

Tetris Royale : SRD
INFO-F—209 : Projet d’Informatique 2

Ali Umar Babar	Ayman Benaim	Antoine Berthion	Mamadou Barry
Oleksandra Omelyanyuk	Rares Bors	Roberto Rabei	Taha Es-saâdouni

8 mai 2025

Table des matières

1	Introduction	2
1.1	Objectif du SRD	2
1.2	Description du Projet	2
1.2.1	Résumé	2
1.2.2	Modes de jeu	2
1.2.3	Fonctionnalités multijoueurs	3
1.2.4	Fonctionnalités hors partie	3
1.3	Glossaire	3
1.4	Historique	3
2	Besoins Utilistateurs	5
2.1	Besoins Fonctionnels	5
2.1.1	Connexion et Création de Compte	5
2.1.2	Gestion des Parties et Profil Utilisateur	6
2.1.3	Déroulement d'une partie	7
2.2	Besoins non-fonctionnels	8
3	Besoins Systèmes	9
3.1	Connexion	9
3.2	Lobby	9
3.3	Partie	10
4	Design et Fonctionnement	11
4.1	Connexion et inscription	11
4.2	Gestion des comptes	12
4.3	Menu principal	12
4.4	Matchmaking	12
4.5	Chat	12
4.6	Jeu	12
4.7	Vue globale	13
4.7.1	Vue globale : MVC	13
4.7.2	Vue globale : Serveur	13
4.7.3	Vue globale : Base de données	14

Chapitre 1

Introduction

1.1 Objectif du SRD

Ce document de spécifications des besoins logiciels (*Software Requirements Document*, **SRD**) vise à décrire les exigences et les aspects clés du projet "Tetris Royale", réalisé dans le cadre du cours d'informatique *INFO-F209*.

L'objectif principal de ce rapport est de fournir une vision claire et structurée des besoins logiciels, des choix architecturaux, et des outils utilisés pour le développement. Par ailleurs, des diagrammes explicatifs suivant les conventions **UML** (*Unified Modeling Language*) seront présentés afin d'illustrer l'implémentation technique et les interactions entre les différentes composantes du projet.

Cette introduction constitue une vue d'ensemble des thèmes abordés dans le document, servant ainsi de guide pour comprendre la portée et les objectifs du projet.

1.2 Description du Projet

1.2.1 Résumé

Le projet consiste à développer **Tetris Royale**, une version multijoueur compétitive du jeu classique *Tetris*. Chaque joueur place des pièces (*tetraminos*) tombant sur une grille en essayant de compléter des lignes horizontales. Lorsqu'une ligne est complétée, elle disparaît et le joueur gagne des points. *Tetris Royale* se distingue des autres versions de *Tetris* en permettant aux joueurs d'envoyer des malus à leurs adversaires sous forme de lignes incomplètes supplémentaires lorsqu'ils complètent plusieurs lignes en un seul coup. Le but du jeu est de survivre le plus longtemps possible. Le dernier joueur en lice, après élimination de tous les autres, est déclaré vainqueur.

1.2.2 Modes de jeu

Quatre modes de jeu sont disponibles : *Endless*, *Classic*, *Duel* et *Royal Competition*.

Mode Endless

Le mode *Endless* est un mode à joueur unique. Le joueur gagne des points en fonction de ses combos, tandis que la vitesse de chute des pièces augmente progressivement. Le score final est enregistré dans un classement en ligne.

Mode Classic

Le mode *Classic* permet à un groupe de 3 à 9 joueurs de s'affronter dans des parties simples, où seuls des malus basiques sont utilisés (ces derniers seront décrits dans une section ultérieure). Le dernier joueur en lice est déclaré gagnant.

Mode Duel

Le mode *Duel* suit les mêmes règles que le mode *Classic*, mais se joue en un contre un.

Mode Royal Competition

Dans le mode *Royal Competition*, les malus ne sont plus envoyés automatiquement. Les joueurs accumulent de l'énergie, et lorsqu'ils en ont suffisamment, ils peuvent choisir d'envoyer un malus ou de s'octroyer un bonus (ces mécanismes seront détaillés ultérieurement). Ce mode se joue, comme le mode *Classic*, en groupes de 3 à 9 joueurs.

1.2.3 Fonctionnalités multijoueurs

Dans les modes multijoueurs, chaque joueur peut visualiser les tableaux des autres participants.

1.2.4 Fonctionnalités hors partie

En dehors des parties, le joueur dispose des fonctionnalités suivantes :

- Consulter une liste d'amis pour discuter ou inviter des joueurs en tant que participants ou observateurs ;
- Créer ou rejoindre une partie ;
- Consulter le classement des meilleurs joueurs du mode *Endless*.

