

Université Libre de Bruxelles

---

Software requirements document

Quoridor

---

Groupe 8

ETIENNE MORIN, AYMAN KITMIR, XHULIO KAPEDANI, KONRAD GRACKI,  
ILIAS AKAJOU, MAMADOU BARRY, MOSAAB OUAKASSE, COSMIN TUDOR

INFO-F209 : Projets d'informatique 2

14 décembre 2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	But . . . . .	2
1.2	Glossaire . . . . .	2
1.3	Historique . . . . .	3
<b>2</b>	<b>Besoins utilisateur</b>	<b>4</b>
2.1	Fonctionnels . . . . .	4
2.1.1	Connexion et inscription . . . . .	4
2.1.2	Liste d'amis . . . . .	4
2.1.3	Tableau de score . . . . .	4
2.1.4	Lancement d'une partie de jeu . . . . .	4
2.1.5	Partie de jeu . . . . .	4
2.2	Non fonctionnels . . . . .	4
<b>3</b>	<b>Besoins système</b>	<b>4</b>
3.1	Fonctionnels . . . . .	4
3.1.1	Connexion et inscription . . . . .	4
3.1.2	Menu principal . . . . .	4
3.1.3	Liste d'amis . . . . .	5
3.1.4	Tableau de scores . . . . .	5
3.1.5	Rejoindre une partie . . . . .	5
3.1.6	Héberger une partie . . . . .	5
3.1.7	Partie de jeu . . . . .	5
3.1.8	Modes de jeux alternatifs . . . . .	5
3.1.9	Gestion du score . . . . .	6
3.2	Non fonctionnels . . . . .	6
3.2.1	Sécurité . . . . .	6
3.2.2	Plateformes . . . . .	6
3.2.3	Réseau . . . . .	6
<b>4</b>	<b>Architecture et fonctionnement du système</b>	<b>6</b>
4.1	Architecture réseau . . . . .	6
4.2	Acteurs . . . . .	6
4.3	Connexion et inscription . . . . .	6
4.4	Interfaces . . . . .	6
4.5	Interface terminal . . . . .	7
4.5.1	Les commandes du terminal . . . . .	7
4.5.2	Gestion des amis . . . . .	7
4.5.3	Tableau de score . . . . .	7
4.5.4	Plateau de jeu . . . . .	7
4.5.5	Le chat . . . . .	7
4.6	Interface graphique (GUI) . . . . .	7
4.6.1	Menu connexion . . . . .	7
4.6.2	Menu inscription . . . . .	7
4.6.3	Menu tableau de score . . . . .	8
4.6.4	Menu principal . . . . .	8
4.6.5	Menu . . . . .	8
4.6.6	Menu Join . . . . .	8
4.6.7	Affichage du jeu . . . . .	8
4.6.8	Panneau de la liste d'amis . . . . .	8

4.6.9	Panneau du chat de partie . . . . .	8
4.6.10	Panneau du chat privé . . . . .	8
<b>5</b>	<b>Annexes</b>	<b>9</b>
5.1	Use case diagram . . . . .	9
5.1.1	Connexion . . . . .	9
5.1.2	Gestion des amis . . . . .	10
5.1.3	Lancement de partie . . . . .	10
5.1.4	Partie de jeu . . . . .	11
5.2	Diagrammes de classe . . . . .	12
5.3	Diagrammes de séquence . . . . .	16
5.4	Exemple d'interface . . . . .	20

## Liste des tableaux

1	Historique des modifications du SRD . . . . .	3
2	Commandes en mode terminal pour se connecter ou créer un compte . . . . .	7
3	Commandes en mode terminal pendant la durée du programme, sur le menu principal et en jeu . . . . .	7
4	Use case : explications du mécanisme de connexion et de création de compte . . . . .	9
5	Use case : explications de la gestion des amis . . . . .	10
6	Use case : explications du mécanisme de création de partie . . . . .	10
7	Use case : explications des fonctionnalités disponibles lors de la partie de jeu . . . . .	11

## Table des figures

1	Use Case . . . . .	9
2	Diagramme de classe : <i>master server</i> . . . . .	12
3	Diagramme de classe : Application . . . . .	13
4	Diagramme de classe : Interface . . . . .	15
5	Diagramme de séquence : Connexion . . . . .	16
6	Diagramme de séquence : Principal . . . . .	17
7	Diagramme de séquence : Amis . . . . .	18
8	Diagramme de séquence : Jeu . . . . .	19
9	Exemple design de la partie en terminal . . . . .	20

# 1 Introduction

Quoridor est un jeu de société se jouant à 2 ou 4 joueurs. La partie se déroule sur un plateau de 9x9 cases séparées par des sillons. Chaque joueur dispose d'un pion et de 10 murs. Les pions se déplacent sur les cases, les murs se placent dans les sillons. Un mur est de deux cases (+une largeur de sillon) de longueur. Lors de son tour de jeu, chaque joueur pose un mur ou déplace son pion au choix. Un pion ne peut pas traverser un mur. Un pion peut se déplacer dans les quatre directions correspondantes aux quatre cotés de la case sur laquelle il se trouve. Lorsque deux pions sont côte à côte, ils peuvent se sauter par-dessus et atterrir de l'autre côté, tant que ce n'est pas bloqué par un mur. On ne peut pas totalement enfermer un pion avec des murs. Chaque joueur commence d'un côté du plateau, face à son adversaire. Le gagnant est le joueur dont le pion arrive en premier à son côté opposé.

## 1.1 But

Ce programme a pour objectif de permettre à 2 ou 4 joueurs de faire une partie de Quoridor sur ordinateur. Les utilisateurs de ce programme pourront :

- créer un compte joueur
- se connecter à leur compte joueur
- consulter leur liste d'amis
- ajouter ou supprimer des amis
- consulter un tableau des meilleurs scores Quoridor
- utiliser une boîte de chat intégrée pour communiquer avec leurs amis
- héberger une partie de Quoridor
- rejoindre une partie de Quoridor hébergée sur une autre machine
- inviter des amis à jouer
- jouer une partie avec 1 ou 3 autres personnes

Ce programme s'adresse à des personnes qui désirent jouer une partie en ligne de Quoridor tout en communiquant par texte avec leurs amis.

Ce programme ne permet pas le chat vocal.

Ce programme ne permet pas de jouer contre des inconnus.

Ce programme est utilisable pour les 3 à 99 ans.

Ce programme ne contient pas de langage abusif ou de contenu pouvant heurter la sensibilité des personnes fragiles.

