

INFO-F209 - PROJETS D'INFORMATIQUE 2

SRD -TOUCHÉ-COULÉ

Auteurs :

Othman EL KAZBANI - 000493194

FatimaZohra LAHRACH - 000536142

Jawad CHERKAOUI - 000576517

Superviseur :

Alexis REYNOUARD

Année académique 2023-2024

Table des matières

Disclaimer	3
1 Introduction	5
1.1 But du projet	5
1.2 But du jeu	5
1.3 Historique des modifications du SRD	6
2 Besoins Utilisateur	7
2.1 Menu de connexion	7
2.1.1 Création de compte	8
2.1.2 Connexion à un compte existant	8
2.2 Menu principal	9
2.2.1 Lancement du Menu de jeu	9
2.2.2 Gérer sa liste d'amis et ses demandes d'amis	10
2.3 Menu de jeu	10
2.3.1 Créer une partie	10
2.3.1.1 Configurer le mode de jeu	11
2.3.1.2 Configurer le temps par tour et par partie	11
2.3.2 Rejoindre une partie	11
2.3.3 Regarder des parties (Mode Observateur)	11
2.3.4 Revoir une partie (Mode Replay)	12
2.3.5 Gérer ses invitations	12
2.4 Commandes	12
2.4.1 Chat	12
2.4.2 Inviter un ami à sa partie créée	13

2.4.3	Afficher les plateaux	13
2.5	Déroulement d'une partie	13
3	Besoins système	15
3.1	Base de données	15
3.2	Battleship	15
3.2.1	Système client-serveur	16
3.2.1.1	Système de communication	18
3.2.2	Threads	18
3.2.3	Affichage	18
3.2.3.1	Terminal	18
3.2.3.2	Interface Graphique (GUI)	19
3.2.4	Triche	19

Disclaimer

Ce SRD a été adapté à partir du SRD rédigé à la fin de la phase 2. Des éléments ont été ajoutés, et plusieurs parties ont été modifiées pour assurer la compatibilité avec le code présent dans la branche "network_temp". Ces modifications ont été réalisées par Lahrach FatimaZohra, Cherkaoui Jawad et El Kazbani Othman.

Table des figures

1	Diagramme use case du menu de connexion	7
2	Diagramme use case du menu principal	9
3	Menu de jeu	10
4	Commandes disponibles	12
5	Affichage du plateau durant une partie	14
6	Structure Client	16
7	Structure Serveur	17

1 Introduction

Ce Software Requirements Document (SRD) présente les besoins et les exigences pour le développement d'une application en C++ du jeu Touché-Coulé, autrement appelé bataille navale ou encore Battleship, destinée à être jouée en réseau local (LAN).

1.1 But du projet

L'objectif fondamental du projet est de créer une version digitale et jouable en réseau local du jeu Battleship. Les utilisateurs devront, entre autres, être capables de se créer un compte, de s'y connecter, de pouvoir envoyer et accepter des demandes d'amis d'autres utilisateurs, et de démarrer des parties avec ces derniers. Pour mettre toutes ces fonctionnalités en place, une bonne maîtrise de l'architecture type client-serveur est requise ainsi qu'une bonne connaissance en base de données.

1.2 But du jeu

Le jeu Battleship est un jeu de stratégie qui est composé de deux phases différentes. La première phase est la création des flottes de combat des deux joueurs. Les joueurs sont invités à remplir leur grille respective en plaçant leurs bateaux aux positions et rotations voulues. La deuxième phase est la phase de combat durant laquelle les joueurs vont devoir s'affronter. Chacun à leur tour, les joueurs désignent une position adverse qu'ils souhaitent attaquer. La partie se termine lorsqu'un des deux joueurs parvient à couler l'entièreté de la flotte adverse.

