UNIVERSITÉ LIBRE DE BRUXELLES

INFO-F-209 : Software Requirements Document

Quoridor

Ayoub Afif (524527), Victor Burton (520439), Romain Delanghe (465497),
Safouan Ehlalouch (514145), Hamza Hajjaj (461105), Stanislas Mondesir (526369),
Lionel Piraux (516308), Aymane Taifour (507877), Safouan Tassi (520501)

13 décembre 2021

Table des matières

1 Introduction

1.1 But

Quoridor est un jeu de plateau qui se joue à 2 ou 4 joueurs en ligne. Le but est de se frayer un chemin jusqu'à la ligne adverse le plus rapidement possible.

Les règles sont assez simples : chaque joueur dispose d'un seul pion et de 10 murs. En début de partie, chaque pion est positionné au centre de sa première ligne, avec donc 4 cases de chaque côté (plateau 9x9). Les joueurs jouent à tour de rôle et ont le choix entre bouger leur pion ou poser un mur. On ne peut pas enfermer un pion entre des murs. Il doit toujours lui rester un moyen d'atteindre la ligne opposée à la sienne. Le pion lui ne peut bouger que dans 4 directions (nord, sud, est, ouest.). Si un pion se trouve sur une case adjacente à un pion adverse, il peut passer au-dessus et aller sur la case derrière lui. Si un mur se trouve derrière le pion adverse, alors le pion peut bifurquer à droite ou à gauche du pion adverse (mouvement diagonal). La partie se termine si un pion arrive sur la ligne opposée à sa ligne de départ.

Deux modes de jeu alternatifs sont ajoutés :

- Pouvoir jouer un tournoi entre amis (les parties se jouent simultanément en parallèle).
- Deux items sont pré-disposés sur le plateau et lorsqu'un pion arrive sur une case qui comprend un item, il obtient une capacité spéciale en fonction de l'item (soit traverser un mur soit supprimer un mur) qu'il peut utiliser lorsqu'il le souhaite dans la partie à une seule reprise. Ce sont des items consommable.

Au lancement du jeu, le joueur a le choix entre lancer le jeu en terminal ou en GUI. La seule différence sera au niveau de l'affichage.

Chaque joueur devra se créer un compte pour pouvoir jouer au jeu. Une fois connecté, le joueur peut :

- Rejoindre une partie
- Créer une partie
- Chatter avec ses amis
- Gérer sa liste d'amis
- Accéder au classement

Ce document est destiné à une audience publique, il est conseillé à toute personne souhaitant jouer à Quoridor.

1.2 Glossaire

GUI : Diminutif de "Graphical User Interface", désigne une interface graphique. Il s'agit de la manière selon laquelle un logiciel est présenté à un utilisateur sur un écran.

Serveur : Le terme serveur désigne le rôle joué par un appareil matériel destiné à offrir des services à des clients en réseau Internet ou intranet. La taille du support physique d'un serveur varie d'un simple boîtier à une ferme de calcul, selon le nombre d'utilisateurs susceptibles de le solliciter simultanément.

Client : Le mot client est employé dans le contexte du modèle client/serveur. C'est un logiciel installé sur le poste de travail qui permet d'accéder à un serveur du même type.

Base de données : Une base de données est un ensemble structuré et organisé de données qui représente un système d'informations sélectionnées de telle sorte qu'elles puissent être consultées par des utilisateurs ou par des programmes.

Item : Un item est un objet qui peut être collecté par un joueur. Les items sont généralement utilisés comme consommables avec des effets positifs ou négatifs.

Mur : Ce sont des obstacles pouvant être placés sur le plateau, ayant pour but de causer des difficultés aux joueurs adverses lors de la partie. Ceux-ci ne peuvent en aucun cas être traversés et donc obligent le joueur à dévier sa trajectoire.

Terminal : Le terminal est un point d'accès de communication entre l'utilisateur, un ordinateur central ou un réseau d'ordinateurs.

1.3 Historique

Version	Nom	Modification	Date
0.1	Groupe	Conception du use case du menu principal en groupe et choix brainstorming mode de jeu.	17/11/2021
0.2	Stanislas	Première esquisse du diagramme de classe du jeu.	18/11/2021
0.3	Safouan Eh et Hamza	Révision du use case du menu principal du jeu, mise au propre et conception du use case de la partie sociale.	19/11/2021
0.4	Safouan Eh et Hamza	Ajout du use case Ami qui gère la partie social.	20/11/2021
0.5	Safouan Eh et Hamza	Ajout de l'explication du Use case Ami	21/11/2021
0.6	Stanislas	Ajout d'un diagramme de séquence de la boucle d'interaction principale.	21/11/2021
0.7	Ayoub, Aymane et Safouan tsi	Ajout du diagramme de use case de la partie lancée avec explications	21/11/2021
0.8	Ayoub	Première version Latex	21/11/2021
0.9	Hamza	Introduction SRD.	22/11/2021
0.10	Stanislas	Ajout du diagramme d'activité pour la connexion	22/11/2021
0.11	Stanislas	Description du diagramme de classe du jeu, du diagramme de séquence et du diagramme de connexion	22/11/2021
0.12	Stanislas	Première version de l'annexe détaillant la sauvegarde de fichier (utilisateur, log, partie sauvée)	22/11/2021
0.13	Ayoub et Aymane	Correction du SRD et mise en page Latex	23/11/2021
0.14	Aymane	Explication du use case menu.	23/11/2021
0.15	Stanislas, Safouan Tsi et Ayoub	diagramme de classe du jeu	26/11/2021
0.16	Hamza et Safouan Eh	Besoins utilisateur fonctionnels et structure de la table des matières	26/11/2021
0.17	Romain et Lionel	Besoins systèmes fonctionnels.	26/11/2021
0.18	Aymane	Besoins systèmes non fonctionnels.	29/11/2021
0.19	Ayoub et Stanislas	diagramme de classe du jeu (suite)	29/11/2021
0.20	Safouan Eh, Lionel et Romain	Besoins systèmes fonctionnels (suite).	29/11/2021
0.21	Safouan Tsi et Victor Burton	Besoins utilisateur non fonctionnels.	29/11/2021
0.22	Aymane et Safouan Tsi	Correction SRD.	03/12/2021
0.23	Hamza	Conception du séquence diagramme de partie en cours	03/12/2021
0.24	Stanislas	Ajout connexion diagramme "activity" et client diagramme	04/12/2021
0.25	Hamza	Correction des diagrammes "gestion de mouvements" et "invitation à rejoindre une partie	04/12/2021
0.26	Safouan Eh	Mise en commun des différentes parties pour la partie design et fonctionnement.	08/12/2021
0.27	Safouan Eh	Modification du use case de la gestion des mouvements.	08/12/2021
0.28	Safouan Eh	Correction de l'orthographe.	08/12/2021
0.29	Ayoub	Création de l'historique Latex.	08/12/2021
0.30	Hamza	Ajout de l'explication du séquence diagramme de partie en cours	09/12/2021
0.31	Aymane et Safouan Tassi	Correction du glossaire	09/12/2021
0.32	Safouan Eh et Hamza	Finalisation du SRD	12/12/2021

