

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
1.1 Objectifs initiaux	2
1.2 Objectifs ajouté pendant la conception	3
2 Règles du Jeu	4
2.1 Chess	4
2.2 Trappist Chess	5
2.3 Dark Chess	6
2.4 Anti-Chess	6
3 Besoin Fonctionnels Utilisateurs	7
3.1 Connexion	8
3.1.1 Authentification	8
3.1.2 Inscription	8
3.2 Avant Jeu	9
3.2.1 Lancer une partie	9
3.2.2 Gérer sa liste d'amis	9
3.2.3 Modifier son profil	10
3.2.4 Déconnexion	10
3.2.5 Consulter le classement	10
3.2.6 Ouvrir un chat	10
3.3 En Jeu	11
3.3.1 Chat	11
3.3.2 Choisir une pièce	11
3.3.3 Choisir un mouvement	11
3.4 Consulter le classement	12
4 Besoin Fonctionnels Système	13

4.1	Connexion	13
4.1.1	Vérifier les Informations	13
4.1.2	Sauvegarder les Informations	13
4.2	Lancer Partie	14
4.2.1	Matchmaking	14
4.2.2	Lancement de la partie	14
4.3	Gérer Partie	15
4.3.1	Indiquer déplacements possibles	15
4.3.2	Déplacement	15
4.3.3	Mise à Jour Chat	15
4.4	Système de publicité	16
5	Besoin Non Fonctionnels	17
5.1	Accessibilité	17
5.2	Sécurité	17
5.3	Exigences de Performance	17
6	Besoin de Domaines	18
6.1	Compétition	18
6.2	Ambiance	18
6.3	Connexion	18
7	Implémentation du Système	19
7.1	Section Jeu	19
7.1.1	GameRules	21
7.1.2	GameType	22
7.1.3	GamePieces	23
7.1.4	GamePlayers	24
7.1.5	GameAffichages	25
7.1.6	GamePosis	26
7.1.7	GameDico	27
7.1.8	GamePlateau	28
7.2	Section Client/Serveur	29
7.2.1	Serveur	29
7.2.2	Client	30
8	Déroulement du Programme	31
8.1	Diagrammes d'activité	31
8.2	Diagramme de Séquence	34

Table des figures	37
Appendices	39
A Références	40
A.1 Variantes d'échecs	40
A.2 Code Source utilisée d'internet	40
A.2.1 Partie Jeu	40
B Vision d'avenir	41
B.1 Idées	41
B.1.1 Idées non-réalisées	41
B.1.2 Idées réalisées	41
B.2 Interfaces	41
C Particularités du projet	43
C.1 Particularités Jeu	43
C.2 Particularités Gui	43
C.3 Particularités Serveur	43
D Remerciements	44

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
1.4.6	01.03.2019	D. Quentin	Remplissage attributs class diagram A
1.4.7	02.03.2019	D. Quentin	Nouvelle référence, modification structure "bullets"
1.4.8	02.03.2019	L. Achraf	Diagramme de séquence du matchmaking
1.4.9	03.03.2019	L. Achraf	Modification Diagramme du déroulement d'un tour de jeu
1.4.10	20.03.2019	D. Quentin	Complétion class diagram A
1.4.11	21.03.2019	D. Quentin	Complétion class diagram A
1.5.0	22.03.2019	D. Quentin	split class diagram A
1.5.1	23.03.2019	D. Quentin	continue split class diagram A
1.5.2	24.03.2019	D. Quentin	continue class diagrams Jeux
1.5.3	30.03.2019	B. Matias	Description pendule et publicité
1.5.4	30.03.2019	D. Quentin	Description classdiagrams
1.5.5	31.03.2019	D. Quentin	Description classdiagrams, ajout structure et complétion annexe
1.5.6	01.04.2019	D. Quentin	Complétion class diagram A et B

Chapitre 1

Objectifs



FIGURE 1.1 – logo projet groupe

1.1 Objectifs initiaux

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.

1.2 Objectifs ajouté pendant la conception

Nouvelle possibilité de jouer à un mode de jeu en temps réel et dans un mode chronométré appelé pendule en plus de mode normale appelé TourParTour. Un système publicitaire est demandé pour le programme, l'objectif principal des publicités étant de faire rire. Un système de pré-move doit être implémenter c.a.d un système permettant de jouer des mouvement a l'avance.

Chapitre 2

Règles du Jeu

2.1 Chess

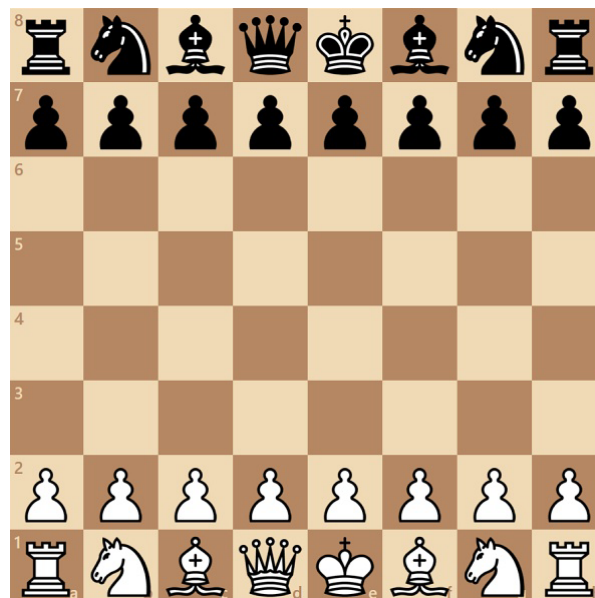


FIGURE 2.1 – Classic chess

Plateau de 8x8 cases.

Le but : Mettre le roi adverse dans une situation telle qu'il ne peut plus bouger sans risque de se faire prendre ou bien de directement prendre celui-ci.

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.

Roque avec déplacement pendant lequel le roi ne peut pas se mettre en danger (ne peut pas passer par une case où il serait en échec).

2.2 Trappist Chess

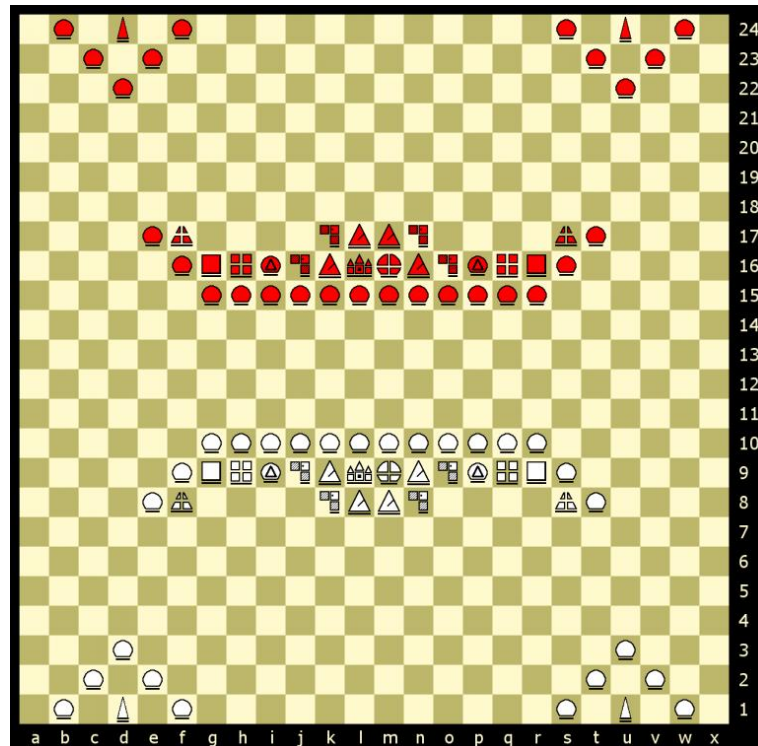


FIGURE 2.2 – Trappist chess

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

Jeu se jouant habituellement sur un plateau de 21x21 ou 24x24. Mêmes règles que les échecs classiques. La version 24x24 semblant plus impressionnante ce fût celle choisit pour être implémenter. 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 (a 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 d'échec se jouant avec les mêmes règles que le jeu original en rajoutant l'apparition d'un brouillard de guerre. On peut voir une pièce si on la possède ou si une de nos pièces pourrait la capturer ou se déplacer sur sa case si elle était vide.

2.4 Anti-Chess

Variantes se jouant avec les mêmes règles que le jeu original en apportant quelques ajouts : Jeu d'échecs classique ou 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

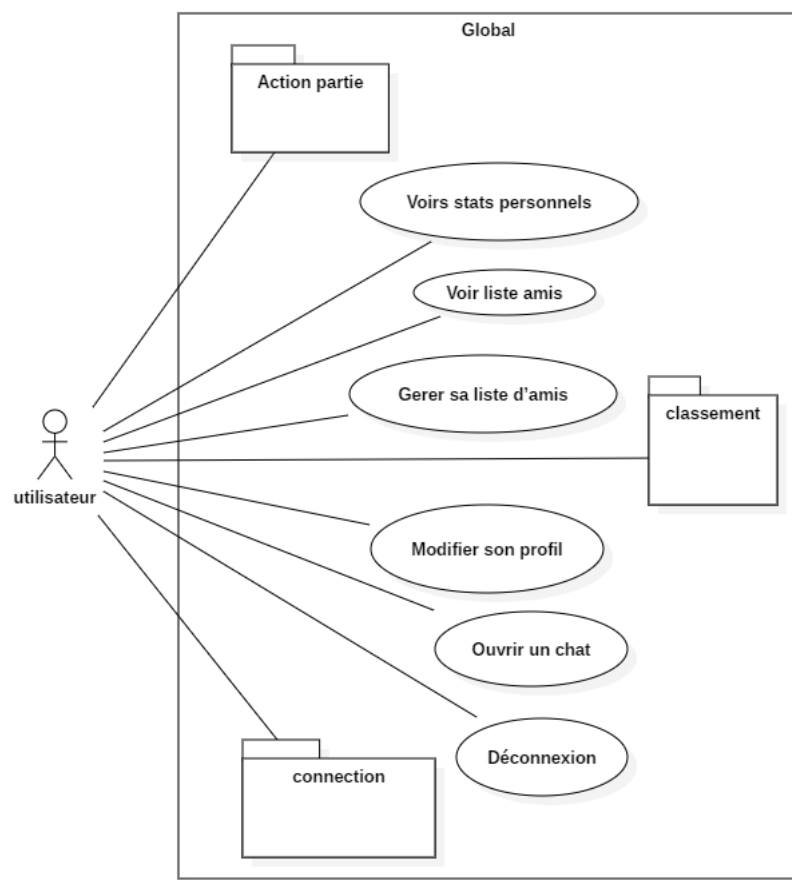


FIGURE 3.1 – Use case global d'utilisateur

Ce use case rassemble toutes les actions pouvant être effectuées par l'utilisateur, certains use cases furent groupés en packages pour plus de lisibilité.