1.3 Historique des modifications du SRD

Introduction, but du projet et but du jeu	Cyril Stembert	8/12/2023
Use case diagramme de placement de bateaux et gestion des tirs	Cyril Stembert Aleksandra Otto Othman El Kazbani	10/12/2023
Diagramme de classes	Othman El Kazbani	11/12/2023
Diagramme de séquence d'une partie de bataille navale	Chris Eid Othman El Kazbani	11/12/2023
Fonctionnalités : Jeu	Cyril Stembert	13/12/2023
Fonctionnalités : Utilisateur	Cyril Stembert	13/12/2023
Relecture et corrections	FatimaZohra Lahrach	15/12/2023
Diagrammes d'activités Connexion et Création de compte utilisateur	Cyril Stembert	21/02/2024
Diagramme Use case Menu et Placement bateau	Cyril Stembert	24/02/2024
Besoins système : Base de données	Cyril Stembert	08/03/2024
Besoins système : Sécurité et menu	Cyril Stembert	10/03/2024
Besoins système : Battleship Système client-serveur Threads - Affichage	Othman El Kazbani FatimaZohra Lahrach	16/03/2024
Relecture et corrections	Cyril Stembert Othman El Kazbani FatimaZohra Lahrach	16/03/2024
Ajout diagrammes de classe (client + serveur) + use case (menu connexion, menu principal, menu de jeu, commandes) , Modification sections Besoins utilisateurs et systèmes	Jawad Cherkaoui Othman El Kazbani FatimaZohra Lahrach	25/04/2024

TABLE 1 – Tableau des modifications du SRD

2 Besoins Utilisateur

Cette section du SRD décrit toutes les fonctionnalités auxquelles un utilisateur doit avoir accès tel que la gestion de son compte et de ses amis ou encore la configuration d'une partie.

2.1 Menu de connexion

Au lancement du programme client par un utilisateur, un écran de connexion s'affiche lui laissant le choix entre deux options : la création d'un nouveau compte (Signup) ou bien la connexion à un compte existant (Login).

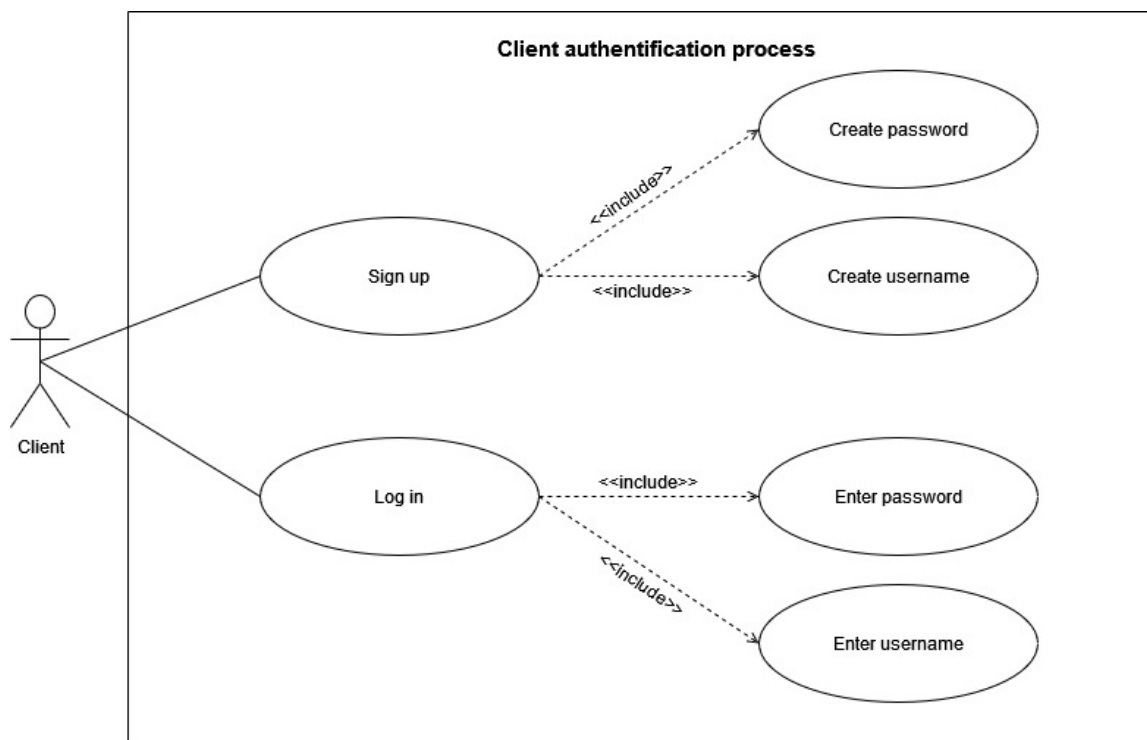


FIGURE 1 – Diagramme use case du menu de connexion

2.1.1 Création de compte

Lorsqu'un utilisateur souhaite créer un nouveau compte, celui-ci doit fournir un nouveau nom d'utilisateur ainsi qu'un mot de passe qui lui sera associé. Le nom d'utilisateur ainsi que le mot de passe entrés doivent tous deux respecter une certaine syntaxe afin de se voir valider (voir ci-dessous). Si la syntaxe n'est pas respectée, l'utilisateur en est informé et est invité à entrer de nouvelles informations.

