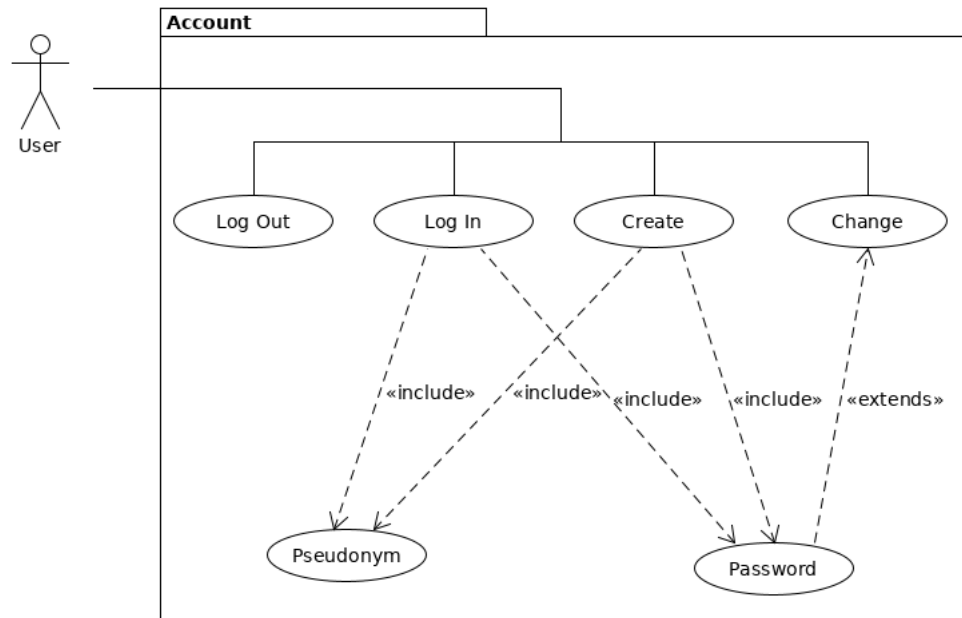


Figure 3: Diagramme de *use case* des actions possibles d'un utilisateur quant à son identification

2.1.1 S'identifier

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Le *user* doit avoir sélectionné le sous-menu *Connexion* puis l'option *Log In* dans le menu principal du jeu.

Post-conditions : Le *user* doit encoder un nom d'utilisateur et un mot de passe d'un compte déjà existant dans le système.

Cas général : Après s'être identifié, le *user* peut profiter de tous les sous-menus du menu principal.

Cas exceptionnels : Si un utilisateur est déjà connecté sur un compte, un autre joueur ne peut se connecter sur ce compte. Également, si l'utilisateur encode mal ses identifiants, le système lui renvoie un message d'erreur afin qu'il les réécrive correctement.

2.1.2 Créer un Compte

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Le *user* doit avoir sélectionné le sous-menu *Connexion* puis l'option *Create* dans le menu principal du jeu.

Post-conditions : Le *user* doit encoder un nom d'utilisateur et un mot de passe afin de créer son compte.

Cas général : Un nom d'utilisateur et un mot de passe correspondant sont encodés dans la base de données

du système.

Cas exceptionnels : Si le nom d'utilisateur encodé est déjà dans la base de données, l'utilisateur doit en réencoder un nouveau.

