

Software Requirement Document

Pour

INFO-F209 : ChessBachelor

El Achouchi Iliass
Laamiri Achraf
Mugrabi Mathieu
Bedaton Antoine
Bassil Matias
Demonceau Quentin
Servais Nikita

*Université Libre de Bruxelles
Science de l'Informatique
Projet d'Informatique 2
Assistants*

2018-2019

Table des matières

Historique des révisions	1
1 Objectifs	2
2 Règles du Jeu	3
2.1 Chess	3
2.2 Trappist Chess	4
2.3 Dark Chess	5
2.4 Anti-Chess	5
3 Besoin Fonctionnels Utilisateurs	6
3.1 Connexion	7
3.1.1 Connexion	7
3.1.2 Inscription	7
3.2 Avant Jeu	8
3.2.1 Lancer une partie	8
3.2.2 Gérer sa liste d'amis	8
3.2.3 Modifier son profil	9
3.2.4 Déconnexion	9
3.2.5 Consulter le classement	9
3.2.6 Ouvrir un chat	9
3.3 En Jeu	10
3.3.1 Chat	10
3.3.2 Choisir une pièce	10
3.3.3 Choisir un mouvement	10
4 Besoin Fonctionnels Système	11
4.1 Connexion	11
4.1.1 Vérifier les Informations	11
4.1.2 Sauvegarder les Informations	11

4.2	Lancer Partie	12
4.2.1	Matchmaking	12
4.2.2	Lancement de la partie	12
4.3	Gérer Partie	13
4.3.1	Indiquer déplacements possibles	13
4.3.2	Déplacement	13
4.3.3	Mise à Jour Chat	13
5	Besoin Non Fonctionnels	14
5.1	Accessibilité	14
5.2	Sécurité	14
5.3	Exigences de Performance	14
6	Besoin de Domaines	15
6.1	Compétition	15
6.2	Ambiance	15
6.3	Connexion	15
7	Implémentation du Système	16
8	Déroulement du Programme	18
8.1	Diagrammes d'activité	18
8.2	Diagramme de Séquence	21
	Table des figures	23
	Appendices	24
A	Références	25
A.1	Variante d'échecs	25
A.2	Code Source utilisée d'internet	25
A.2.1	Partie Jeu	25
B	Vision d'avenir	26
B.1	Idées	26
B.1.1	Idées non-réalisées	26
B.1.2	Idées réalisées	26
B.2	Interfaces	26

Historique des révisions

Version	Date	Auteur(s)	Modifications
1.0	21.11.2018	D. Quentin	Création du SRD et des diagrammes
1.1	05.12.2018	D. Quentin	Modification du SRD et des diagrammes après réunion
1.2	8.12.2018	N. Servais	Modification du SRD (réécriture en LaTeX)
1.2.1	10.12.2018	M. Mathieu	Diagramme de séquence ajouté
1.2.2	10.12.2018	L. Achraf	Rédaction de l'introduction
1.2.3	10.12.2018	E.A. Iliass,D. Quentin,B. Antoine	Use case + description des besoins
1.2.4	13.12.2018	B. Matias	Diagramme de séquence modifié
1.2.5	13.12.2018	D. Quentin	Diagramme d'activité + modifié diagramme de classes
1.3	14.12.2018	tout le monde	Modifications schémas, textes et structure document
1.4	17.12.2018	tout le monde	Compléter + finition
1.4.1	17.12.2018	E.A. Iliass	Correction orthographe
1.4.2	10.02.2019	D. Quentin	Diagramme d'activité + modifié diagramme de classes
1.4.3	26.02.2019	D. Quentin	Ajouts de références partie jeu
1.4.4	27.02.2019	D. Quentin	complétion class diagram A
1.4.5	28.02.2019	D. Quentin	complétion class diagram B, images en grilles

Chapitre 1

Objectifs

L'objectif du projet est d'implémenter un jeu d'échec multijoueur supportant le jeu d'échec classique 8x8 ainsi que plusieurs variantes de ce jeu : "Trapist chess", "Dark Chess" et "Anti-Chess".

Le jeu étant multijoueur il peut être joué en réseau entre plusieurs utilisateurs.

Chaque utilisateur doit s'inscrire et a la possibilité d'avoir une liste d'amis et est capable d'effectuer les actions suivantes sur ceux-ci : possibilité de rejoindre leur partie, de regarder la partie d'un ami, d'inviter un ami à jouer, d'envoyer un message privé à un ami ainsi que la possibilité de consulter les statistiques personnelles et la place dans le classement d'un ami.

Le jeu dispose aussi d'un classement selon les différents modes de jeu ainsi que plusieurs critères différents de classement tels que le ratio victoires/défaites.

Il est également possible de regarder une partie en cours à partir du classement ainsi que de consulter le profil de chaque joueur.

Ces informations sont utilisées dans le système de matchmaking mis en place pour chaque joueur désirant affronter un adversaire au hasard, le matchmaking a pour but de concevoir des parties où les adversaires ont un niveau équitable.

Une base de donnée est disponible pour contenir le nom d'utilisateur de chaque utilisateur, son mot de passe, ses informations, ainsi que les informations présentes dans le classement.

Des comptes admins sont également disponibles à fin de faciliter la gestion des utilisateurs et de la base de donnée.

Chapitre 2

Règles du Jeu

2.1 Chess

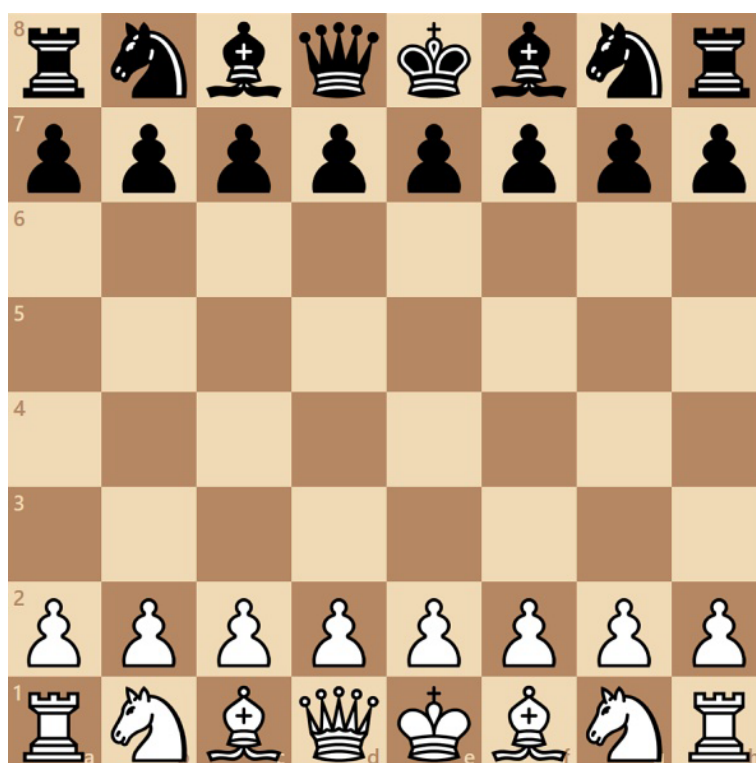


FIGURE 2.1 – Classic chess

Plateau de 8x8 cases.

Le but : Mettre le roi dans une situation telle qu'il ne peut plus bouger sans risque de se faire prendre.

Le principe général : Les joueurs déplacent tour-à-tour une pièce leur appartenant, selon les déplacements possibles de cette pièce. Une case ne peut être occupée que par 1 seule pièce. Si une pièce atterrit sur une case occupée par une pièce adverse elle capture la pièce adverse, c.a.d qu'on retire la pièce capturée du plateau et par conséquent le joueur adverse ne peut plus l'utiliser.