## 1.2 Glossaire

- chat : système permettant de communiquer par écrit en instantané.
- *host* : utilisateur dont la machine sera utilisée pour héberger une partie.
- *master server* : est un serveur tiers qui se charge d'authentifier et d'enregistrer les utilisateurs, de maintenir leurs listes d'amis, de la communication inter-utilisateurs lorsqu'ils ne sont pas en train de jouer, de la gestion des scores, et d'envoyer les invitations pour rejoindre des parties de jeu.
- utilisateur / joueur : personne bénéficiant du programme. Cette personne a la possibilité de se connecter et de jouer avec ses amis. Elle ne s'occupe pas de la partie fonctionnelle du programme.
- interopérabilité : possibilité de communication entre plusieurs systèmes différents. Et aussi entre l'interface terminal et graphique.
- vignette : petite indication visuelle sur les bords d'un bouton.
- panneau : une zone dédiée de la fenêtre disposant d'un affichage individuel.
- serveur autoritativ : un serveur qui a autorité absolue sur ses clients, il doit traiter et approuver chaque demande de ses clients avant de répondre.

### 1.3 Historique

Numéro de version	Nom	Modifications	Date
0.1	Etienne, Oury	Structure, Intro, But	23/11/2021
0.2	Tous	Besoins, Design	24/11/2021
0.3	Tous	Besoins utilisateur	06/12/2021
0.4	Tous	Besoins système	07/12/2021
0.5	Oury, Etienne, Cosmin	Design	08/12/2021
0.6	Tous	Relecture, schémas	10/12/2021
0.7	Xhulio, Ilias, Cosmin	Commandes, Use case	11/12/2021
0.9	Tous	Modes de jeu, diagrammes	13/12/2021
0.9	Tous	Annexes et relecture finale	13/12/2021

TABLE 1: Historique des modifications du SRD

## 2 Besoins utilisateur

### 2.1 Fonctionnels

#### 2.1.1 Connexion et inscription

L'utilisateur peut se connecter avec son identifiant et son mot de passe. Dans le cas où ce dernier n'a pas de compte, il peut en créer un pour pouvoir jouer. L'utilisateur ne peut pas récupérer son mot de passe de manière automatisée en cas d'oubli.

#### 2.1.2 Liste d'amis

Lorsqu'il est connecté, l'utilisateur peut consulter sa liste d'amis. Dans cette liste d'amis, il peut faire une recherche, ajouter ou supprimer des amis.

#### 2.1.3 Tableau de score

L'utilisateur peut consulter le classement des meilleurs scores Quoridor mis à jour

#### 2.1.4 Lancement d'une partie de jeu

L'utilisateur peut ensuite décider de lancer ou de rejoindre une partie d'un ami en étant invité.

#### 2.1.5 Partie de jeu

Durant la partie, l'ensemble des utilisateurs ont accès au chat de partie.

Le *host* peut effectuer les actions suivantes : mettre en pause, sauvegarder ou charger une partie. Lorsque c'est son tour, le joueur peut soit déplacer son pion soit placer un mur.

Tous les joueurs peuvent déclarer forfait et quitter à n'importe quel moment la partie.

### 2.2 Non fonctionnels

Pour le moment, le client n'a pas précisé de besoins non fonctionnels particuliers.

## 3 Besoins système

### 3.1 Fonctionnels

#### 3.1.1 Connexion et inscription

Le système doit demander des informations de connexion qui sont le nom d'utilisateur et le mot de passe par le biais d'un panneau. Si l'utilisateur n'a pas de compte, un bouton doit pouvoir lui permettre d'en créer un. Il devra entrer un nouveau nom d'utilisateur et un mot de passe qu'il devra écrire une deuxième fois, pour vérifier qu'il ne s'est pas trompé. Si le nom d'utilisateur entré lors de la création est celui d'un compte déjà existant, l'utilisateur doit en être informé et devra en choisir un nouveau.

#### 3.1.2 Menu principal

Lorsque la connexion est réussie, la fenêtre principale doit contenir une liste d'amis à droite et au centre une liste de boutons `JOIN`, `HOST`, `SCORES` et `QUIT`. Si l'utilisateur est invité par un ami à jouer, une vignette apparaîtra près du bouton `JOIN`. Cette dernière indiquera le nombre d'invitations. Ce menu doit également permettre à l'utilisateur de régler le volume du jeu. Ce dernier, sera aussi présent sur toutes les fenêtres du programme.

### 3.1.3 Liste d'amis

Dans la liste d'amis, un bouton `ADD FRIEND` doit pouvoir permettre d'ajouter un ami. Chaque ami de la liste doit contenir un bouton qui permet de le supprimer et un autre qui permet de discuter avec lui par écrit seulement s'il est connecté. Une autre fenêtre apparaîtra alors près de la liste d'amis. Les amis sont triés par ordre alphabétique en deux catégories non séparées : ceux qui sont connectés sont au dessus et ceux qui ne le sont pas en-dessous. Si un ami envoie un message à l'utilisateur, une vignette apparaîtra près de son nom dans la liste.

### 3.1.4 Tableau de scores

Le tableau des scores doit afficher les utilisateurs (triés par score décroissant). Plus bas, une ligne avec le rang et le score du joueur actuel doit y figurer.

### 3.1.5 Rejoindre une partie

Le bouton `JOIN` du menu principal doit amener l'utilisateur sur une nouvelle fenêtre dans laquelle il peut rejoindre les parties auxquelles il a été invité par ses amis. Si l'utilisateur n'a aucune invitation, il doit en être averti.

### 3.1.6 Héberger une partie

Le bouton `HOST` du menu principal doit amener l'utilisateur sur une nouvelle fenêtre dans laquelle il pourra créer sa partie. Dans cette fenêtre, la liste d'amis dispose d'un bouton supplémentaire par ami, qui doit permettre de l'inviter. Cette fenêtre doit également permettre à l'utilisateur de choisir son mode de jeu. Les boutons de mode de jeu sont les suivants : `NORMAL`, `RUMBLE`, `BOOSTER`, `CAT & MOUSE`. Ceux pour le choix du nombre de joueurs sont `2 PLAYERS` et `4 PLAYERS`. Un panneau `COUNTER` dénombre les joueurs ayant déjà accepté l'invitation. Un bouton `LOAD` permet de charger une partie préalablement sauvegardée via un fichier.

