
Software Requirements Document

Coia Pascal

Lo Cascio Rosario

Malouch Nikita

Mekhiouba Islam

Goossens Joël

Mertens Bryan

Hernalsteen Christian

Sanchez Espinosa Carlos

Renard Simon

Speilers Capucine

Truong Nha

Veyret Danae

Table des matières

1	Introduction	1
1.1	But	1
1.2	Glossaire	2
1.3	Historique	3
2	Fonctionnalités	4
2.1	Écran de connexion	4
2.1.1	Création de compte (Register)	4
2.1.2	Connexion(Login)	4
2.1.3	Fonctionnement du système	5
2.2	Menu principal	5
2.2.1	Accéder à la liste d'amis	5
2.2.2	Accès au jeu	5
2.2.3	Se déconnecter	5
2.2.4	GUI	6
2.3	Gestion de la liste d'amis	6
2.3.1	Envoyer une demande d'ami	6
2.3.2	Accepter une demande d'ami	6
2.3.3	Refuser une demande d'ami	6
2.3.4	Retirer un ami	7
2.3.5	Conversation	7
2.3.6	Accepter une invitation à jouer	8
2.3.7	Fonctionnement du système	8
2.3.8	Menu - Demande d'amis reçues[GUI]	8
2.3.9	Menu - Demandes d'amis envoyées [GUI]	8
2.4	Navigateur de Partie	9
2.4.1	Créer sa partie	9
2.4.2	Rejoindre une partie déjà existante	9
2.5	Lobby de partie	9
2.5.1	En tant qu'hôte	10
2.5.2	En tant que participant	10
2.5.3	Pour aller plus loin	10
2.6	Déroulement d'une partie	11
2.6.1	Sélection de faction	11
2.6.2	Phase de placement	11
2.6.3	Phase de jeu	11

2.6.4	Mode classique	13
2.6.5	Mode commandant	13
2.7	Serveur	13
2.7.1	Démarrage	13
2.7.2	Connexion	14
2.7.3	Communication	14

3 Fonctionnement du système 15

3.1	Côté serveur	15
3.2	Côté jeu	16

Ce document regroupe les informations nécessaires pour la réalisation du jeu en ligne **Touché-Coulé**, un jeu de plateau tour par tour opposant deux joueurs, dont l'objectif est de détruire l'entièreté de la flotte adverse.

Une partie démarre lorsque chaque joueur a placé ses différents navires sur une grille de 10x10 visible uniquement par lui-même. Chaque joueur choisit une case adverse à attaquer et est informé s'il a touché un navire ennemi, n'a rien atteint, ou l'a coulé (dès que l'intégralité de celui-ci a été touché). La partie prend fin dès qu'un joueur parvient à couler toute la flotte adverse.

De plus, il existe deux modes de jeu : le mode **Classique**, où il est interdit de placer des bateaux adjacents les uns aux autres, et le mode **Commandant**, où tout placement est autorisé et des points d'énergie procurent des avantages aux joueurs.

1.1 But

L'objectif du projet est de permettre à deux personnes de jouer à une version en ligne du jeu **Touché-Coulé**. Lorsqu'un joueur crée une partie, il est capable de configurer les paramètres suivants :

1. le mode de jeu,
2. le temps maximal autorisé par tour,
3. la durée maximale d'une partie.

Tout joueur est capable de :

- créer un compte associé à un pseudonyme et un mot de passe,
- se connecter à son compte (*nom d'utilisateur, mot de passe*),
- gérer sa liste d'amis,
- créer et configurer sa propre partie,
- rejoindre une partie déjà créée (avec ou sans mot de passe),
- inviter un ou plusieurs amis à sa partie créée,
- discuter avec ses amis,
- jouer au jeu Touché-Coulé en suivant les règles du jeu sans avoir la possibilité de tricher.

Finalement, le projet propose deux versions du jeu. La première, une version terminal qui, comme son nom l'indique, permet de jouer depuis le terminal, ainsi qu'une seconde version, graphique, permettant de jouer au travers d'une interface graphique (GUI).

1.2 Glossaire

- **Adresse IP** : Sert à identifier les périphériques d'un même réseau informatique.
- **Board** : Représente l'état d'une partie en cours. Un board a deux côtés, un par joueur. Le côté d'un joueur est celui qui contient ses navires.
- **Display** : Affichage. C'est ce qui est appelée "Vue" dans le modèle MVC.
- **GUI - Graphical User Interface** Interface permettant l'affichage graphique d'une application.
- **Plateaux** Dans le cadre de ce projet, un "plateau" est uniquement représenté par les coordonnées des bateaux ainsi que des coups tirés lors de la partie. Il n'est donc pas question d'une matrice représentant chaque case du plateau, mais bien uniquement d'un ensemble d'objets présents sur ce plateau et de leur coordonnées
- **Plateau d'affichage** Plateau tenu par le client, sert à l'affichage de la partie. Chaque joueur tient deux plateaux d'affichage, le sien et celui de l'adversaire (face cachée).
- **Plateau de référence** Plateau tenu par le serveur, sert de référence pour vérifier la validité de différents placement (bateaux, attaques) lors de la partie.
- **Protocole TCP** : Un protocole de transmission de données entre deux ordinateurs.
- **Pseudonyme** : Nom désignant un utilisateur.
- **Timestamp** : Façon de traquer le temps comme le nombre de secondes écoulé depuis un certain événement.
- **View** : Classe ou interface pour accéder à des informations (comme un `std::string_view`). Une classe / interface "View" n'aura en général que des méthodes constantes comme des getters. Une view peut, soit simplement retourner des informations qui ne doivent être accédées qu'en lecture, soit calculer des informations à la volée.