2 Besoins utilisateur : Fonctionnels

2.1 Menu principal

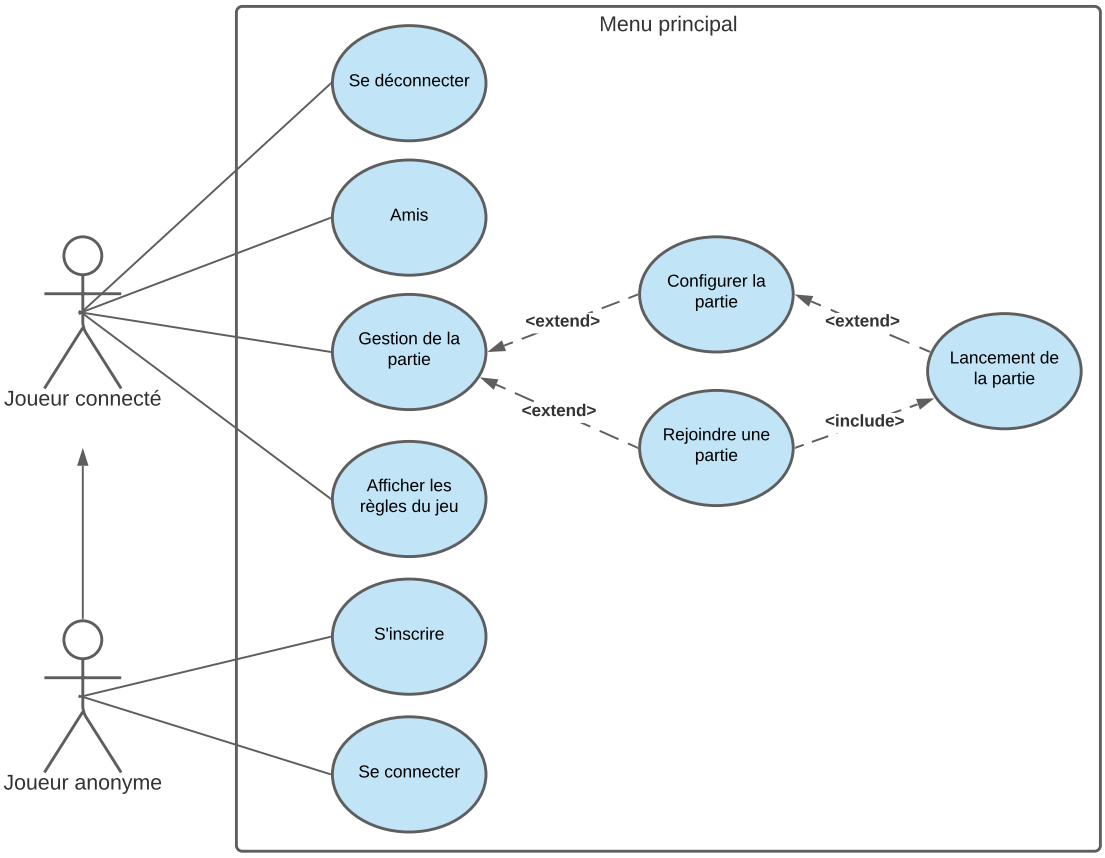

FIGURE 1 – Use case - Menu principal

2.1.1 S'incrire/se connecter

L'utilisateur doit se connecter à son compte à l'aide d'un pseudonyme ainsi que d'un mot de passe. De ce fait, il s'agit d'une pré-condition que le joueur doit remplir afin de bénéficier de toutes les fonctionnalités qu'offre l'application.

Dans le cas où le nouvel utilisateur ne possèderait pas de compte client, il aura seulement la possibilité de s'inscrire. À partir de quoi, il pourra se connecter au jeu à l'aide de son pseudonyme et de son mot de passe.

2.1.2 Gestion de la partie

Pour jouer à Quoridor, le joueur peut gérer une partie à l'aide de deux fonctionnalités :

— Configurer une nouvelle partie : l'utilisateur configure donc une partie avec le choix des options de jeu qui lui sont proposées. Il aura également l'occasion d'inviter un ou plusieurs amis à sa partie. Cependant, ceci n'entraine pas nécessairement le lancement de la partie.
— Rejoindre une partie en cours de jeu : ne permet pas à l'utilisateur de décider du choix d'options lors de la création de partie. Ainsi, il rejoint donc une partie configurée au préalable et en cours de jeu. De ce fait, cette fonctionnalité implique forcément au joueur le lancement de la partie.

2.1.3 Amis

L'application offre une dimension sociale aux joueurs connectés et ainsi ces derniers peuvent être "amis" avec d'autres joueurs. De ce fait, ceci implique que le jeu offre des fonctionnalités spécifiques au sujet de ces amis en questions.

Une section détaillée concernant la liste d'amis est dédiée dans le document (voir table des matières).

2.1.4 Afficher les règles du jeu

Cette fonctionnalité a pour but d'afficher au joueur les règles de jeux de Quoridor.

2.1.5 Déconnexion

La déconnexion permet au joueur de quitter l'espace client lié à son compte. La pré-condition relative à la déconnexion est que le joueur doit être connecté. De plus, la post-condition implique que l'utilisateur ne soit plus connecté et ne dispose donc plus des fonctionnalités dont il dispose lorsque celui-ci est connecté.

2.2 Liste d'amis

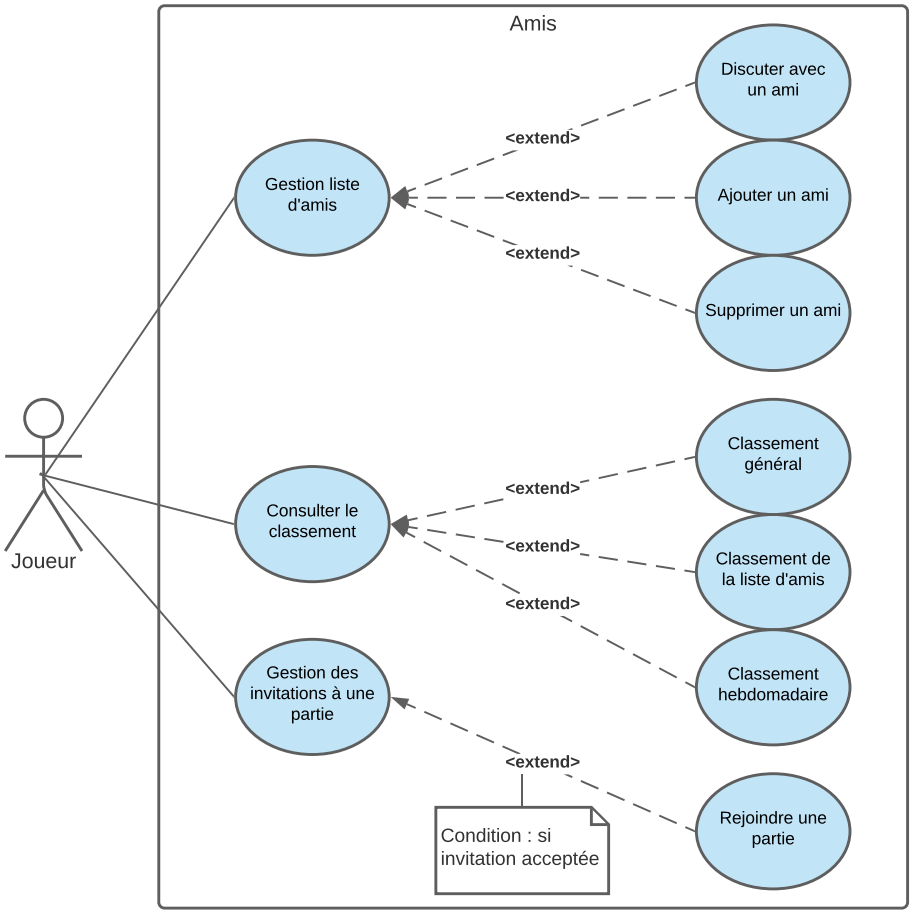

FIGURE 2 – Use case - Amis

2.2.1 Gestion de la liste d'amis

Cette fonctionnalité permet la gestion de sa liste d'amis. La pré-condition est que le joueur doit être connecté à son compte. Trois options en découlent. Le joueur peut premièrement chatter avec un autre ami. Il a ensuite la capacité d'ajouter un joueur, à l'aide de son pseudonyme, à sa liste d'amis. Finalement, il peut aussi supprimer un ami de sa liste d'amis.

2.2.2 Consulter le classement

Nous mettrons en place un classement (avec filtres en options) qui regroupe tous les joueurs inscrits selon leur classement. Il y aura trois filtres différents. La pré-condition est que le joueur doit être connecté à son compte. Le premier consiste à montrer la totalité du classement, il affichera tous les joueurs sans exceptions (qu'ils soient amis ou non). Le deuxième permet d'afficher le classement hebdomadaire. Enfin, le dernier permet d'afficher le classement de sa liste d'amis. Lors de chaque fin de partie, le classement est mis à jour.

2.2.3 Gestion des invitations à une partie

Le joueur a la possibilité de gérer l'invitation à une partie. Ceci signifie qu'il peut accepter l'invitation à rejoindre une partie ce qui lui impliquera le lancement d'une partie créée au préalable. De plus, ces invitations ne peuvent provenir que de joueurs présents dans sa liste d'amis. Cette partie ne permet pas d'inviter des amis à jouer mais seulement à recevoir des invitations. L'envoi d'invitations se fait dans la configuration de partie (décrit ci-dessus).