Cas particuliers : Un pion arrivant de l'autre côté du tableau se verra octroyé la possibilité d'évoluer en tour, fou, cavalier ou dame selon le choix du joueur. Le roc : Action particulière permettant de déplacer une

tour et un roi si aucune de ces pièces n'ont encore bougées de la partie.

Déplacements de chaque pièce :

2.2 Trappist Chess

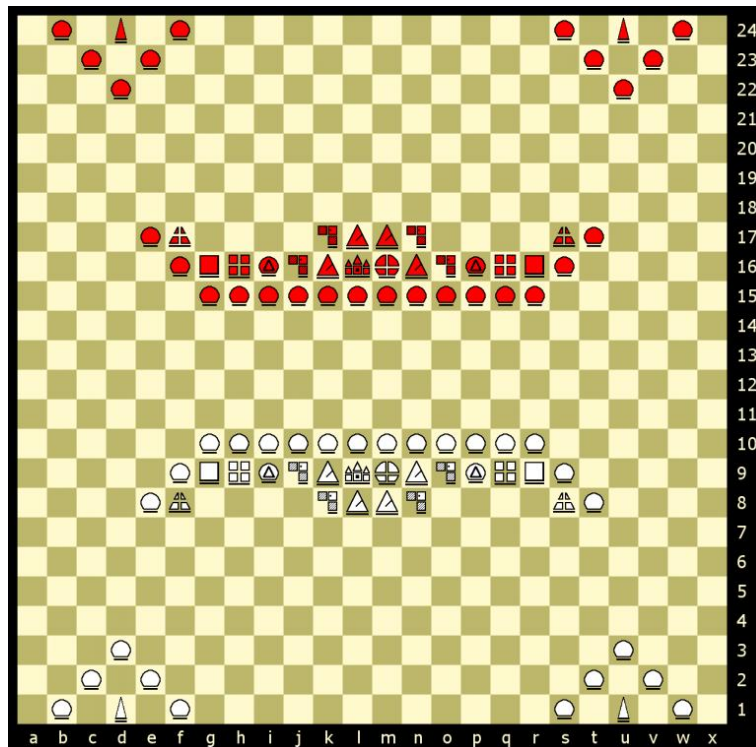


FIGURE 2.2 – Trappist chess

Variantes se jouant avec les mêmes règles que le jeu original en apportant quelques ajouts :

Jeu se jouant sur un plateau de 21x21. Mêmes règles que les échecs classiques. Nouvelles pièces inédites :

- Chancelier : déplacement combinés de la tour et du chevalier
- Faucon : saut de 2 ou 3 cases dans n'importe quelle direction.
- Garde : même déplacement que le roi, il n'est par contre pas affecté par la situation d'échec (et mat).
- Fonctionnaire : fait des sauts de nombre premiers (à partir de 5) horizontalement ou verticalement

Note : Un saut est différent d'un déplacement car on saute au-dessus d'autres pièces. On n'est donc pas gênés par la présence d'autres pièces.

2.3 Dark Chess



FIGURE 2.3 – Dark chess

Variante se jouant avec les mêmes règles que le jeu original en apportant quelques ajouts :

Echecs avec brouillard de guerre . Seules les cases où une pièce pourrait se déplacer sont visibles.

2.4 Anti-Chess

Variantes se jouant avec les mêmes règles que le jeu original en apportant quelques ajouts : Jeu d'échecs classique où le but est de perdre toutes ses pièces, il y a obligation de prendre des pièces si l'occasion se présente. La notion d'échec et mat est perdue.