1.3 Historique

Version	Modification	Auteur	Date
0.1	Ajout de l'introduction et du but	Tous	25/11/2023
0.2	Ajout des diagrammes initiaux	Tous	04/12/2023
0.3	Mise à jour des diagrammes et ajout texte	Pascal, Nha, Danae	10/12/2023
0.4	Mise à jour des diagrammes	Pascal, Nha, Danae	11/12/2023
0.5	Mise à jour des diagrammes et ajout texte des diagrammes de classe	Nha, Danae	13/12/2023
0.6	Suppression du nom de boucles dans les diagrammes de séquence et update du diagramme 2.9	Nha, Danae	14/12/2023
0.7	Correction de quelques coquilles	Nha	15/12/2023
1.0	Ajout description base de données	Danae	29/02/2024
1.1	Modifications des use cases + correction orthographique et reformulation base de données	Nha	02/03/2024
1.2	Modification diagramme de classe serveur	Pascal	03/03/2024
1.3	Restructuration du SRD et écriture serveur-client	Pascal, Nha, Danae	14/03/2024
1.4	Écriture serveur-client	Capucine	14/03/2024
1.5	Détails à propos des uses cases et diagrammes de classe	Pascal, Nha	15/03/2024
1.6	Ajout diagramme de classe Game-Server-CD et update descriptions et autres CD	Danae	16/03/2024
1.7	Fin écriture des fonctionnalités et ajouts glossaire	Pascal, Nha	16/03/2024
2.0	Mise à jour diagramme de séquence login	Nha	22/03/2024
2.1	Correction conversation	Nha	25/03/2024
2.2	Modification chat	Nha	18/04/2024
3.0	Révision du SRD (1)	Capucine	18/04/2024
3.1	Révision du SRD (2)	Capucine	19/04/2024
3.2	Révision du SRD (3), diagrammes d'usage	Capucine	24/04/2024
3.3	Ajout des dernières fonctionnalités implémentées, diagrammes d'usage	Capucine	25/04/2024
3.4	Adaptations dues aux changements, diagrammes de séquence	Capucine	26/04/2024
3.5	Ajout de précision pour le placement en mode gui	Pascal	26/04/2024

Remarque préliminaire : Certains cas d'usage diffèrent entre mode console et mode interface graphique (GUI). Le premier mode sera représenté en bleu, le second en jaune. Si un diagramme ne diffère pas, il sera représenté en blanc.

2.1 Écran de connexion

Le premier menu à apparaître est l'écran de connexion, illustré à la figure 2.1, offrant la possibilité à l'utilisateur de créer un compte s'il n'en possède pas encore ou de se connecter s'il en a déjà un. L'utilisateur reste sur cet écran tant qu'il ne parvient pas à se connecter ou à s'inscrire.

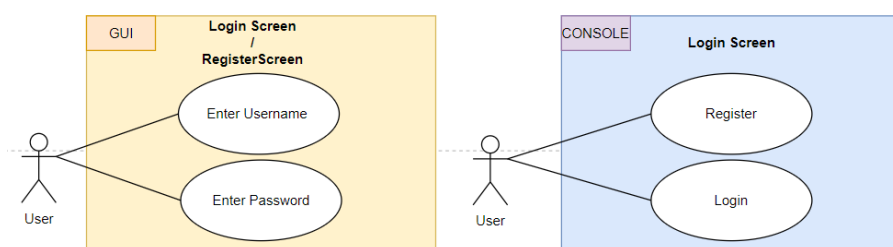


FIGURE 2.1: Écran de connexion

2.1.1 Création de compte (Register)

Lorsqu'un utilisateur souhaite créer un compte, le serveur attend un pseudonyme et un mot de passe, puis procède à une vérification dans la base de données afin de déterminer si ce nom d'utilisateur est déjà utilisé. Si c'est le cas, le compte n'est pas créé. Une fois qu'un nom unique et un mot de passe sont fournis par l'utilisateur, le compte est créé et enregistré dans la base de données. L'utilisateur est connecté et a désormais accès au menu principal.

2.1.2 Connexion(Login)

Pour se connecter, l'utilisateur doit fournir ses informations d'identification, à savoir son pseudonyme et son mot de passe. Le serveur effectue ensuite une vérification dans la base de données afin de déterminer si les informations données correspondent à l'un des comptes enregistrés. Si aucun pseudonyme ne correspond ou si le mot de passe n'est pas le même que celui associé au pseudonyme fourni, la connexion échoue. Si les informations sont valides, l'authentification réussit et l'utilisateur accède au menu principal.

En cas d'oubli du pseudonyme ou du mot de passe, aucune option de récupération n'est disponible. Dans ce cas, l'utilisateur doit créer un nouveau compte s'il souhaite continuer à jouer au jeu.

2.1.3 Fonctionnement du système

Les explications précédentes concernant les différents scénarios envisageables pour les étapes de création de compte et de connexion sont illustrées dans la figure 2.2.

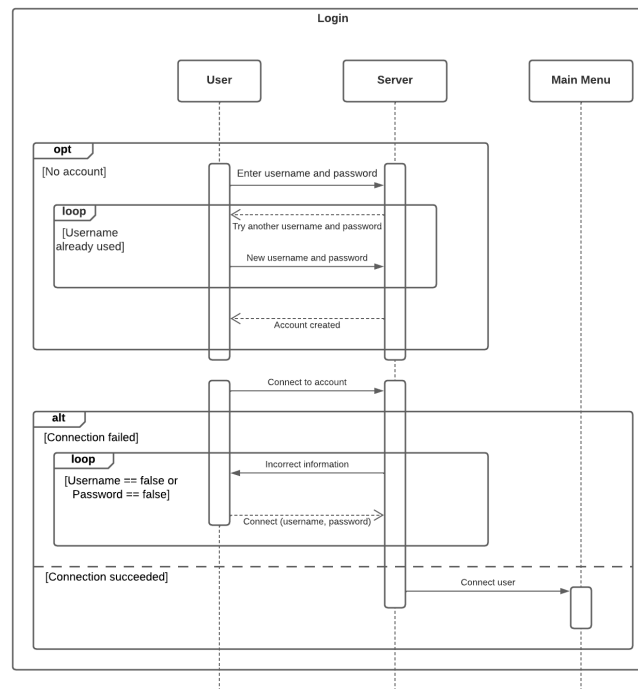


FIGURE 2.2: Connexion

2.2 Menu principal

La figure 2.3 montre l'écran d'accueil vers lequel le joueur est dirigé après s'être connecté. A partir de cet écran, il a la possibilité de consulter et gérer sa liste d'amis, accéder au menu de jeu ou se déconnecter.

2.2.1 Accéder à la liste d'amis

Remarque : Cette option n'existe qu'en mode terminal, l'accès aux options d'amis en mode GUI est décrit plus loin dans cette section.

L'utilisateur peut accéder au menu dédié, détaillé au point 2.3, lui permettant de gérer sa liste d'amis.

2.2.2 Accès au jeu

L'utilisateur peut accéder au navigateur de partie (browser), afin de lancer ou rejoindre une partie de **Touché-Coulé**. Ce menu est détaillé en section 2.4.

2.2.3 Se déconnecter

L'utilisateur peut choisir de se déconnecter, ce qui le ramènera à l'écran de connexion.