### 3.1.7 Partie de jeu

Lorsque la partie commence, le plateau et le chat général du jeu doivent s'afficher. Si l'utilisateur est le *host* de la partie, le programme doit lui permettre de mettre en pause et de sauvegarder l'état de la partie via les boutons `PAUSE` et `SAVE`. Lors de la partie, le programme doit réagir aux commandes de l'utilisateur et déplacer les pions. La fenêtre doit également afficher deux icônes accompagnées d'un nombre indiquant le nombre total de murs restants pour chaque joueur. Si un coup n'est pas permis par les règles du jeu, le programme doit permettre au joueur de recommencer son coup. Lors de la victoire, un message apparaît félicitant le gagnant. Le système doit alors mettre les scores de chaque joueur à jour.

### 3.1.8 Modes de jeux alternatifs

L'utilisateur peut changer de mode de jeu s'il désire une expérience différente. Sont proposés les modes : RUMBLE, BOOSTER, CAT & MOUSE. Ces modes de jeu peuvent être joués avec 2 ou 4 joueurs.

- **Rumble** : Désormais on peut sacrifier des murs de sa réserve pour acquérir des coups spéciaux. (coups spéciaux : glace = bloque un tour à l'adversaire, double-saut = permet de parcourir deux cases durant son tour, Bélier = détruit un mur ennemi).
- **Booster** : Le plateau fait aléatoirement apparaître des boosters sur certaines cases, si un joueur déplace son pion dessus il peut disposer de coups spéciaux. (coups spéciaux : charmer = permet de faire jouer le pion adverse à la place de l'adversaire, plus les coups du RUMBLE)
- **Cat & Mouse** : A chaque fois, la première moitié des joueurs sera définie comme étant le/les chat(s), et la deuxième moitié comme étant la/les souris. Le but de la souris est de fuir, le chat. Pour ce faire, la

souris peut se déplacer une seule fois et placer un mur, alors que le chat lui peut se déplacer deux fois mais ne peut pas placer de murs. Pour que la souris puisse gagner, elle doit atteindre l'autre extrémité du plateau. Pour que le chat puisse gagner, celui-ci doit pouvoir se trouver sur la même case que la souris.

Ces modes de jeu peuvent être équilibrés au fur et à mesure du développement.

### 3.1.9 Gestion du score

Le score est augmenté de 1 par victoire. A ce stade, le score est le nombre de victoires du joueur.

## 3.2 Non fonctionnels

### 3.2.1 Sécurité

Comme l'architecture du réseau est de type serveur autoritatif, la triche est impossible sans un accès privilégié à la machine sur laquelle tourne le serveur, le *host*. En l'occurrence, seul l'accès physique à la machine de l'utilisateur peut altérer les données de la partie.

### 3.2.2 Plateformes

Le programme est multiplateforme : il fonctionne sur un système Windows, Linux et MacOS.

### 3.2.3 Réseau

L'utilisateur doit disposer d'une connexion internet et de partenaires de jeu car une IA n'est pas implémentée. Pour diriger le tout, il faut que le *master server* soit hébergé sur une machine tierce (voir ci-dessus). Le *host* doit ouvrir les ports de son modem afin de permettre aux autres joueurs de se connecter sur sa machine.

## 4 Architecture et fonctionnement du système

### 4.1 Architecture réseau

Voir *Class diagram*, explication en progrès.

### 4.2 Acteurs

Le *master server* (voir glossaire) 1.2.

L'utilisateur lambda peut rejoindre une partie de jeu en étant invité par un *host* ami, il peut interagir avec sa liste d'amis et donc les autres utilisateurs au moyen du *master server*.

Le *host* est un utilisateur qui héberge une partie de jeu sur sa machine, il peut inviter d'autres utilisateurs amis au moyen du *master server*.

### 4.3 Connexion et inscription

Cet aspect est géré par un programme tiers tournant sur une machine : le *master server*. Les données de connexion entre le *master server* et l'utilisateur sont chiffrées.

### 4.4 Interfaces

L'utilisateur a le choix entre deux interfaces, soit terminal soit graphique (GUI). Interopérabilité des interfaces : deux joueurs peuvent donc s'affronter avec deux interfaces différentes



## 4.5 Interface terminal

Le jeu dispose d'un affichage terminal et l'utilisateur aura la capacité d'utiliser les commandes définies par le programme afin d'interagir avec ce dernier.

### 4.5.1 Les commandes du terminal

Connexion
<code>login <i>username password</i></code>
<code>signin <i>username password</i></code>

TABLE 2: Commandes en mode terminal pour se connecter ou créer un compte

Globales	Menu principal	En partie
<code>help</code>	<code>helphost</code>	<code>gchat</code>
<code>chat <i>username</i></code>	<code>host [save] [load] <i>players gamemode</i></code>	<code>quit</code>
<code>disconnect</code>	<code>pendingrequests</code>	<code>pause</code>
<code>friend list</code>	<code>invite <i>username</i></code>	<code>save</code>
<code>friend add <i>username</i></code>	<code>join <i>username</i></code>	
<code>friend remove <i>username</i></code>		
<code>score</code>		

TABLE 3: Commandes en mode terminal pendant la durée du programme, sur le menu principal et en jeu

### 4.5.2 Gestion des amis

Affiche la liste complète des amis. Une colonne pour les amis connectés et l'autre pour les amis non connectés.

### 4.5.3 Tableau de score

Affiche dans le terminal tous les utilisateurs ainsi que leurs scores triés de manière décroissante.

### 4.5.4 Plateau de jeu

Le plateau est affiché en terminal, chaque ligne dispose d'un chiffre et chaque colonne d'une lettre. Les pions des joueurs sont représentés par des caractères du type \*, +, H, 8 (temporaire). Les stocks de murs sont représentés par le symbole du pion et le nombre de murs dans leurs réserves respectives.

### 4.5.5 Le chat

La partie inférieure du terminal affiche les messages privés, les notices de partie ainsi que le chat de partie. Le type du message est précisé entre crochets avant celui-ci.

## 4.6 Interface graphique (GUI)

### 4.6.1 Menu connexion

Une petite fenêtre login comprenant deux espaces de saisie, un pour le nom d'utilisateur et l'autre pour le mot de passe. Un bouton `CONNEXION`. Un bouton `REGISTER`.

### 4.6.2 Menu inscription

Affiche trois espaces de saisie, un pour le nom d'utilisateur et deux pour le mot de passe (première saisie et confirmation).

#### 4.6.3 Menu tableau de score

La fenêtre contient une liste de tous les utilisateurs. Ils sont triés en fonction du score. Au dessus un panneau indique le score de l'utilisateur et son classement actuel. Cet écran possède le bouton retour au menu principal.