2.3 Partie en cours

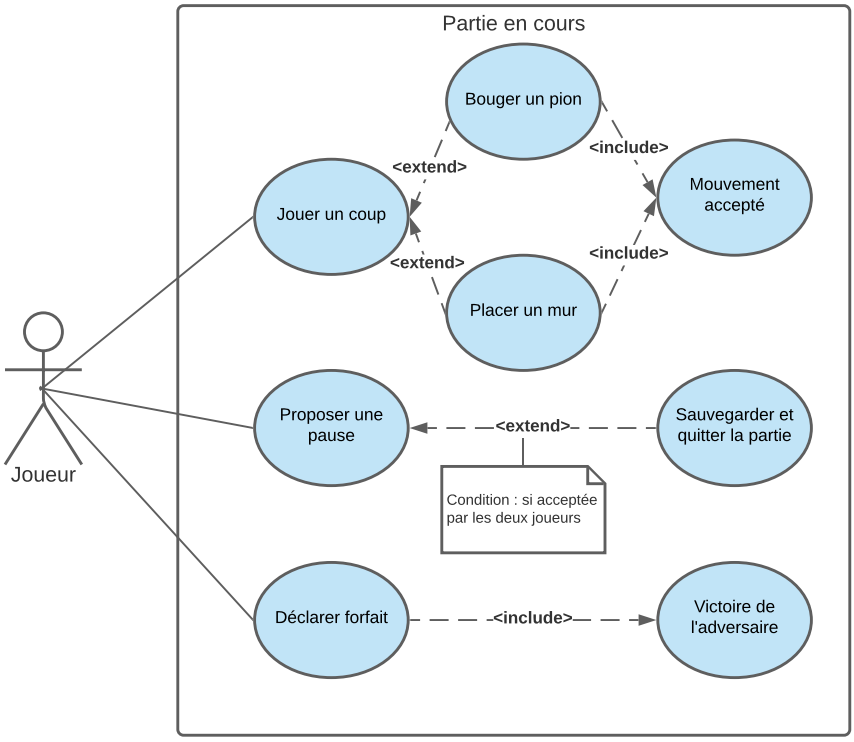

FIGURE 3 – Use case - Partie lancée

2.3.1 Jouer un coup

Action principale lors d'une partie : implique à l'utilisateur de bouger un pion ou de placer un mur. Cette action est refusée, et n'est donc pas effectuée, si elle ne respecte pas les règles du jeu. Après avoir jouer un coup, le joueur doit attendre que l'adversaire ait également jouer un coup.

2.3.2 Proposer une pause à l'adversaire

Cette fonctionnalité permet de mettre en pause la partie en cours de jeu. Dans ce cas de figure, aucun gagnant n'est déclaré et la partie est sauvegardée et quittée.Elle peut être reprise à n'importe quel moment à partir du menu principal, à condition que les deux joueurs soient connectés et qu'ils l'acceptent tous les deux de la reprendre. Le joueur peut demander une pause seulement lorsque c'est à son tour de jouer le coup.

2.3.3 Déclarer forfait

Cette option permet au joueur de quitter la partie sans demander l'autorisation de son adversaire. Il déclare donc forfait, son adversaire gagne automatiquement la partie et celle-ci est terminée (dans le cas d'une partie à 2 joueurs)

Dans le cas d'une partie à 4 joueurs, celui qui déclare forfait a une pénalité de -1 point dans le classement. Les autres joueurs ne gagnent ni perdent de points.

3 Besoins utilisateur : Non fonctionnels

Pour pouvoir accéder au jeu, le joueur doit pouvoir s'identifier à l'aide d'un pseudonyme et d'un mot de passe sur le serveur. De plus, l'utilisateur doit disposer d'une configuration minimale à savoir :

— Un accès internet
— Un écran permettant l'affichage d'une fenêtre de jeu
— Un clavier ainsi qu'une souris, permettant l'interaction avec le jeu.

En ce qui concerne la disponibilité du jeu, celui-ci sera disponible gratuitement.

4 Besoins système : Fonctionnels

4.1 Connexion

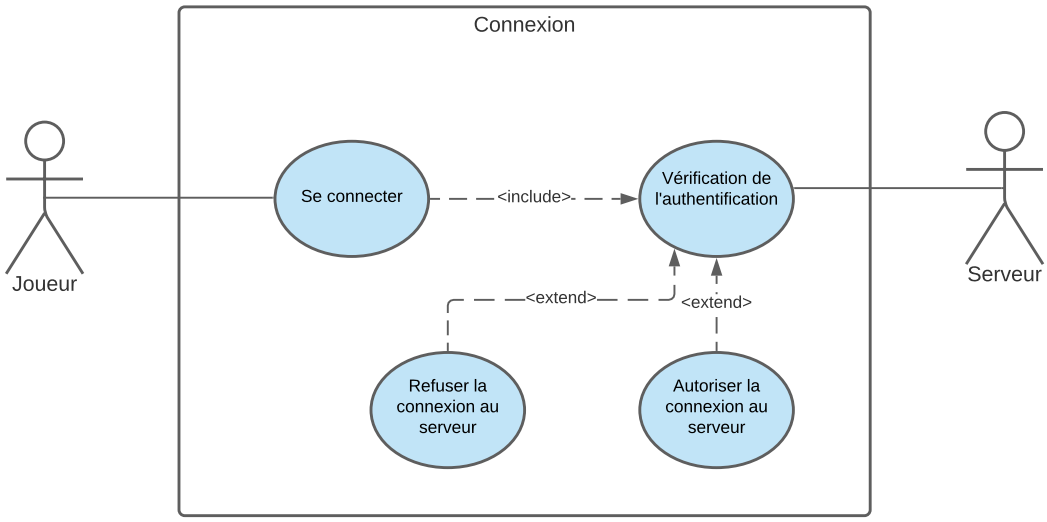

FIGURE 4 – Use case - Connexion à l'application

Lorsqu'un utilisateur décide de se connecter à Quoridor à l'aide de son pseudonyme et son mot de passe, le serveur vérifie son authentification. Si celle-ci est valide, le joueur est connecté et a accès à son compte. Si l'authentification n'est pas valide pour une des deux raisons suivantes, ce serait : soit un compte lié au pseudonyme entré ne figure pas dans la base de données ou soit celui-ci existe mais le mot de passe est incorrect. Dans les deux cas, le serveur refuse l'accès au compte du joueur.

4.2 Gestion des comptes

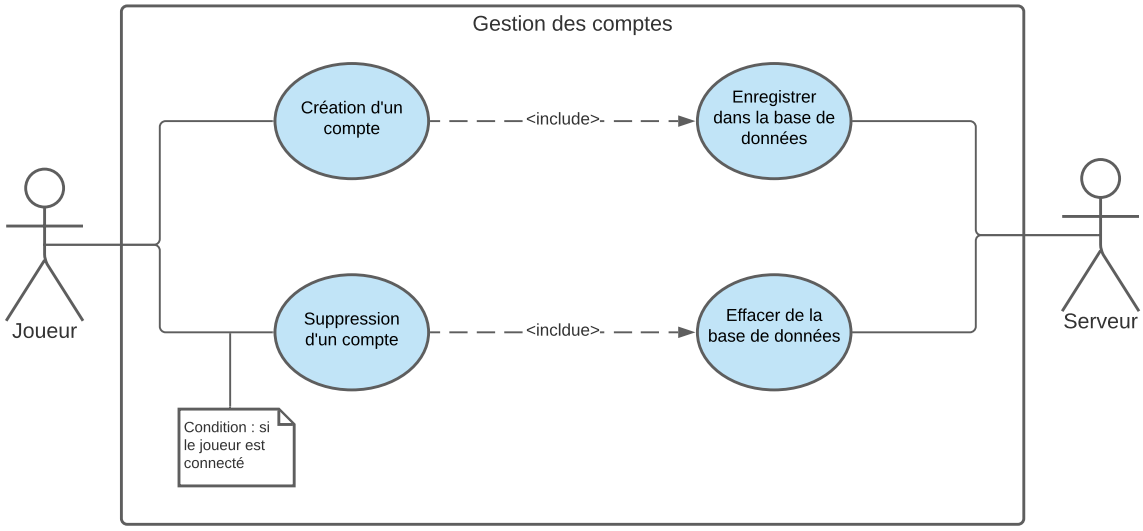

FIGURE 5 – Use case - Gestion des comptes

4.2.1 Création d'un compte

Lorsqu'un utilisateur s'inscrit à l'application, il renseigne un pseudonyme ainsi qu'un mot de passe.Cependant, pour créer un compte le pseudonyme ne doit pas exister dans la base de données. Ces données sont enregistrées dans la base de données du jeu afin qu'il puisse se connecter à son compte à n'importe quel moment après son inscription effectuée.