2.2 Menu Principal

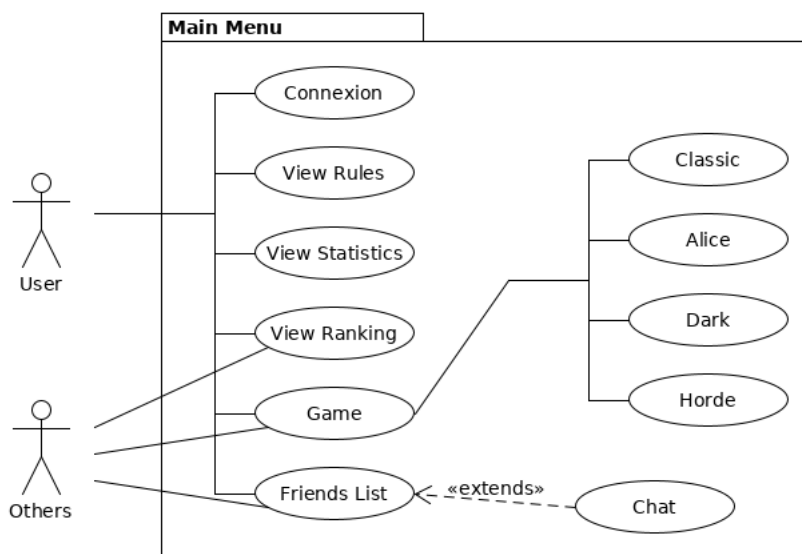


Figure 4: Diagramme de *use case* des actions possibles d'un utilisateur

2.2.1 Jouer

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Modes de jeu proposés (*Classic*, *Alice*, *Dark* ou *Horde*).

Pré-conditions : Le *user* doit avoir sélectionné l'option *Game* dans le menu principal du jeu.

Post-conditions : Le *user* sélectionne un mode de jeu parmi ceux proposés (*Classic*, *Alice*, *Dark* ou *Horde*).

Cas général : Le *user* peut sélectionner un mode de jeu parmi ceux proposés (*Classic*, *Alice*, *Dark* ou *Horde*), ce qui lance une requête de *matchmaking* (voir Figure 6) au serveur.

Cas exceptionnels : Néant.

2.2.2 Voir les Règles

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Le *user* doit avoir sélectionné l'option *View Rules* dans le menu principal du jeu.

Post-conditions : Le *user* accède aux règles du jeu.

Cas général : Le *user* peut consulter les règles du jeu en envoyant une requête au serveur.

Cas exceptionnels : Néant.

2.2.3 Voir les Statistiques Personnelles

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Le *user* doit avoir sélectionné l'option *View Statistics* dans le menu principal du jeu.

Post-conditions : Le *user* accède à ses statistiques personnelles.

Cas général : Le *user* peut consulter ses statistiques en envoyant une requête au serveur.

Cas exceptionnels : Néant.

2.2.4 Voir le Classement

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Le *user* doit avoir sélectionné l'option *View Ranking* dans le menu principal du jeu.

Post-conditions : Le *user* accède au classement global des joueurs (*user* et *others*).

Cas général : Le *user* peut consulter le classement global des joueurs (*user* et *others*) en envoyant une requête au serveur.

Cas exceptionnels : Néant.

2.2.5 Voir sa Liste d'Amis

Acteur : *User*.

Relations avec d'autres cas d'utilisation : *Chat*.

Pré-conditions : Le *user* doit avoir sélectionné l'option *Friends List* dans le menu principal du jeu.

Post-conditions : Le *user* accède à sa liste d'utilisateurs "amis", avec lesquels il peut choisir de discuter via l'option *chat*.

Cas général : Le *user* peut consulter sa liste d'utilisateurs "amis" en envoyant une requête au serveur. Il peut aussi discuter avec eux via messages instantanés et le serveur.

Cas exceptionnels : Néant.

2.3 Durant une Partie

2.3.1 Avancer une Pièce

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant

Pré-conditions : C'est au tour du *user* de jouer. Il doit préalablement choisir une pièce et indiquer un déplacement possible et valide.

Post-conditions : La pièce se retrouve à la position désignée par le *user* après déplacement.

Cas général : Le *user* doit déplacer un pion lorsque c'est à son tour de jouer. Il envoie ainsi une requête de déplacement au serveur qui vérifiera à son tour la validité ou non du coup.

Cas exceptionnels : Le coup enregistré par le *user* n'est pas valide. Il doit donc rejouer son tour.