#### 4.6.4 Menu principal

Un menu composé des boutons centraux :

- 
- 
- 
- 

En haut à gauche se trouve un bouton permettant de régler le volume sonore.

#### 4.6.5 Menu

Affiche les boutons :

- 
- 
- 
- avec chemin d'accès
- 

Affiche également la liste d'ami avec le bouton supplémentaire pour pouvoir en inviter et le panneau de compteurs de personnes en train d'attendre le début de la partie.

#### 4.6.6 Menu Join

Affiche une liste des invitations

#### 4.6.7 Affichage du jeu

Affiche le plateau de jeu dans le centre. Représente les murs lorsqu'ils sont placés. Les pions de chaque joueur disposent d'une couleur spécifique. Affiche le compteur de murs avec un logo représentant les murs et une indication numérique indiquant combien il en reste dans chaque réserve. Affiche deux boutons  et  uniquement lorsque l'utilisateur est l'*host*. Cet écran possède le bouton retour au menu principal, qui revient à abandonner la partie.

#### 4.6.8 Panneau de la liste d'amis

Visible tout le temps, il s'agit d'un panneau à droite de la fenêtre, pouvant s'étendre pour montrer les notifications. Chacun des noms d'utilisateurs des amis de la liste est suivi d'un bouton pour le supprimer et un bouton pour discuter. Si l'utilisateur est dans le menu *host* il voit également apparaître un troisième bouton qui permet d'inviter. Cet écran possède aussi un bouton qui permet le retour au menu principal.

#### 4.6.9 Panneau du chat de partie

Ce panneau se trouve en dessous du plateau de jeu dans la partie inférieur gauche.

#### 4.6.10 Panneau du chat privé

Ce panneau se trouve dans la partie inférieure du panneau de la liste d'amis.

## 5 Annexes

### 5.1 Use case diagram

Le diagramme Use Case permet d'avoir une vue d'ensemble du programme depuis le point de vue de l'utilisateur. Une explication détaillée se trouve en dessous de la figure.

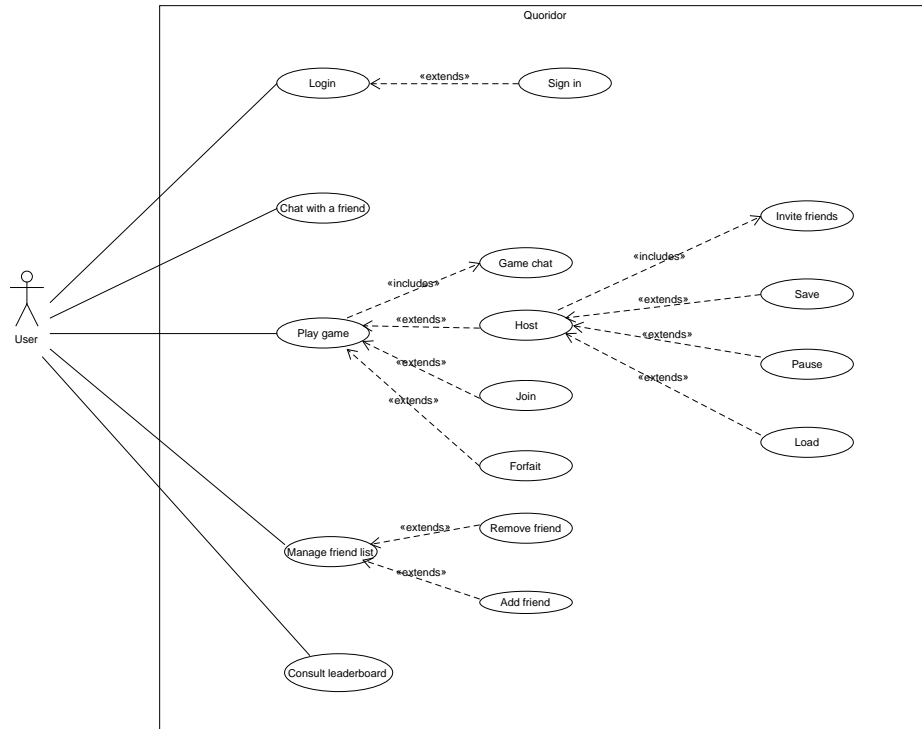


FIGURE 1 – Use Case

#### 5.1.1 Connexion

Acteur	Pré-conditions	Post-conditions	Cas général	Cas exceptionnels
Log in	L'utilisateur possède déjà un compte	L'utilisateur est connecté	L'utilisateur indique son nom d'utilisateur et son mot de passe	Nom d'utilisateur incorrect Mot de passe incorrect
Sign in	L'utilisateur n'a pas de compte	L'utilisateur a créé un compte	L'utilisateur entre un nom d'utilisateur et deux fois son mot de passe	Le nom d'utilisateur existe déjà Le mot de passe ne respecte pas les conditions

TABLE 4: Use case : explications du mécanisme de connexion et de création de compte

### 5.1.2 Gestion des amis

Acteur	Pré-conditions	Post-conditions	Cas général	Cas exceptionnels
Chat with friend	L'utilisateur doit avoir l'ami avec lequel il veut communiquer dans sa liste d'amis L'ami doit être connecté	L'utilisateur a envoyé un message à un ami	L'utilisateur envoie un message à l'ami de son choix	Néant
Consult leaderboard	L'utilisateur doit être connecté	Néant	L'utilisateur consulte le tableau des scores	Néant
Add friend	L'ami possède un compte	Néant	L'ami est ajouté dans la liste d'amis de l'utilisateur	L'utilisateur est déjà ami avec cette personne
Remove friend	L'utilisateur connecté doit cibler un ami	Néant	L'ami cible est retiré de la liste	Néant

TABLE 5: Use case : explications de la gestion des amis

### 5.1.3 Lancement de partie

Acteur	Pré-conditions	Post-conditions	Cas général	Cas exceptionnels
Host Game	L'utilisateur doit être connecté L'utilisateur ne doit pas se trouver en partie	Néant	L'utilisateur peut choisir les modes de jeu et inviter des amis	Néant
Load	Être host de la partie en cours	Neant	Une partie est antérieurement sauvegardée est chargée	Fichier de sauvegarde incompatible
Invite friends	Être host de la partie en cours Avoir des amis connectés dans sa liste d'amis	L'ami a le choix d'accepter ou de refuser une invitation	Les amis sont invités à rejoindre une partie	L'ami invité est déjà dans une partie.
Join Game	L'utilisateur doit être connecté L'utilisateur ne doit pas se trouver en partie L'utilisateur doit avoir accepté une invitation	Néant	L'utilisateur rejoint une partie	L'utilisateur tente de rejoindre une partie déjà pleine
Play Game	Le nombre de joueurs présents est 2 ou 4	Néant	L'utilisateur joue une partie avec ses amis	Néant