3.1 Connexion

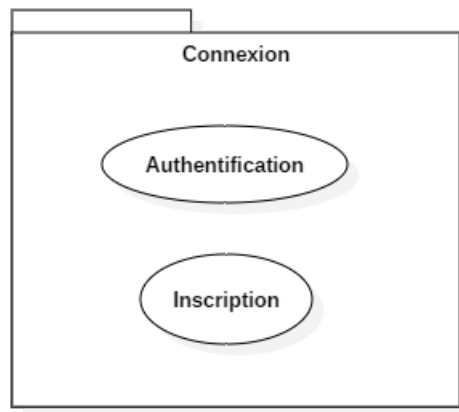


FIGURE 3.2 – Use case connexion

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

3.1.1 Authentification

- Acteur : Utilisateur
- Relations avec d'autres cas d'utilisation : /
- 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

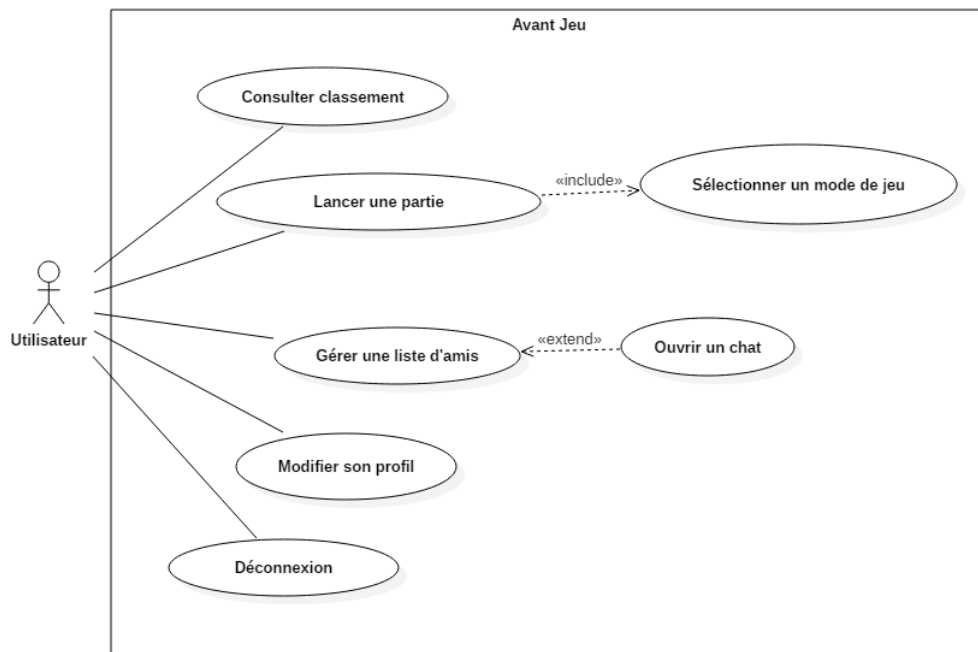


FIGURE 3.3 – Use Case avant une partie

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

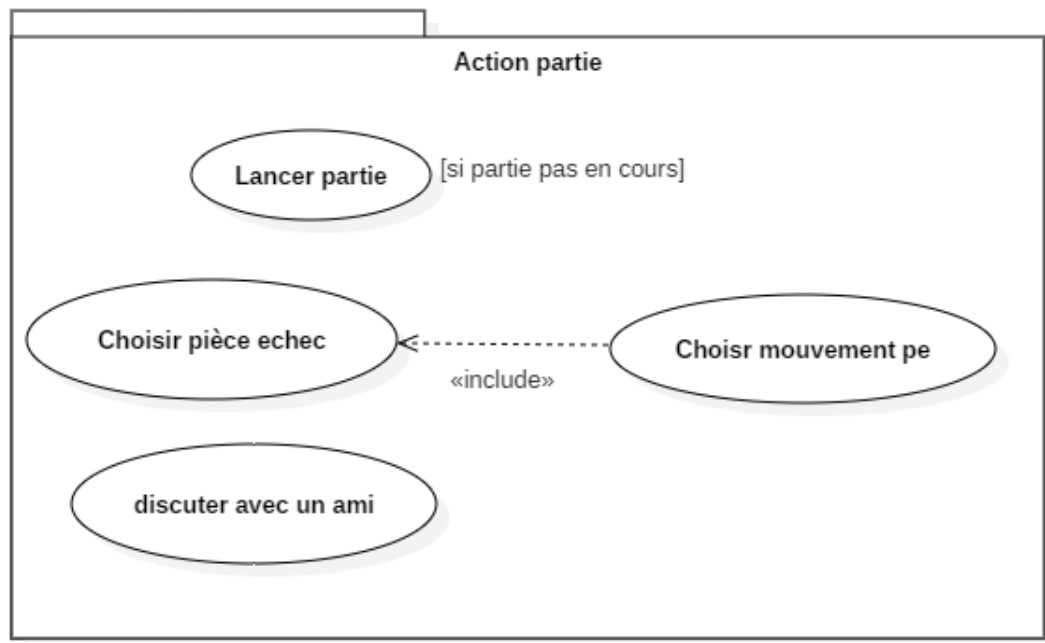


FIGURE 3.4 – Use case en partie

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 mouvement 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 »

3.4 Consulter le classement

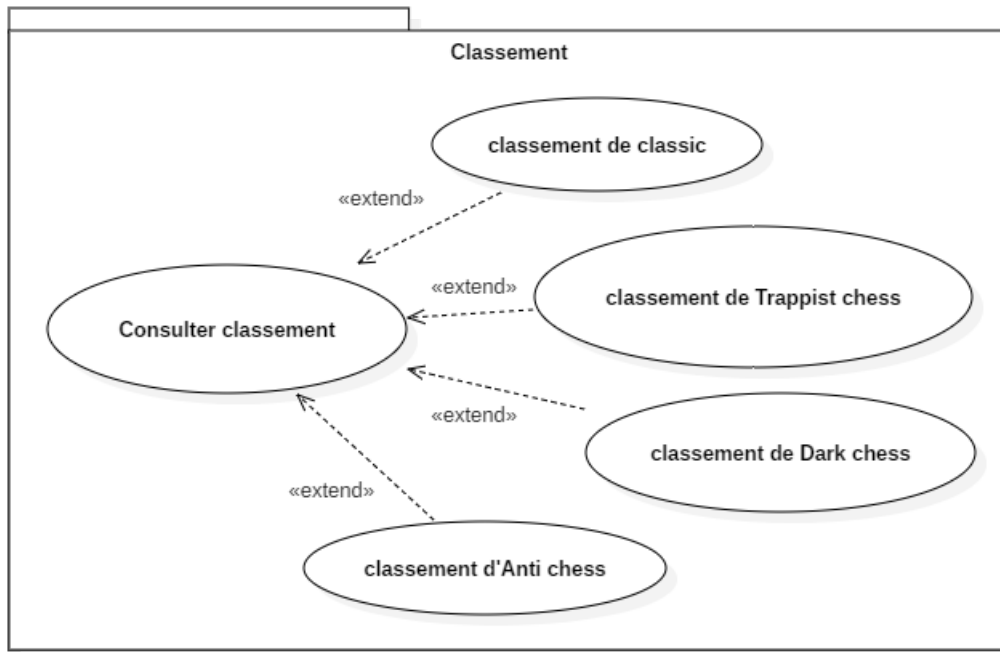


FIGURE 3.5 – Use case class

Chapitre 4

Besoin Fonctionnels Système

4.1 Connexion

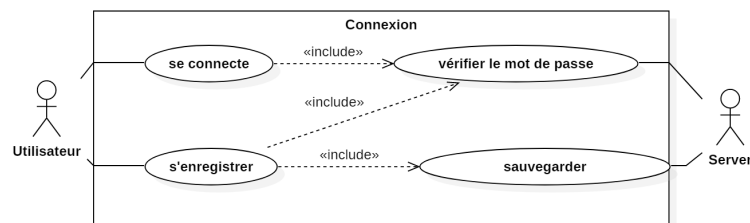


FIGURE 4.1 – Use case de la connexion

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

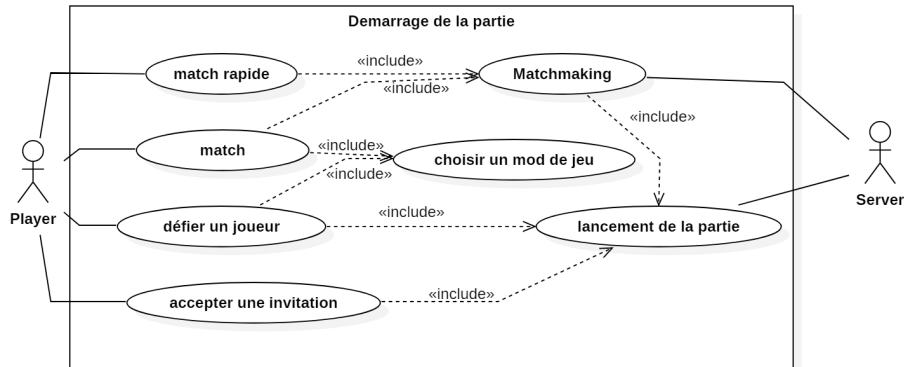


FIGURE 4.2 – Use case du lancement d'une partie

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

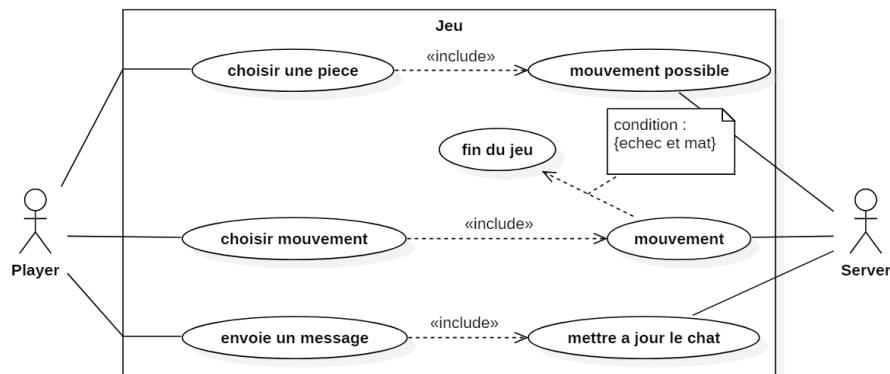


FIGURE 4.3 – Use case de la gestion d'une partie

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 : /

4.4 Système de publicité

- Acteurs : Client
- Relations avec d'autres cas d'utilisation : /
- Pré-Conditions : L'utilisateur est en partie en mode GUI.
- Post-Conditions : La publicité affichée à l'écran est mise à jour à intervalle régulier.
- Cas général : Des publicités sont affichées à l'utilisateur.
- 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