2.3.2 Discuter dans le Chat

Acteur : *User*.

Relations avec d'autres cas d'utilisation : *Friends List*.

Pré-conditions : Avoir un ami ou "adversaire" et qu'il soit connecté.

Post-conditions : Néant.

Cas général : Le *user* ouvre avec un bouton (à définir) le chat et peut écrire sur un textfield pour

communiquer avec les autres personnes.

Cas exceptionnels : Néant.

2.3.3 Déclarer Forfait

Acteur : *User*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : Il faut que le *user* soit dans un jeu.

Post-conditions : Informer l'adversaire sur la déclaration de forfait et terminer le jeu.

Cas général : Lorsque le joueur veut terminer la partie.

Cas exceptionnels : Lorsque le joueur perd la connexion le joueur déclare automatiquement forfait.

3 Besoins Utilisateurs : Non-Fonctionnels

Afin d'avoir des parties de jeux dynamiques, les joueurs se verront imposer une limite de temps pour pouvoir jouer leur tour.

4 Besoins Systèmes : Fonctionnels

4.1 Connexion au Serveur

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Log In*.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Utilisateur connecté, redirection vers le menu principal, sous-menus réservés aux utilisateurs débloqués, remplacement du sous-menu *Log In/Create an account* par *Log Out*.

Cas général : Utilisateur connecté.

Cas exceptionnels : Si un utilisateur est déjà connecté sur un compte, un autre joueur ne peut se connecter sur ce compte. De même, si l'utilisateur encode mal ses identifiants, le système lui renvoie un message d'erreur afin qu'il les récrive correctement.

4.2 Enregistrement d'un Nouveau Compte

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : Créer un compte.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Compte enregistré, message de succès de création de compte envoyé à l'utilisateur.

Cas général : Compte enregistré.

Cas exceptionnels : Si le compte est déjà existant, un message d'erreur est envoyé à l'utilisateur.

4.3 Création d'une Partie

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : Choisir un mode de jeu.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Vérifie dans la file d'attente si un joueur dans la même fourchette de classement et

désirant jouer au même mode est disponible. Si oui, le *Server* crée la partie. Sinon, l'utilisateur est mis en file d'attente.

Cas général : Utilisateur mis en file d'attente si aucun adversaire n'est trouvé, sinon une partie est créée.

Cas exceptionnels : Néant.

4.4 Voir les Statistiques Personnelles

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Envoi des statistiques personnelles à l'utilisateur correspondant.

Cas général : Envoi des statistiques personnelles de l'utilisateur.

Cas exceptionnels : Néant.

4.5 Voir le Classement

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : Néant.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Envoi du classement.

Cas général : Envoi du classement.

Cas exceptionnels : Néant.

4.6 Voir sa Liste d'Amis

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Envoi de messages*.

Pré-conditions : *Server* en ligne et réception du signal correspondant.

Post-conditions : Envoi de la liste d'amis de l'utilisateur en question.

Cas général : L'utilisateur reçoit sa liste d'amis et peut interagir avec elle via des requêtes serveur (ajout d'un ami, suppression, etc).

Cas exceptionnels : Néant.

4.7 Envoi de Messages

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Friends List*, jouer.

Pré-conditions : *Server* en ligne et réception du message à envoyer.

Post-conditions : Envoi du message.

Cas général : Les utilisateurs peuvent s'envoyer des messages durant une partie mais également en dehors d'une partie s'ils sont amis.

Cas exceptionnels : Néant.

4.8 Gestion d'une Partie

4.8.1 Premier à Jouer

Le joueur blanc commence. Celui-ci est le premier joueur connecté et qualifiable de la file d'attente pour ce jeu.

4.8.2 Déplacer une Pièce

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Avancer un pion*.

Pré-conditions : Réception du signal correspondant, coup valide (pas de pièces qui bloquent, etc).

Post-conditions : Pièce déplacée.

Cas général : Pièce déplacée.