Règle syntaxique	Nom d'utilisateur	Mot de passe
Longueur	1-15 caractères	1-30 caractères
Unique	Oui	Non

TABLE 2 – Tableau des règles syntaxiques

2.1.2 Connexion à un compte existant

Lorsqu'un utilisateur souhaite se connecter à un compte, celui-ci doit fournir un nom d'utilisateur (pseudonyme) ainsi que le mot de passe qui lui est associé (password). Si les informations entrées par l'utilisateur sont valides, la connexion au compte s'effectue et l'utilisateur se retrouve dans le menu principale.

2.2 Menu principal

Une fois dans le menu principal, l'utilisateur peut choisir parmi plusieurs options. La figure 2 montre les différentes actions disponibles dans ce menu. Les principales catégories incluent : Commencer à jouer (Start playing), qui mène au menu de jeu détaillé dans la section suivante, consulter sa liste d'amis (Check friends), envoyer une demande d'ami (Send friend request), consulter ses demandes d'amis (Check friend requests), ou encore une commande (help) qui affiche quelques fonctionnalités telles que comment envoyer un message à ses amis, ou inviter un ami à une partie créée,...

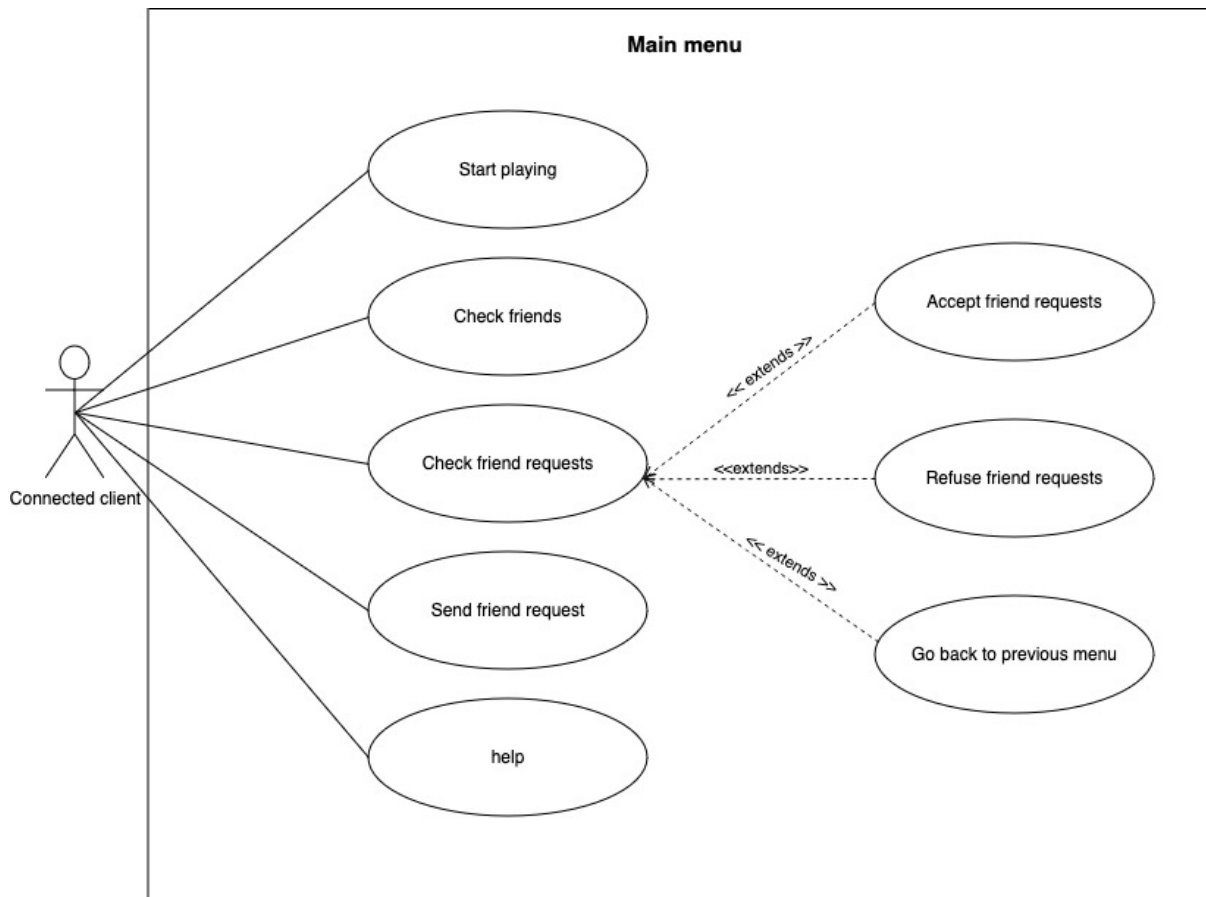


FIGURE 2 – Diagramme use case du menu principal

2.2.1 Lancement du Menu de jeu

En sélectionnant (Start Playing) dans le menu principal, l'utilisateur est redirigé vers un menu de jeu, où l'utilisateur peut choisir parmi plusieurs options pour interagir avec le jeu, il peut ainsi créer une partie, rejoindre une partie déjà créée, regarder une partie en cours, ou encore, consulter les invitations reçues pour rejoindre ou regarder une partie.

2.2.2 Gérer sa liste d'amis et ses demandes d'amis

Chaque utilisateur possède d'une liste d'amis qu'il peut consulter (Check friends), et enrichir en envoyant une demande d'ami à d'autres utilisateurs via une recherche de leurs pseudonymes (Send friend request). Une fois la demande envoyée, elle apparaît dans la liste des demandes d'amis de l'utilisateur destinataire. Celui-ci peut alors consulter ses demandes d'amis (Check friend requests) et choisir d'accepter (Accept friend request) ou de refuser la demande (Refuse friend request). Si la demande est acceptée, les deux utilisateurs deviennent amis et sont ajoutés à leurs listes d'amis respectives.