7.1 Section Jeu

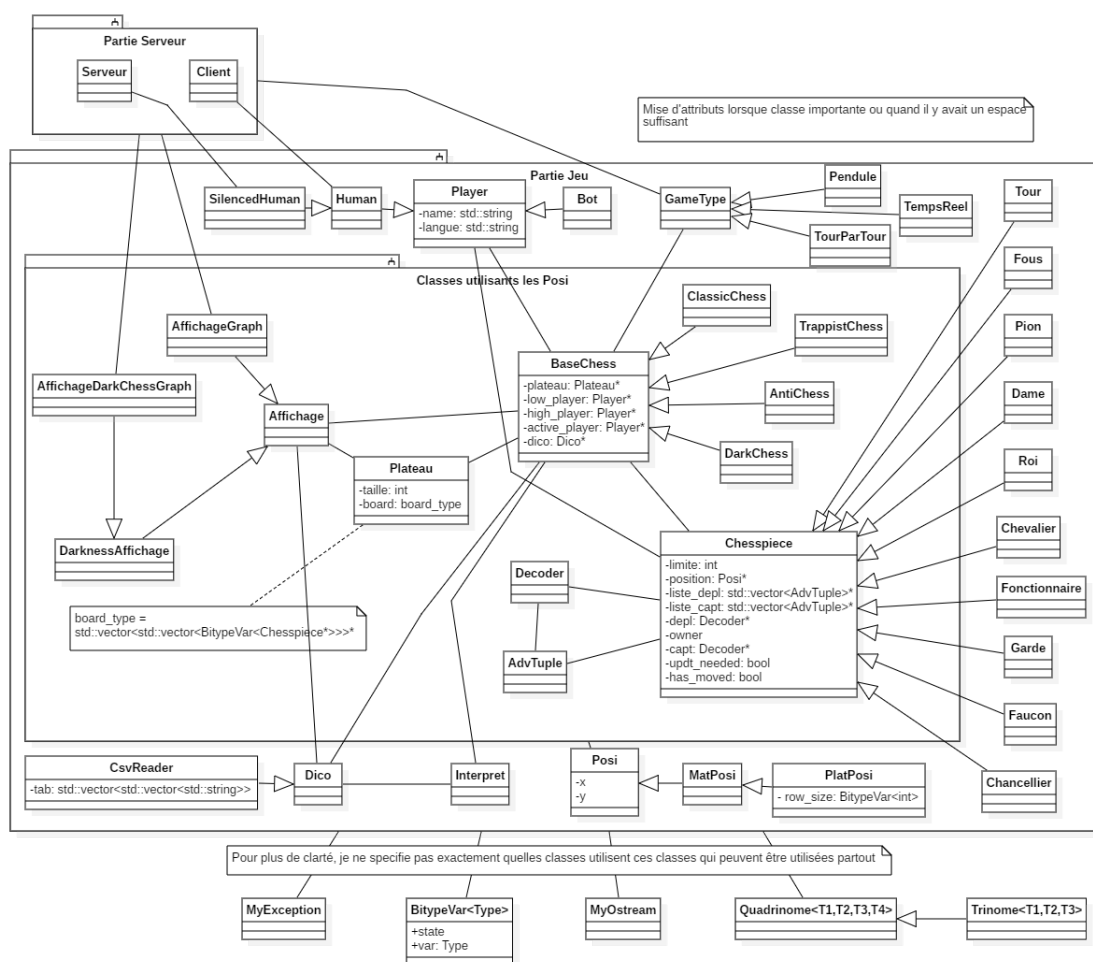


FIGURE 7.1 – Diagramme de classes Partie Jeu Complet

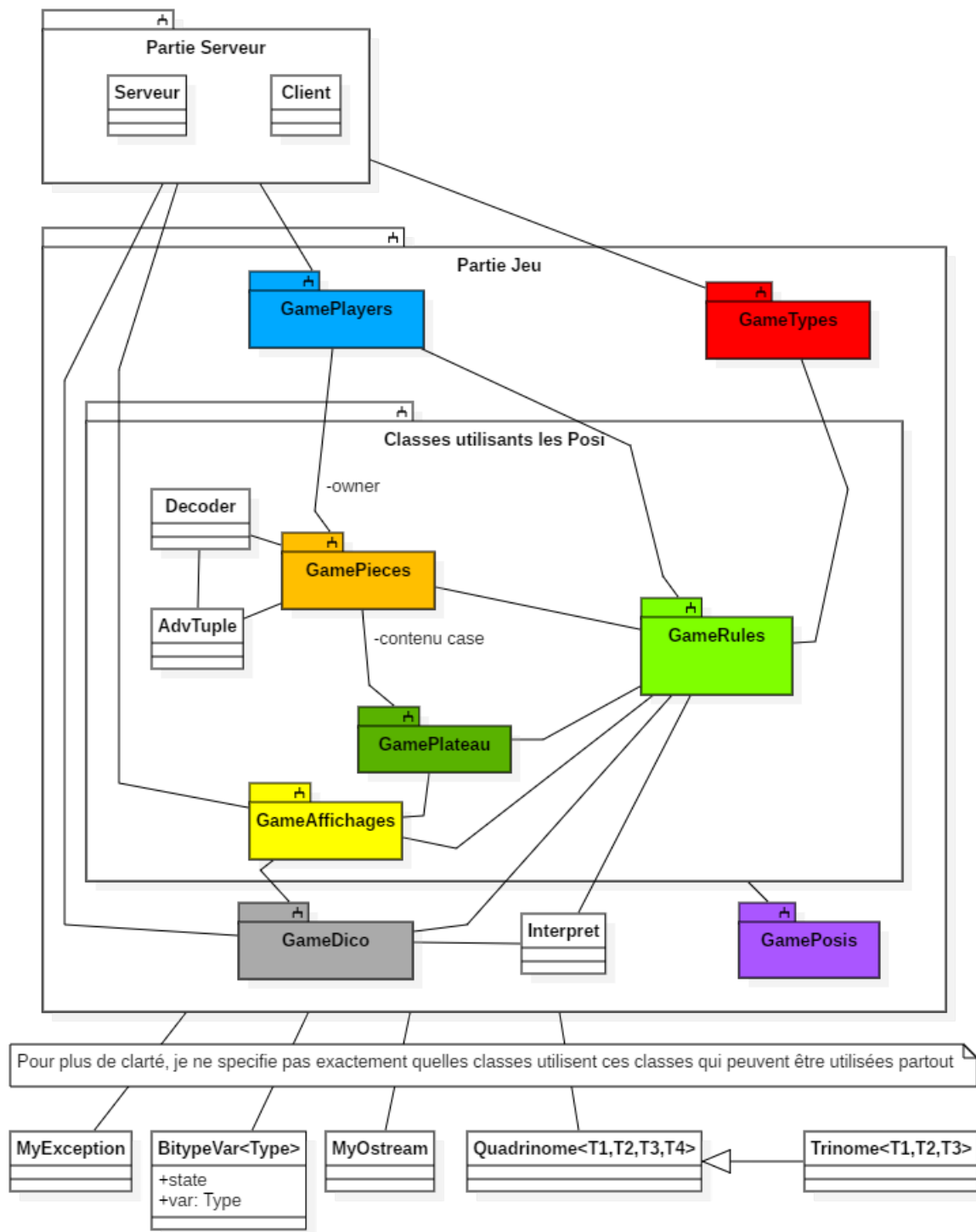


FIGURE 7.2 – Diagramme de classes Partie Jeu Global

7.1.1 GameRules

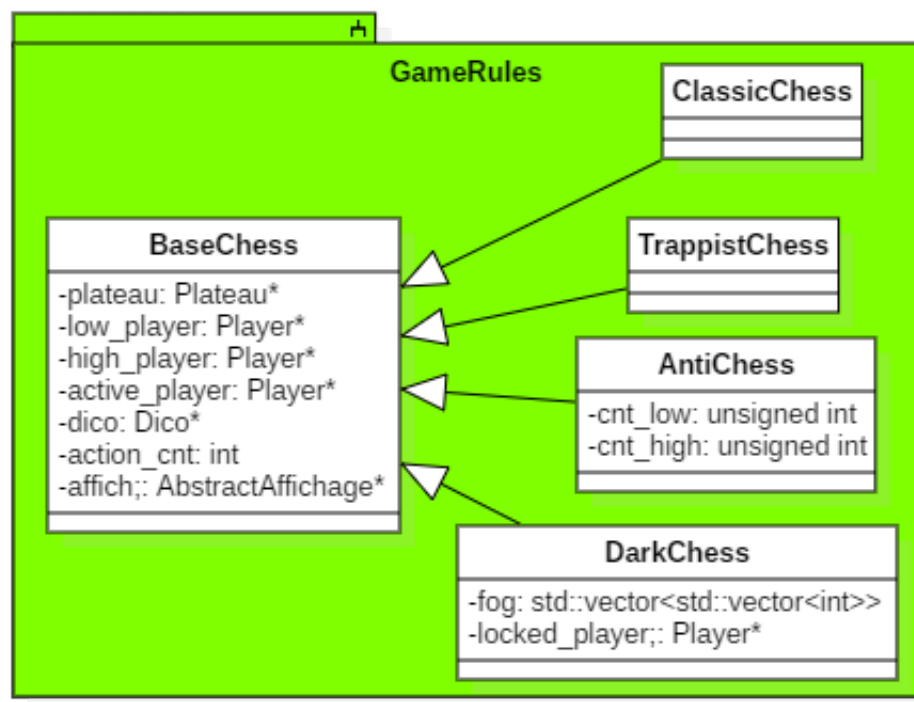


FIGURE 7.3 – Diagramme de classes Partie Jeu GameRules

GameRules

GameRules est le nom donné au regroupement de classes implémentant les règles d'échecs

BaseChess

Classe mère implémentant les bases des 4 classes filles ClassicChess, TrappistChess, AntiChess et DarkChess. Classe principale de la partie Jeu car implémente toutes la logique d'une partie d'échec ainsi que la gestion de ses composantes.

ClassicChess

Classe implémentant les règles classique d'échec.

TrappistChess

Classe implémentant les règles du TrappistChess. Notamment fonction override pour enlever le roque.

DarkChess

Classe implémentant les règles du DarkChess. Nouvel attribut par rapport aux autres classes filles de BaseChess : locked_player qui indique sur quel joueur l'affichage doit afficher le brouillard.