TABLE 6: Use case : explications du mécanisme de création de partie

#### 5.1.4 Partie de jeu

Acteur	Pré-conditions	Post-conditions	Cas général	Cas exceptionnels
Save	Être host de la partie en cours et l'avoir mise en pause	Néant	La partie est sauvegardée dans un fichier	Néant
Forfeit	Être en partie	L'utilisateur n'est plus présent dans une partie	L'utilisateur a perdu	Néant
Pause	Être host de la partie en cours	Les joueurs ne peuvent pas interagir pour le moment	La partie se met en pause	Néant
Game chat	Être en partie	Néant	Permet de parler avec les joueurs de la partie en cours	Le message est trop volumineux

TABLE 7: Use case : explications des fonctionnalités disponibles lors de la partie de jeu

## 5.2 Diagrammes de classe

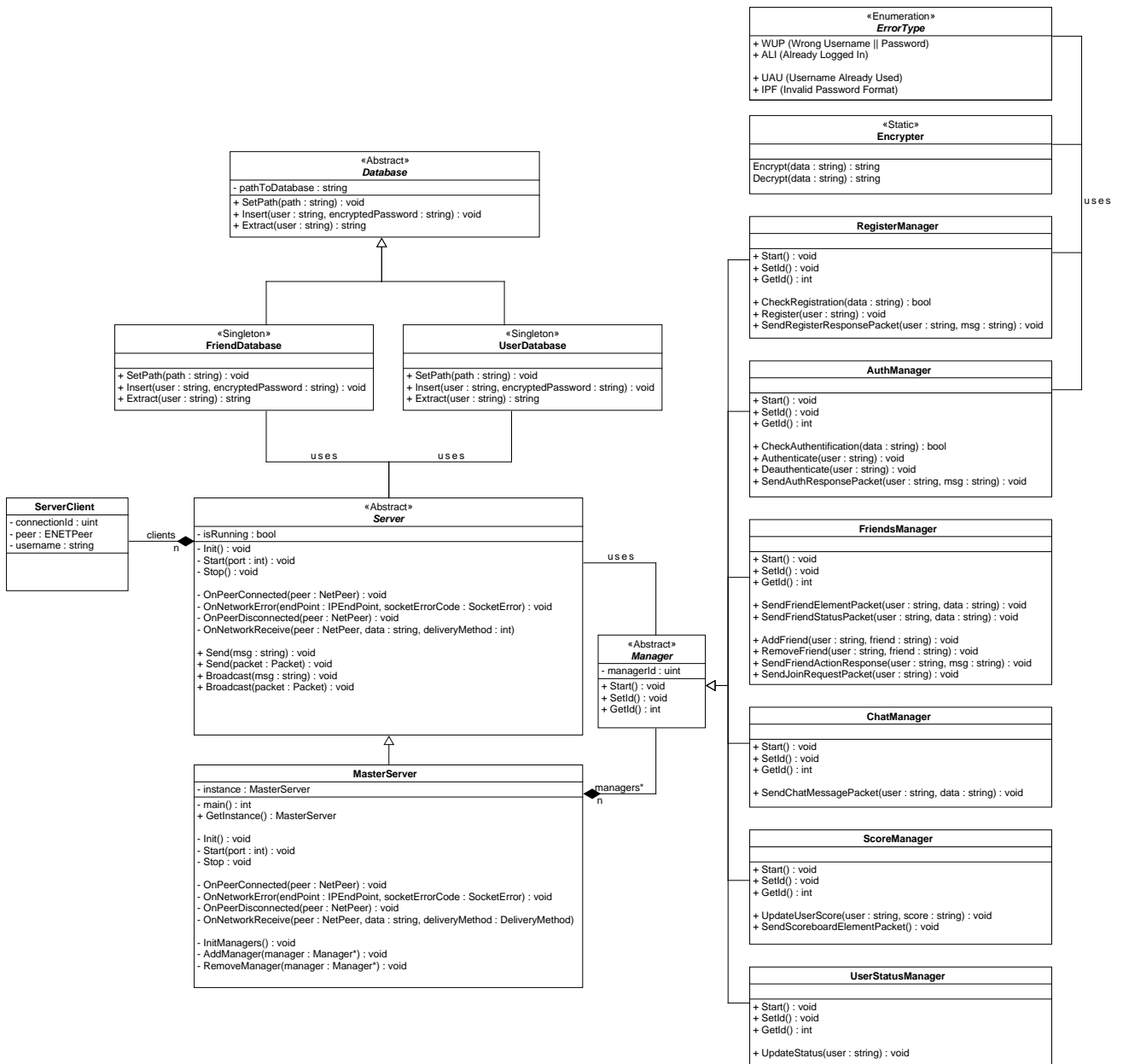


FIGURE 2 – Diagramme de classe : *master server*

**MasterServer** gère l'authentification, l'enregistrement, les invitations de parties, les amis ainsi que le chat entre joueurs. **Manager** Plug-in lancé par le **MasterServer** permettant de gérer un tâche spécifique.



serveur (hebergé par le host), d'envoyer et de recevoir des informations depuis le réseau. **RemoteClient** est la représentation des autres clients pour un client.

**Server** représente la partie réseau du *host* mais pas seulement. Il s'occupe aussi de lancer le serveur de la partie, envoie et reçoit des informations des clients. **ServerClient** est la représentation des clients pour le client.

**Game** représente la logique du jeu. Il gère le tour des joueurs, les coups, les défaites et la victoire.

**Board** représente la structure du plateau.

**Board** représente le plateau dans le jeu. **Wall** représente un mur, son état et ses actions dans le jeu. **Pawn** représente un pion, son état et ses actions dans le jeu. **Cell** est le conteneur des **Piece Controller** permet à l'utilisateur de contrôler.