2.3 Menu de jeu



FIGURE 3 – Menu de jeu

2.3.1 Créer une partie

Un utilisateur peut créer une nouvelle partie en sélectionnant un des deux modes de jeu, et en définissant le temps maximal autorisé pour un tour, et la durée maximale pour une partie. Une fois la partie créée, le joueur est mis en attente d'un autre joueur pour lancer la partie. De plus, une fois mis en attente, le joueur peut inviter ses amis à la partie en temps que joueur ou en tant qu'observateur.

2.3.1.1 Configurer le mode de jeu

Le jeu de la bataille navale possède deux modes distincts :

- **Classique** : Mode de jeu classique où chaque joueur possède chacun une flotte composée de cinq bateaux placés sur la grille de manière horizontale ou verticale. Les dimensions respectives des cinq bateaux sont 1x2, 1x3, 1x3, 1x4 et 1x5. Les adversaires tirent au tour à tour sur la grille ennemie, si un coup touche un des bateaux adverses, le joueur ayant effectué le tir peut rejouer.

- **Commandant** : Mode de jeu plus complet utilisant un système de factions, chacune d'entre elles possédant des bateaux et des compétences propres. Les joueurs ont chacun une barre d'énergie qui se remplit à chaque début de tour, permettant aux joueurs d'effectuer des actions plus stratégiques nécessitant une quantité plus ou moins importante d'énergie.

À l'issue du projet, seul le mode classique a été implémenté.

2.3.1.2 Configurer le temps par tour et par partie

L'utilisation et la configuration d'un timer est nécessaire pour définir la durée maximale d'une partie ainsi que la durée maximale d'un tour durant lequel un joueur doit effectuer une action. La durée de ces timers est configurée par l'utilisateur lors de la création de la partie. Concernant le temps maximal autorisé par tour, il doit être compris entre 15 et 60 secondes. Concernant le temps maximal autorisé par partie, il doit être compris entre 650 et 1200 secondes.