7.1.2 GameType

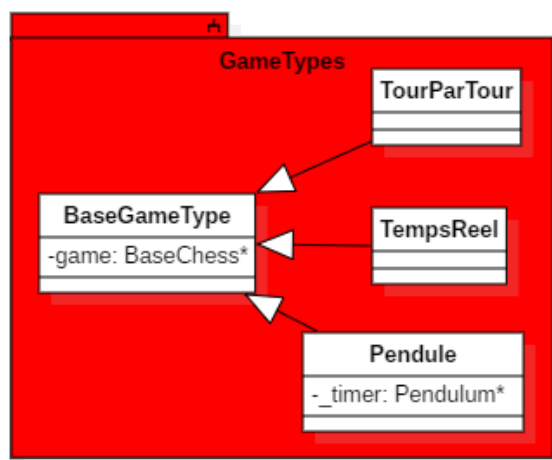


FIGURE 7.4 – Diagramme de classes Partie Jeu GameTypes

GameType

GameType est le nom donnée au regroupement de classe manipulant un objet de type BaseChess lui donnant une dimension temporelle et implémentant une structure d'appel de celui-ci

BaseGameType

Classe mère implémentant les bases des 3 classes filles Pendule, TourParTour et TempsReel. Classe manipulant un jeu de type BaseChess, lui donnant le comportement temporel voulu.

Pendule

La pendule d'échec est un mode de jeu supplémentaire qui peut être vu comme une difficulté ajoutée à la partie.

C'est un système comprenant un chronomètre par joueur à fin de limiter leurs temps de jeu, si un des joueurs dépasse le temps autorisé alors celui-ci se voit déclaré perdant de la partie.

La pendule oblige donc au joueur à prendre le temps en considération lors de sa partie et donc de jouer et prendre des décisions plus rapidement.

TourParTour

Classe manipulant un jeu de type BaseChess, permettant chacun a leur tour un input de coordonnée dans le jeu. Input qui lorsque est erronée est redemandé automatiquement par BaseChess

TempsReel

Classe vide implémenté pour une cohérence de structure, en réalité n'ajoute rien a BaseGameChess mais c'est plus logique de ne pas devoir hériter de TempsReel mais d'une classe commune

7.1.3 GamePieces

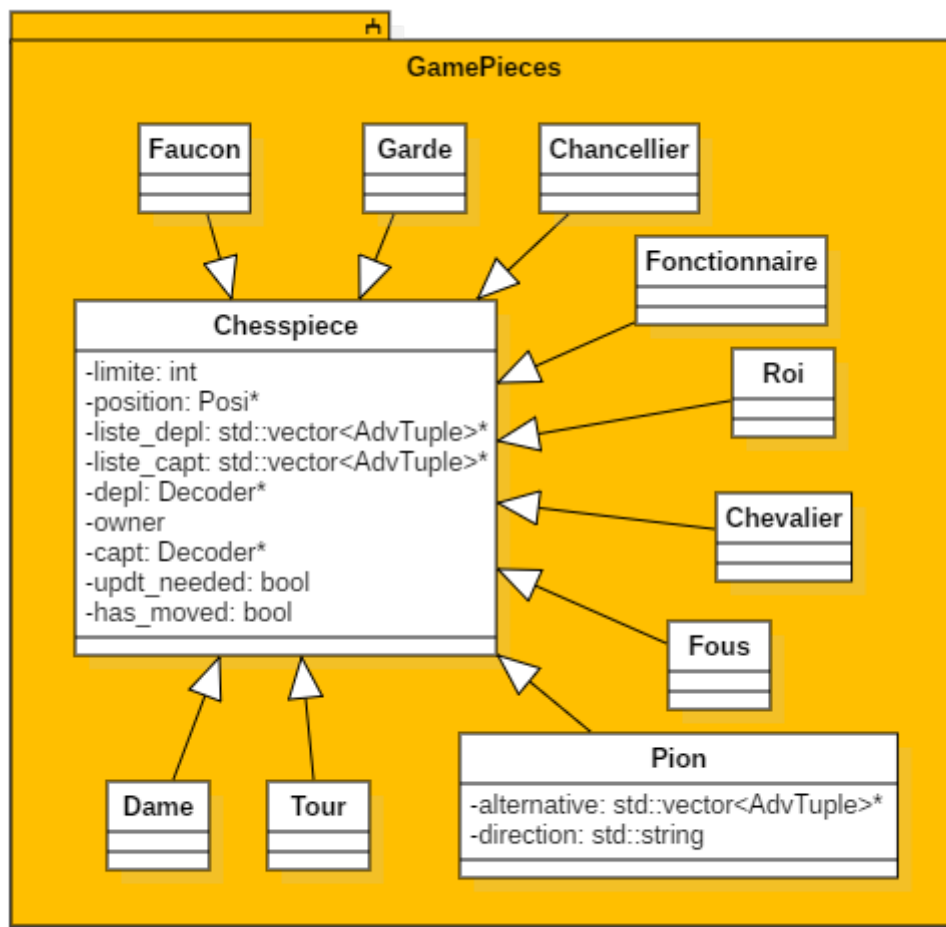


FIGURE 7.5 – Diagramme de classes Partie Jeu GamePieces

GamePieces

GameType est le nom donné au regroupement de classe implémentant des pièces d'échecs.

Chesspiece

Classe mère implémentant une pièce abstraite, différent d'un pion a la conception puisqu'un peut évoluer alors que les autres pièces ne le peuvent pas.

Classes filles de Chesspiece

Le reste des classes présentées sur le diagramme sont des classes implémentant des comportements spécifiques aux pièces d'échec.

7.1.4 GamePlayers

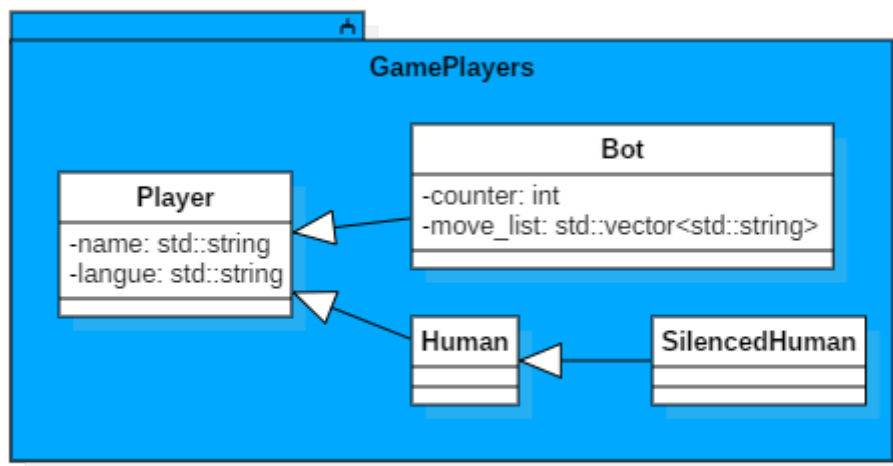


FIGURE 7.6 – Diagramme de classes Partie Jeu GamePlayers

GamePlayers

GamePlayers est le nom donné au regroupement de classe implémentant les différents joueurs du jeu.

Player

Classe mère implémentant le bases des joueurs du jeu.

Human

Classe effectuant des cin et des cout lorsque demandé.

SilencedHuman

Classe Human auquel on enlève la capacité de parler, cependant classe fille car doit contenir tout ce que Human contient aussi et peut recevoir d'autres facultés non disponibles pour human. Silenced est bien une sorte d'Human donc l'héritage est valable malgré tout.

Bot

Classe récupérant des inputs d'un fichier csv lorsqu'on lui demande un cin. Classe crée pour uniformisation windows et linux suite au fait que exécutable«fichier_texte.txt ne fonctionnerait pas dans tout les cas et/ou de façon uniforme/identique.

7.1.5 GameAffichages

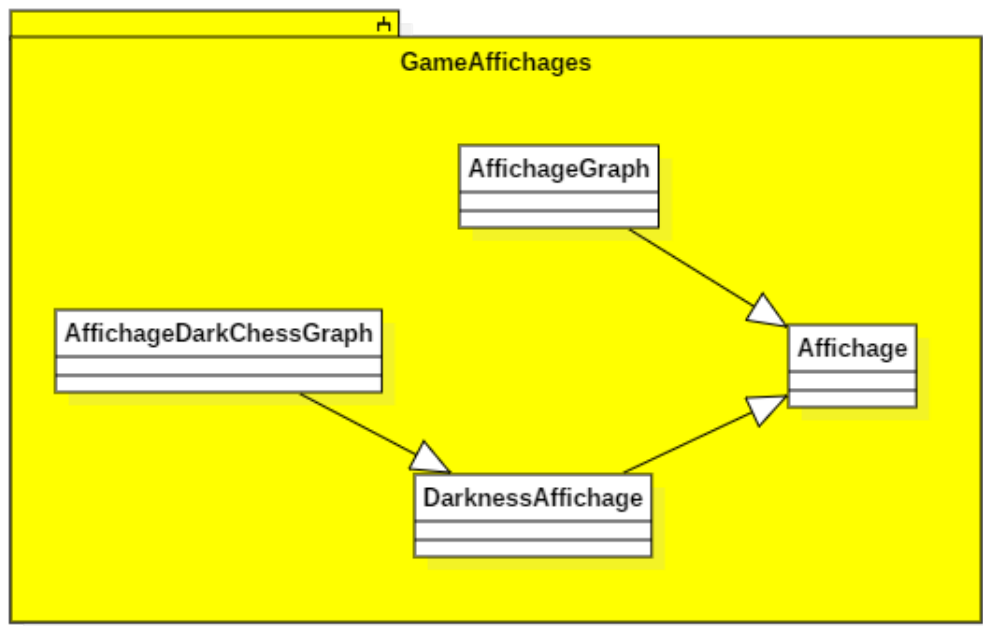


FIGURE 7.7 – Diagramme de classes Partie Jeu GameAffichages

GameAffichages

GameAffichages est le nom donnée au regroupement de classe implémentant les classes liant les interfaces du jeu aux interfaces globales terminales ou graphiques.

Affichage

Classe de lien entre le jeu et la classe globale d’affichage terminal pour BaseChess, ClassicChess, AntiChess et TrappistChess.

DarknessAffichage

Classe de lien entre le Darkchess et la classe globale d’affichage terminal. hérite de la classe Affichage.

AffichageGraph

Classe de lien entre le jeu et la classe globale d’affichage graphique pour BaseChess, ClassicChess, AntiChess et TrappistChess.

AffichageDarkChessGraph

Classe de lien entre le Darkchess et la classe globale d’affichage graphique. hérite de la classe AffichageGraph.