Chapitre 3

Besoin Fonctionnels Utilisateurs

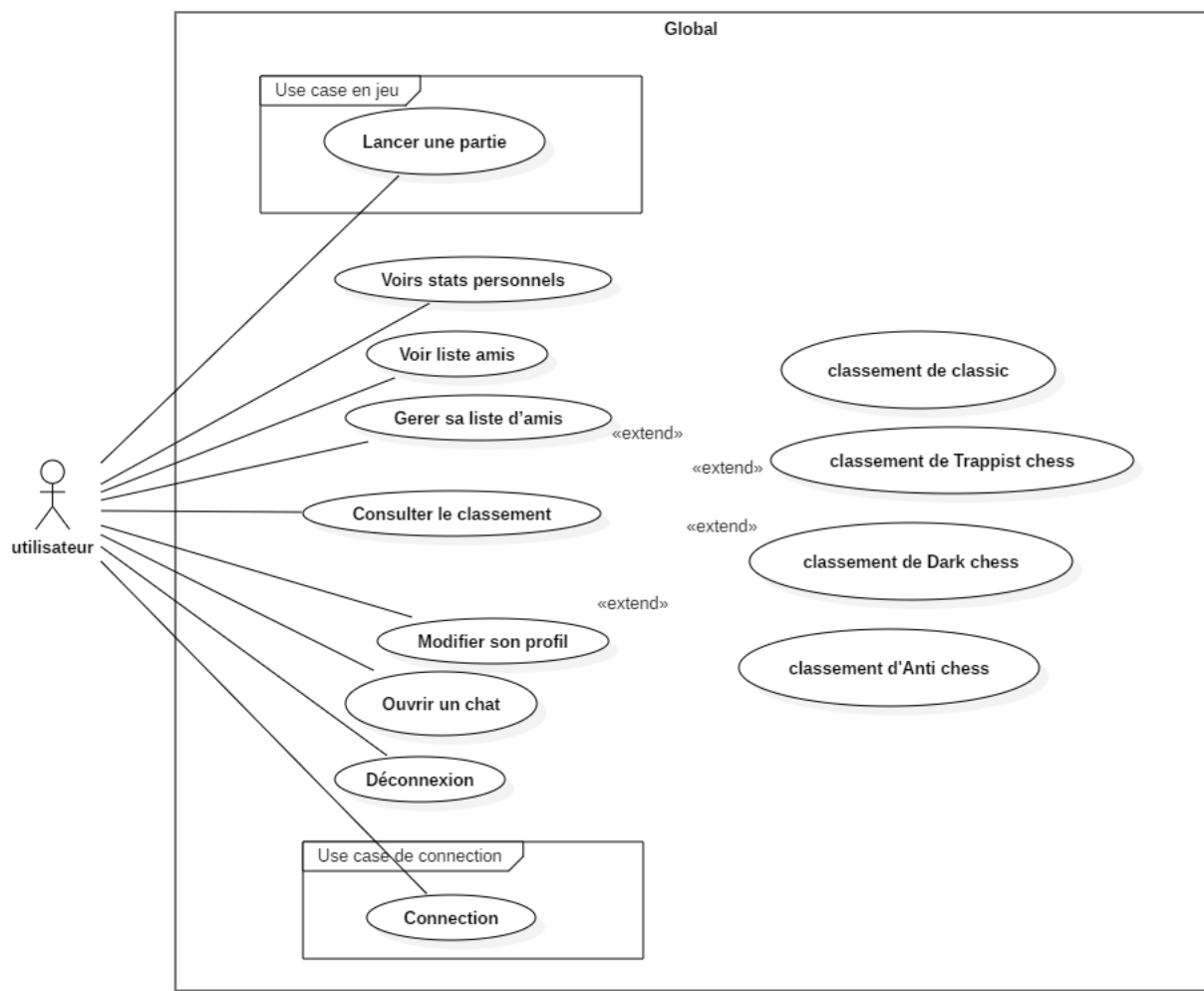


FIGURE 3.1 – Use case global d'utilisateur

3.1 Connexion

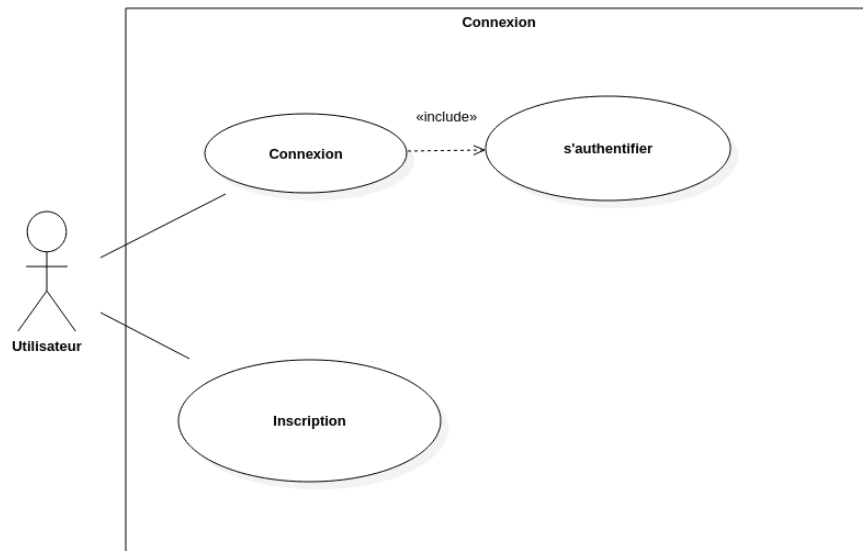


FIGURE 3.2 – Use case connexion

Nous précisons ici que le serveur doit être en ligne, il devra l'être pour toutes les autres options du jeu.

3.1.1 Connexion

- Acteur : Utilisateur
- Relations avec d'autres cas d'utilisation : S'authentifier
- Pré-conditions : L'utilisateur possède un compte.
- Post-conditions : L'utilisateur est connecté à son compte et accède au menu du jeu.
- Cas général : Un utilisateur se connecte pour accéder à son compte
- Cas exceptionnels : Le serveur ne reconnaît pas les informations entrées par l'utilisateur.

3.1.2 Inscription

Obligation pour un utilisateur sans compte qui souhaite accéder au jeu.

- Acteur : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- Pré-conditions : /
- Post-conditions : L'utilisateur possède désormais un compte et accède au menu de connexion.
- Cas général : Un utilisateur veut créer un compte pour pouvoir se connecter et le fait donc depuis l'interface d'inscription.
- Cas exceptionnels : Le nom d'utilisateur demandé par le nouvel inscrit est déjà existant, on redemande à l'utilisateur dans ce cas d'entrer un nom.

3.2 Avant Jeu

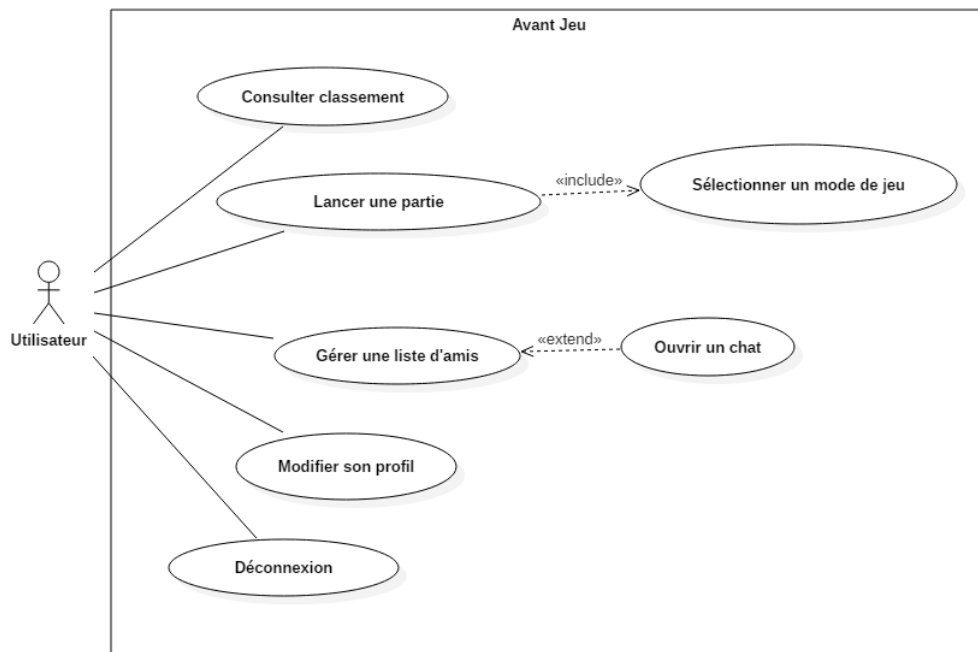


FIGURE 3.3 – Use Case avant une partie

3.2.1 Lancer une partie

- Acteur : Utilisateur
- Relations avec d'autres cas d'utilisation : Sélectionner un mode de jeu
- Pré-conditions : L'utilisateur a sélectionné "Lancer une partie".
- Post-conditions : Un matchmaking a lieu et le jeu sera lancé quand un autre joueur est trouvé.
- Cas général : Un utilisateur lance une partie et choisit un mode de jeu parmi les suivants :
 1. Mode classique
 2. Mode trappist
 3. Mode dark
 4. Mode anti
- Cas exceptionnels : /

3.2.2 Gérer sa liste d'amis

- Acteurs : Utilisateur
- Relations avec d'autres use case : Chat / Gérer sa liste d'amis
- Pré-conditions : L'utilisateur est connecté et a appuyé sur le bouton « Liste d'amis »
- Post-Conditions : Une liste d'amis s'affiche et propose plusieurs fonctionnalités
- Cas général : L'utilisateur consulte sa liste d'amis afin de chatter avec un tiers ou gérer sa liste d'amis à sa guise. C'est-à-dire, ajouter un ami, supprimer un ami ou chatter.
- Cas exceptionnel : /

3.2.3 Modifier son profil

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- Pré-conditions : L'utilisateur est connecté et a cliqué sur « Profil »
- Post-Conditions : Le profil de l'utilisateur est affiché et lui propose certaines fonctionnalités
- Cas général : L'utilisateur peut modifier son mot de passe ou son pseudo
- Cas exceptionnel : /

3.2.4 Déconnexion

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- Pré-conditions : L'utilisateur est connecté et appuie sur déconnexion
- Post-Conditions : L'utilisateur est déconnecté et n'a plus accès à aucune fonctionnalité du programme
- Cas général : voir post condition
- Cas exceptionnel : /

3.2.5 Consulter le classement

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- Pré-conditions : L'utilisateur est connecté et appuie sur "Consulter le classement".
- Post-Conditions : L'utilisateur est confronté à l'apparition du classement et peut le consulter à sa guise.
- Cas général : Un classement est affiché avec les "scores" des joueurs inscrits, l'utilisateur peut le consulter autant qu'il veut.
- Cas exceptionnel : /

3.2.6 Ouvrir un chat

- Acteurs : Utilisateur
- Relations avec d'autres use case : Consulter liste d'amis
- Pré conditions : L'utilisateur a au moins un ami et en a choisi un
- Post-Conditions : Un onglet chat s'ouvre
- Cas général : L'utilisateur a la possibilité de communiquer avec son ami
- Cas exceptionnel : Si l'ami se déconnecte, le chat se ferme

3.3 En Jeu

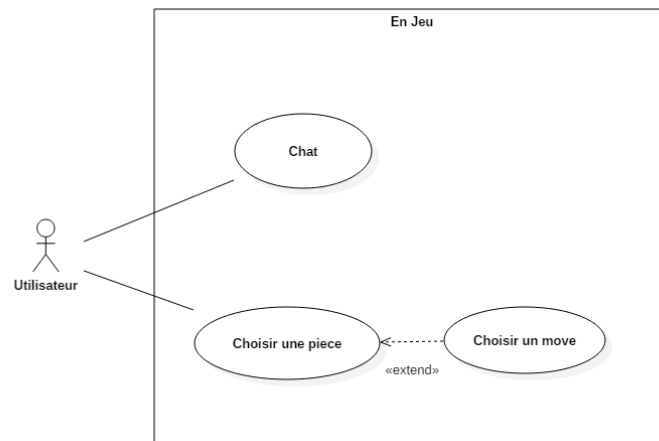


FIGURE 3.4 – Use case en partie

3.3.1 Chat

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- Pré-Conditions : Utilisateur en partie
- Post-Conditions : L'utilisateur peut interagir avec son adversaire
- Cas général : L'utilisateur peut communiquer directement avec son adversaire.
- Cas exceptionnel : /