Cas exceptionnels : Pièces qui bloquent ou une pièce est déjà sur la case. Dans tous ces cas, on redemande au joueur de faire déplacer une de ses pièces.

4.8.3 Promotion

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Avancer un pion*.

Pré-conditions : Un pion est arrivé à l'autre bout du plateau.

Post-conditions : Pion promu en dame, fou, cavalier ou tour au choix du joueur qui possède le pion en question. Le jeu est réanalysé pour identifier une potentielle mise en échec due au changement de statut du pion.

Cas général : Pion promu.

Cas exceptionnels : Néant.

4.8.4 Roque

Acteur : *Server*.

Relations avec d'autres cas d'utilisation : *Avancer un pion*.

Pré-conditions : Un joueur déplace le roi sur une case provoquant le roque, le roi et la tour en question n'ont pas bougé depuis le début de la partie. Aucune pièce ne bloque.

Post-conditions : Tour déplacée, position du roi mise à jour, le roi et la tour marqués comme "ayant déjà bougé".

Cas général : Roque effectué.

Cas exceptionnels : Si l'une des pré-conditions n'est pas respectée, on redemande au joueur de faire déplacer une de ses pièces.

4.8.5 Déroulement de la Partie

Voici le déroulement de la partie représentée à l'aide d'un diagramme de séquence :