4.3 Création d'une partie

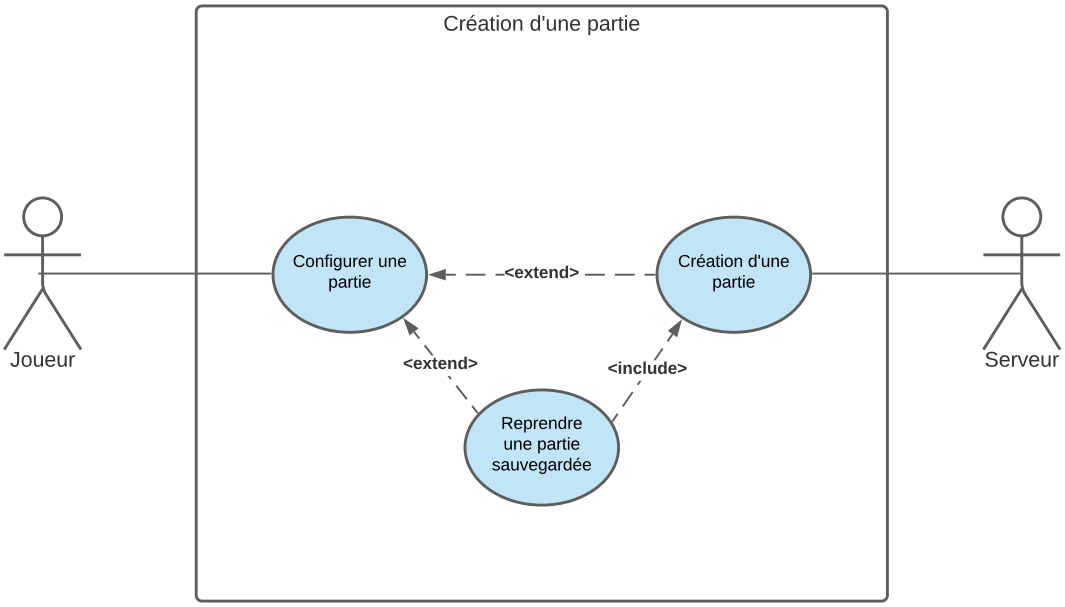

FIGURE 6 – Use case - Création d'une partie

Le serveur a la possibilité de :

— Créer une nouvelle partie : la création d'une partie se fait suite à une requête d'un joueur. Le joueur lui envoie la configuration qu'il souhaite pour sa partie. Le serveur crée une nouvelle partie et connecte le joueur à celle-ci.

— Charger une partie sauvegardée : le serveur charge les données de la partie sauvegardée, puis crée une nouvelle partie avec les données chargées.

— Envoyer les informations d'une sauvegarde : lorsque le joueur se rend dans le menu de création de partie, le serveur vérifie si une sauvegarde existe. Si c'est le cas, il envoie un message, informant qu'une sauvegarde existe, au joueur et lui permet de reprendre la partie.

4.3.1 Invitation à rejoindre une partie

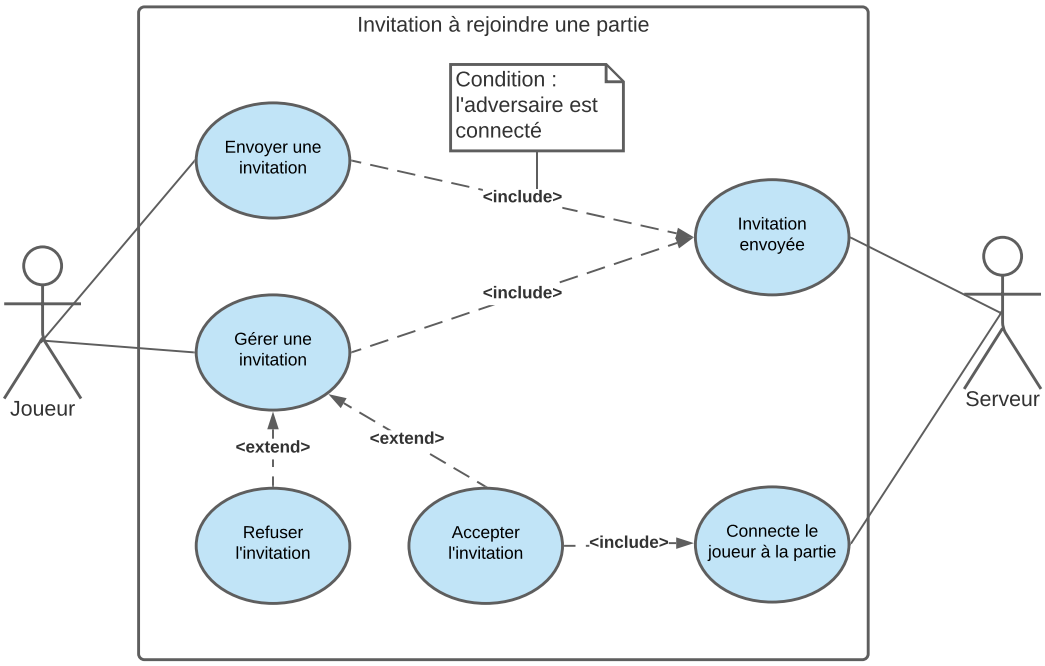

FIGURE 7 – Use case - Invitation à rejoindre une partie

Lorsque que la partie est créée, le joueur peut envoyer des invitations à ses amis. L'invitation est alors envoyée au serveur qui vérifie si le destinataire est connecté. Si oui, le serveur lui envoie le message. Si non, le serveur met le message en attente jusqu'à ce que le destinataire se connecte. L'ami peut décider de refuser l'invitation ou de l'accepter. S'il accepte, le joueur est automatiquement redirigé et connecté à la partie.

4.3.2 Reprise d'une partie sauvegardée

Une sauvegarde est stockée sous forme de fichier texte sur le serveur. Le serveur lit le fichier et en récupère les informations. Il créé ensuite une nouvelle partie avec ces données et connecte le joueur.

4.4 Gestion d'une partie

Le serveur crée, héberge la partie et y connecte les joueurs.

4.4.1 Gestion des mouvements

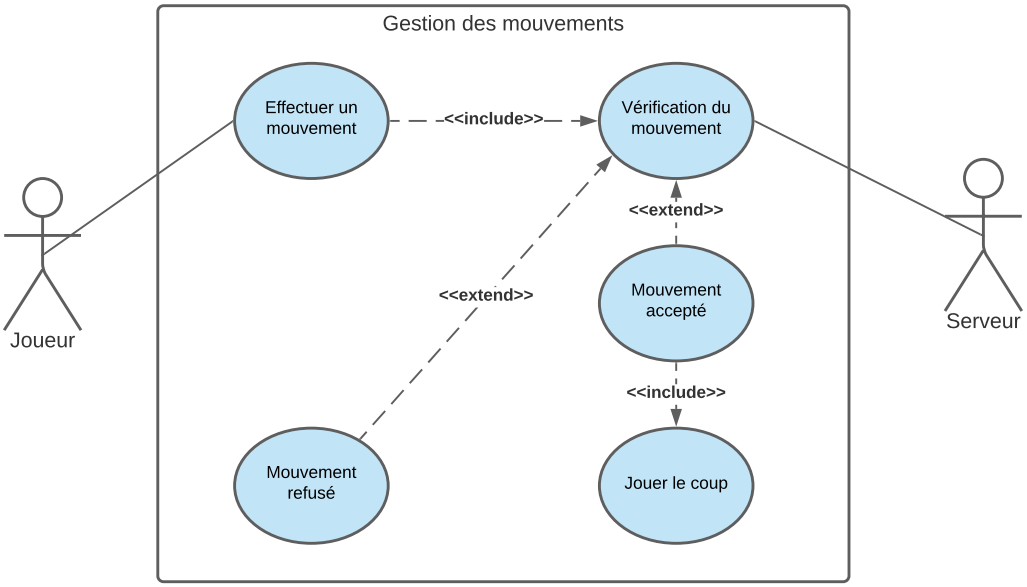

FIGURE 8 – Use case - Gestions des mouvements

Le joueur a le choix entre 2 types de mouvements différents. Il peut, soit déplacer son pion, soit placer un mur. Lorsque le joueur a choisi son coup, l'information est transmise au serveur, qui vérifiera si le coup est valide ou non, suivant les règles du jeu.