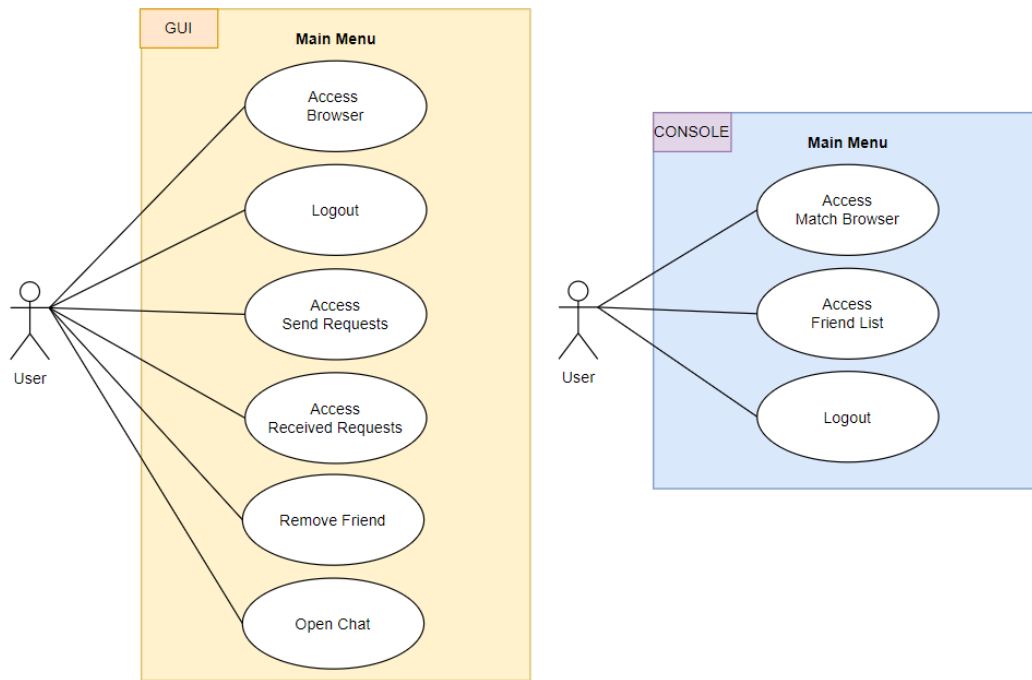


FIGURE 2.3: Menu principal

2.2.4 GUI

En mode graphique, l'utilisateur peut ouvrir une conversation avec un ami directement depuis le menu principal. De même, il peut directement retirer un utilisateur de sa liste d'amis. La gestion des requêtes se fait depuis des menus uniques au GUI (sections 2.3.8 et 2.3.9).

2.3 Gestion de la liste d'amis

Concernant la gestion de la liste d'amis, un utilisateur peut envoyer des demandes d'ami, accepter ou refuser les demandes reçues, supprimer des amis de sa liste, ou débiter une conversation avec eux. En mode terminal, cela se fait depuis le menu dédié tandis qu'en mode GUI, une partie des actions est gérée directement depuis le menu principal. Le reste des actions est géré dans les menus dédiés, décrit en sous-section 2.3.8 et 2.3.9.

2.3.1 Envoyer une demande d'ami

Quand un utilisateur souhaite envoyer une demande d'ami, il doit fournir le pseudonyme de l'utilisateur à qui il souhaite envoyer la requête. Ensuite, le serveur vérifie dans la base de données si cet utilisateur existe et s'ils ne sont pas déjà amis. Si c'est le cas, la demande est envoyée, sinon elle est rejetée.

2.3.2 Accepter une demande d'ami

Lorsque l'utilisateur reçoit une demande d'ami, il peut l'accepter. Cela ajoute alors la personne ayant envoyé la demande à sa liste d'amis. De même, celui qui a envoyé la demande voit également l'utilisateur ajouté à sa liste d'amis. En effet, deux utilisateurs deviennent amis dès lors que la demande d'ami envoyée est acceptée par l'utilisateur qui l'a reçue.

2.3.3 Refuser une demande d'ami

L'utilisateur a également le choix de refuser une demande d'ami, ce qui la supprime.

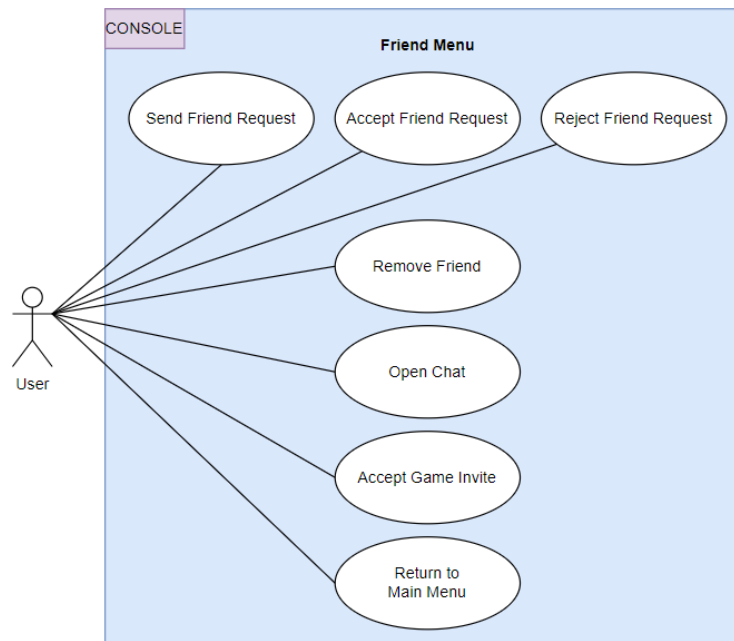


FIGURE 2.4: Gestion de la liste d'amis

2.3.4 Retirer un ami

L'utilisateur a la possibilité de supprimer un ami de sa liste, ce qui entraîne également sa suppression de la liste de l'autre utilisateur. Le serveur vérifie l'existence du nom d'utilisateur donné dans la base de données et s'assure qu'il figure effectivement dans la liste d'amis, et retire ensuite l'utilisateur des deux listes.

2.3.5 Conversation

L'utilisateur peut décider de discuter par message avec ses amis. Pour chaque conversation distincte entre deux personnes, un fichier texte est créé et le timestamp (horodatage) est enregistré à la première ligne. Si le timestamp excède les 30 jours, les conversations entre utilisateurs sont automatiquement supprimées par le serveur.