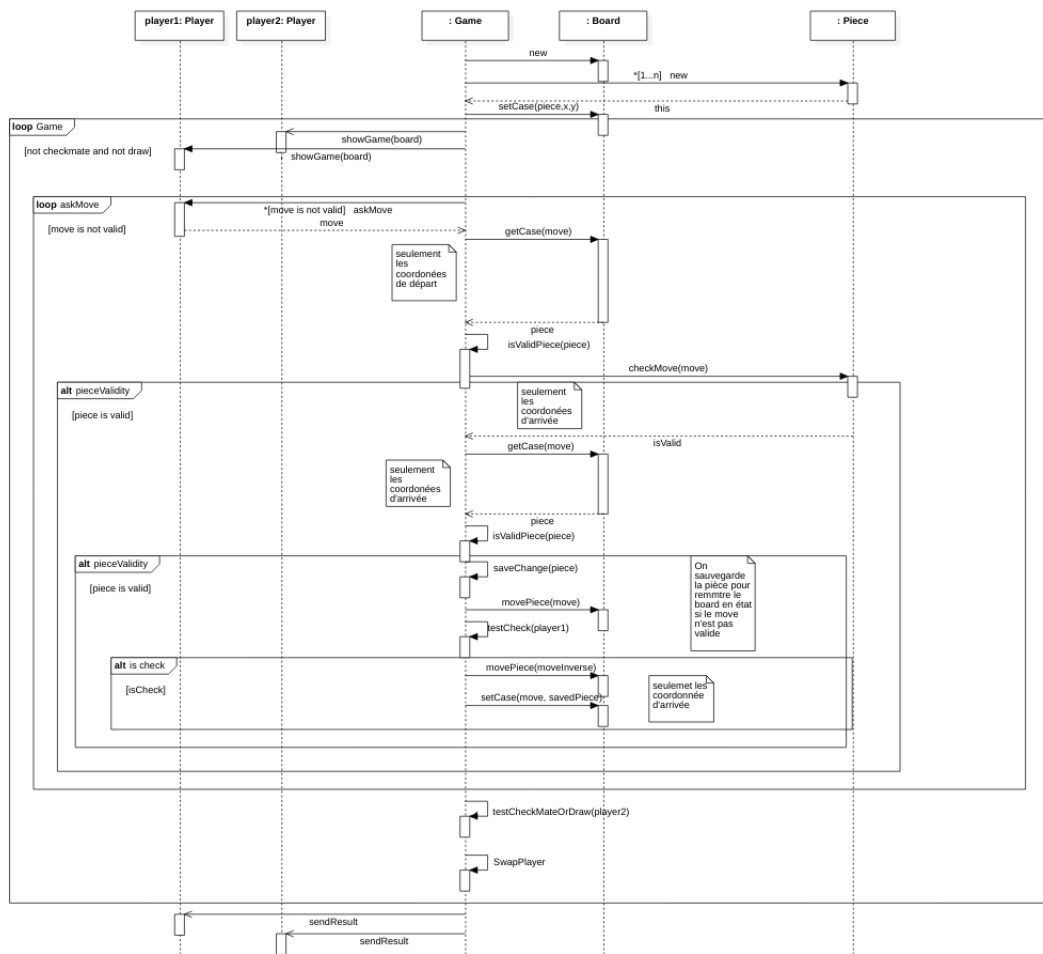


Figure 5: Diagramme de séquence d'une partie

4.8.6 Fin de la Partie

À la fin de chaque tour de jeu, la logique du jeu se trouvant côté serveur vérifie s'il y a un gagnant ou si l'on a affaire à un Pat. Si oui, le résultat est envoyé à chacun des joueurs et la partie se termine.

5 Besoins Systèmes : Non-Fonctionnels

5.1 Système d'Exploitation

Le programme doit être capable de tourner sous Unix.

5.2 Réseau

Le jeu se joue en réseau local. Une connexion internet est donc requise.

5.3 Mise à Jour des Différentes Fonctionnalités

Le système se doit de mettre assez rapidement à jour les statistiques personnelles et le classement global après chaque partie.

6 Design du Système

6.1 Les Différentes Classes

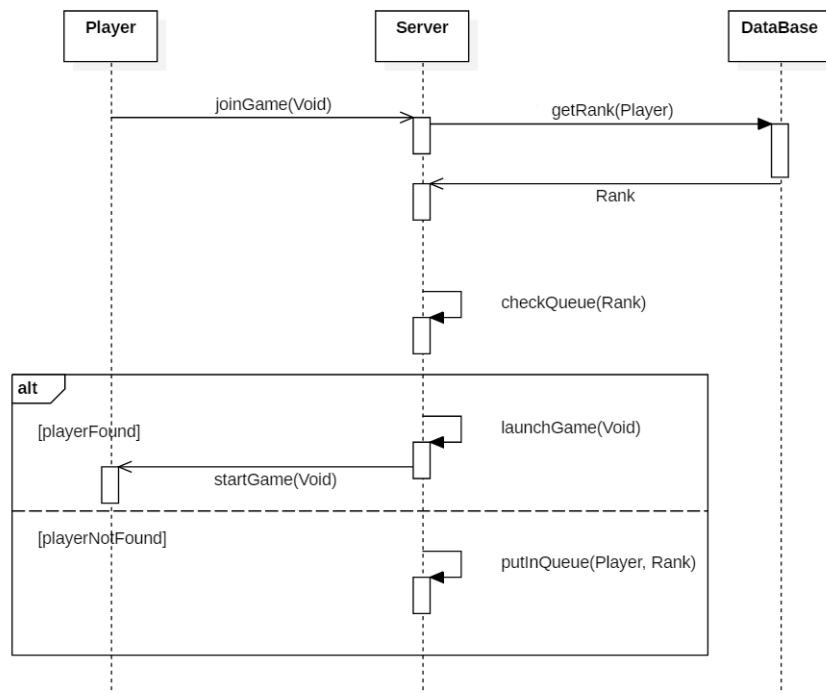


Figure 6: Diagramme de séquence pour le *Matchmaking*

6.2 Connexion du Joueur

6.3 Les Actions du Joueur

6.4 Client-Serveur

Dans le cadre de ce projet, nous avons décidé que la partie serveur s'occupera de toute la logique du jeu tandis que la partie client s'assurera d'afficher correctement l'interface du jeu en temps réel et pour se faire, enverra des requêtes au serveur pour connaître les mouvements de pièces.