2.3.2 Rejoindre une partie

Lorsqu'un utilisateur décide de rejoindre une partie déjà créée, il reçoit une liste de toutes les parties actuellement en attente d'un second joueur. L'utilisateur n'a qu'à indiquer la partie qu'il souhaite rejoindre. Une fois la partie rejointe, elle se lance.

2.3.3 Regarder des parties (Mode Observateur)

Lorsqu'un utilisateur décide de regarder une partie, il reçoit une liste de toutes les parties actuellement en attente ou en cours. Similairement à la section précédente, l'utilisateur n'a qu'à choisir une partie. Une fois la partie rejointe, tous les coups déjà joués sont envoyés à l'observateur afin qu'il rattrape l'état actuel de la partie. Une fois le retard rattrapé, il sera notifié à chaque nouveau coup joué. Les observateurs ne voient pas les bateaux placés par les deux joueurs durant la partie pour éviter toute situation de triche.

2.3.4 Revoir une partie (Mode Replay)

Lorsqu'un utilisateur souhaite revoir une partie à laquelle il a participé, en tant que joueur ou en tant qu'observateur, il peut en faire la demande auquel cas, il recevra une liste de tous ses replays disponibles. Une fois une partie choisie, l'utilisateur n'a plus qu'à envoyer 'n' au serveur pour avancer dans la partie au coup par coup. S'il souhaite quitter le replay, il peut à n'importe quel moment envoyer 'q'.

2.3.5 Gérer ses invitations

Lorsqu'un utilisateur souhaite gérer ses invitations, il peut consulter qui l'invite à rejoindre une partie, que ce soit en tant que joueur ou en tant qu'observateur. Il peut alors choisir d'accepter ou de refuser l'invitation.

2.4 Commandes

Un système de commande a été créé pour permettre à l'utilisateur d'effectuer certaines actions spéciales. Toutes les commandes commencent par le caractère '/'. Il existe une commande */help* permettant d'afficher toutes les commandes disponibles.

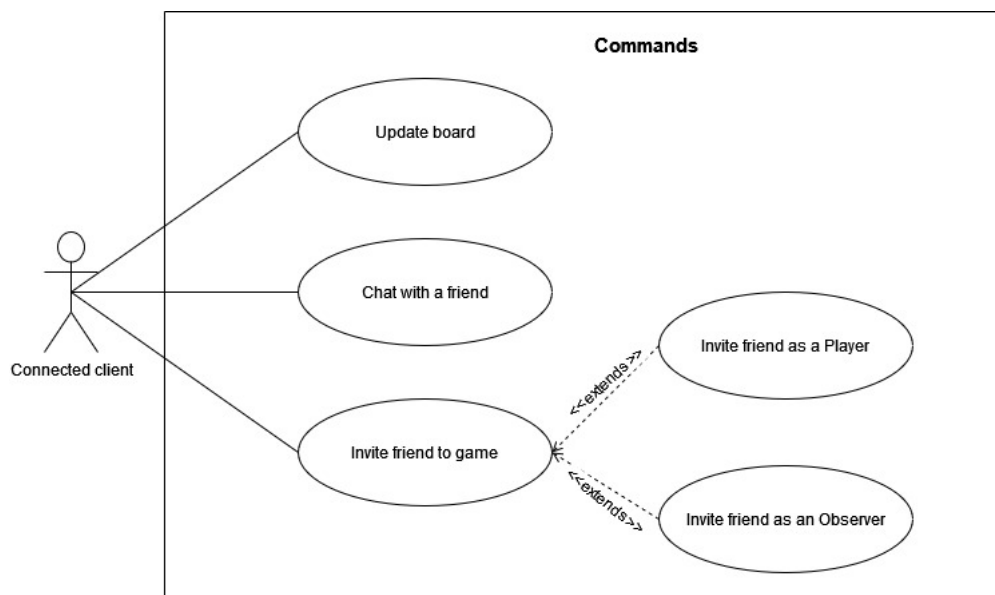


FIGURE 4 – Commandes disponibles

2.4.1 Chat

Il existe une commande pour envoyer un message à un ami à n'importe quel moment du programme. Il suffit d'utiliser la commande */mp "friend_name" "message"*. L'ami est notifié instantanément du message ainsi que de son expéditeur.