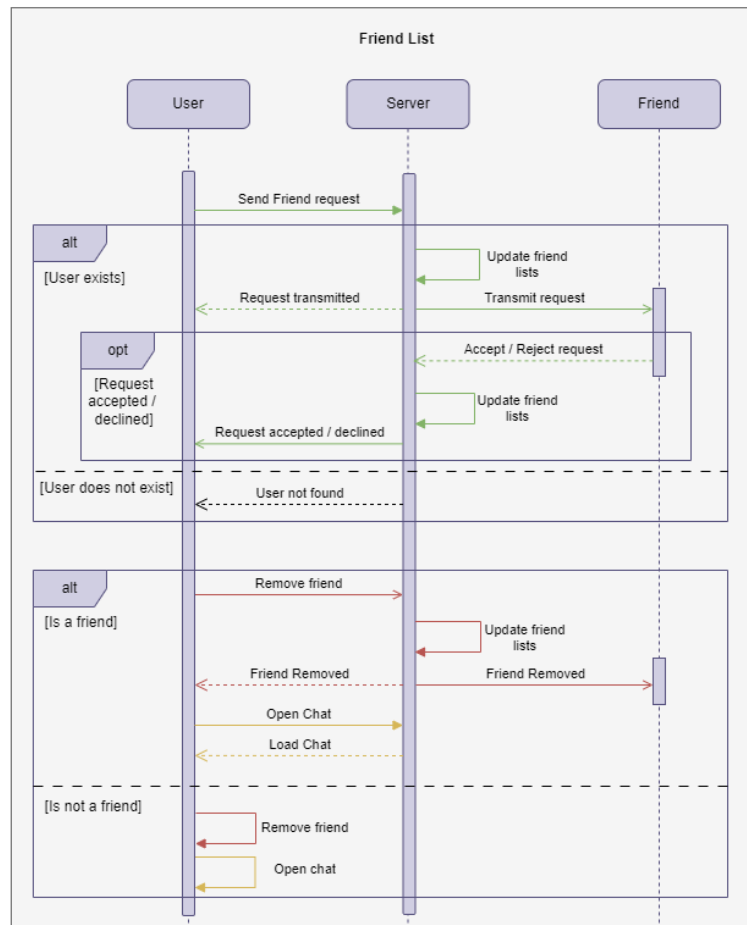


FIGURE 2.5: Gestion de la liste d'amis

2.3.6 Accepter une invitation à jouer

Afin de rejoindre la partie d'un ami, l'ami en question doit avoir envoyé une invitation, comme expliqué dans le point 2.5.3. L'utilisateur ayant été invité peut donc accepter ces demandes depuis le menu ami. Dans ce cas, le serveur traitera la requête en cherchant le lobby appartenant à l'ami proposant l'invitation. Ensuite le serveur déclenche lui-même une requête de join avec les informations nécessaires.

2.3.7 Fonctionnement du système

La figure 2.5 présente les différentes options disponibles dans le menu de gestion de la liste d'amis de l'utilisateur, ainsi que la manière dont le serveur les gère.

2.3.8 Menu - Demande d'amis reçues [GUI]

Ce menu n'apparaît qu'en mode GUI. Il permet de gérer les demandes d'ami reçues par l'utilisateur. Depuis ce menu il peut donc les consulter, les accepter ou les refuser.

2.3.9 Menu - Demandes d'amis envoyées [GUI]

Ce menu exclusif au mode GUI permet à l'utilisateur d'envoyer des demandes d'amis ainsi que de voir les demandes d'amis déjà envoyées.

Remarque : Les deux menus précédents ne sont que des interfaces, leur fonctionnement avec le serveur est exactement le même que celui décrit par le figure 2.5.

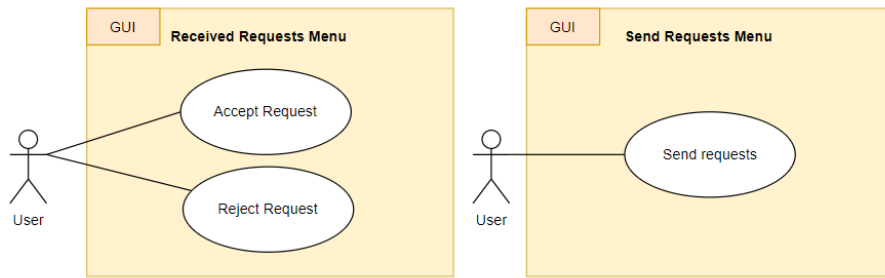


FIGURE 2.6: Menu - Demande d'amis reçues et envoyées [GUI]

2.4 Navigateur de Partie

Depuis ce menu, l'utilisateur a la possibilité de voir la liste des parties en cours, de créer sa propre partie ou d'en rejoindre une déjà existante.

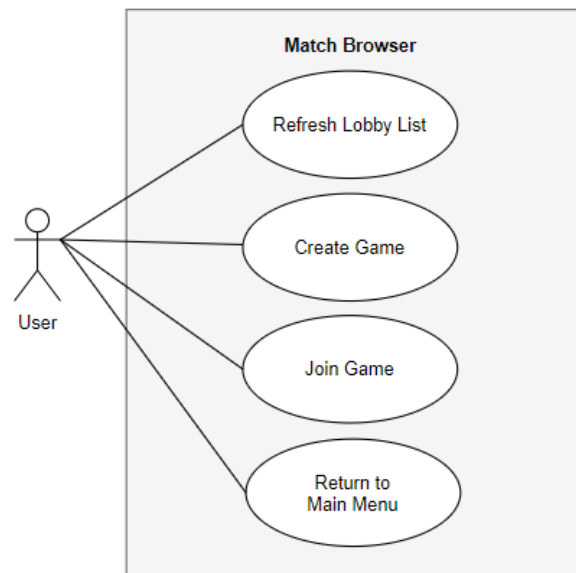


FIGURE 2.7: Navigateur

2.4.1 Créer sa partie

S'il souhaite être l'hôte de la partie, l'utilisateur peut créer sa propre session en lui donnant un nom. De plus, il peut éventuellement décider d'un mot de passe afin d'en restreindre l'accès. Une fois la partie créée, l'utilisateur est redirigé vers le lobby de partie, détaillé en section 2.5.

2.4.2 Rejoindre une partie déjà existante

L'utilisateur peut rejoindre la partie de son choix en entrant le nom de la partie choisie ainsi que, le cas échéant, le mot de passe. Si la partie existe, que le mot de passe (s'il y en a un) est valide et que la partie n'est pas en cours, l'utilisateur est redirigé vers le lobby de la partie, détaillé en section 2.5.

2.5 Lobby de partie

Ce menu est dédié au paramétrage de la partie par son hôte. Il permet également à l'hôte d'inviter ses amis ainsi qu'à chaque utilisateur l'ayant rejointe de choisir son rôle dans la partie (joueur ou spectateur).