3.3.2 Choisir une pièce

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : Choisir un mouvement
- Pré-Conditions : Utilisateur en partie, c'est le tour de l'utilisateur, et il clique sur une pièce
- Post-Conditions : Les différents mouvements possibles sont affichés sur l'écran de l'utilisateur
- Cas général : L'utilisateur choisit une pièce, les différents mouvements possibles sont affichés et l'utilisateur peut en choisir un.
- Cas exceptionnel : La partie se termine quand l'adversaire est mis en échec et mat

3.3.3 Choisir un mouvement

- Acteurs : Utilisateur
- Relations avec d'autres cas d'utilisation : Echec et mat / Choisir une pièce
- Pré-Conditions : L'utilisateur a choisi un mouvement parmi les mouvements affichés à la sélection de la pièce
- Post-Conditions : Le mouvement est effectué
- Cas général : Le mouvement sélectionné par l'utilisateur est effectué, et il y a possibilité d'avoir terminé la partie si l'adversaire est en échec et mat. Après que l'utilisateur ait joué, le tour passe à l'adversaire.
- Cas exceptionnel : Si l'utilisateur est en échec, il est obligé de choisir parmi certains mouvements spécifiques. Si l'utilisateur clique sur une case, qui n'est pas dans celle des mouvements possibles, on retourne au cas « Choisir une pièce »

Chapitre 4

Besoin Fonctionnels Système

4.1 Connexion

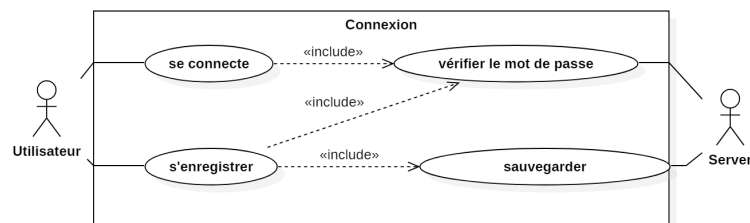


FIGURE 4.1 – Use case de la connexion

4.1.1 Vérifier les Informations

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : Connexion / Inscription
- Pré-Conditions : Un utilisateur tente de se connecter via l'interface de login, ou veut se créer un compte
- Post-Conditions : Connexion de l'utilisateur s'il entre le bon mot de passe, ou création de l'utilisateur s'il n'existe pas
- Cas général :
 - Si l'utilisateur tente de se connecter et entre le mot de passe correspondant à celui sauvegardé dans la base de données, il est connecté.
 - Si l'utilisateur veut se créer un compte qui n'existe pas déjà, il est créé
- Cas exceptionnel :
 - Si le mot de passe est incorrect, il a tout de même la possibilité d'en entrer un autre.
 - Si l'utilisateur existe déjà, rien n'est fait.

4.1.2 Sauvegarder les Informations

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : S'enregistrer
- Pré-conditions : L'utilisateur veut se créer un compte et il a rempli les champs correspondants
- Post-Conditions : Le compte est créé et sauvegardé dans la base de données.
- Cas général : L'utilisateur veut se créer un compte, il entre les données correspondantes et elles sont sauvegardées dans la base de données

- Cas exceptionnel : L'identifiant est déjà pris et il doit en entrer un autre

4.2 Lancer Partie

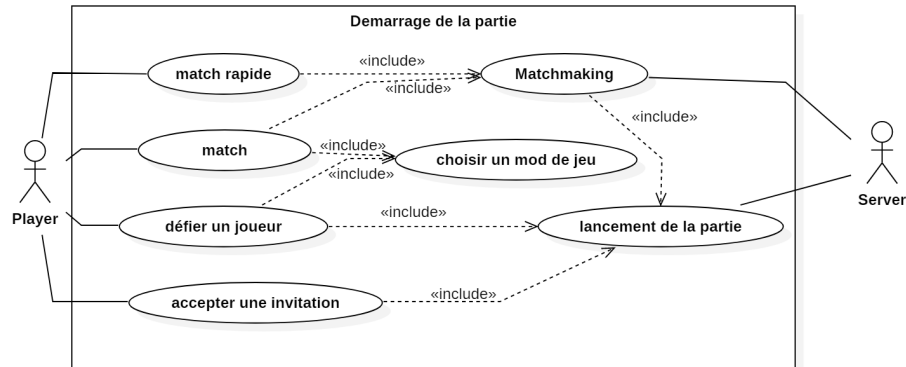


FIGURE 4.2 – Use case du lancement d'une partie

4.2.1 Matchmaking

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : Match Rapide / Match / Lancement de la partie
- Pré-Conditions : Plusieurs utilisateurs sont connectés au serveur, et ont demandé à lancer une partie.
- Post-Conditions : Les clients sont en attente et le serveur essaye d'en relier deux.
- Cas général : Deux utilisateurs sont mis en relation afin qu'ils puissent disputer un match
- Cas exceptionnel : Si quelqu'un d'un certain niveau ne peut pas être relié à quelqu'un du même niveau, le serveur le relie à un autre utilisateur d'un autre niveau.

4.2.2 Lancement de la partie

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : matchmaking / Défier un joueur / Accepter une invitation
- Pré-conditions : Deux joueurs ont été reliés suite au matchmaking ou à une demande amicale.
- Post-Conditions : La partie est lancée.
- Cas général : Le jeu est lancé pour deux utilisateurs qui souhaitent jouer
- Cas exceptionnel : /

4.3 Gérer Partie

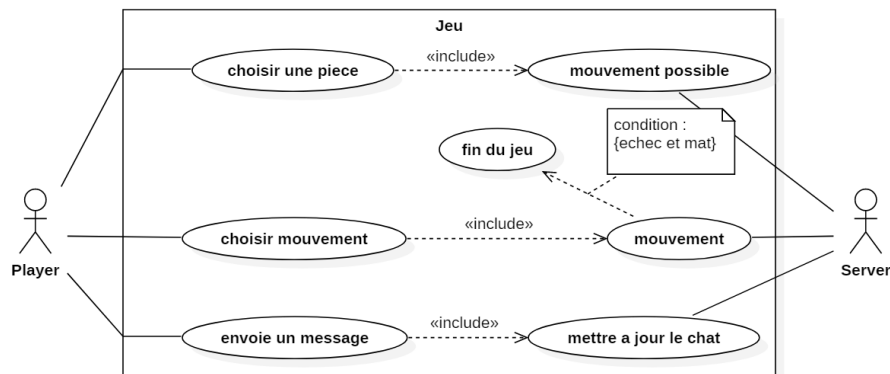


FIGURE 4.3 – Use case de la gestion d'une partie

4.3.1 Indiquer déplacements possibles

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : Choisir une pièce
- Pré-Conditions : Un des joueurs en partie a cliqué sur une pièce
- Post-Conditions : Une liste de tous les déplacements possibles pour la pièce en question sont affichés
- Cas général : Le joueur sélectionne une pièce, et une liste de toutes les possibilités de déplacement pour cette dernière est affichée
- Cas exceptionnel : /

4.3.2 Déplacement

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : Fin du jeu
- Pré-conditions : Le joueur a sélectionné un mouvement
- Post-Conditions : La pièce est déplacée chez les deux joueurs
- Cas général : Le joueur sélectionne une pièce et ensuite un mouvement, le serveur s'occupe de déplacer cette pièce.
- Cas exceptionnel : Si à la suite d'un mouvement, l'adversaire est en état d'échec et mat la partie est finie.

