Université Libre de Bruxelles

Projets d'informatique II

Tetristitude

SRD 3V-CUBE

Auteurs:
Derras Eiman
Drion Elisha
Bouali Abderrahmane

Rouaux Thomas
Libert Alexandre
Jouniaux Alexane

March 29, 2018

Contents

1 Introduction

1.1 But du projet

Le but du projet est de développer un jeu en réseau à l'aide d'outils acquis depuis le début de notre cursus. Celui-ci doit être une reproduction du célèbre "Tetris", disposant d'un mode multijoueur ainsi que d'un système de gestion des challengers. Se nommant 3V-CUBE, il est intégralement écrit en C++.

Il est demandé d'accorder une attention particulière aux concepts vus aux cours d'INFO-F-201 et d'INFO-F-204. Le client-serveur gérant le jeu et le SRD seront donc les points phares de ce projet.

Comme tout jeu en réseau, le joueur a la possibilité de créer un compte pour suivre ses statistiques, trouver des parties adaptées à son niveau mais également de discuter avec ses amis ainsi que d'autres fonctionnalités s'offrant à lui.

Il y a 4 modes de jeu disponibles : classique, marathon, sprint et VS (en ligne).

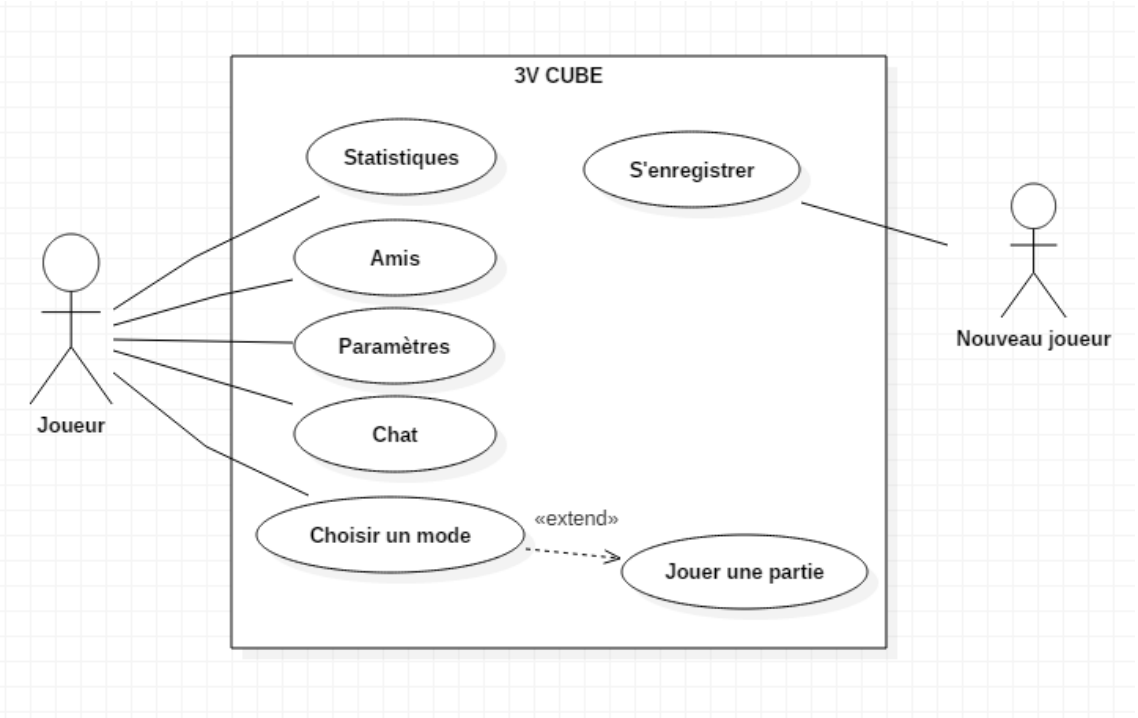
1.2 Glossaire

DAS : Acronyme de "Delayed Auto Shift". Comportement du tetriminos lorsque le joueur maintient la touche gauche ou droite enfoncée. Le tetriminos va bouger d'un cran, attendre, puis le rebouger jusqu'à ce que le joueur lâche la touche.
Ghost : Tetriminos légèrement transparent qui représente l'endroit où le tetriminos en train de tomber atterrira.
Gravité : Constante qui détermine la vitesse de chute des tetriminos.
Salon : Endroit virtuel où des joueurs attendent de jouer/jouent ensemble.
Matchmaking : Processus de mise en relation de deux joueurs afin de créer une partie.