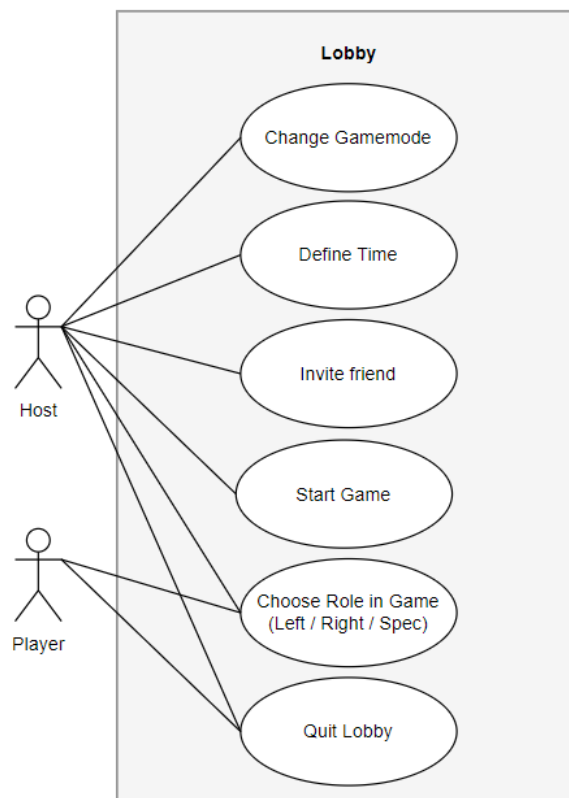


FIGURE 2.8: Lobby

2.5.1 En tant qu'hôte

A travers différentes commandes, l'hôte peut modifier les paramètres de partie. Il peut définir le mode de jeu, déterminer la durée maximale soit d'un tour soit de la partie et décider de commencer la partie en confirmant les paramètres choisis. De plus, l'hôte a la possibilité d'inviter ses amis à rejoindre sa partie. À ces privilèges s'ajoutent les actions de base à savoir choisir son rôle dans la partie et pouvoir quitter la session.

Une fois les paramètres de partie confirmés et si deux joueurs sont présents, un de chaque côté du plateau, la partie se lance. Pour un aperçu du déroulement d'une partie, voir le point 2.6.

2.5.2 En tant que participant

Les participants n'ont aucun droit quant au paramétrage de la partie, ils ont uniquement la possibilité de choisir leur rôle (joueur ou spectateur) ainsi que la possibilité de quitter le lobby.

2.5.3 Pour aller plus loin

À chaque modification du paramétrage, si le paramètre demandé est valide, le serveur est informé de la modification. Le serveur en informe alors tous les participants, chaque joueur a alors une version mise à jour de paramètres.

Pour ce qui est de l'envoi d'invitation à rejoindre une partie, le serveur ajoute cette demande dans la liste dédiée de l'invité. L'expéditeur quant à lui, n'a aucune trace la demande faite.

Enfin, quand un joueur change de rôle, le serveur vérifie que le rôle est libre. Si c'est le cas l'utilisateur est déplacé et les autres participants en sont informés. À nouveau, chaque participant a la version mise à jour du lobby.

Finalement, quand l'hôte est prêt à lancer la partie, le serveur vérifie qu'il y ait bien un joueur de chaque côté du plateau, après quoi le serveur lance la partie dans le lobby et informe chaque joueur présent que la partie commence. Les participants passent tous alors en mode 'en jeu', ce qui déclenche l'affichage de partie.

2.6 Déroulement d'une partie

2.6.1 Sélection de faction

Les détails de la partie sont décidés par l'hôte dans le lobby (section 2.5). Si le mode Commandant a été choisi, les joueurs commencent par choisir leur faction parmi celle disponibles. Une fois qu'un joueur a validé sa faction, il est invité à placer ses navires sur son plateau. Si le mode Classique qui a été choisi, alors les joueurs accèdent directement à la phase de placement.

2.6.2 Phase de placement

Remarque : En mode classique il n'est pas possible de placer un bateau directement à côté d'un autre cependant en mode commandant cela est tout à fait possible.

Chaque joueur doit placer tous ses navires. Tant qu'il reste des bateaux dans la flotte, la phase continue. Le joueur doit d'abord sélectionner le bateau à placer, ensuite il peut ajuster sa position sur la grille en le déplaçant et en le tournant. Une fois satisfait du placement du bateau, le joueur confirme son choix. Le contrôleur vérifie d'abord la validité du placement et si c'est bon, acte le choix sur le plateau de référence (tenu par le serveur).

Une fois toute la flotte placée, le joueur est mis en attente si l'adversaire n'a pas encore fini la phase de placement. Ensuite la partie commence.

En mode gui pour placer ses bateaux ils faut les sélectionner en cliquant sur le bouton correspondant à celui souhaité, choisir la position en cliquant sur son plateau, pour faire tourner le bateau il faut utiliser les touches "a" et "e", respectivement pour tourner vers la gauche et vers la droite, et enfin pour confirmer il faut appuyer sur la touche enter du clavier.

2.6.3 Phase de jeu

En début de tour, la joueur sélectionne la capacité à utiliser, la positionne sur la grille adverse (qui lui apparaît cachée) et valide son coup. Si l'entrée est correcte, le contrôleur vérifie avec le plateau de référence le résultat du coup, et met à jour le plateau d'affichage. Le contrôleur communique le résultat du coup au joueur et met à jour le plateau d'affichage. L'adversaire reçoit également les résultats des attaques subies. Tant que l'attaque touche un bateau, le joueur continue de jouer. Si elle tombe à l'eau, l'adversaire prend la main. La partie continue tant que le jeu n'est pas terminé et qu'un joueur n'a pas gagné.

Remarque : La phase de jeu décrite précédemment correspond au mode de jeu classique. En mode commandant la notion d'énergie est introduite. Les joueurs ont accès à de nouvelles capacités mais doivent désormais tenir compte du coût de celles-ci.