4.4.2 Choix du vainqueur

Pendant l'entièreté de la partie, les conditions de victoire sont testées pour chaque joueur. Si les conditions sont validées pour l'un des joueurs, celui-ci sort vainqueur de la partie, et l'autre en sort donc perdant. Leurs scores sont ensuite mis à jour dans la base de données.

4.4.3 Gestion des items

Lors d'une partie avec items, les joueurs ont la possibilité de récupérer des objets qui leur seront bénéfiques. Quand un joueur arrive sur une case avec un objet, l'effet correspondant est appliqué au joueur, et la partie continue. Il existe deux types d'items : un permettant la suppression d'un mur et l'autre permettant la traversée d'un mur. Ces deux items ne sont utilisable qu'une seule fois.

4.4.4 Sauvegarde de la partie

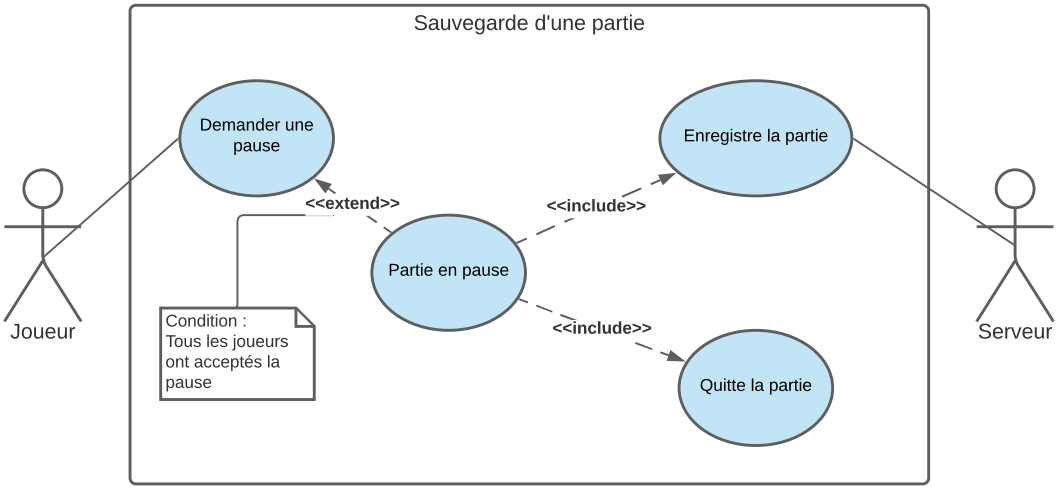

FIGURE 9 – Use case - Sauvegarde de la partie

Lorsque une pause est demandée et acceptée par les deux joueurs, la partie est alors sauvegardée et quittée afin d'être reprise plus tard. Dans ce cas, le serveur enregistre l'état de la partie dans la base de données. Lorsque les joueurs désirent reprendre leur partie, le serveur reprend la sauvegarde et la recharge.

5 Besoins système : Non fonctionnels

5.1 Disponibilité

Le jeu est compatible en version terminal et sur une interface graphique. L'utilisateur a le choix entre ces deux alternatives. Un client en terminal peut jouer une partie, chatter, etc.. avec un client en GUI. Le serveur se comporte de la même manière avec les deux types de clients (GUI ou terminal).

5.2 Sécurité

Le joueur doit pouvoir s'identifier à l'aide d'un pseudonyme et d'un mot de passe sur le serveur. Toutes les données personnelles du joueur seront stockées sur une base de données

5.3 Système d'exploitation

Le jeu est compatible sur une machine ayant comme système d'exploitation GNU/Linux.

6 Design

6.1 Design du système

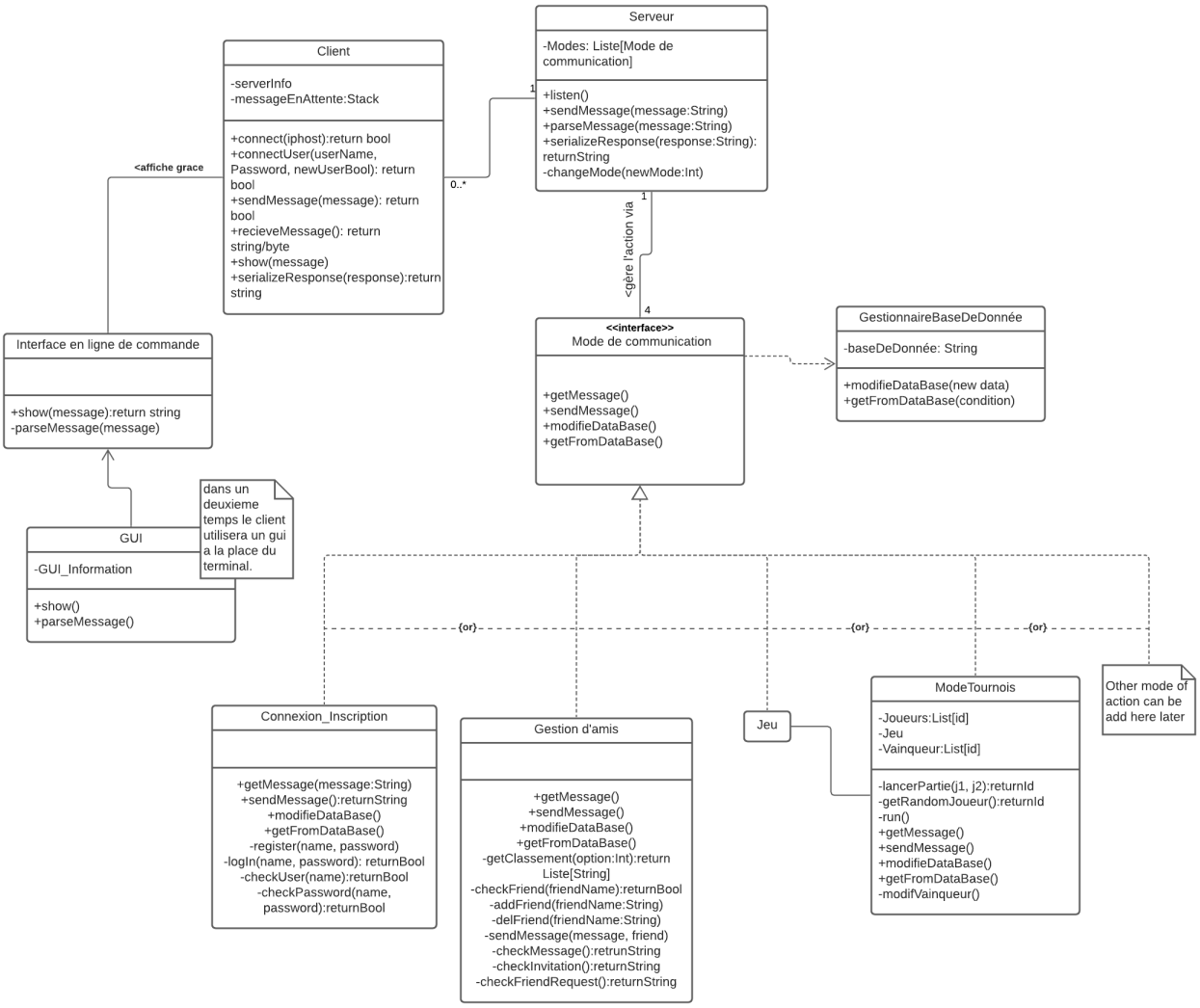

FIGURE 10 – Diagramme de classe

Serveur et *Client* sont les classes principales des 2 programmes du système : serveur et client respectivement.

Le *Client* sert à afficher les informations, transmises par le serveur, grâce à l'interface en ligne de commande ou sa spécialisation, l'interface graphique (GUI). Il a aussi comme rôle d'envoyer des messages au serveur.

La classe *Jeu* a été séparée dans un autre diagramme de classe pour alléger le diagramme de classe du système.