7.1.6 GamePosis

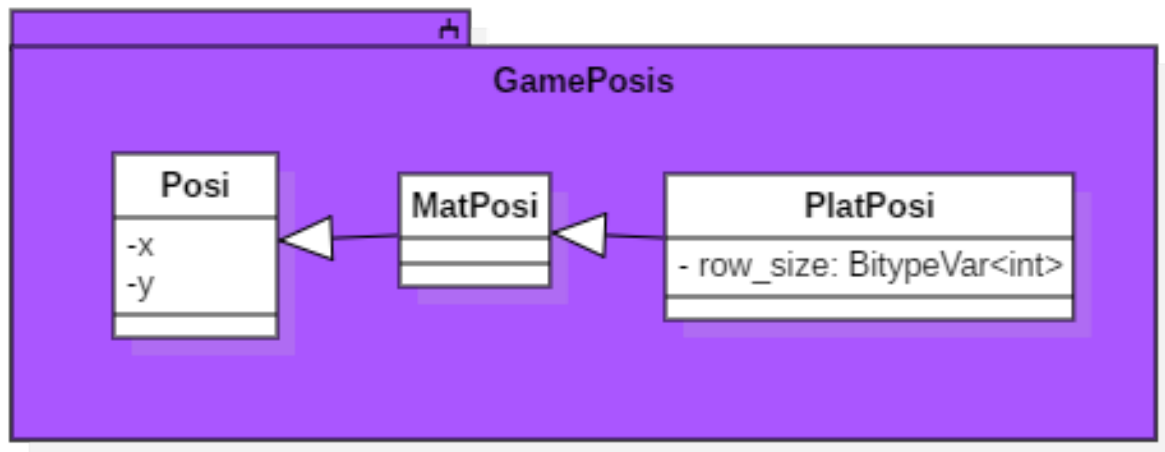


FIGURE 7.8 – Diagramme de classes Partie Jeu GamePosis

GamePosis

GamePosis est le nom donné au regroupement de classe de gestions de coordonnées utilisées par le jeu. L'utilisation de ces objets permet de ne plus se préoccuper de la représentation de la coordonnée et permet la conversion facile entre les différentes représentation possibles.

Posi

La classe Posi intègre la notion de coordonnée partielle x et y. La coordonnée est représenté sous forme de paire d'entier.

MatPosi

La classe MatPosi hérite de Posi et y ajoute la notion de colonne et de ligne. La coordonnée peut maintenant se représenter sous forme de string en plus de la forme en paire d'entiers. ex : A1 = (0,0)

PlatPosi

La classe PlatPosi hérite de MatPosi et y ajoute des contraintes limitant les coordonnées dans des limites données par la taille d'un plateau. La coordonnée peut maintenant être représenté sous forme d'un entier unique grâce a la taille d'un plateau, en effet cet entier est obtenu comme suit : $n^{\circ}\text{ligne} * \text{taille_ligne} + n^{\circ}\text{colonne}$. pour une matrice carrée $n=3$ on aura B2 = (2,2) = 4 [ici ligne 1 car commence a 0 $\rightarrow (1*2)+2 = 4$]

7.1.7 GameDico

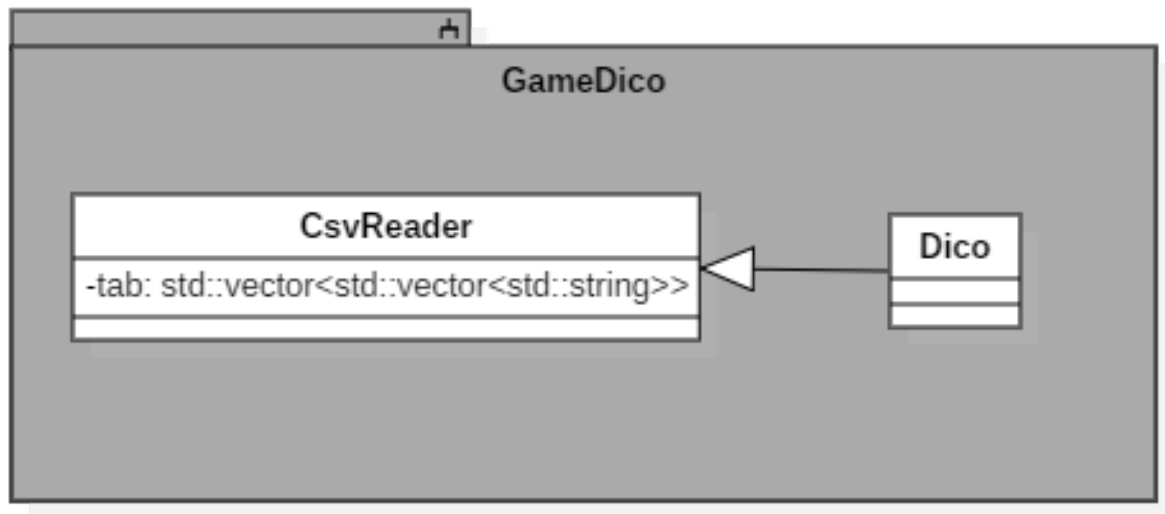


FIGURE 7.9 – Diagramme de classes Partie Jeu GameDico

GameDico

GameDico est le nom donné au regroupement de classe de récupération de données a partir d'un fichier csv.

CsvReader

Classe permettant la lecture de fichier csv avec une légende horizontale sur la 1e ligne du fichier, chaque colonne représentant un vecteur de données liée a la légende.

Dico

Classe héritant de CsvReader et ajoute la notion d'une 2e légende cette fois ci verticale sur la 1e colonne. Dico permet la recherche de données a l'intersection des 2 légendes ainsi que la récupération de vecteur de données.

7.1.8 GamePlateau

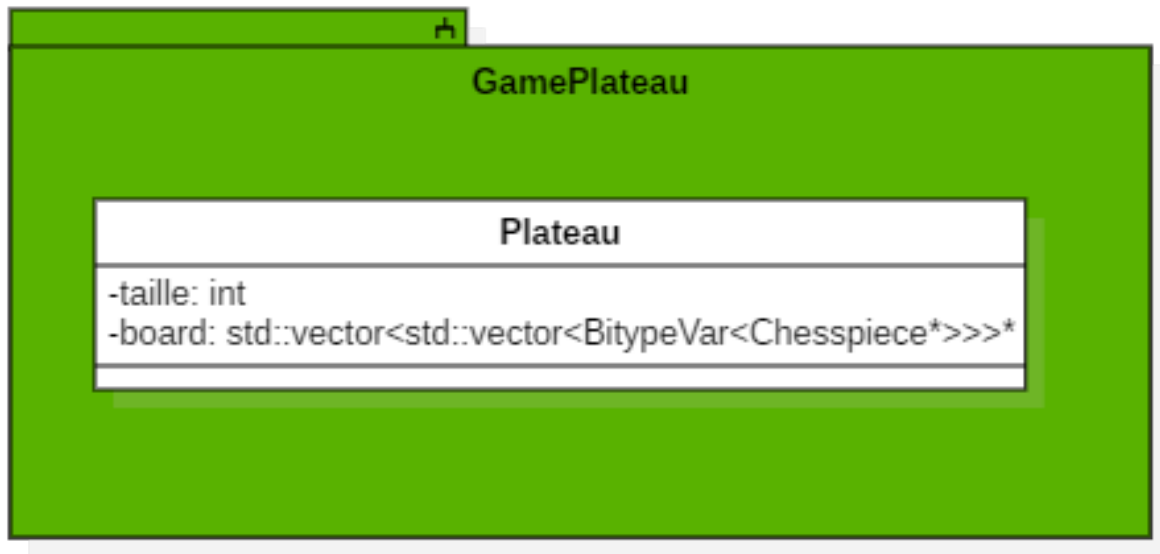


FIGURE 7.10 – Diagramme de classes Partie Jeu GamePlateau

GamePlateau

GamePlateau est le nom donnée au regroupement de classe de gestion de plateau de jeu. Groupe crée afin d'aérer le diagramme de classe Global. En effet la grande taille du type de board encombre bcp ce diagramme.

Le plateau contiendra des Chesspiesses, pièces d'échecs abstraites dont il ignorera l'existence exacte. Le plateau contient en réalité de BityeVar<> qui sont des objets permettant de représenter le None de Python en c++. En effet ces objets contiennent un bool caché (l'état rempli ou vide) ainsi qu'une variable seulement valable si l'état est rempli (true).