1.3 Glossaire

Dans cette courte section, nous définirons quelques mots que nous utiliserons au long de ce **SRD**. Ce glossaire est composé d'éléments liés à *Tetris Royale*, et plus généralement au développement de celui-ci.

- **Pseudo** : Nom d'utilisateur choisi par celui-ci lors de la création de son compte.
- **Combo** : Action de compléter plusieurs lignes simultanément.
- **Matchmaking** : Mécanisme de mise en relation des joueurs afin de créer une partie en ligne.
- **Lobby** : Interface ou lieu où l'on regroupe des joueurs avant de démarrer une partie.
- **User (Utilisateur)** : Client qui est connecté en réseau et lié à un compte.
- **Tetramino** : Pièce de *Tetris*, composée de 4 blocs carrés disposés selon diverses configurations.
- **Battle Royale** : Mode de compétition où chaque joueur s'affronte pour être le dernier en vie.
- **Ligne** : Une rangée complète de blocs horizontaux sur la grille. Une ligne complétée disparaît.
- **Grille** : L'aire de jeu où tombent les *tetraminos*. Elle mesure 10 cases de large et 22 cases de haut.
- **Tetris** : Compléter quatre lignes en une seule action à l'aide d'une pièce en forme de barre (*I*).

1.4 Historique

Dans cette courte section, nous présenterons un historique des différentes versions de ce **SRD**. La figure ci-dessous illustre l'évolution du document.

Version	Nom	Modifications	Date
0.1	Oleksandra Omelyanyuk	Introduction + besoins utilisateurs	05/12/2024
0.2	Roberto Rabei	Besoins système	08/12/2024
0.3	Roberto Rabei	Squelette Design système	09/12/2024
0.4	Oleksandra Omelyanyuk	Modifications besoins système + diagrammes	09/12/2024
1.0	Berthion Antoine	Restructuration du SRD	13/12/2024
1.1	Oleksandra Omelyanyuk	Début adaptation aux changements	10/03/2025

Version	Nom	Modifications	Date
1.2	Oleksandra Omelyanyuk	Modification/rajout diagrammes	12/03/2025
2	Oleksandra Omelyanyuk	Modifications diagrammes	14/03/2025
2.1	Oleksandra Omelyanyuk	Adaptation texte	08/05/2025

Chapitre 2

Besoins Utilistateurs

Dans cette section du SRD, il s'agira de décrire les besoins spécifiques à l'utilisateur. En particulier, nous distinguerons les besoins fonctionnels, c'est-à-dire les exigences techniques et liées à l'implémentation du *Tetris Royale*, ainsi que les besoins non-fonctionnels, qui concernent l'expérience utilisateur et les attentes en termes de performance, de sécurité et d'ergonomie.

2.1 Besoins Fonctionnels

2.1.1 Connexion et Création de Compte

Lorsque l'utilisateur lance le jeu, il est d'abord accueilli par une fenêtre de connexion. Cette fenêtre lui permet de se connecter à son compte en saisissant son identifiant et son mot de passe. Si l'utilisateur ne possède pas encore de compte, il a la possibilité d'en créer un en cliquant sur un bouton de création de compte.

Lors de la création d'un compte, l'utilisateur doit fournir les informations suivantes :

- **Pseudo** : L'utilisateur doit choisir un pseudo qui lui est unique. Le système vérifiera que le pseudo choisi n'est pas déjà utilisé par un autre joueur.
- **Mot de Passe** : L'utilisateur doit également choisir un mot de passe sécurisé pour protéger son compte. Le mot de passe sera requis pour la connexion ultérieure.
- **Validation du Pseudo** : Si le pseudo choisi est déjà pris, l'utilisateur recevra un message d'erreur lui demandant de choisir un autre pseudo. Une fois un pseudo disponible sélectionné, l'utilisateur pourra poursuivre la création de son compte.

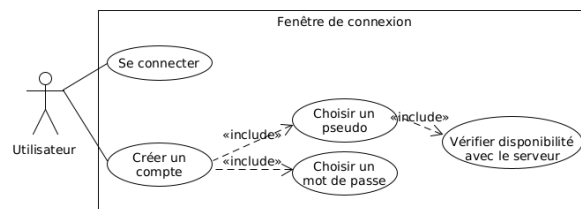


FIGURE 2.1 – Use-Case : Connexion et Création de Compte

Une fois ces informations validées, le compte est créé et l'utilisateur est connecté à son profil. Le joueur peut alors accéder aux différentes fonctionnalités du jeu, telles que rejoindre des parties, consulter ses statistiques et interagir avec d'autres utilisateurs.