6.1.1 Jeu

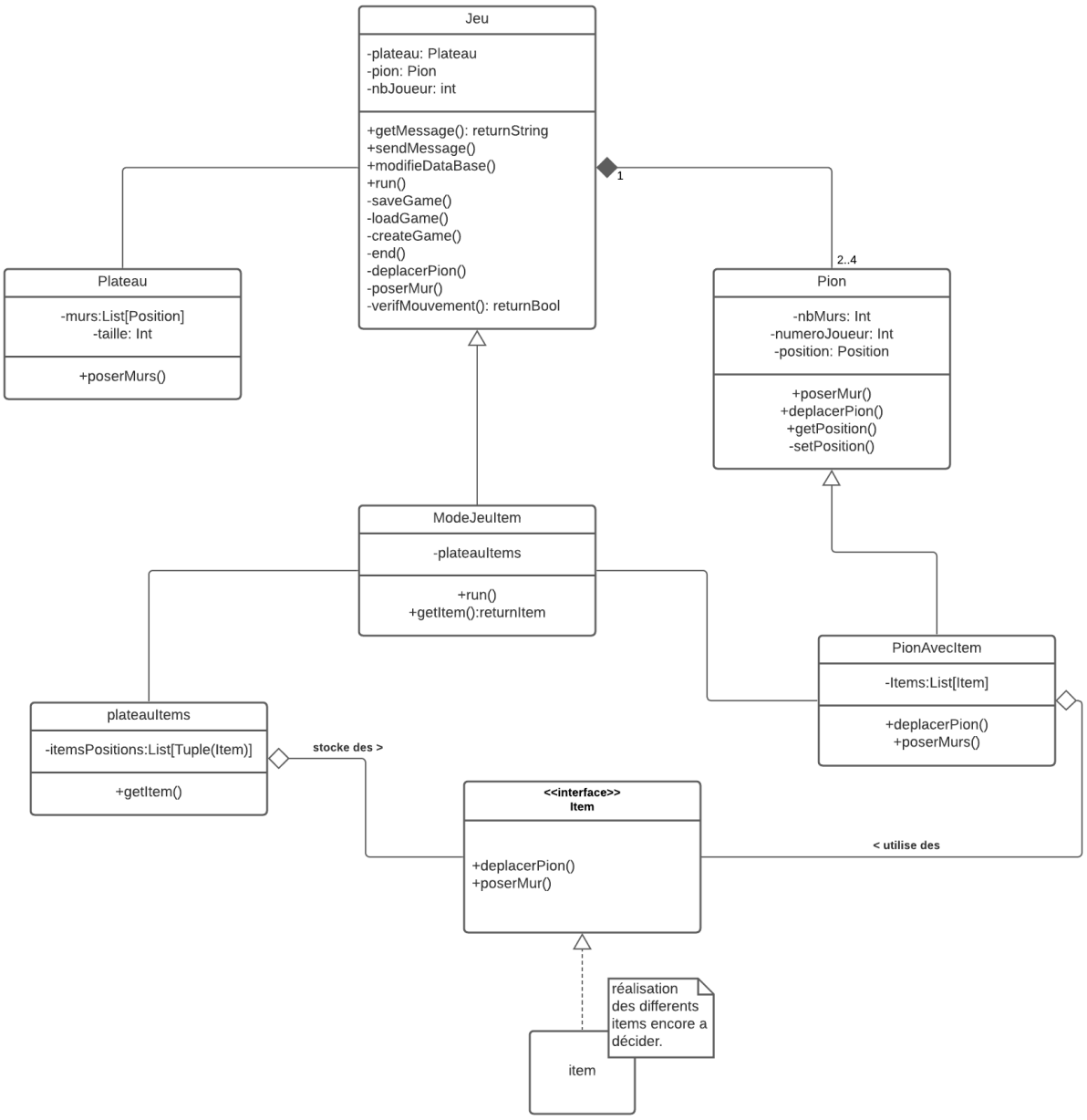

FIGURE 11 – Diagramme de classe

Le jeu a deux types d'objets pour pouvoir fonctionner : un plateau gardant en mémoire la position des murs placés. Ainsi que plusieurs pions (2 ou 4), avec leur position et le nombre de murs restant de chaque joueur.

Le modes de jeux "item" demande de rajouter différents objets gérant les items. Premièrement, les différents items doivent suivre l'implémentation de la classe abstraite *Item*. Ils permettent de modifier les actions de la classe *PionAvecItem* qui est un enfant de la classe *Pion*. Leurs emplacements sur le plateau

sont stockés grâce à *PlateauItem*.

7 Fonctionnement du système

Le serveur et le client communiquent entre eux à travers le réseau via le protocole TCP. Les messages du serveur permettent au client d'afficher les informations à l'utilisateur. Le client encode la réponse et l'envoie au serveur. Celui-ci réagit ensuite en conséquence.
La classe *Serveur* est le cœur des opérations, elle récupère les demandes du client, les traite, et lui renvoie les informations pour faire le prochain choix. La classe *Serveur* possède différents modes d'utilisation représentés par la classe abstraite *Mode*. Celle-ci a deux méthodes principales : celle de recevoir les messages et celle de les envoyer. Le traitement du message va se faire par les instances des classes enfants.

7.1 Point de vue du serveur

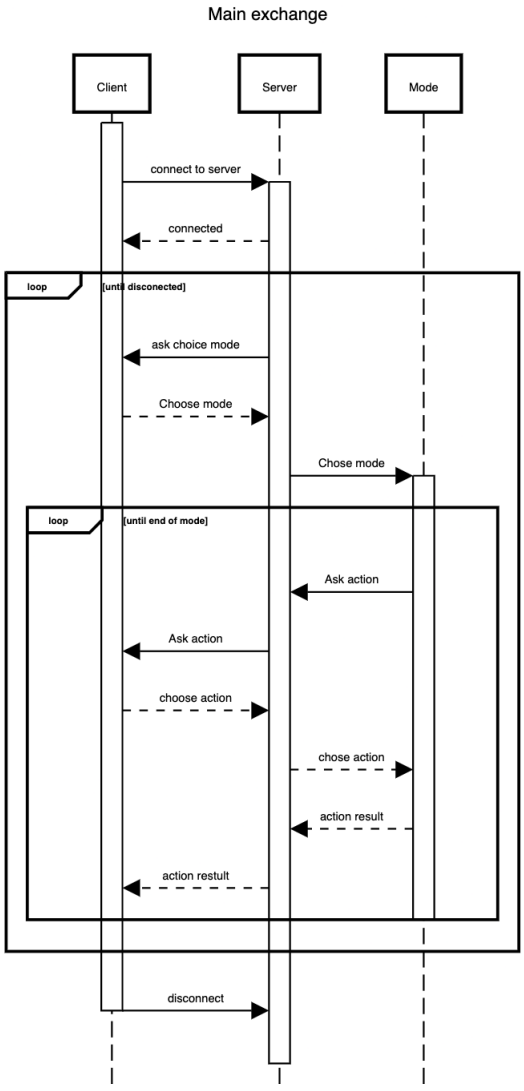

Figure 12 – Diagramme de séquence

Le système contient trois phases principales :

— La connexion.
— Le choix du mode : le serveur a plusieurs modes différents demandant diverses actions possibles. Le client devra en choisir un à utiliser.
— Les actions du mode choisi.