2.4.2 Inviter un ami à sa partie créée

Lorsqu'un joueur crée une partie, il a la possibilité d'inviter un de ses amis à rejoindre celle-ci. Il peut l'inviter en tant que joueur ou observateur en précisant respectivement '0' ou '1'. La commande est donc telle : `/inv "friend_name" "0/1"`. Une fois invité, le joueur peut accepter l'invitation dans le menu de jeu, dans la partie gestion des invitations.

2.4.3 Afficher les plateaux

Un joueur peut à tout moment, utiliser la commande `/u` pour afficher l'état actuel des plateaux de jeu.

2.5 Déroulement d'une partie

Lorsque deux utilisateurs ont rejoint le même lobby en tant que joueur, la partie est lancée. Les deux joueurs sont alors invités à placer leurs bateaux les uns à la suite des autres jusqu'à ce que leurs flottes soient entièrement composées. Pour placer un bateau, l'utilisateur doit entrer une position dans la grille suivie d'une orientation (verticale ou horizontale) (exemple "A1h"). La position choisie correspondra à la case la plus haute du bateau ou la plus à gauche en fonction de l'orientation choisie par l'utilisateur.

La grille est mise à jour avec les positions rentrées par le joueur. Le système attend que la flotte du premier joueur soit complète avant de demander au deuxième utilisateur de constituer la sienne. Une fois l'entière des bateaux des joueurs placés, la partie peut démarrer. Si un joueur dépasse le temps maximal autorisé par tour lors du placement d'un bateau, il perdra celui-ci. S'il ne parvient à placer aucun bateau, il perdra la partie.

Les joueurs doivent chacun effectuer des tirs à tour de rôle en indiquant la position de la case adverse qu'ils souhaitent attaquer. Un joueur dispose de sa propre grille remplie avec ses bateaux placés précédemment mais il possède également une deuxième grille, vide au départ, représentant la grille adverse. Après un tir, la case visée par le joueur sur la grille ennemie lui est dévoilée. Une case dévoilée peut prendre 3 formes différentes : une croix si la case attaquée par le joueur était un bout d'océan, une case grisâtre indiquant que le tir a touché une partie d'un bateau ennemi, et une case foncée indiquant qu'un bateau ennemi a été coulé (lorsque qu'une case de ce type est dévoilée, l'entière des anciennes cases dévoilées précédemment contenant le bateau changent également de couleur, laissant le bateau apparaître entièrement coulé).

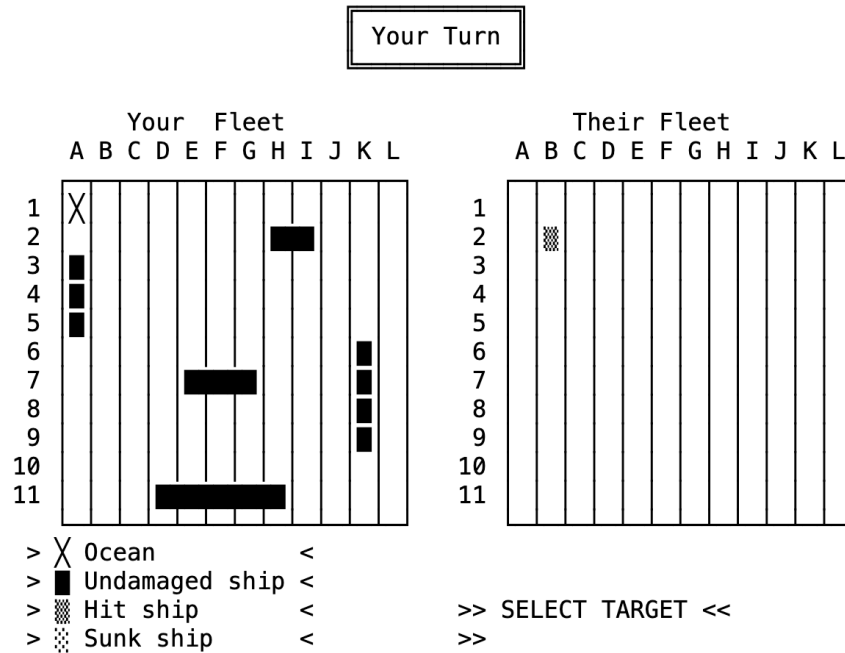


FIGURE 5 – Affichage du plateau durant une partie

Si un joueur touche un bateau ennemi, ce dernier obtient le droit de rejouer pour attaquer à nouveau, et ce jusqu'à ce qu'un de ses tirs finissent à l'eau. Si un joueur prend trop de temps à jouer un coup, son tour est passé.