7.2 Section Client/Serveur

7.2.1 Serveur

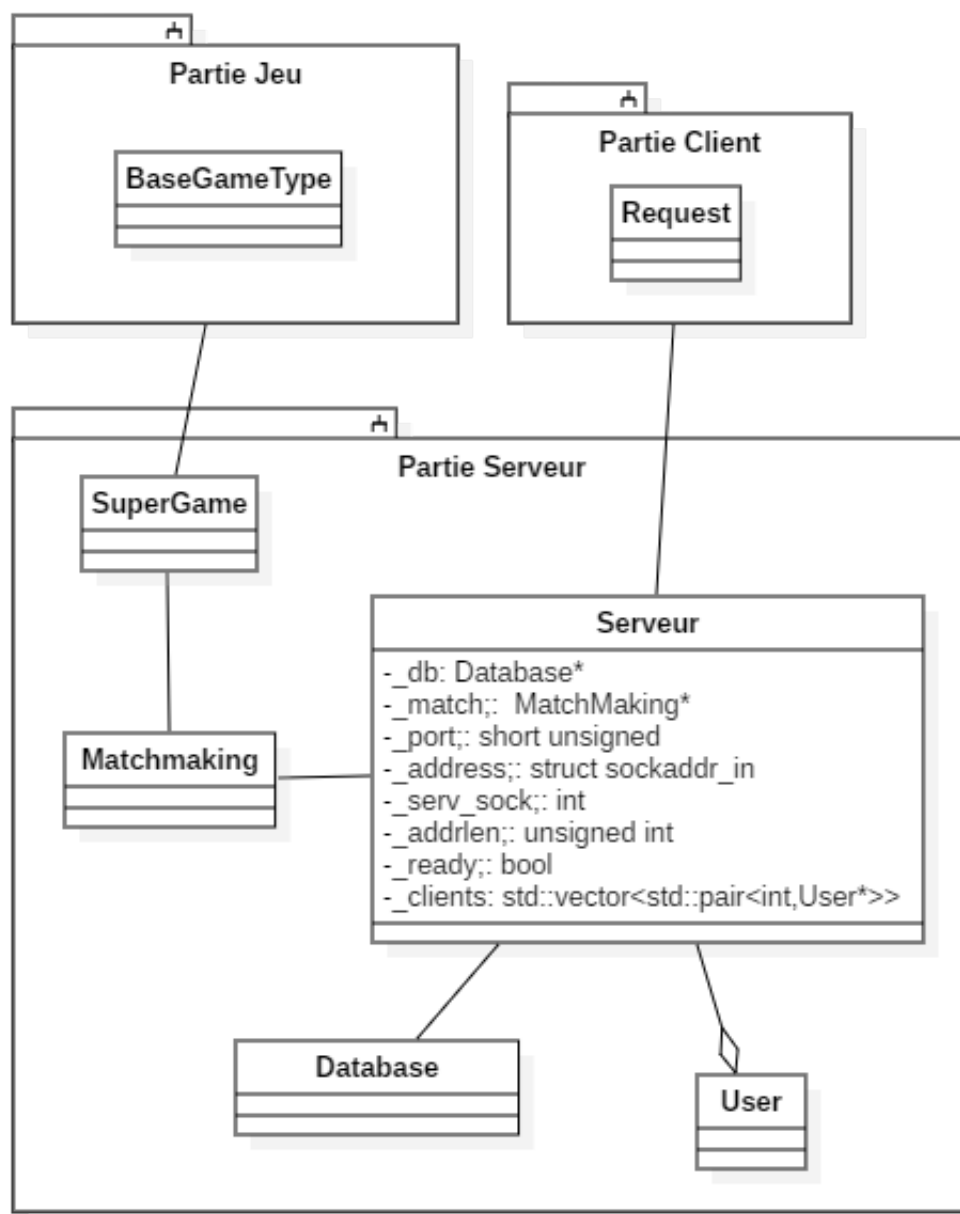


FIGURE 7.11 – Diagramme de classe du Serveur

Le serveur est une passerelle reliant plusieurs clients via un matchmaking. Lui seul a connaissance de la database et par conséquent il est le seul à avoir accès aux données enregistrées. Un système de jeu en parallèle dans le client et le serveur a été mis en place de sorte que le serveur peut détecter toute forme de triche venant du client.

7.2.2 Client

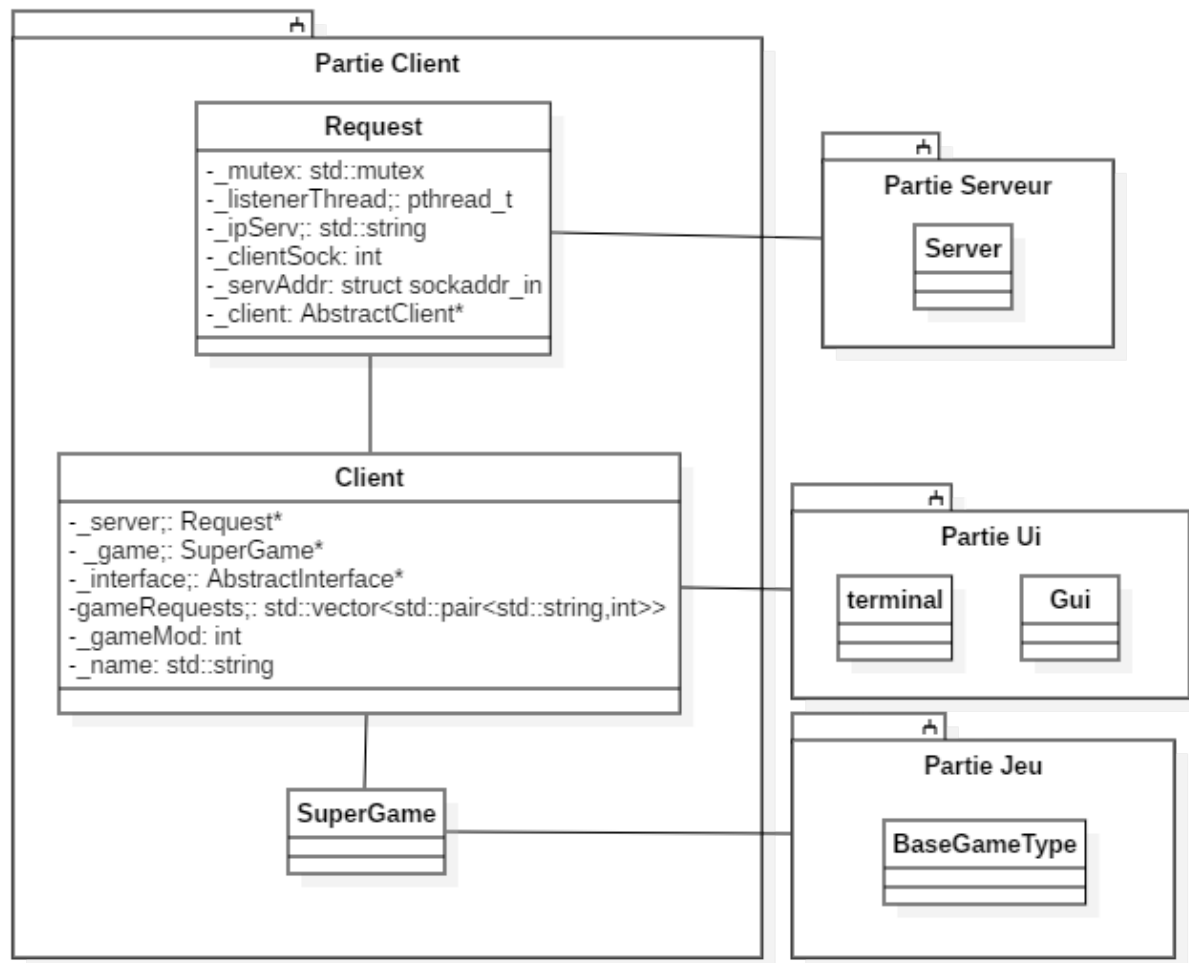


FIGURE 7.12 – Diagramme de classe du Client

Le Client utilise une classe request pour communiquer avec le serveur. Le client possède un jeu en attribut ainsi qu'un lien vers l'interface qu'elles soit terminale ou graphique.