7.1.1 La connexion

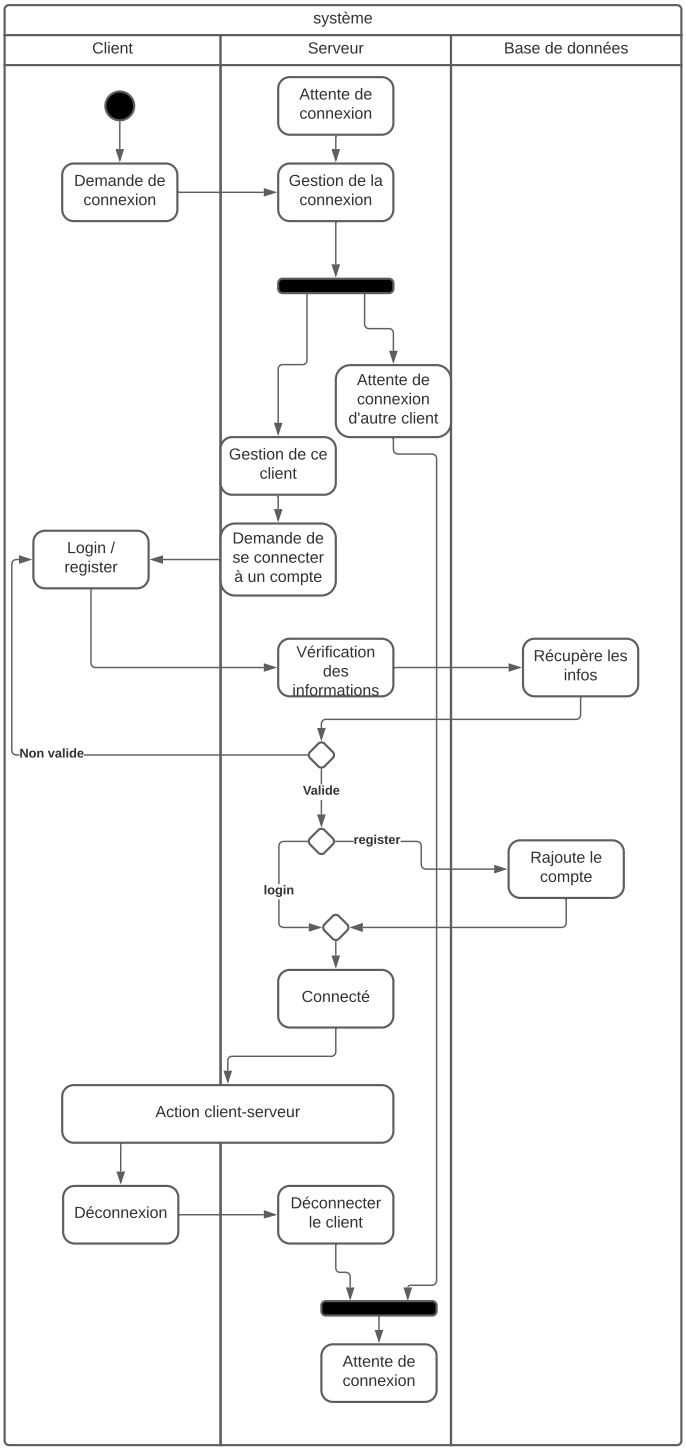

FIGURE 13 – Diagramme d'action

La connexion se fait en deux temps :

1. L'ordinateur du client se connecte au serveur de façon anonyme. Nous lui laissons un laps de temps pour tenter de se connecter. Si le serveur ne répond pas dans ces délais, le programme-client timeout et l'utilisateur pourra réessayer plus tard.
2. Le client se connecte avec son login et mot de passe ou s'enregistre s'il n'a pas de compte. Pendant ce temps, le serveur reste à l'écoute en vue de nouvelles connexions.

7.1.2 Déroulement d'une partie

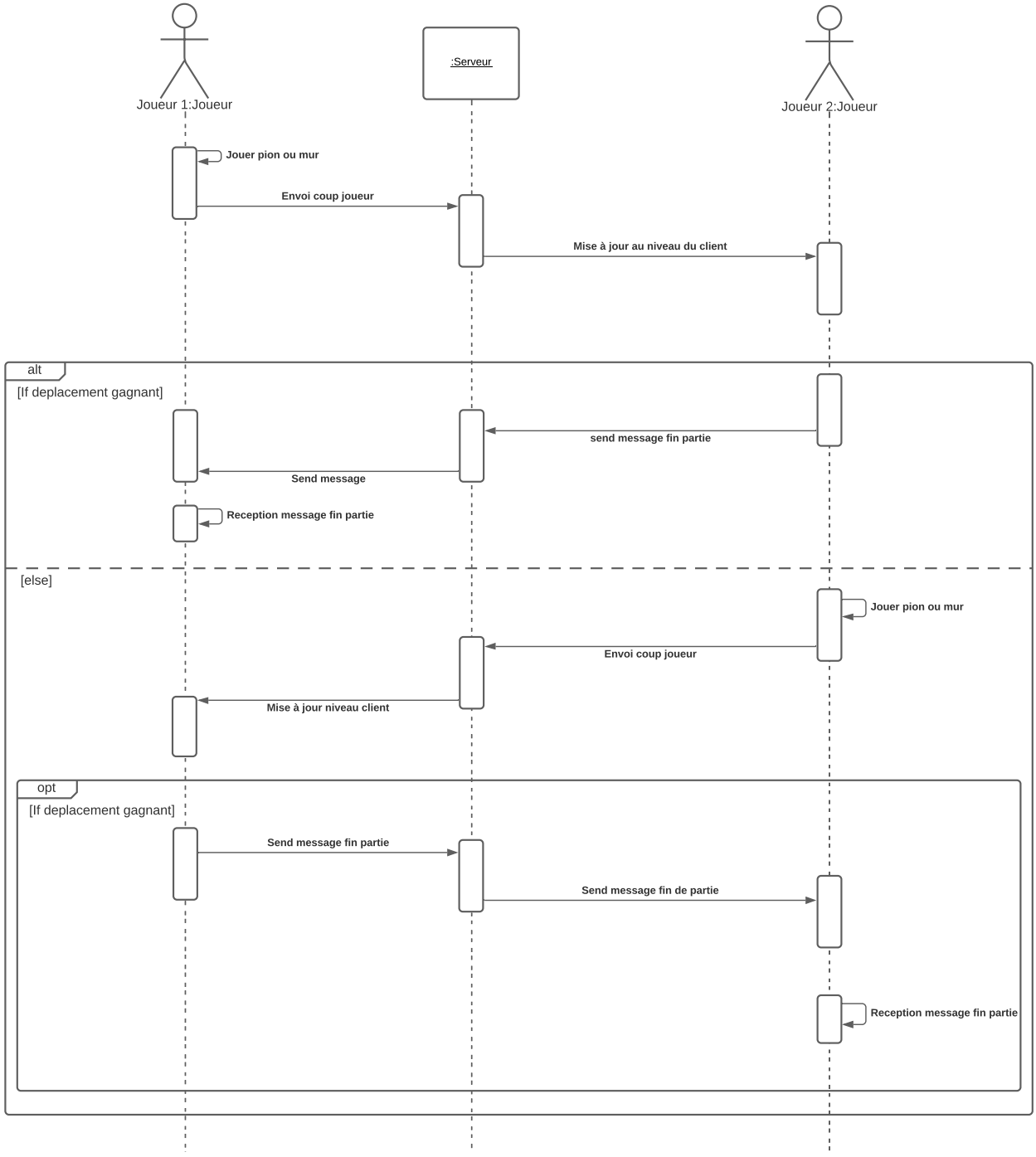

FIGURE 14 – Diagramme de séquence - Déroulement d'une partie

Le joueur 1 commence par jouer son premier coup et le client l'envoie au serveur. Après traitement du coup, celui-ci l'envoie au client du joueur 2 qui se chargera de mettre à jour son plateau. À chaque coup joué,

un test s'effectue pour vérifier si un gagnant est déclaré. Si c'est le cas, un message est envoyé au serveur afin que lui-même prévienne le client du joueur adverse. Ce n'est qu'après vérification que le joueur suivant peut jouer son coup. Nous avons omis la possibilité de proposer une pause du diagramme par souci de clarté. Mais nous avons décidé qu'un joueur ne peut proposer une pause que si c'est son tour de jeu. Ce diagramme est aussi valable pour les parties à 4 joueurs.