La partie progresse et les échanges de tirs entre les deux joueurs s'enchaînent. La partie prend fin une fois que l'entièreté d'une des flottes est détruite ou lorsqu'un joueur n'a plus de temps disponible pour sa partie. Le joueur ayant coulé tous les bateaux ennemis est alors déclaré vainqueur et son adversaire perdant.

3 Besoins système

3.1 Base de données

Une base de données SQLite (stockée sur le serveur) est utilisée pour contenir les comptes des utilisateurs. Pour chaque joueur, la base de donnée possède un pseudo ainsi qu'un mot de passe associé, une liste d'amis ainsi qu'une liste des demandes d'amis en attente. La connexion ou création de compte utilisateur s'effectue via des requêtes SQL sur la base de donnée. Le programme serveur vérifie le bon format des informations avant un ajout dans la base de donnée.

3.2 Battleship

Cette section explique le déroulement d'un jeu à partir de son lancement que ce soit au niveau des communications réseau ou alors au niveau de la gestion des threads jusqu'à la fin d'une partie.

3.2.1 Système client-serveur

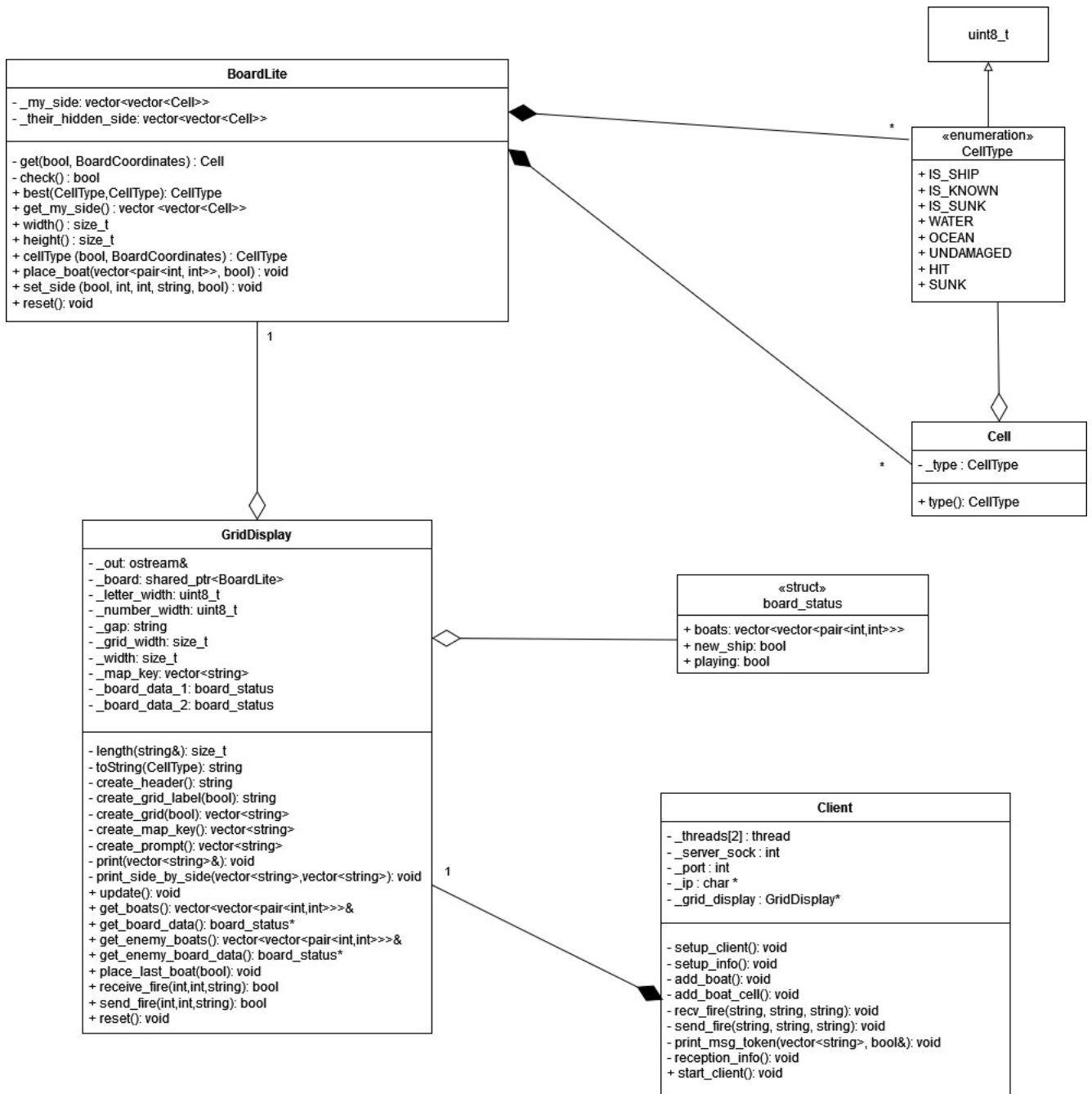


FIGURE 6 – Structure Client

qui l'a placé. Dans le cas d'un tir effectué, le résultat du tir est envoyé aux deux joueurs.

3.2.1.1 Système de communication

Un système de communication robuste et fiable était nécessaire pour ce projet. Il a donc été décidé d'employer un système de tokens pour cela. Un token est, dans le contexte de ce projet, une chaîne de caractère séparée par un caractère délimitant ("~" dans ce cas-ci). Le premier token correspond toujours à une action et les tokens qui suivent correspondent aux arguments de celle-ci. Pour plus d'informations, il est conseillé de se référer au README.

3.2.2 Threads

Le serveur possède tout un tas de threads qu'il utilise pour gérer l'ensemble du jeu. D'abord, il possède un thread dont la mission est de constamment attendre la connexion de potentiels clients (`handle_connection()`). Ce thread s'assure qu'aucun client ne soit jamais en attente et puisse toujours se connecter au serveur. Ensuite, un second thread (`handle_clients()`) vérifie en permanence si un nouveau client s'est connecté, si c'est