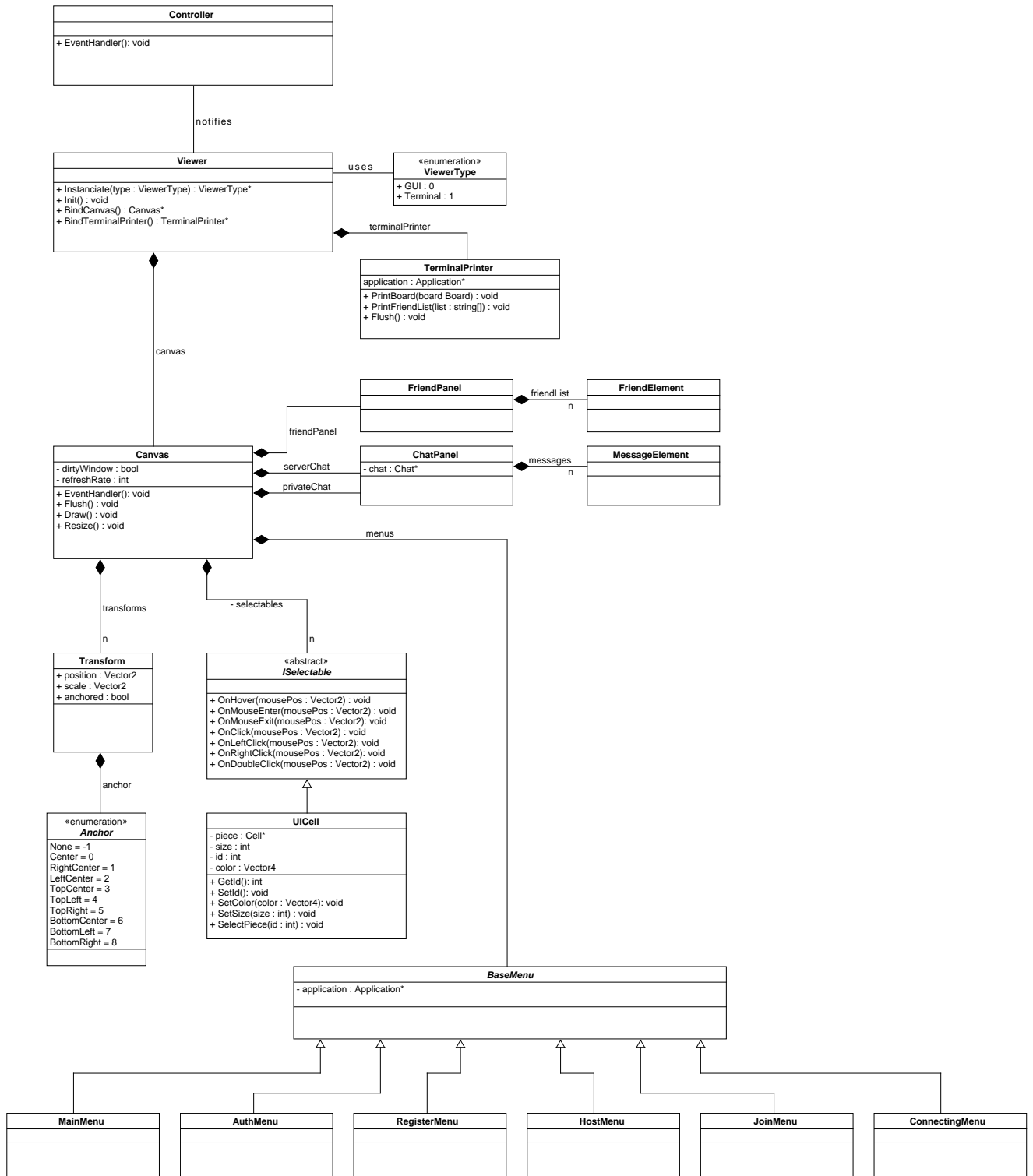


FIGURE 4 – Diagramme de classe : Interface

**Viewer** permet à l'utilisateur de voir une representation de l'application via une interface. **TerminalPrinter** Gère l'interface terminal. **Canvas** Gère l'interface graphique GUI.