Chapitre 8

Déroulement du Programme

8.1 Diagrammes d'activité

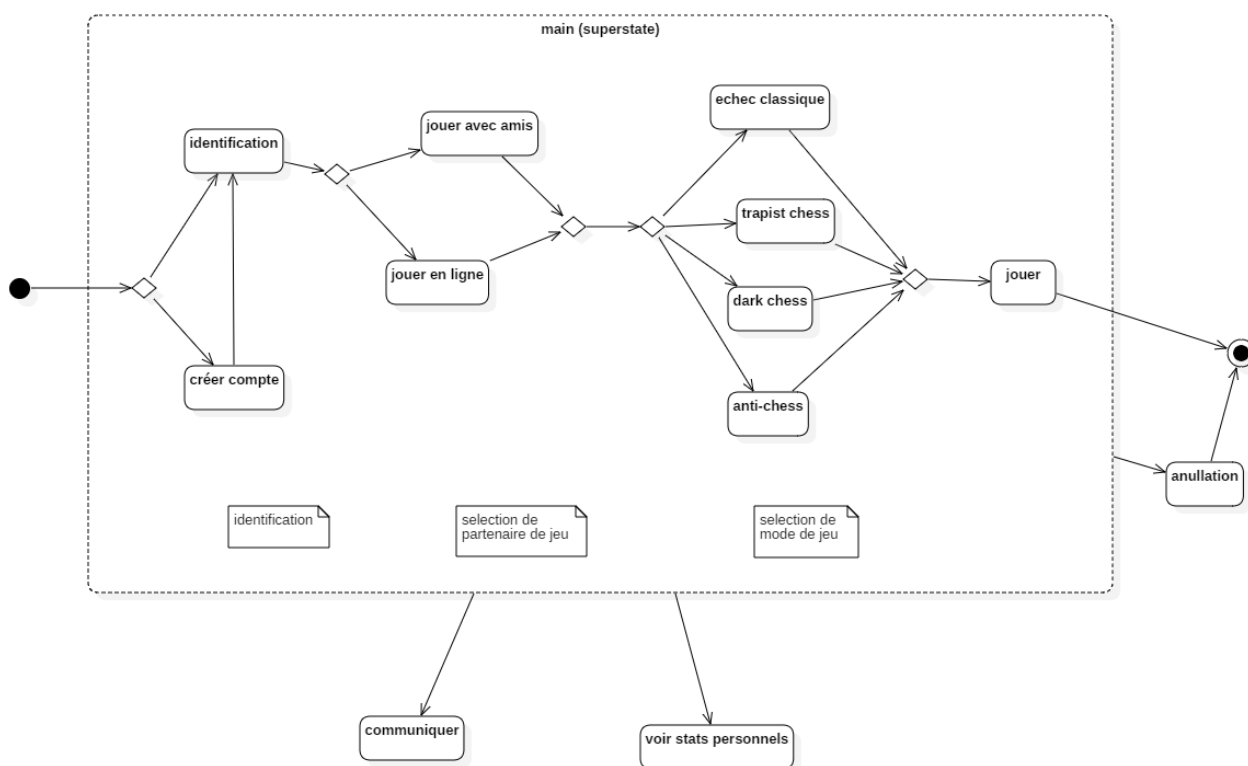


FIGURE 8.1 – Diagramme d'activités du programme global

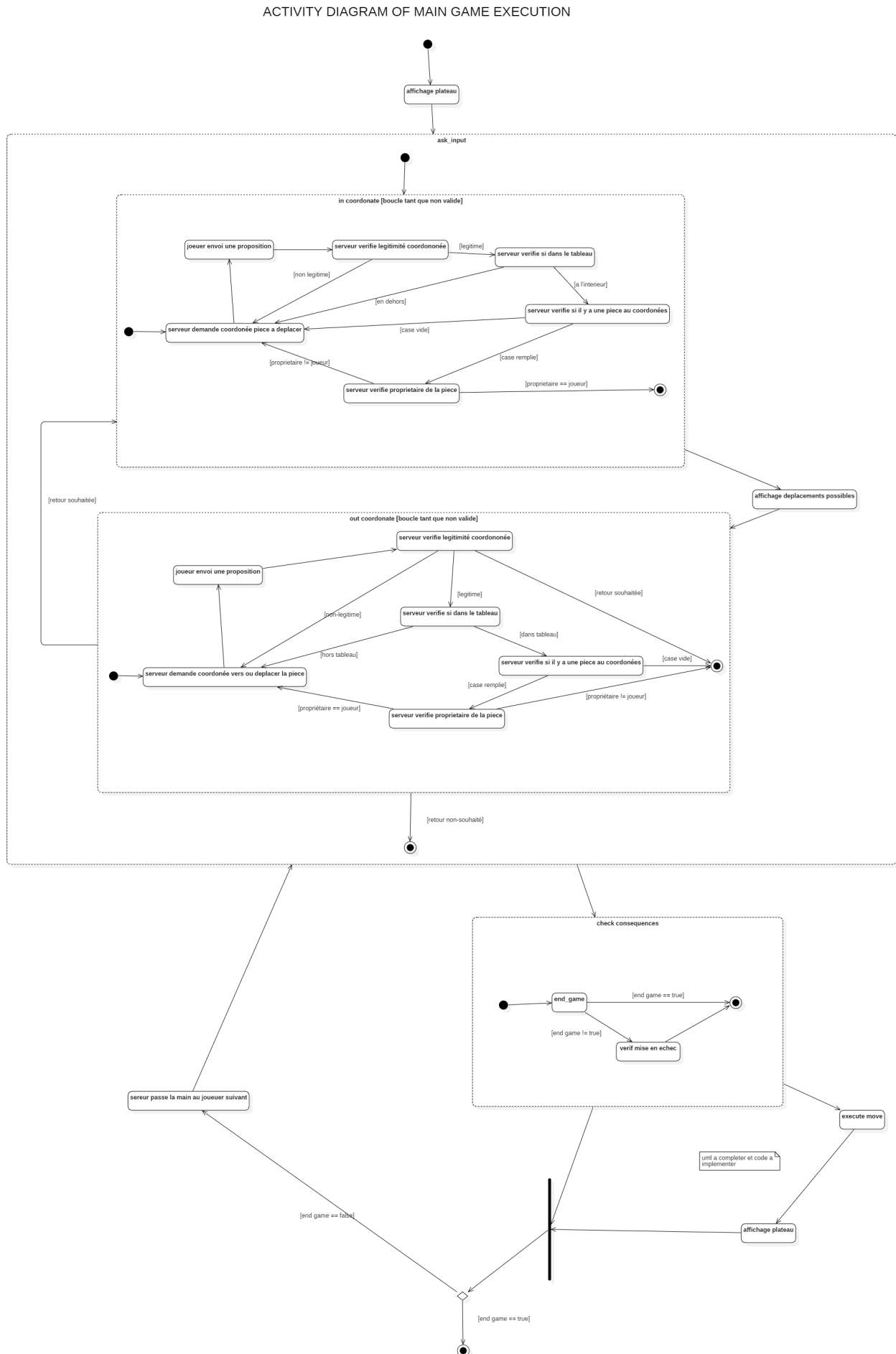


FIGURE 8.2 – Diagramme d'activités de la boucle principale du jeu

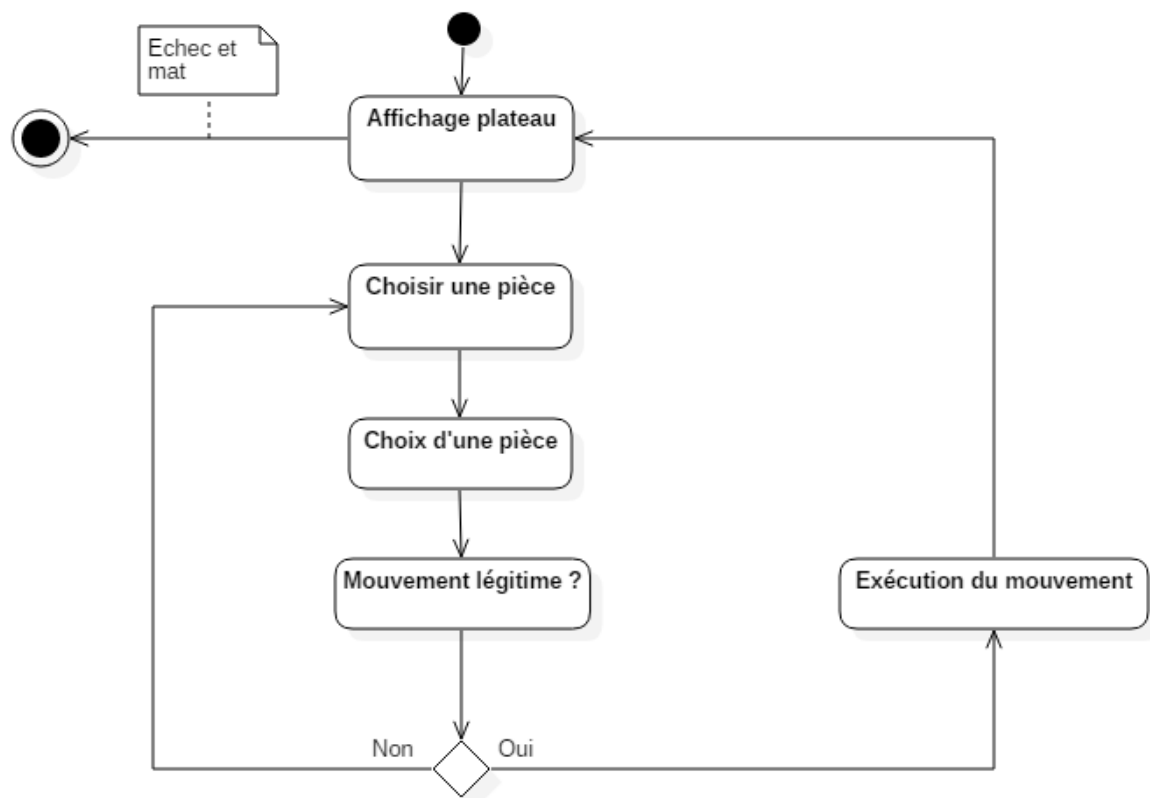


FIGURE 8.3 – Déroulement d'une partie

8.2 Diagramme de Séquence

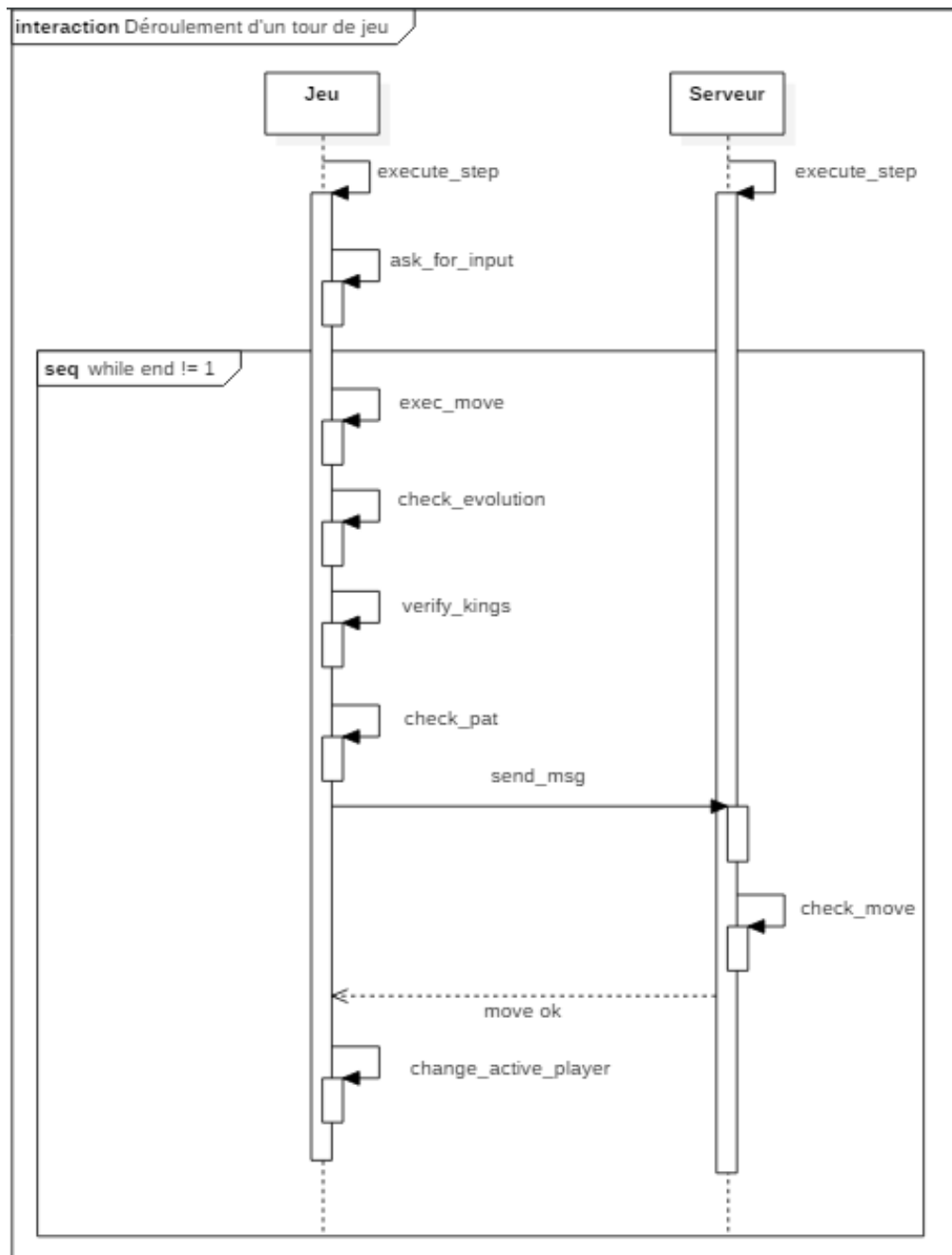


FIGURE 8.4 – Diagramme de séquence du tour de jeu

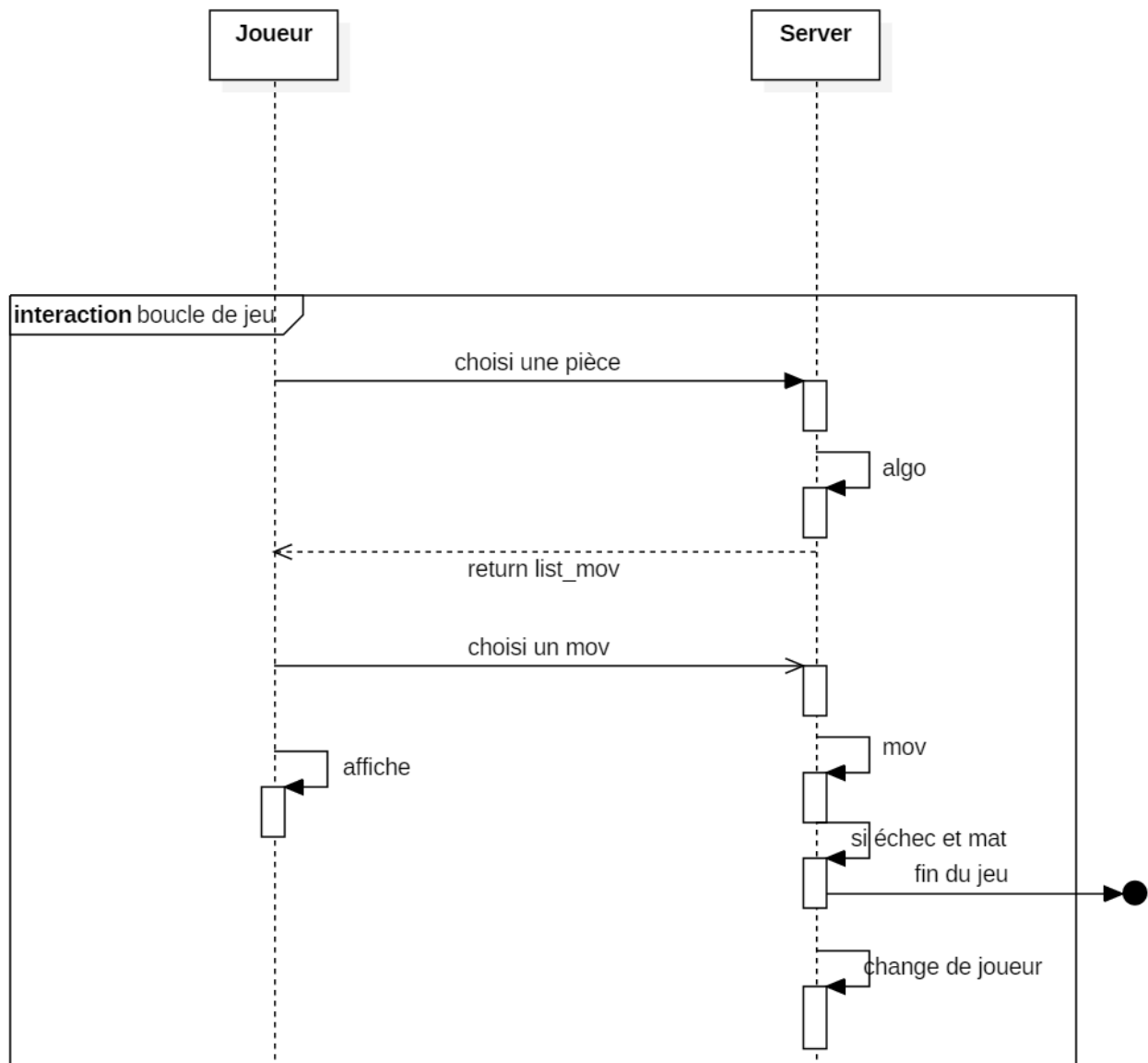


FIGURE 8.5 – Diagramme de séquence de la boucle de jeu

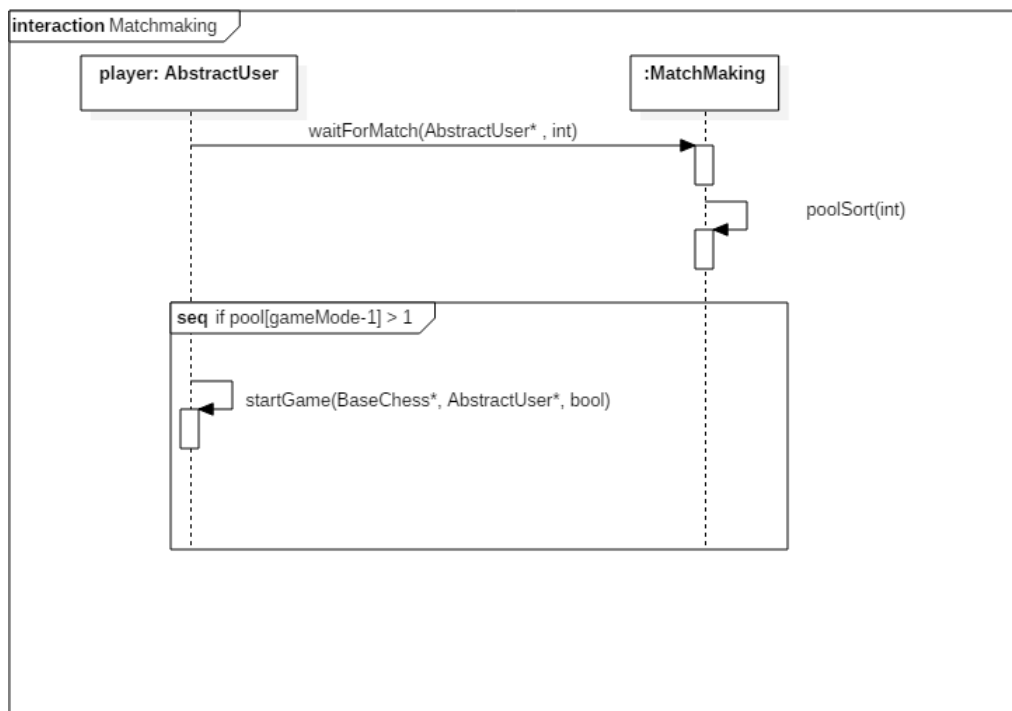


FIGURE 8.6 – Diagramme de séquence du matchmaking

Table des figures

1.1	logo projet groupe	2
2.1	Classic chess	4
2.2	Trappist chess	5
2.3	Dark chess	6
3.1	Use case global d'utilisateur	7
3.2	Use case connexion	8
3.3	Use Case avant une partie	9
3.4	Use case en partie	11
3.5	Use case class	12
4.1	Use case de la connexion	13
4.2	Use case du lancement d'une partie	14
4.3	Use case de la gestion d'une partie	15
7.1	Diagramme de classes Partie Jeu Complet	19
7.2	Diagramme de classes Partie Jeu Global	20
7.3	Diagramme de classes Partie Jeu GameRules	21
7.4	Diagramme de classes Partie Jeu GameTypes	22
7.5	Diagramme de classes Partie Jeu GamePieces	23
7.6	Diagramme de classes Partie Jeu GamePlayers	24
7.7	Diagramme de classes Partie Jeu GameAffichages	25
7.8	Diagramme de classes Partie Jeu GamePosis	26
7.9	Diagramme de classes Partie Jeu GameDico	27
7.10	Diagramme de classes Partie Jeu GamePlateau	28
7.11	Diagramme de classe du Serveur	29
7.12	Diagramme de classe du Client	30
8.1	Diagramme d'activités du programme global	31
8.2	Diagramme d'activités de la boucle principale du jeu	32

8.3	Déroulement d'une partie	33
8.4	Diagramme de séquence du tour de jeu	34
8.5	Diagramme de séquence de la boucle de jeu	35
8.6	Diagramme de séquence du matchmaking	36
B.1	interface identification	41
B.2	interface inscription	41
B.3	interface menu	42
B.4	interface sélection amis	42
B.5	interface mode de jeu	42
B.6	interface liste amis	42
B.7	interface classement	42

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 [quasi 100%] :
<https://stackoverflow.com/questions/1120140/how-can-i-read-and-parse-csv-files-in-c>
- MyOstream [quasi 100%] :
<https://stackoverflow.com/questions/14155364/redirect-the-copy-of-stdout-to-the-file>
- Parcourir des dossiers [quasi 100%] :
<http://www.martinbroadhurst.com/list-the-files-in-a-directory-in-c.html>
- MyException [peu] :
<https://stackoverflow.com/questions/134569/c-exception-throwing-stdstring>
- split_string [quasi 100%] :
<https://stackoverflow.com/questions/14265581/parse-split-a-string-in-c-using-string-delimiter-standard-c>

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