Si l'utilisateur a déjà un compte, il peut simplement saisir ses identifiants (pseudo et mot de passe) pour se connecter.

2.1.2 Gestion des Parties et Profil Utilisateur

Cette sous-section décrit en détail les interactions disponibles pour l'utilisateur via l'interface du menu, illustrée dans la figure 2.2 ci-dessous.

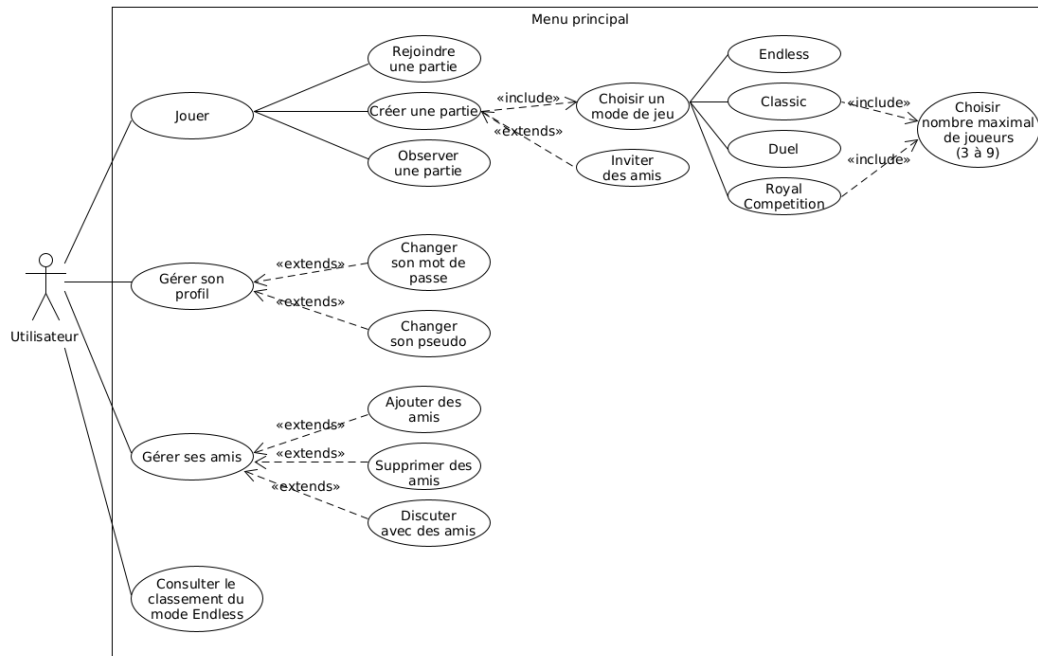


FIGURE 2.2 – Use-Case : Menu Principal

Lorsque l'utilisateur décide de jouer, il peut choisir entre rejoindre une partie déjà créée ou créer sa propre partie.

Création de Partie

Si l'utilisateur choisit de créer sa propre partie, il dispose des options suivantes :

- **Inviter des amis** : L'utilisateur peut inviter ses amis à participer à la partie, soit pour jouer, soit pour observer. Une invitation correspond à un code que le joueur invité devra rentrer pour accéder au jeu.
- **Choisir un mode de jeu** : L'utilisateur doit obligatoirement sélectionner un mode de jeu avant de lancer la partie. Il doit ensuite se mettre prêt, en attendant que le bon nombre de joueurs rejoigne sa partie.
- **Choisir un nombre maximal de joueurs** : Pour les modes *Classic* et *Royal Competition*, l'utilisateur doit définir un nombre maximal de joueurs qui peuvent rejoindre la partie. Ce nombre doit être compris entre 3 et 9 joueurs inclus.

Rejoindre une Partie

L'utilisateur peut également choisir de rejoindre une partie existante. Il peut accéder à un menu contenant les parties en cours, ou qui n'ont pas encore atteint leur nombre souhaité de joueurs, et peut alors choisir de les rejoindre ou de les observer.

Liste d'Amis et Discussions

Chaque utilisateur possède une liste d'amis, qu'il peut gérer de la manière suivante :

- **Ajouter des amis** : L'utilisateur peut envoyer des demandes d'ami à d'autres joueurs. Une fois la demande acceptée, ils deviennent amis et peuvent interagir dans le jeu.
- **Supprimer des amis** : L'utilisateur peut retirer des amis de sa liste à tout moment.