4.3.3 Mise à Jour Chat

- Acteurs : Serveur
- Relations avec d'autres cas d'utilisation : /
- Pré-Conditions : Deux joueurs doivent être en partie
- Post-Conditions : Le chat est constamment mis à jour en fonction des messages reçus.
- Cas général : Le chat est constamment mis à jour en fonction des messages reçus.
- Cas exceptionnel : /

Chapitre 5

Besoin Non Fonctionnels

5.1 Accessibilité

Le jeu doit être accessible à n'importe qui et être facile à utiliser. Pas de complications dans l'interface, les différentes possibilités du jeu sont présentées clairement.

5.2 Sécurité

Les comptes des utilisateurs doivent être protégés par un mot de passe.

5.3 Exigences de Performance

Le serveur doit être stable. Le programme doit être stable. Le temps d'attente pour trouver une partie ne doit pas être trop long. La connexion des clients doit être suffisamment rapide pour que la connexion avec le serveur soit fonctionnelle.

Chapitre 6

Besoin de Domaines

le Domaine est le Jeu-vidéo

6.1 Compétition

La compétition doit être prise en compte :

- explorer facilement les scores de sorte que des comparaison peuvent facilement avoir lieu
- des observateurs
- les replays doivent être accessibles (ex : strings des coups)
- un rythme pourrait être donné (ex : timer)

6.2 Ambiance

Le Jeu d'échec est un jeu de stratégie :

- Une ambiance calme doit être garantie, pas d'entraves à la concentration
- pas de couleurs trop vives
- pas de mouvements intempestifs (ex : pas de layout dynamique en fond)
- possibilité de désactiver les communications extérieures

6.3 Connexion

Les aléatoires de connexion doivent être gérés :

- matchmaking intelligent regardant aussi le ping
- si déconnexion possibilité de re-connexion avec un certain délai