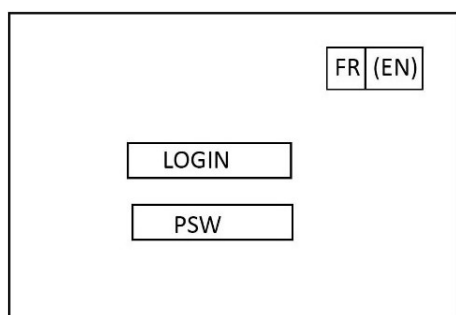


FIGURE B.1 – interface identification

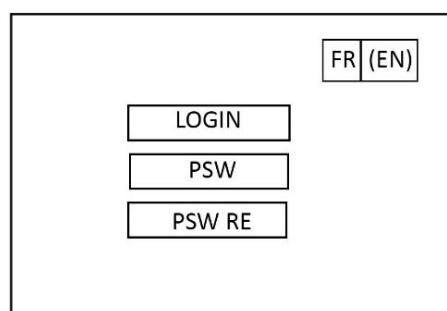


FIGURE B.2 – interface inscription

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

FIGURE B.3 – interface menu

mon compte		leaderboards
sélectionner votre ami		
amis1 amis2 amis3 ...		retour

FIGURE B.4 – interface sélection amis

MON COMPTE		LEADERBOARDS
CHES 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

Annexe C

Particularités du projet

C.1 Particularités Jeu

Un historique de la partie se génère dans un fichier texte dans les dossiers du jeu, effectué par un détournement de std : :cout via un Osteam personnel.

Une traduction du jeu est présente et la possibilité d'autre langues est très facile, en effet toutes les messages du jeu sont stocké dans un fichier csv avec une colonne par langue. pour ajouter un langue il suffit d'ajouter et de remplir une nouvelle colonne.

Le jeu est fonctionnel indépendamment du client/serveur en effet il est possible de le lancer localement.

Un bot recevant des ordres d'un fichier csv a été crée pour permettre des tests facile de différentes configuration nécessitant parfois bcp de mouvements, autrement encodé a la main.

Une batterie de tests ainsi qu'une classe de test ont été développé pour tester bcp de fonctionnalités du jeu.

C.2 Particularités Gui

Plusieurs pools d'images possibles a la sélection lors de la sélection de partie. Tradition Trump continuée, en effet une pool lui est dédiée ou il est représenté manipulant de pièces arborant des skins de mur face a un adversaire utilisant des pièces représentants des mexicains.

Les pubs insolites crée sur base essentiellement de pubs existante modifié en y ajoutant des têtes de participants du projet aussi bien étudiants qu'assistant. Alors que la censure de nudité fût une évidence pour le créateur de ces images détournées la censure d'imagination fût hélas présente. Il fût strictement demandé de ne pas diffusé ces pubs, car étant parfois trop bien réalisés frôlant le réalisme ou bien trop dégradante pour la personne.

C.3 Particularités Serveur

Le Serveur implémente le prémove, c.a.d la possibilité de jouer des tours d'avances qui seront appliqué au lieu de demander un mouvement lors du tour du joueur, si le prémove se trouve être invalidé par le mouvement adverse, un nouveau mouvement est redemandé.

Annexe D

Remerciements

Merci au professeurs nous ayant donné ce projet. Le thème, malgré qu'il soit classique, fût apprécié par le groupe. Ce projet nous a permis de nous confronter a pas mal de soucis nous préparant bien pour notre avenir d'informaticien. L'aide et le soutien apporté au groupe fût grandement apprécié.

Merci De Stefani Jacopo // Merci Merckx Keeno // Merci Gérard François