Il convient également de spécifier que l'échange d'information entre le serveur et le client se fera sous le format de chaînes de caractères. Par exemple, Le serveur enregistrera les noms de compte et mots de passe des joueurs dans un fichier texte et non une base de données. Et cela pour au moins 3 raisons décrites ci-dessous :

1. Comme la plupart des tâches effectuées par nos programmes se font à l'aide de chaînes de caractères et par soucis d'homogénéité, il est préférable et plus simple que l'échange d'information entre le serveur et le client se fasse également sous cette forme.
2. Une chaîne de caractère n'a pas besoin d'être traduite en binaire, ce qui diminue la complexité et le temps d'exécution de nos programmes.
3. La taille du message envoyée n'est pas un problème, car une taille de 100 octets suffit pour envoyer toutes les informations d'un message. Le plus grand message actuel est le plateau, qui ne consiste que de 98 octets lorsqu'il est complet (voir 6.5.2)

Le serveur aura une copie du board et tâchera de vérifier les différentes actions des joueurs. Il veillera également à mettre à jour les board des joueurs.

6.5 Déroulement d'une partie

1. *Assignment de la couleur des pièces aux joueurs*

Mise en pratique : Le serveur envoie la couleur (noir ou blanc) au client.

Protocole : Le serveur envoie '1' pour la couleur 'blanc' et '0' pour 'noir'. Pour l'instant, le premier connecté reçoit la couleur 'blanc'.

2. *Affichage de l'échiquier*

Mise en pratique : Le serveur envoie le plateau aux joueurs.

Protocole : Le message est une suite de toutes les pièces présentes sur l'échiquier, chacune représentée par 3 caractères. Le premier caractère est une lettre désignant le type de la pièce considérée. Les 2 autres, sa position, en commençant par la colonne, selon la représentation classique d'un échiquier, c'est à dire, les rangées sont numérotées de 1 à 9 et les colonnes de "A" à "H". Les pions blancs sont séparés des noirs par '!' et à la fin des pions noirs, on ajoute un '#'. Exemple : pA2hA4rD5qC7kE1bF8!bA8qG4kD7#. L'envoi d'un plateau contenant toutes les pièces classiques est donc de 98 octets: 1 octet par caractère, 3 caractères par pièces, 32 pièces sur un plateau, soit $3 \times 32 = 96$, plus les 2 symboles de séparation.

3. *Numéro du tour de jeu*

Mise en pratique : Le serveur envoie le numéro du tour actuel de jeu aux joueurs.

Protocole : A partir d'un compteur dans la boucle de jeu, le serveur envoie le chiffre au client.

4. *Tour du joueur*

Mise en pratique : Chaque client vérifie si c'est bien au tour de son joueur de jouer, si oui, celui-ci envoie en message le mouvement de la pièce au serveur.

Protocole : Le message est composé de 4 caractères, les 2 premiers sont l'emplacement du début de tour de la pièce et les 2 derniers sont la position future de la pièce demandée par le joueur.

5. *Mouvement de la pièce*

Mise en pratique : Après avoir reçu la position future de la pièce indiquée par le joueur, le serveur vérifie sa validité et envoie son résultat au client. Tant que la position n'est pas validée par le serveur, le client doit renvoyer une position au serveur. Lorsque que celle-ci est validée, le serveur envoie aux 2 clients l'échiquier mis à jour.

Protocole : Le serveur renvoie une boolée True/False.

6. *État de la partie*

Mise en pratique : Le serveur envoie un message de statut : "<fin de jeu-match nul>" ou "<continue>".

Protocole : Le serveur renvoie True pour terminer la partie ou False pour la continuer.

7. Le joueur suivant commence son tour selon la procédure ci-dessus.

8. Si la partie est terminée, le joueur est déconnecté vu qu'il n'y a pas encore de menu.

NB: Ce protocole n'est pas définitif

6.6 Menu

7 Annexe