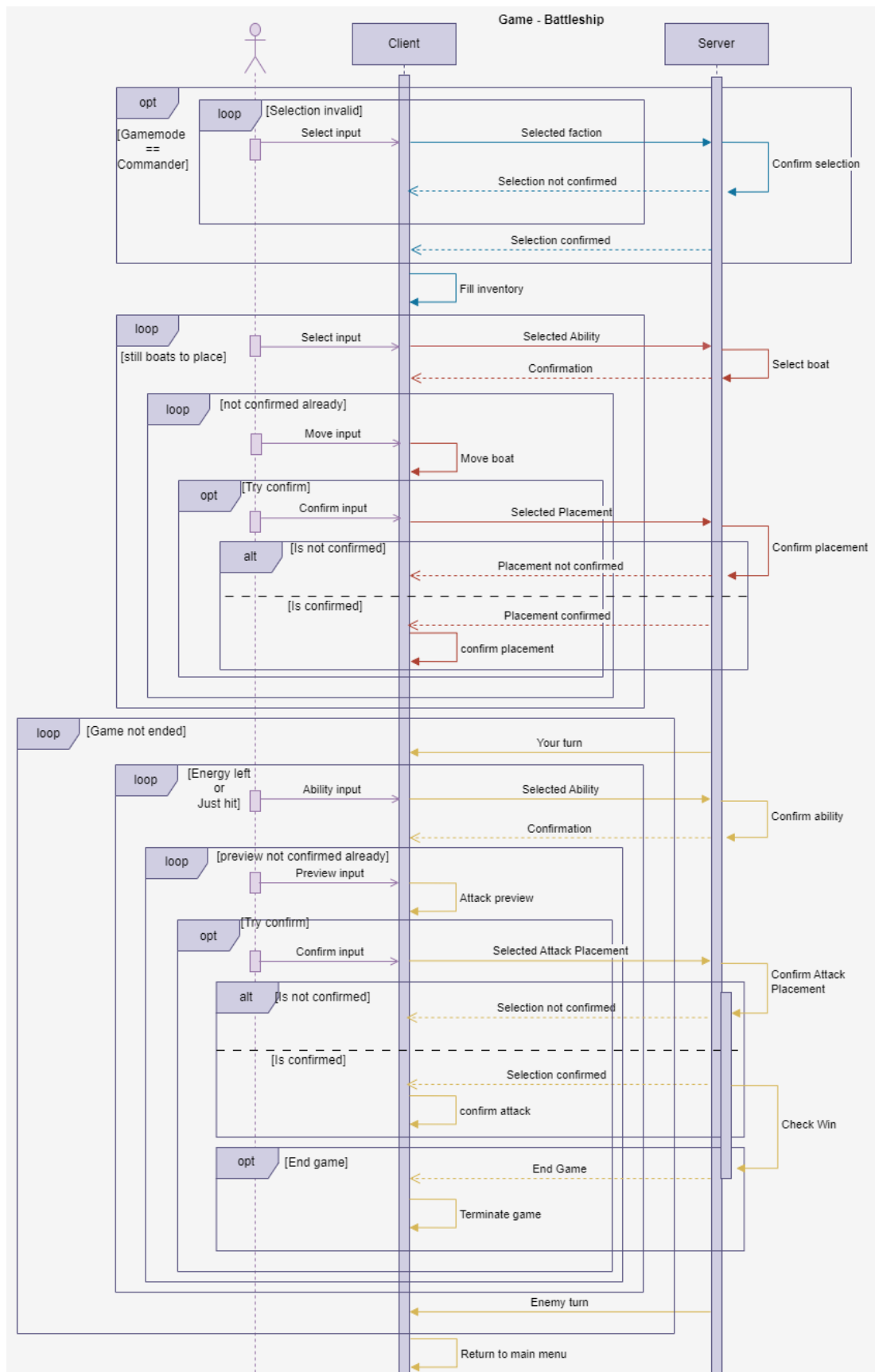


FIGURE 2.9: Déroulement d'une partie

2.6.4 Mode classique

Les bateaux ne peuvent pas être positionnés sur des cases adjacentes. Pendant son tour, le joueur n'a accès qu'à un type de coup et tant que son tir résulte en "touché", il peut continuer d'attaquer.

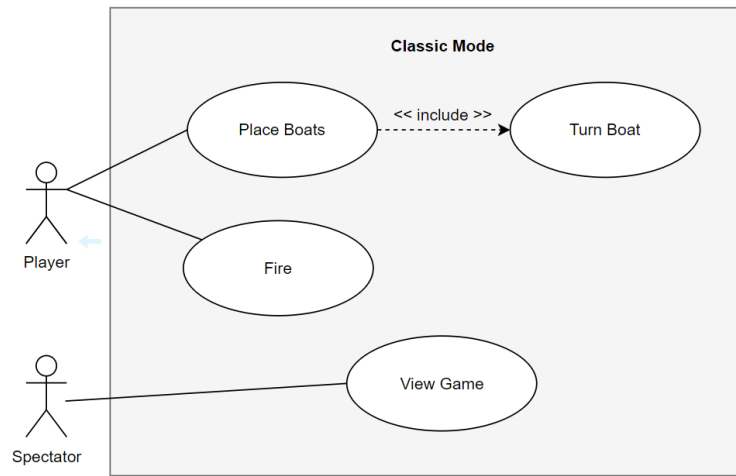


FIGURE 2.10: Partie en mode classique

2.6.5 Mode commandant

Les joueurs choisissent une faction au début de la partie. Leurs bateaux peuvent être adjacents. Pendant son tour, le joueur a la possibilité d'attaquer avec le tir par défaut mais aussi avec des pouvoirs distincts selon la faction choisie. Chaque pouvoir a un coût en énergie, si le joueur qui à la main tombe à cours d'énergie, c'est au tour de l'adversaire.

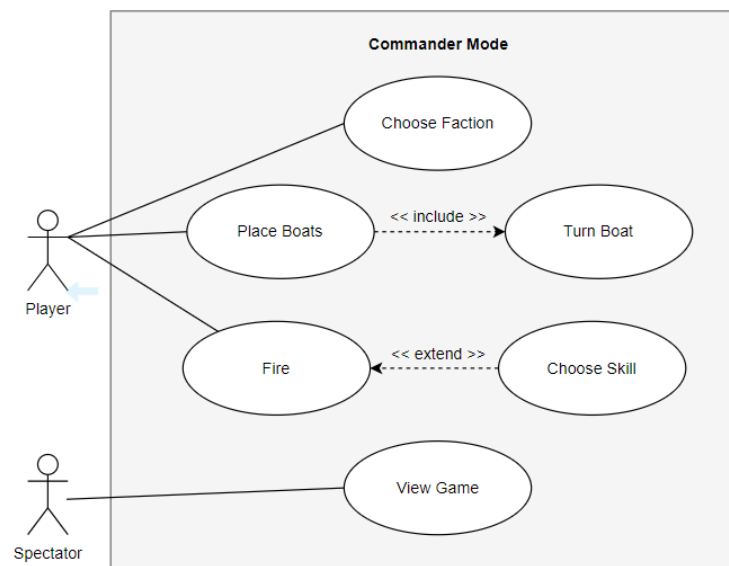


FIGURE 2.11: Partie en mode commandant

2.7 Serveur

2.7.1 Démarrage

Au lancement, en plus de préparer et d'établir la connexion, le serveur ouvre également la base de données. Étant donné que ce projet ne devrait pas requérir une base de données de grande taille, nous estimons

qu'il serait plus efficace de l'ouvrir une seule fois au lancement plutôt que de le faire à la demande. En effet, le faible volume de données, telles que les informations de compte, ne devrait pas poser de problème mémoire. De plus, lorsque la base de données est fermée, le fichier texte est mis à jour pour sauvegarder tout le contenu.