- **Lancer une discussion** : L'utilisateur peut démarrer une conversation avec un ami à partir de la liste d'amis.

Classement du Mode Endless

L'utilisateur peut accéder à un classement en ligne des joueurs du mode *Endless*. Ce classement présente tous les joueurs ainsi que leurs scores, classés dans l'ordre décroissant. Cela permet à l'utilisateur de voir sa position par rapport aux autres joueurs et de suivre la progression des meilleurs joueurs du mode.

2.1.3 Déroulement d'une partie

Une fois que le joueur a lancé une partie, il peut déplacer un tetramino pour le positionner sur son tableau de jeu. Il peut également utiliser le *bag*, un espace dans lequel il peut ranger un tetramino ou utiliser celui qui y est déjà stocké.

De plus, le joueur a la possibilité d'utiliser des malus, mais seulement s'il joue dans les modes *Classic*, *Duel* ou *Royal Competition* (dans ce mode des bonus sont aussi disponibles). Dans ces trois modes, il peut également voir les tableaux des autres joueurs en temps réel.

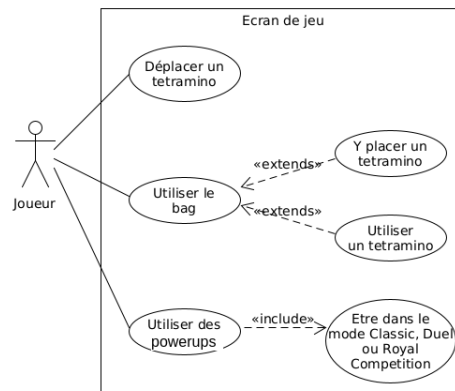


FIGURE 2.3 – Use-Case : Déroulement d'une partie

Malus et bonus

Dans les modes de jeu autres que *Royal Competition*, où des bonus et des malus avancés sont disponibles, seuls des malus basiques sont envoyés automatiquement lorsqu'un joueur réalise un combo. Ces malus se traduisent par l'apparition de lignes additionnelles sur le tableau d'un joueur adverse, selon les règles suivantes :

- 1 ligne complétée n'envoie aucun malus.
- 2 lignes complétées envoient 1 ligne de malus.
- 3 lignes complétées envoient 2 lignes de malus.
- 4 lignes complétées (Tetris) envoient 4 lignes de malus.

Dans le mode *Classic*, le joueur peut également changer le joueur actuellement ciblé par ses malus, lui offrant ainsi un certain contrôle stratégique sur l'adversaire.

Le mode *Royal Competition* introduit des malus plus avancés, incluant des bonus (avantages), permettant une interaction plus dynamique avec les autres joueurs. Voici quelques exemples des *power-ups* disponibles dans ce mode :

- **Inverser les commandes** : Inverse les commandes du joueur ciblé pour la pose des trois prochains blocs.
- **Bloquer les commandes** : Bloque les commandes du joueur ciblé pour la pose du prochain bloc.

- **Réduire la vitesse de chute** : Réduit la vitesse de chute des pièces sur son propre plateau pendant un moment.
- **Augmenter la difficulté** : Accélère la chute des pièces sur le plateau de l'adversaire.
- **Éclair** : Envoie un éclair qui supprime les blocs dans une zone de 2x2 chez le joueur ciblé.
- **Éteindre la lumière** : Éteint la lumière, rendant le tableau du joueur ciblé invisible pendant un court moment.
- **Blocs de 1x1** : Les deux prochaines pièces du joueur se transforment en blocs de 1x1.

2.2 Besoins non-fonctionnels

Le jeu doit offrir une interface à la fois simple d'utilisation et agréable à l'œil. Cela est particulièrement important dans les modes multijoueurs, où les tableaux des adversaires doivent être affichés de manière claire et disposés de façon à ne pas encombrer l'interface principale du jeu. C'est pourquoi il est possible d'afficher un seul tableau adverse à la fois, avec une possibilité de changer d'adversaire visé.

De plus, le jeu doit être accessible à travers différentes interfaces, permettant à l'utilisateur de jouer aussi bien sur un terminal que sur une interface graphique (GUI).

Un autre besoin crucial est la rapidité de l'envoi d'information entre les joueurs. Pour garantir une expérience de jeu fluide et réactive, il est essentiel qu'il n'y ait aucun délai perceptible lorsque le joueur consulte les grilles de ses adversaires.