Chapitre 7

Implémentation du Système

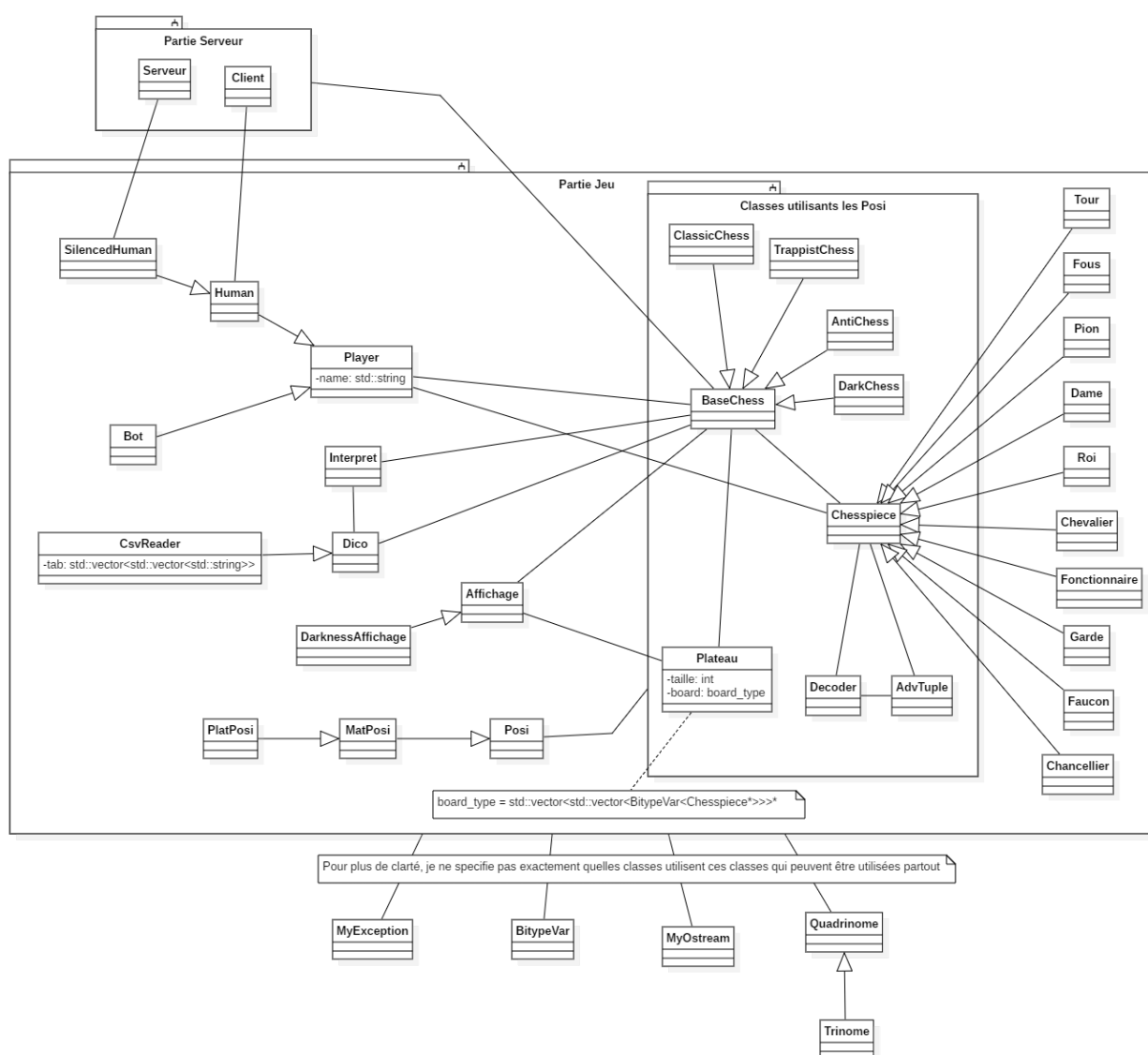


FIGURE 7.1 – Diagramme de classes Partie Jeu

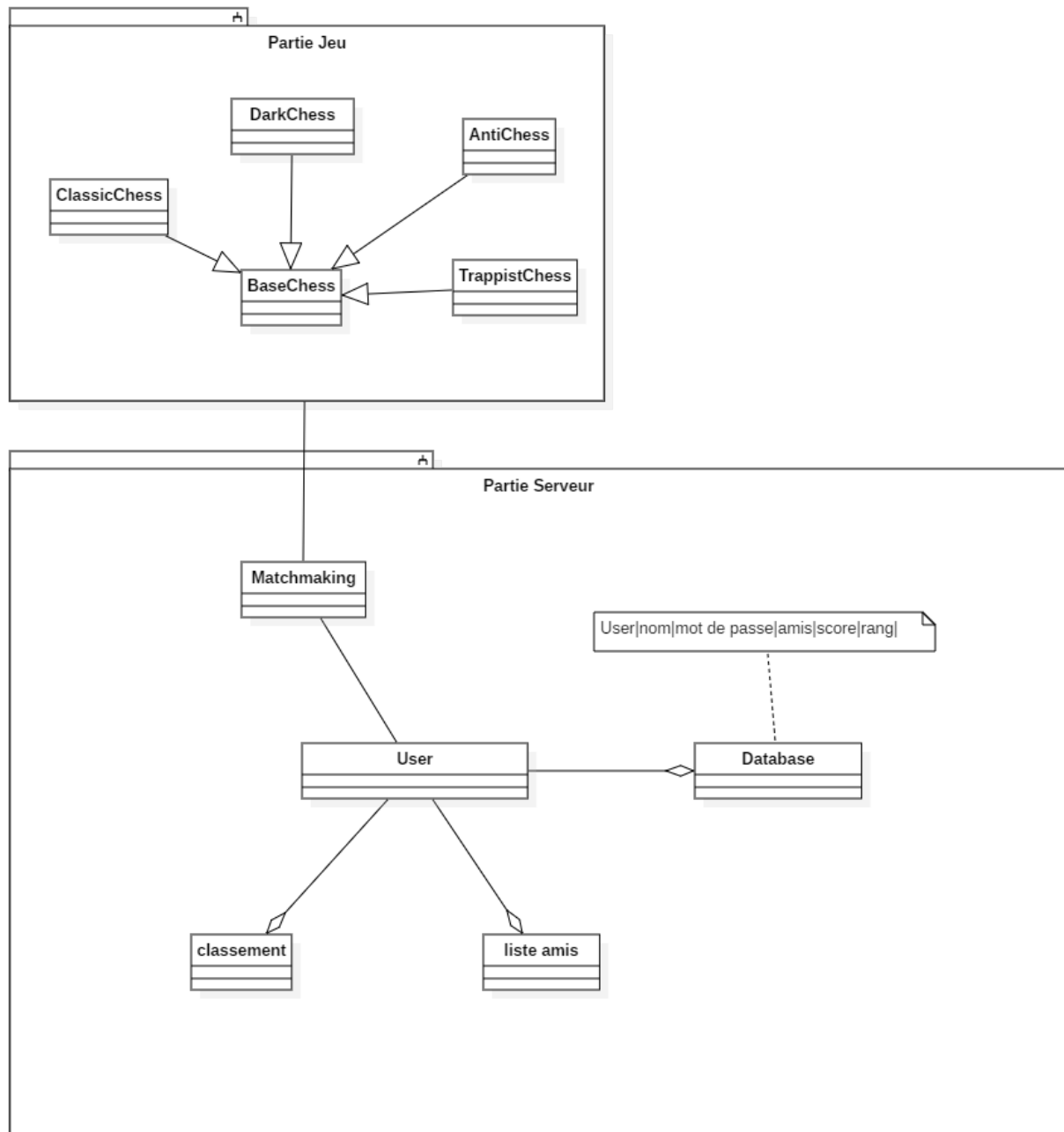


FIGURE 7.2 – Diagramme de classes Partie Serveur

Chapitre 8

Déroulement du Programme

8.1 Diagrammes d'activité

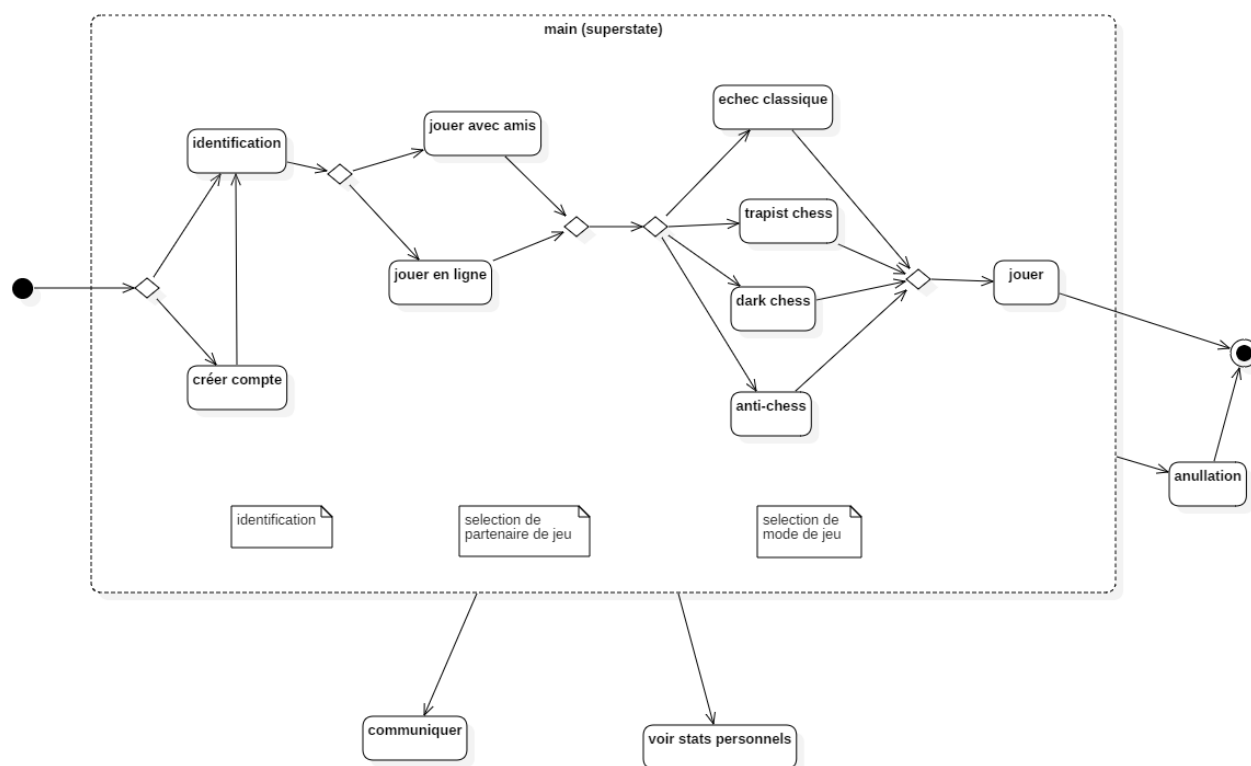


FIGURE 8.1 – Diagramme d'activités du programme global

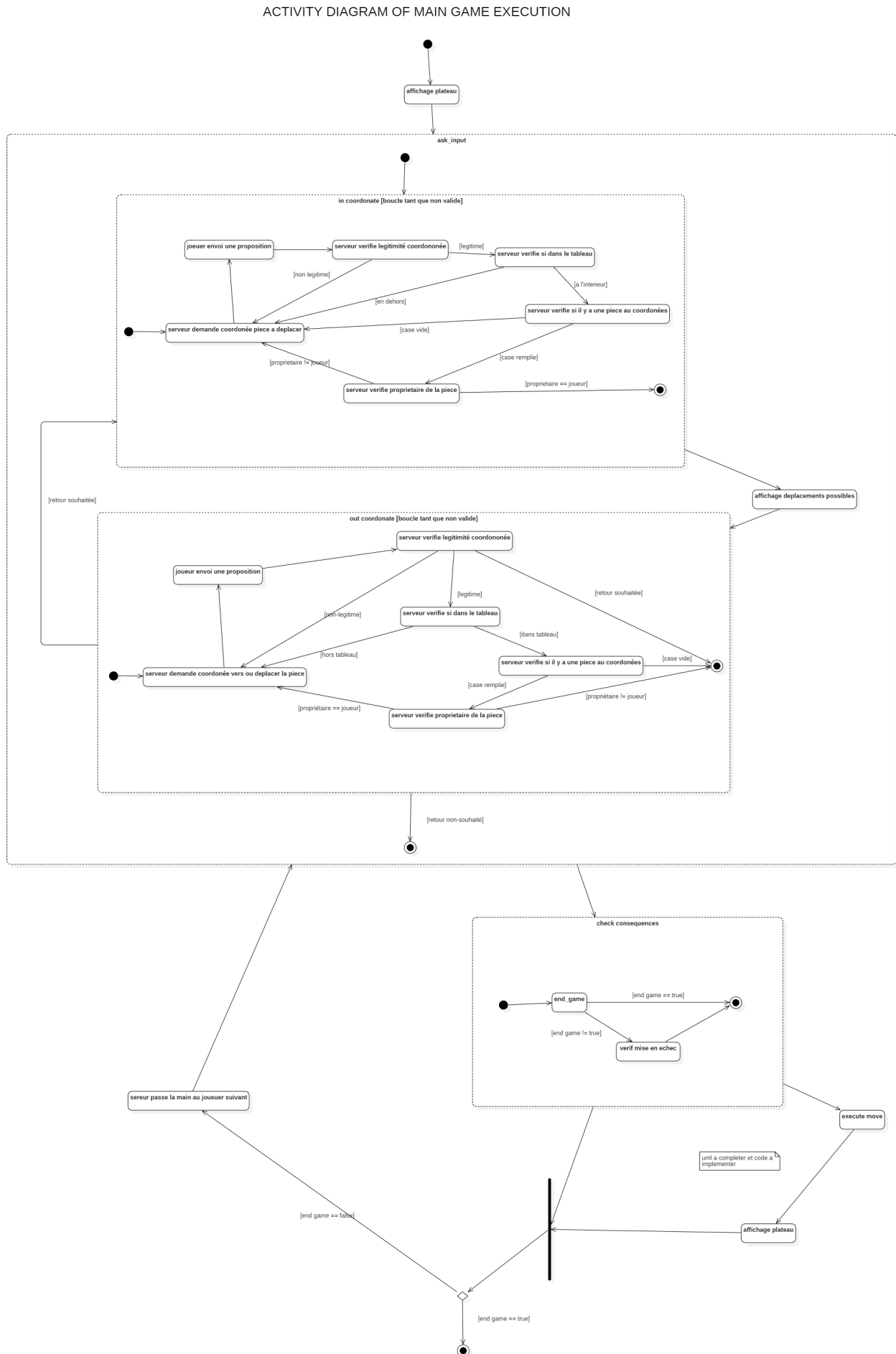


FIGURE 8.2 – Diagramme d'activités de la boucle principale du jeu

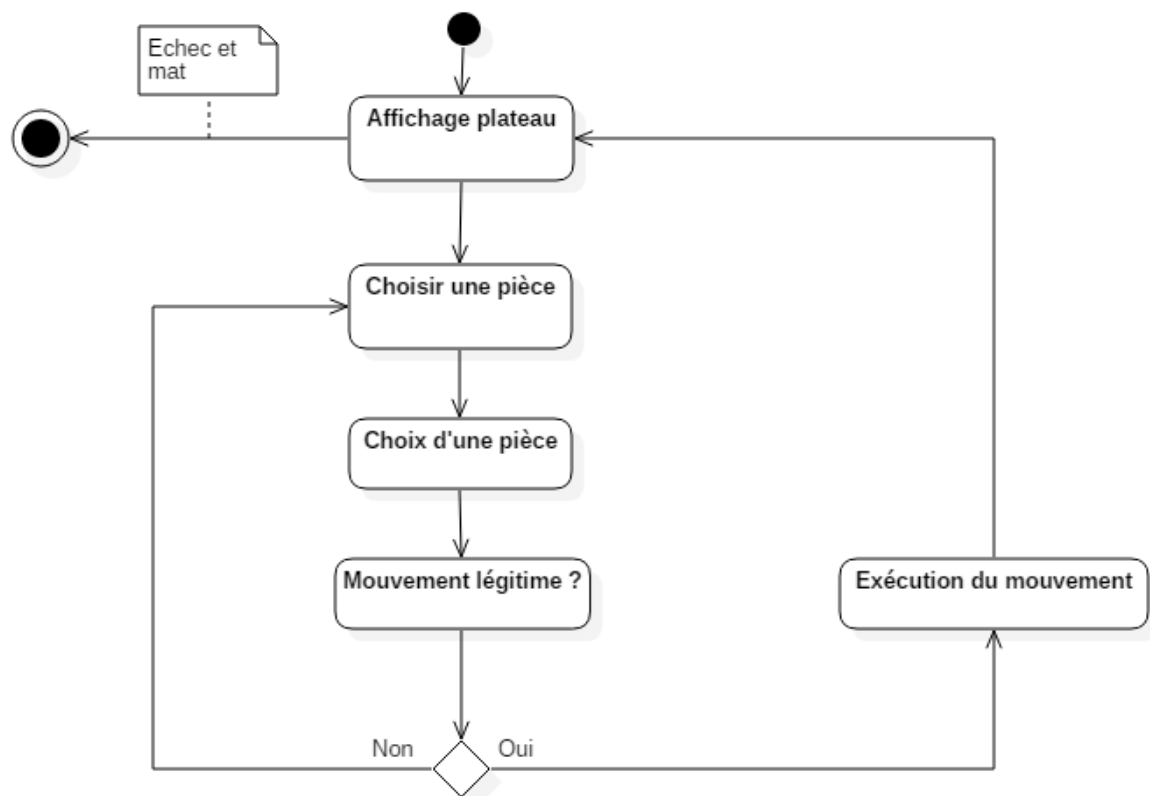


FIGURE 8.3 – Déroulement d'une partie

8.2 Diagramme de Séquence

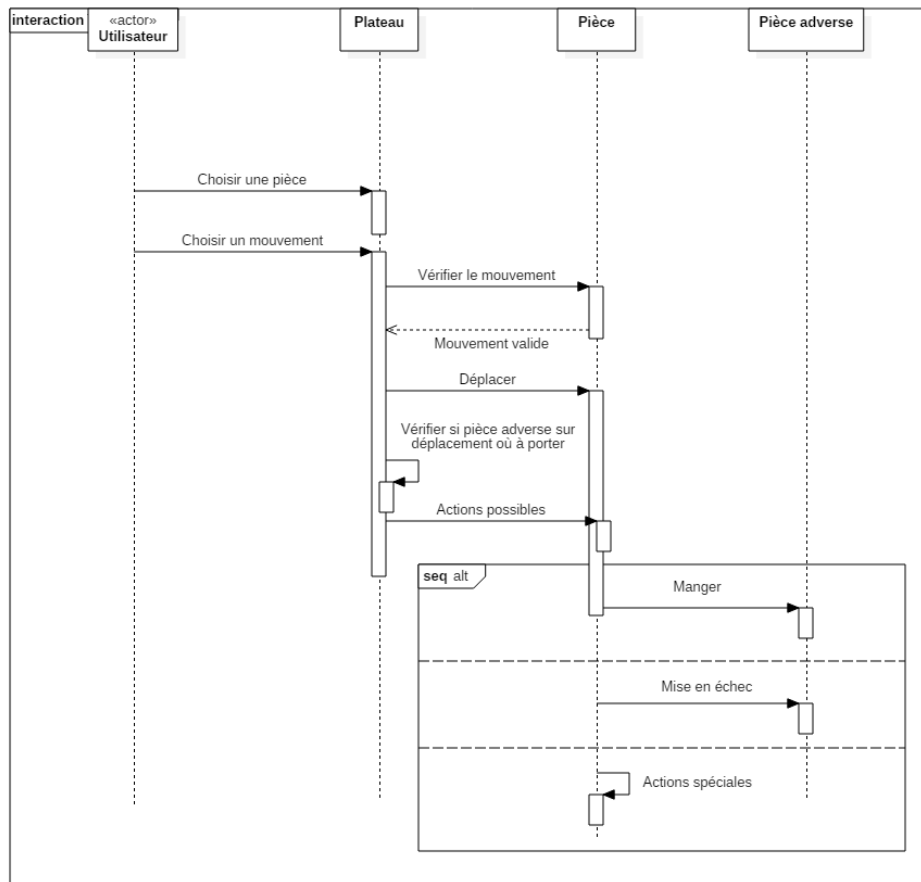


FIGURE 8.4 – Diagramme de séquence du tour d'un utilisateur en jeu

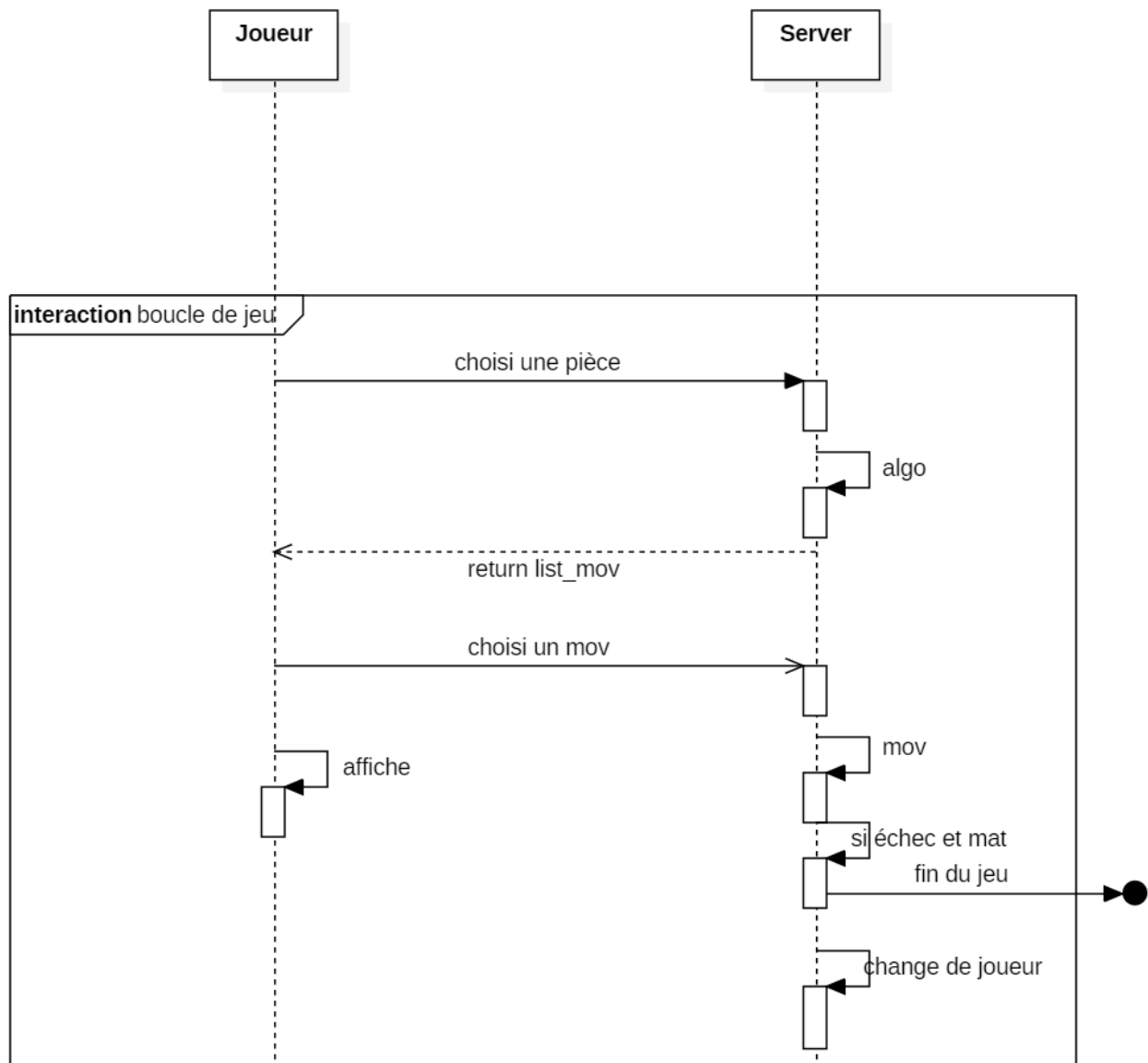


FIGURE 8.5 – Diagramme de séquence de la boucle de jeu

Table des figures

2.1	Classic chess	3
2.2	Trappist chess	4
2.3	Dark chess	5
3.1	Use case global d'utilisateur	6
3.2	Use case connexion	7
3.3	Use Case avant une partie	8
3.4	Use case en partie	10
4.1	Use case de la connexion	11
4.2	Use case du lancement d'une partie	12
4.3	Use case de la gestion d'une partie	13
7.1	Diagramme de classes Partie Jeu	16
7.2	Diagramme de classes Partie Serveur	17