2.7.2 Connexion

La connexion s'établit selon le protocole TCP. Le client a la possibilité d'entrer l'adresse IP de son choix afin d'accéder à un serveur distant. Sinon, elle est définie par défaut à "127.0.0.1".

A la connexion, l'utilisateur commence à l'état initial "LOGIN", il peut ensuite évoluer vers les états "MENU" (en cas d'authentification réussie) et "PLAYING" (quand il commence à jouer). Cet état est partagé avec le serveur qui utilise des modules dédiés pour chaque état du client (menu et game). En outre, le client maintient un second état permettant de changer d'affichage entre les menus et donc de proposer les différentes actions.

2.7.3 Communication

Le serveur utilise la méthode **epoll** pour la gestion des événements. Cette décision découle de la capacité d'**epoll** à permettre au serveur de surveiller un grand nombre de descripteurs de fichiers simultanément et de manière non bloquante. Ainsi, cela améliore l'efficacité dans le traitement des requêtes émises par les utilisateurs. De plus, le système de messagerie établi au sein de notre réseau, conjugué aux trois états définis pour les clients (« LOGIN », « MENU » et « PLAYING »), permet au serveur de gérer efficacement toutes les demandes des clients sans nécessiter d'informations supplémentaires sur l'état des clients.

Fonctionnement du système

3.1 Côté serveur

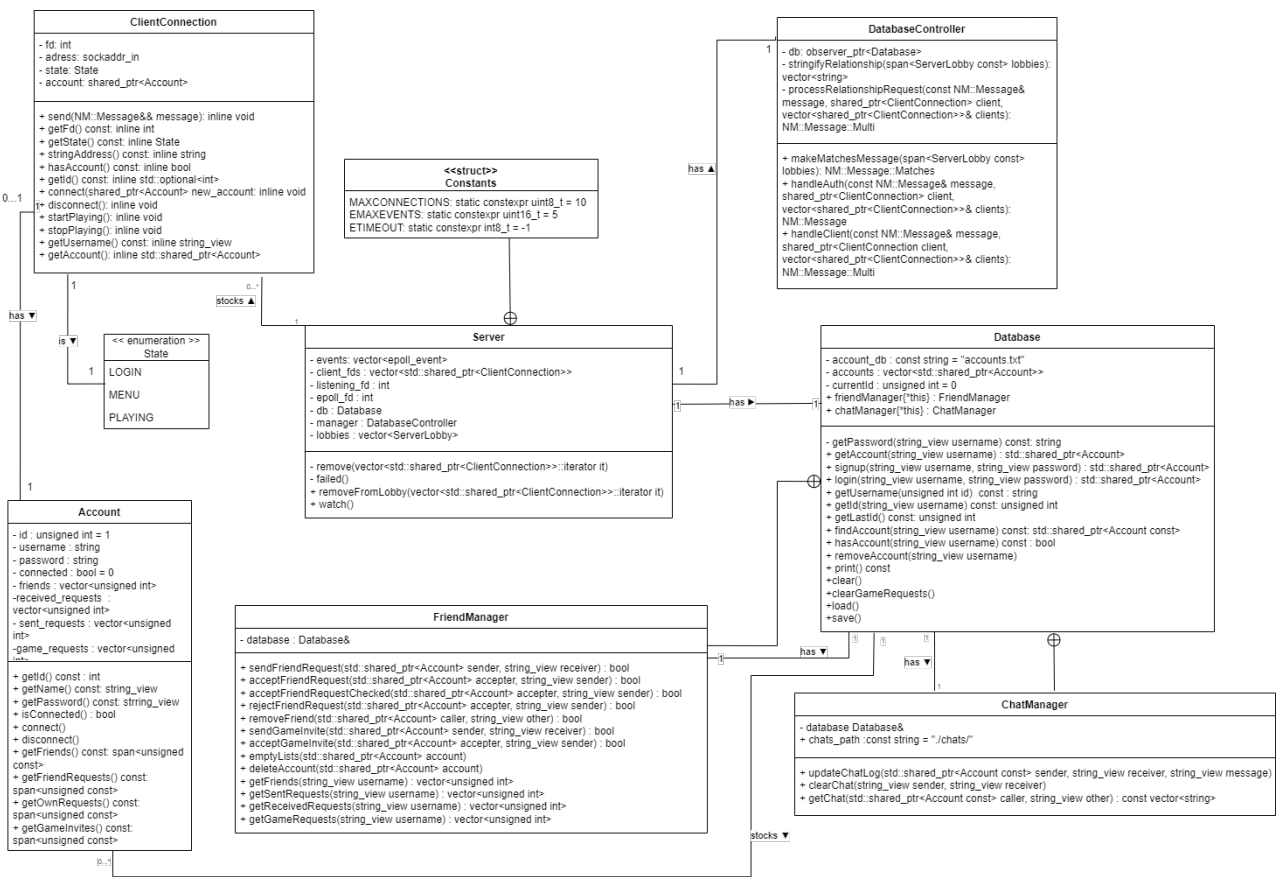


FIGURE 3.1: Côté serveur

Le diagramme à la figure 3.1 représente les relations entre les classes constituant l'infrastructure du serveur. La classe `Server` interagit avec une classe `Database` qui est responsable du stockage des comptes. La gestion des amis se fait via la classe `FriendManager`, tandis que la gestion des conversations entre deux utilisateurs est assurée par la classe `ChatManager`. Le client et serveur communiquent grâce aux changements d'état (`State`).