Chapitre 3

Besoins Systèmes

3.1 Connexion

Le système doit permettre la création de nouveaux comptes tout en vérifiant les informations fournies par les utilisateurs. Lors de l'inscription, il doit s'assurer que le pseudonyme choisi n'est pas déjà utilisé par un autre utilisateur. Une fois les informations validées, elles doivent être enregistrées sur le serveur. Pour la connexion, le système doit vérifier l'authenticité des données en les comparant à celles stockées sur le serveur.

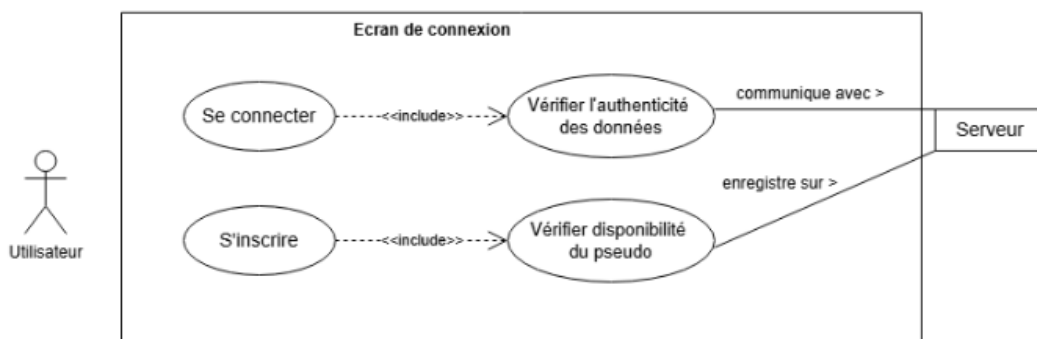


FIGURE 3.1 – Use-Case : Connexion

3.2 Lobby

Lorsqu'un joueur souhaite lancer une partie, il est d'abord placé dans un lobby. Dans cet espace, il peut inviter des amis à rejoindre sa partie. Le lobby est également accessible aux joueurs aléatoires à la recherche d'une partie. Le serveur s'assure que le nombre de joueurs minimum et maximum est respecté. Lorsque tous les joueurs sont prêts, le serveur lance la partie.

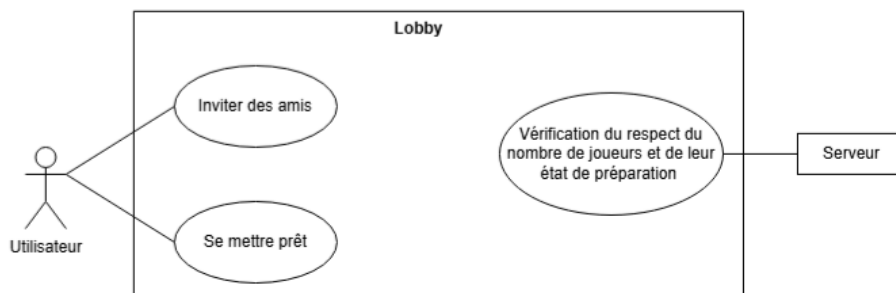


FIGURE 3.2 – Use-Case : Lobby

3.3 Partie

Le système doit être capable de suivre en temps réel les actions de chaque joueur dans la partie afin d'assurer une expérience fluide, immersive et compétitive. Selon le mode de jeu sélectionné, l'interface doit s'adapter pour offrir un affichage personnalisé :

- **Endless** : Ce mode solo affiche la grille de jeu avec le nombre de points accumulés durant la partie.
- **Classique** : L'affichage est similaire au mode *Endless*, avec, en plus, une grille d'un joueur adverse visible à côté, permettant ainsi de choisir à qui envoyer les malus.
- **Duel** : Le même affichage que le mode *Classique*, mais sans possibilité de changer d'adversaire visé (car il y en a qu'un).
- **Royal Competition** : L'affichage du mode *Classique* est utilisé, mais cette fois, l'utilisateur peut choisir d'utiliser des bonus ou d'envoyer des malus plus avancés, sans pouvoir envoyer le malus de base.

Chapitre 4

Design et Fonctionnement

Design du système

Le système est divisé en six entités principales, chacune ayant un rôle précis dans le fonctionnement global du jeu. Ces entités sont :

- Connexion et Inscription
- Gestion des comptes
- Menu principal
- Matchmaking
- Chat
- Jeu

Ces entités partagent une structure commune, en raison de l'utilisation du design *MVC* (Modèle-Vue-Contrôleur). Chaque entité du système est composée de trois composants principaux :

- **La vue** : Elle correspond à l'interface utilisateur. Elle permet d'afficher les informations à l'utilisateur et de recevoir ses inputs qu'elle transmet au contrôleur.
- **Le contrôleur** : Il gère les requêtes envoyées par la vue et les communique au serveur. Il met à jour le serveur en fonction des actions des utilisateurs.
- **Le serveur** : Il contient toutes les informations relatives aux utilisateurs et aux parties en cours. Il est responsable d'envoyer les informations requises par le contrôleur et de mettre les joueurs en relation pour la création de parties.