### 5.3 Diagrammes de séquence

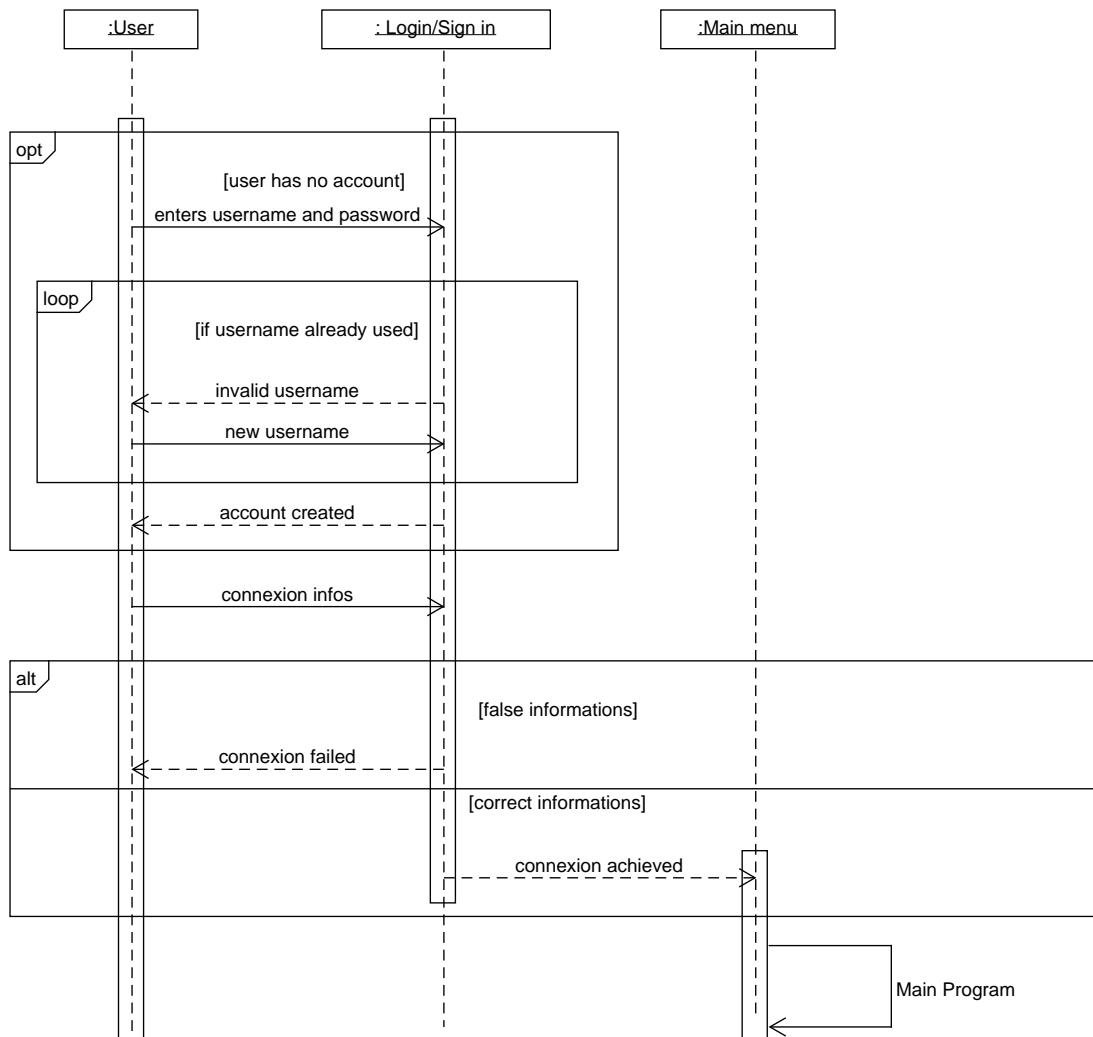


FIGURE 5 – Diagramme de séquence : Connexion

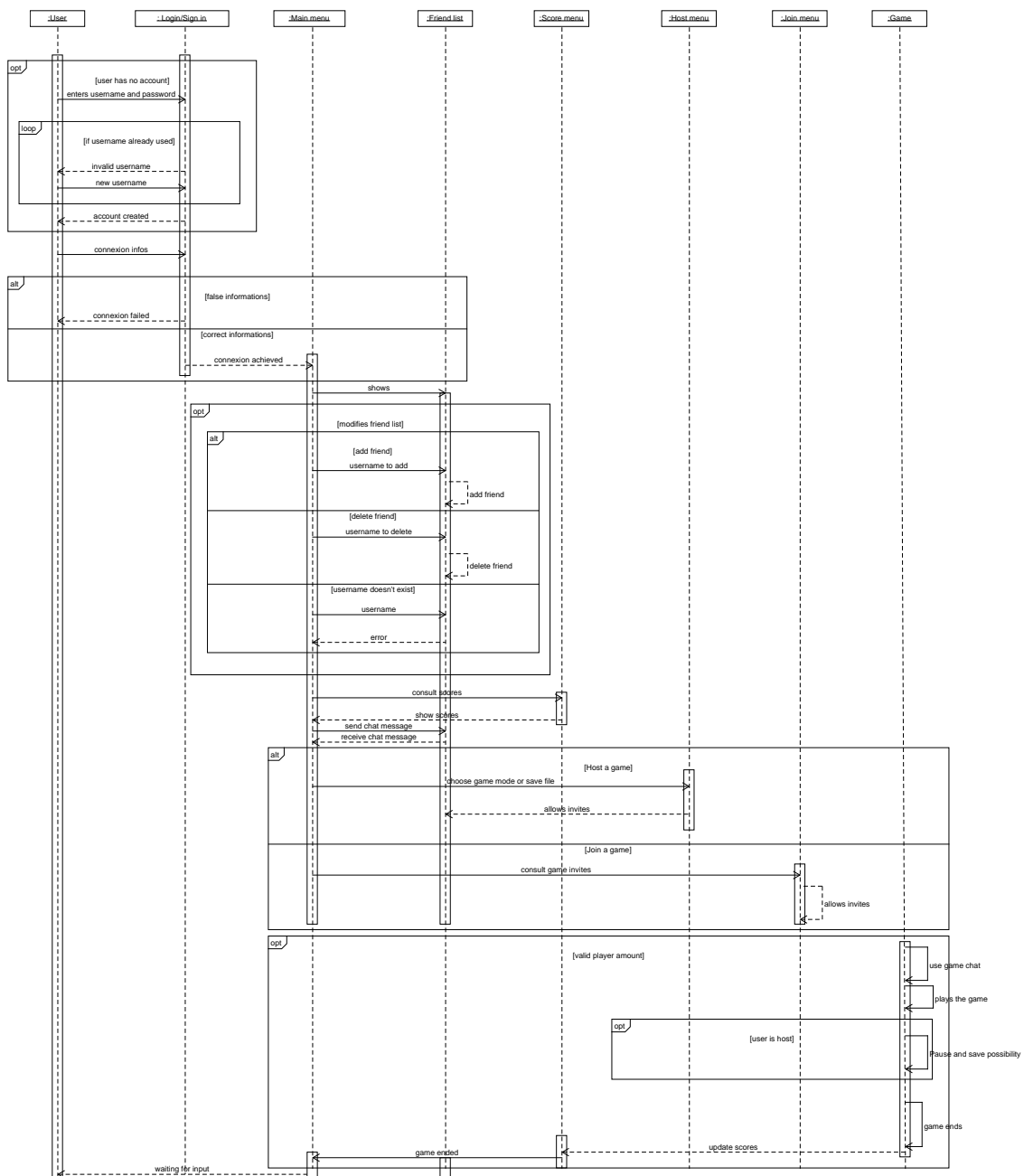


FIGURE 6 – Diagramme de séquence : Principal