8.1	Diagramme d'activités du programme global	18
8.2	Diagramme d'activités de la boucle principale du jeu	19
8.3	Déroulement d'une partie	20
8.4	Diagramme de séquence du tour d'un utilisateur en jeu	21
8.5	Diagramme de séquence de la boucle de jeu	22
B.1	interface identification	26
B.2	interface inscription	26
B.3	interface menu	27
B.4	interface sélection amis	27
B.5	interface mode de jeu	27
B.6	interface liste amis	27
B.7	interface classement	27

Appendices

Annexe A

Références

A.1 Variantes d'échecs

Variante d'échec « Trappist chess » :

<https://www.chessvariants.com/invention/trappist-1>

Variante d'échec « Dark chess » :

https://en.wikipedia.org/wiki/Dark_chess

Variante d'échec « Antichess / Losing chess » :

https://en.wikipedia.org/wiki/Losing_Chess

A.2 Code Source utilisée d'internet

A.2.1 Partie Jeu

- lecteur CSV : <https://stackoverflow.com/questions/1120140/how-can-i-read-and-parse-csv-files-in-c> [quasi 100%]
- MyOstream <https://stackoverflow.com/questions/14155364/redirect-the-copy-of-stdout-to-the-file> [quasi 100%]
- Parcourir des dossiers : <http://www.martinbroadhurst.com/list-the-files-in-a-directory-in-c.html> [quasi 100%]