4.1 Connexion et inscription

Lorsqu'un joueur démarre le jeu, il peut choisir entre créer un nouveau compte ou se connecter à un compte existant. Deux interfaces distinctes sont proposées pour ces actions. Lors de la connexion, les données de l'utilisateur sont vérifiées par rapport à la base de données. Lors de l'inscription, les nouvelles données de l'utilisateur sont enregistrées dans la base de données.

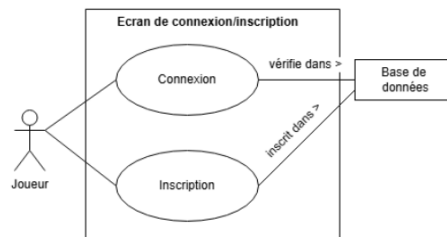


FIGURE 4.1 – Use-Case : Connexion

4.2 Gestion des comptes

Une fois qu'un joueur crée un compte, celui-ci est stocké dans la base de données. Cela permet à l'utilisateur d'accéder aux fonctionnalités du jeu, telles que le chat et la gestion de sa liste d'amis.

4.3 Menu principal

Le menu principal permet d'accéder à différentes sections du jeu, telles que le lobby pour créer ou rejoindre une partie, ainsi que le classement des meilleurs joueurs. Ce menu est l'interface de navigation principale dans le jeu.

4.4 Matchmaking

Le serveur gère une liste de parties actives et une liste de parties en état de *matchmaking*, c'est-à-dire des parties en attente de joueurs pour être complétées. Lorsqu'un joueur souhaite rejoindre une partie, le serveur vérifie s'ils existent déjà des parties en *matchmaking*. Si au moins une partie est disponible, le joueur peut choisir de la rejoindre. Sinon, le serveur crée une nouvelle partie avec ce joueur comme premier participant et marque la partie comme étant en *matchmaking*.

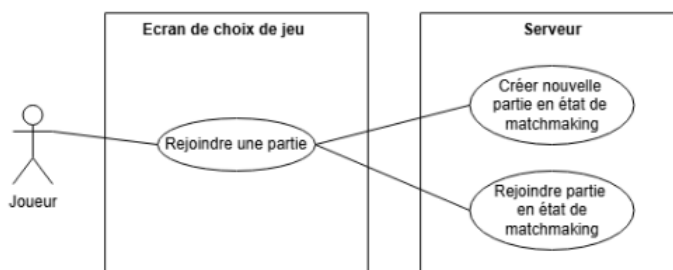


FIGURE 4.2 – Use-Case : Matchmaking

4.5 Chat

Le jeu propose un système de chat permettant aux joueurs d'envoyer des messages privés à ceux présents dans leur liste d'amis. Les messages envoyés passent par un contrôle pour bloquer les liens malicieux ou les insultes, assurant ainsi une expérience de communication respectueuse et adaptée à tous.

4.6 Jeu

Dans le mode de jeu, les joueurs débutent avec un plateau vide sur lequel des tétraminos tombent. Grâce aux commandes du jeu, ils doivent déplacer les tétraminos pour les positionner de manière stratégique. Les états des parties sont sauvegardés côté serveur, et les actions des joueurs sont envoyées au serveur pour validation.

Le joueur n'a jamais d'accès direct à sa partie. Toutes les interactions passent par le *GameEngine*, qui gère la logique du jeu afin d'assurer la sécurité et d'empêcher la triche. Le serveur garde une vue d'ensemble des parties des joueurs dans la session en cours, ce qui lui permet de déterminer la fin d'une partie (par exemple, lorsque seulement un joueur reste en vie dans le mode Royal).

Même en mode solo, les joueurs doivent se connecter au serveur, ce qui permet de bénéficier de certaines fonctionnalités, comme l'observation des parties en cours.

4.7 Vue globale

Dans cette section, nous présenterons les diagrammes de classes et les diagrammes de cas d'utilisation (*Use-Case*) associés à la vue globale du projet.