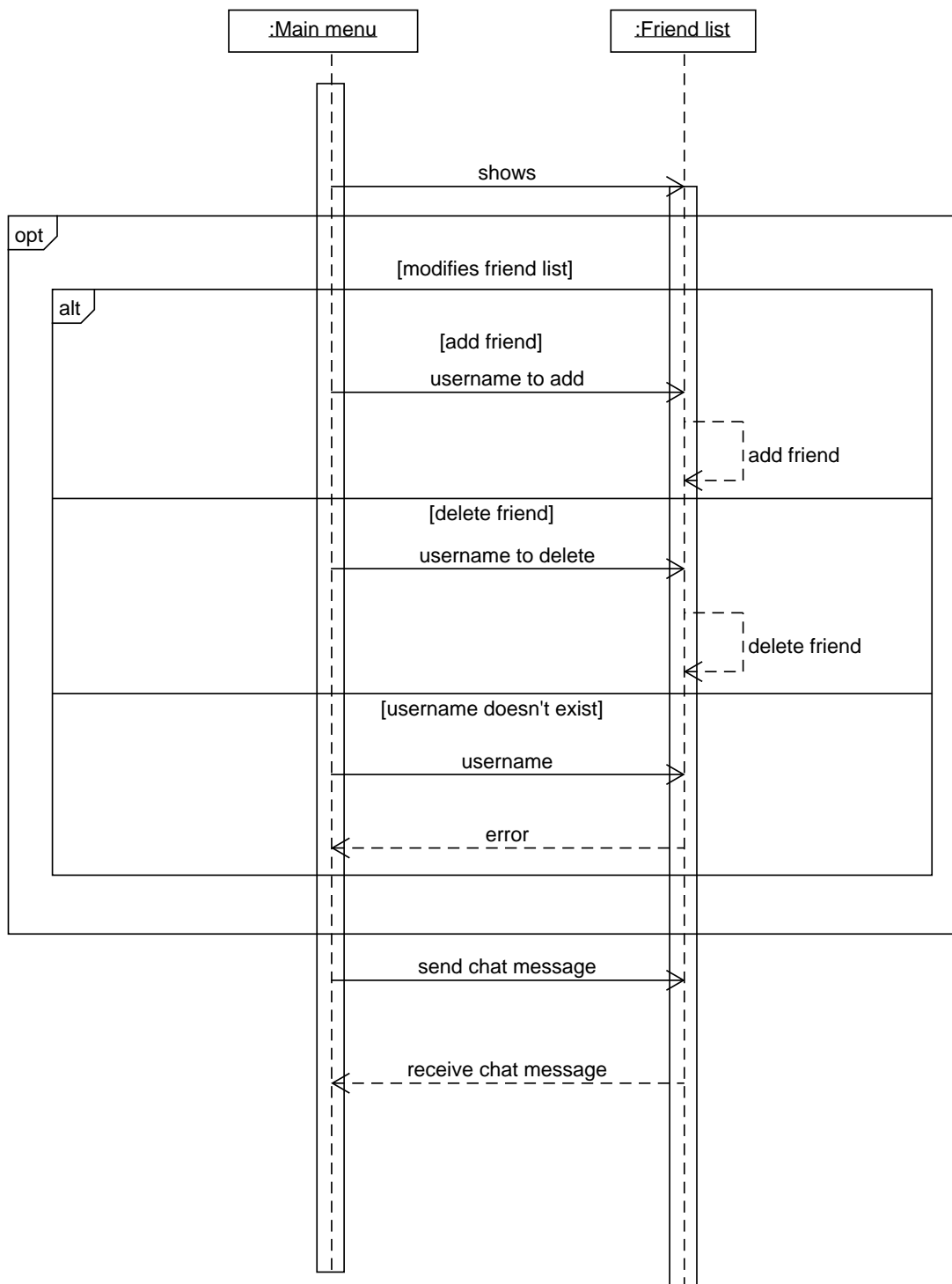


FIGURE 7 – Diagramme de séquence : Amis

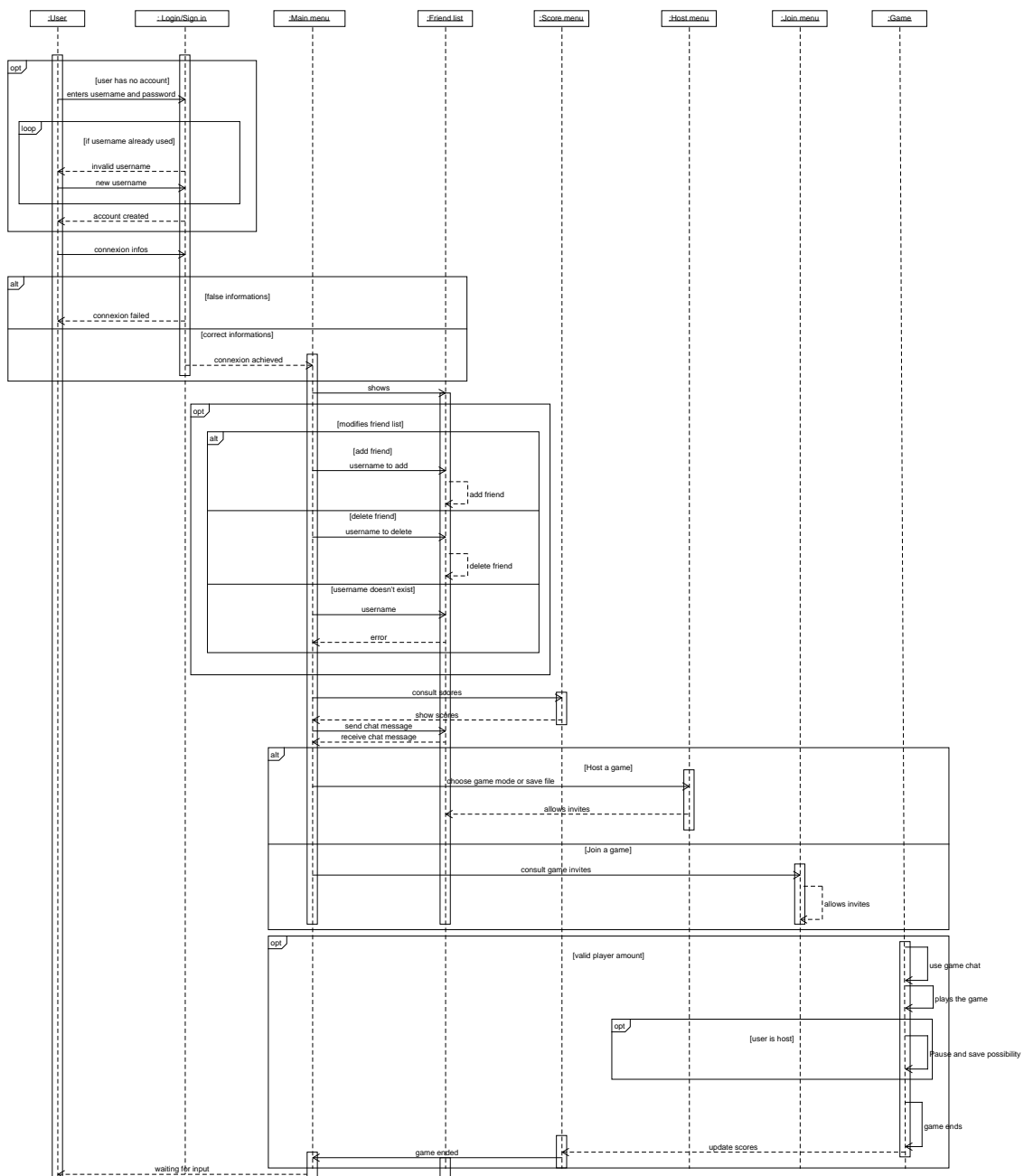


FIGURE 8 – Diagramme de séquence : Jeu

## 5.4 Exemple d'interface



FIGURE 9 – Exemple design de la partie en terminal