- MyException : <https://stackoverflow.com/questions/134569/c-exception-throwing-stdstring> [peu]

Annexe B

Vision d'avenir

B.1 Idées

B.1.1 Idées non-réalisées

- variante bonus : 3d chess
- mode solo : bot
- menu "help" avec explication des règles

B.1.2 Idées réalisées

- traduction en anglais (découplage de tel sorte que rajouter une langue est facile [ex : texte dans un fichier csv])

B.2 Interfaces

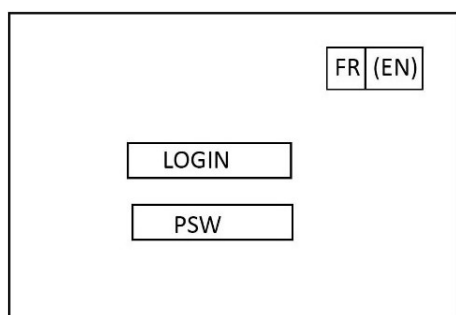


FIGURE B.1 – interface identification

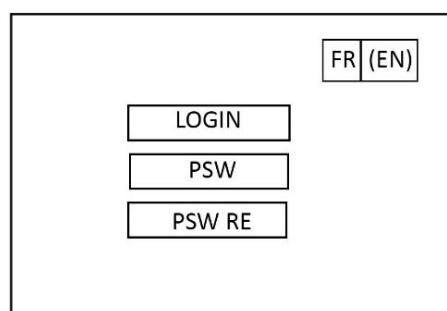


FIGURE B.2 – interface inscription

mon compte	leaderboards
Bienvenue {nom utilisateur} , que voulez-vous faire? <div> <div>jouer contre un ami</div> <div>jouer en ligne</div> </div>	

FIGURE B.3 – interface menu

mon compte	leaderboards
sélectionner votre ami <div> <div> amis1 amis2 amis3 ... </div> <div>retour</div> </div>	

FIGURE B.4 – interface sélection amis

MON COMPTE	LEADERBOARDS
CHESS CLASSIC	VARIANTE 1
VARIANTE 2	VARIANTE 3

FIGURE B.5 – interface mode de jeu

LISTE AMIS	+
Utilisateur1	
Utilisateur2	
...	

FIGURE B.6 – interface liste amis

CLASSEMENT	
Score1	Utilisateur1
Score2	Utilisateur2 Utilisateur3
...	...

FIGURE B.7 – interface classement