7.1.3 La gestion d'amis

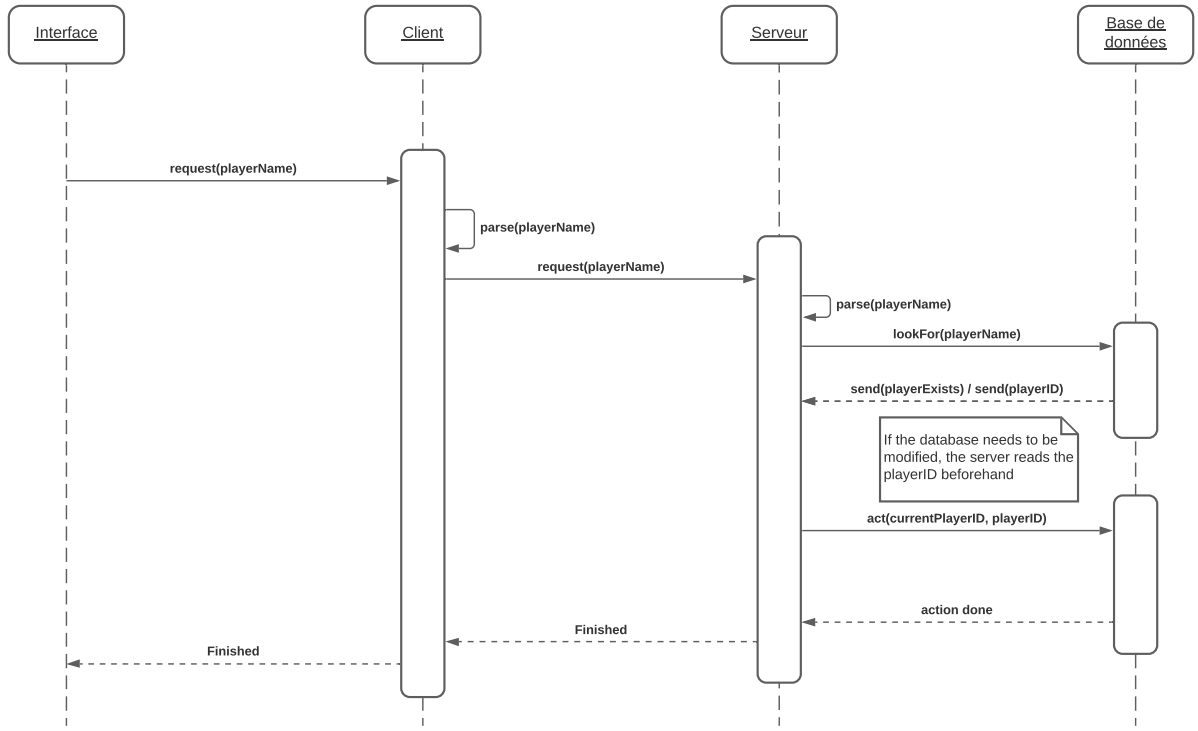

FIGURE 15 – Diagramme de séquence - Gestion d'amis

Tout d'abord, via l'interface, le joueur utilisateur peut rechercher un joueur, ajouter un joueur en ami ou en retirer un. La requête du joueur utilisateur est transmise au client qui va ensuite la transmettre au serveur. Dans le cas de la recherche d'un joueur, le serveur cherche simplement dans la base de données s'il s'y trouve ou non, et renvoie l'information au client, qui l'affiche dans l'interface. Dans le cas de l'ajout et du retrait d'un ami, le serveur cherche le numéro d'identification du joueur à ajouter/retirer dans la base de données, et modifie ensuite celle-ci en fonction de l'action demandée par le joueur.

7.1.4 Chat

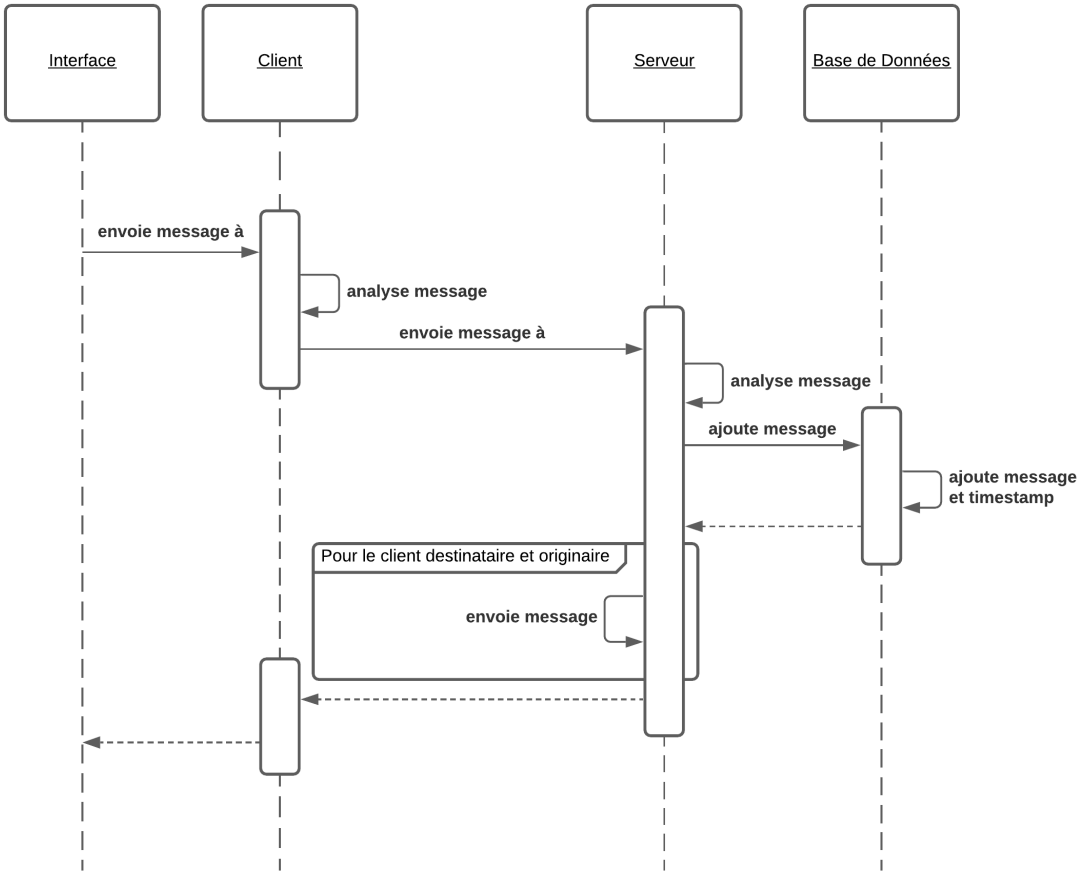

FIGURE 16 – Diagramme de séquence - Chat

Pré-conditions :
— Les deux clients sont connectés au serveur
— Les deux clients sont amis

Tant qu'un joueur écrit un nouveau message, l'interface envoie le message au client qui le traite puis l'envoie à son tour au serveur. Le serveur analyse le message puis demande à la base de données de l'ajouter ainsi que le timestamp auquel il est arrivé. Le serveur transmet ensuite le message au client destinataire et originaire qui traite le message puis l'affiche via l'interface.

7.2 Point de vue du client

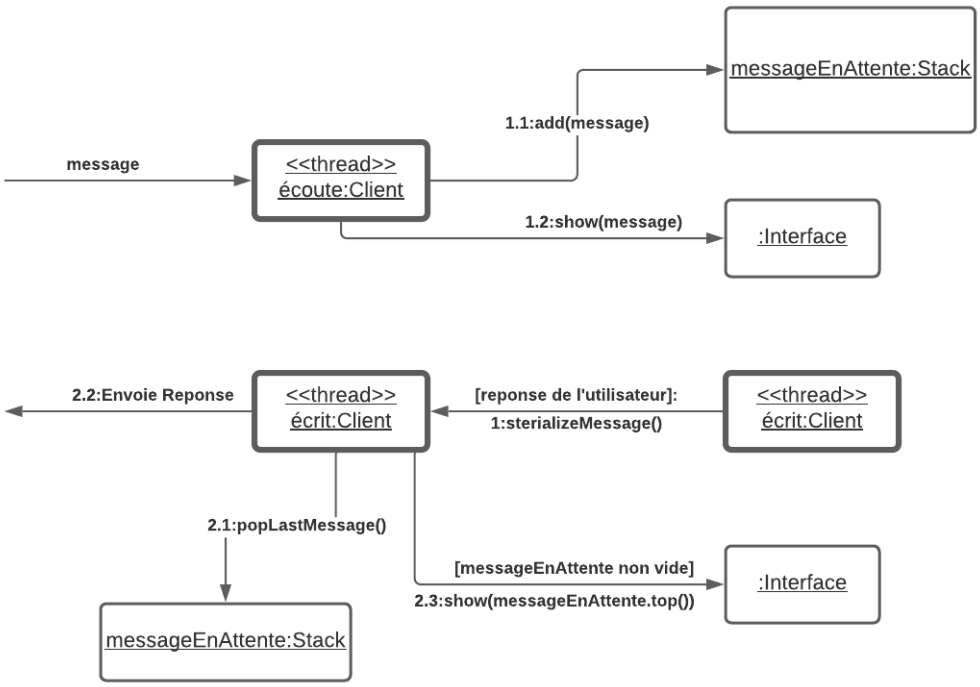

FIGURE 17 – Diagramme de collaboration

Le client est divisé en deux thread de travail, un gérant la réception de message (écoute) et un autre pour l'envoi de message. Le message venant du serveur est décomposé selon ses différents composants (ParseMessage()) afin de décrire à l'interface comment afficher les informations à l'utilisateur. Ensuite, il est stocké dans un stack de messages en attente. Une fois que l'utilisateur a répondu, il est sérialisé selon le type de message étant en haut du stack de messages en attente. Le message en haut est alors enlevé et le nouveau message au top est affiché.

7.2.1 Interface

Terminal
En mode terminal, le message reçu du client va simplement être écrit sur la sortie standard. La réponse est récupérée sur l'entrée standard.

GUI
En mode graphique, le message reçu va être divisé en deux parties distinctes, l'affichage et les choix représentant les réponses possibles de l'utilisateur.

8 Annexes