4.7.1 Vue globale : MVC

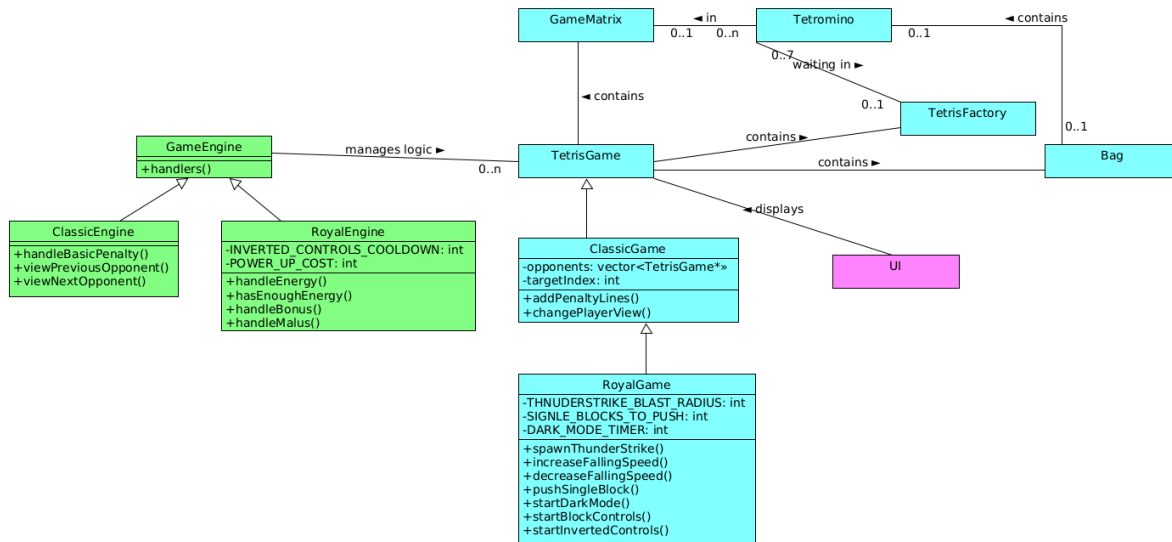


FIGURE 4.3 – Diagramme de classes : Vue globale du système en architecture MVC

4.7.2 Vue globale : Serveur

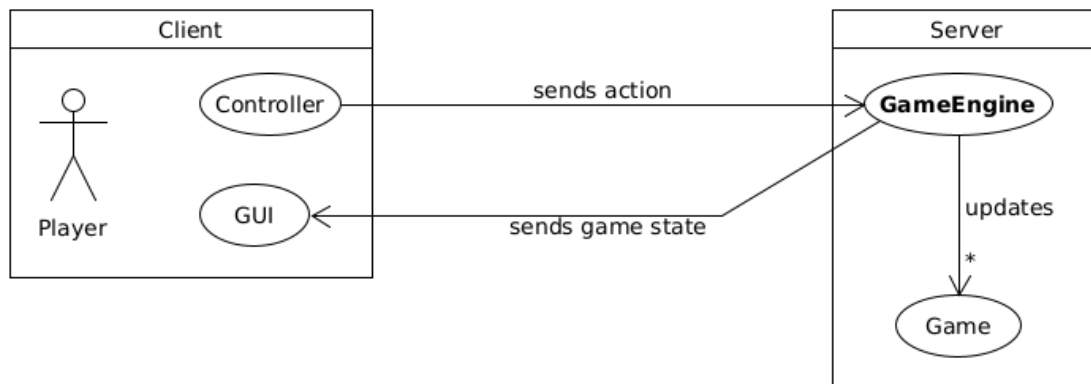


FIGURE 4.4 – Diagramme de cas d'utilisation : Interactions client-serveur