3.2 Côté jeu

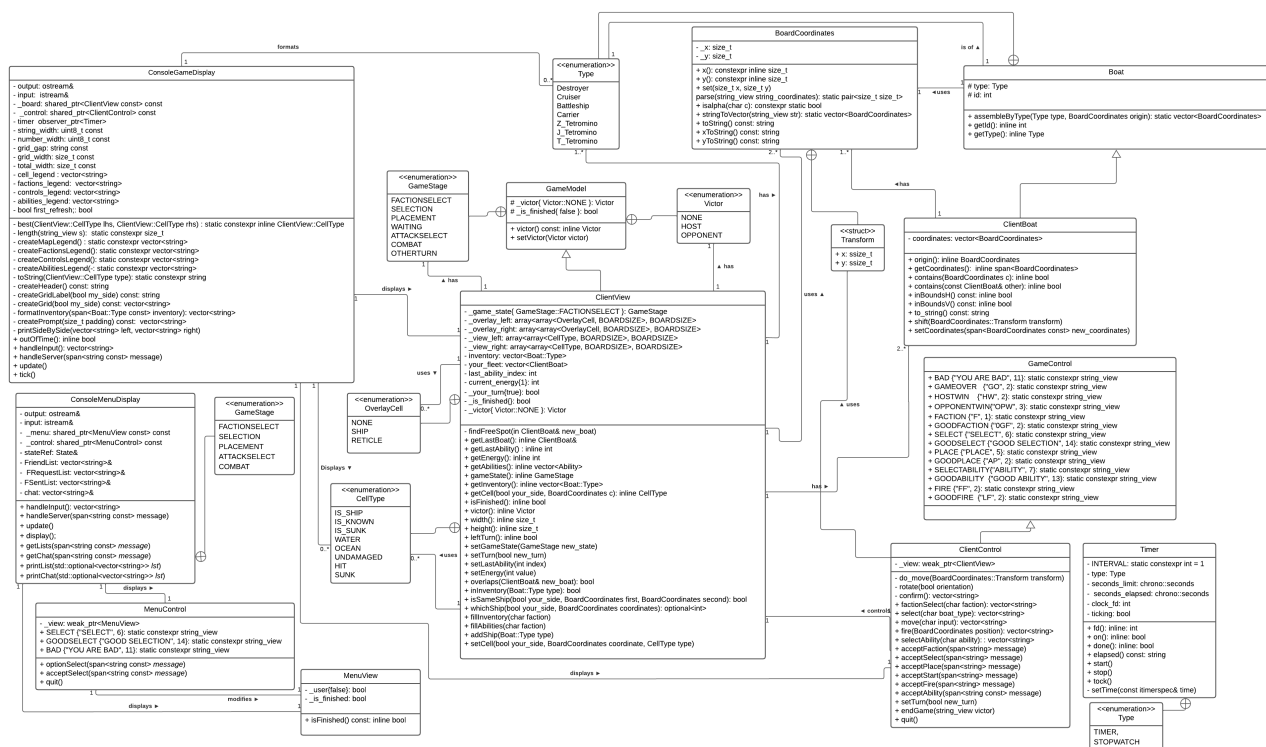


FIGURE 3.2: Côté jeu-client

Le diagramme présenté à la figure 3.2 illustre les interactions entre les classes responsables du déroulement d’une partie du côté du client. La classe `ClientView` contient les données de la partie et interagit ainsi avec la classe `ClientBoat` qui contient les données des bateaux, lesquelles sont contrôlées par `ClientControl`. Pour l’affichage des éléments relatifs au jeu lui-même, tel que le plateau, la classe `BoardConsoleDisplay` est sollicitée. Enfin, les classes `GameMenu` et `MenuView` interagissent pour afficher le menu et gérer les données entrées, étant toutes deux illustrées par `ConsoleMenuDisplay`.

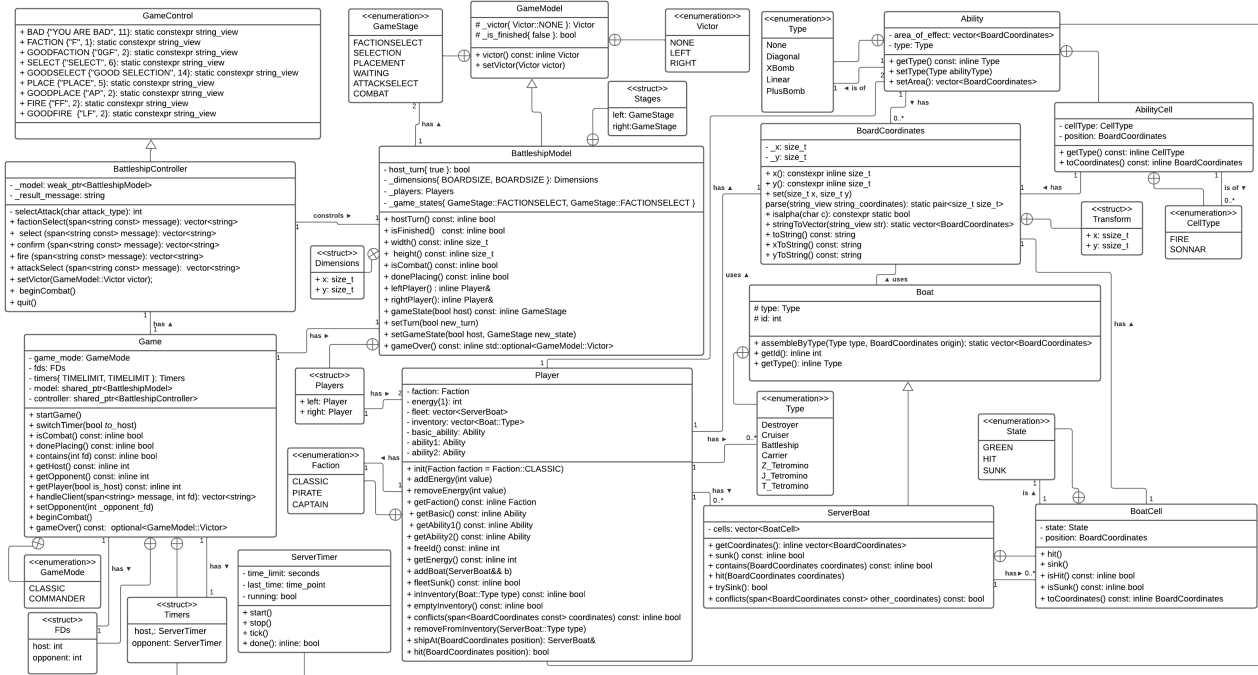


FIGURE 3.3: Côté jeu-serveur

Le diagramme à la figure 3.3 montre les interactions entre les classes impliquées dans le déroulement d'une partie du côté du serveur. La classe `BattleshipModel` agit comme modèle principal, contenant la majorité des données du jeu. Elle est contrôlée par `BattleshipController`, qui est à son tour contenu dans la classe `Game`, offrant ainsi des détails plus spécifiques sur la partie en cours. À l'intérieur de `BattleshipModel`, on retrouve deux instances de la classe `Player`, chacune possédant des instances différentes des classes `ServerBoat` et `Ability`, selon le mode de jeu et la faction.