le cas il lui assigne un thread d'écoute (`handle_client()`). Le serveur assigne donc un thread d'écoute pour chaque client, ce qui lui permet de constamment être à l'affût de requête du client sans délai et sans devoir attendre qu'un autre client termine une action. Enfin lors d'une partie, le serveur demande et reçoit des inputs de plusieurs clients. Pour ne pas créer de délai d'attente entre plusieurs parties et afin de pouvoir jouer concurremment, un thread est créé pour gérer chaque partie (`play_game()`).

Chaque client quant à lui possède deux threads. Un thread lui permettant de constamment recevoir des messages du serveur (`reception_info()`) et un second grâce auquel il peut faire des demandes au serveur en permanence (`send_info()`).

3.2.3 Affichage

L'application se doit d'être fonctionnelle aussi bien sur un terminal que via l'utilisation d'une interface graphique. Il est donc important de séparer la partie logique de la partie affichage pour permettre de facilement passer d'un affichage à l'autre. C'est pour cela que seul le client gère l'affichage de son plateau.

3.2.3.1 Terminal

Deux classes sont utilisées pour gérer l'affichage : *BoardLite* et *GridDisplay*. *BoardLite* est une classe qui permet de stocker l'état actuel du jeu et *GridDisplay* permet d'afficher les deux plateaux.

3.2.3.2 Interface Graphique (GUI)

Les interfaces graphiques assurent un confort pour les utilisateurs. Par exemple, dans la gestion des entrées, il est beaucoup plus intuitif d'utiliser un périphérique externe comme une souris ou un trackpad pour placer des bateaux ou alors effectuer des tirs.

Mallheureusement, cette partie n'a pas pu être développée.

3.2.4 Triche

Pendant une partie, les entrées des utilisateurs sont transmises au serveur pour validation avant d'être renvoyées aux joueurs, afin d'éviter toute possibilité de triche. De plus, le client ne dispose pas de la capacité de modifier le courant du jeu étant donné que celui-ci se trouve dans le serveur. Comme cité dans la section concernant le mode observateur (2.3.3), un spectateur ne voit pas les bateaux lorsqu'il observe une partie pour éviter qu'il communique l'information à un des joueurs.