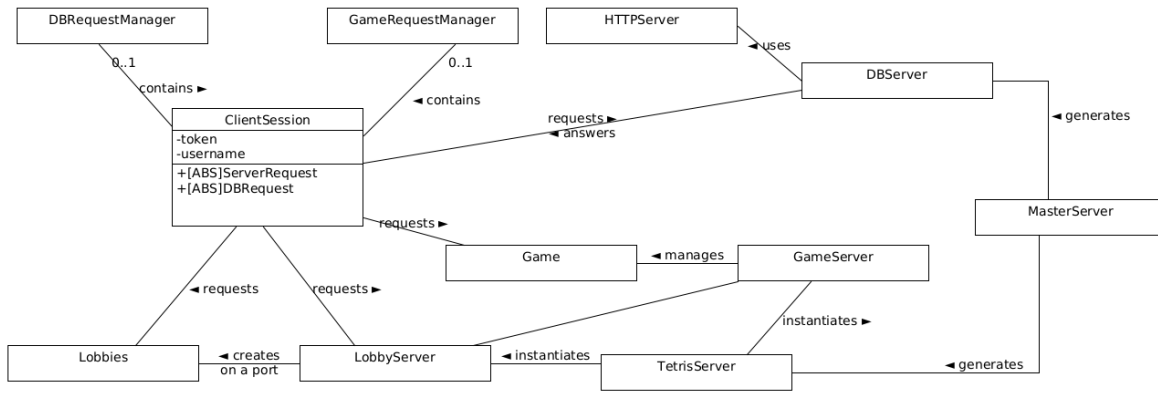


FIGURE 4.5 – Diagramme de classes : Interactions classes du serveur

4.7.3 Vue globale : Base de données

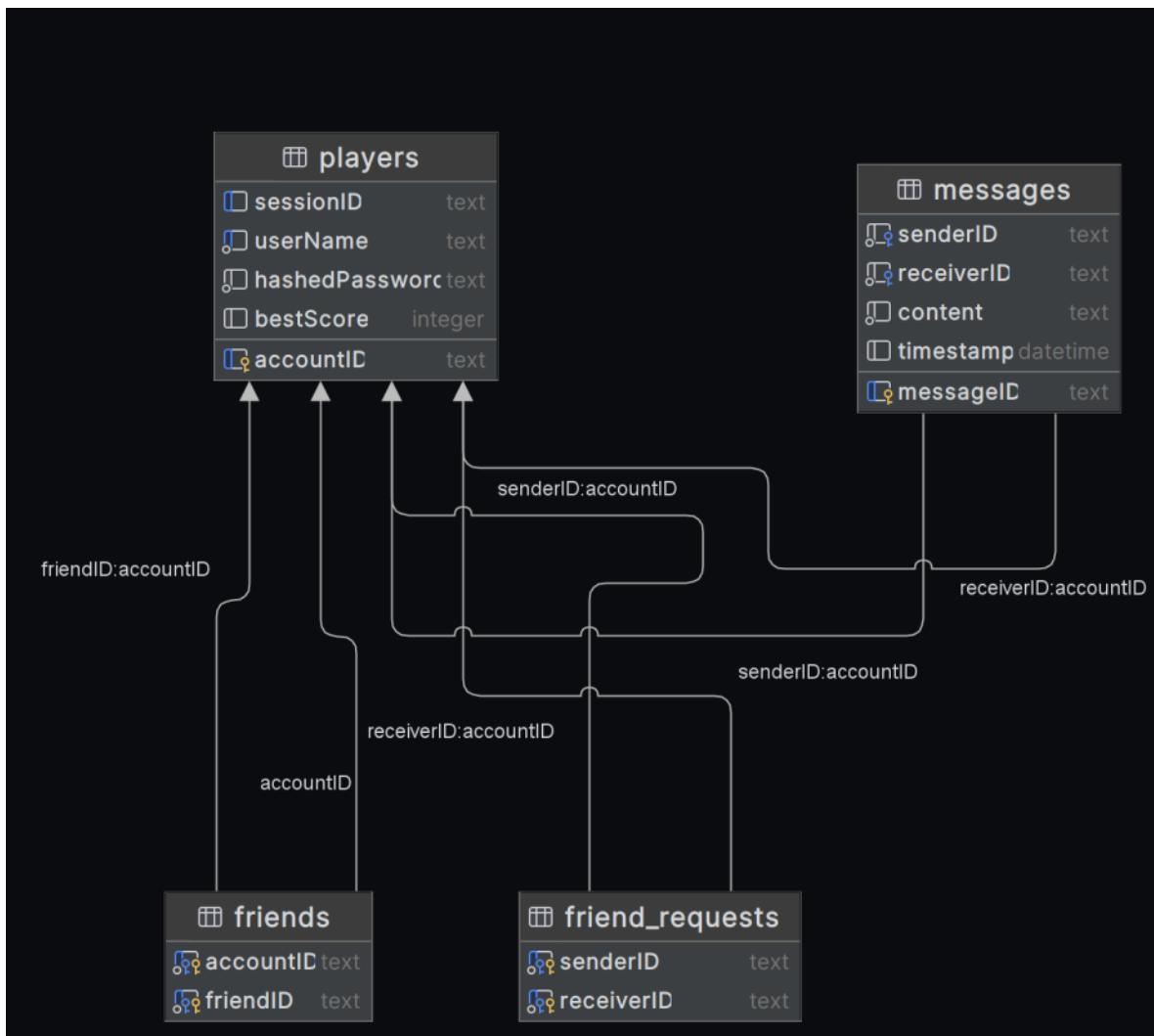


FIGURE 4.6 – Diagramme entité-relation : Base de données