Tetrimino : Forme géométrique de 4 carrés que le joueur manipule.

1.3 Historique du document

Version	Date de modification	Auteur	Résumé
0.1	26/11/2017	Eiman	Ajout de l'introduction, de la grille d'historique et des USE CASE
0.2	27/11/2017	Elisha	Ajout glossaire, remaniement structure besoins fonctionnels, màj introduction
0.3	28/11/2017	Abderr, Thomas, Alexandre, Alexane	Use case matchmaking
0.4	01/12/2017	Eiman	Ajout de besoins fonctionnels, remaniement structure matchmaking
0.5	13/02/2018	Abderr, Alexane	Ajout des diagrammes de classes, de séquence et d'activité et remaniement structure globale
0.6	22/02/2018	Alexane	Mise à jour diagramme de séquence et d'activité
0.7	24/02/2018	Alexane	Remaniement texte et structure
0.8	28/02/2018	Alexane	Ajout des diagrammes de classe,des use-case (et remaniement de certains) et texte complété
0.9	04/03/2018	Alexane	Ajout des diagrammes de séquence de boucle de jeu, explications et schéma de la base de données et diverses explications (chat, liste d'amis, ...)
1.0	15/03/2018	Alexane	Description des différents modes de jeu
1.1	28/03/2018	Thomas, Elisha	Modification générale du SRD

2 Diagramme général

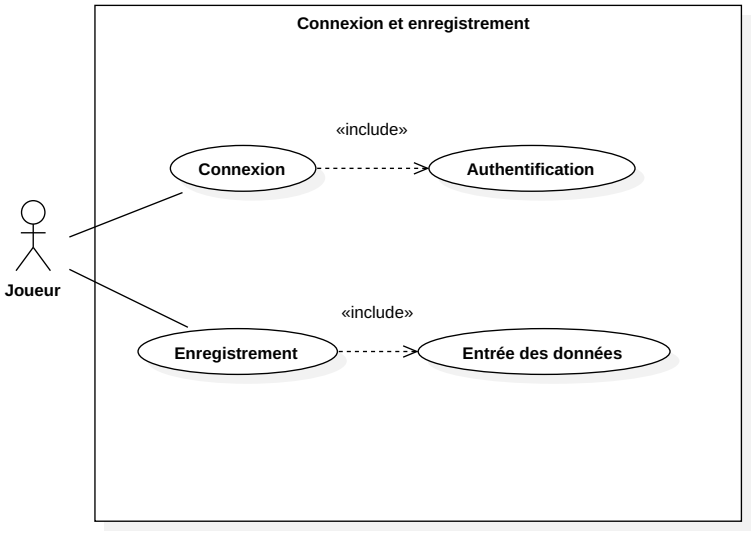


L'utilisateur aura une multitude de choix devant lui. Il pourra entre autre consulter ses statistiques personnelles et son classement mondial, il pourra aussi chatter avec ses différents amis et évidemment regarder les statistiques de ceux-ci. Mais également, avoir le choix de changer les paramètres comme il l'apprécie. Enfin, il pourra choisir un mode de jeu et jouer en solo ou faire une recherche de salon favorable à celui-ci.

3 Besoins fonctionnels de l'utilisateur

3.1 Connexion et enregistrement

Condition générale : Serveur doit être en ligne.



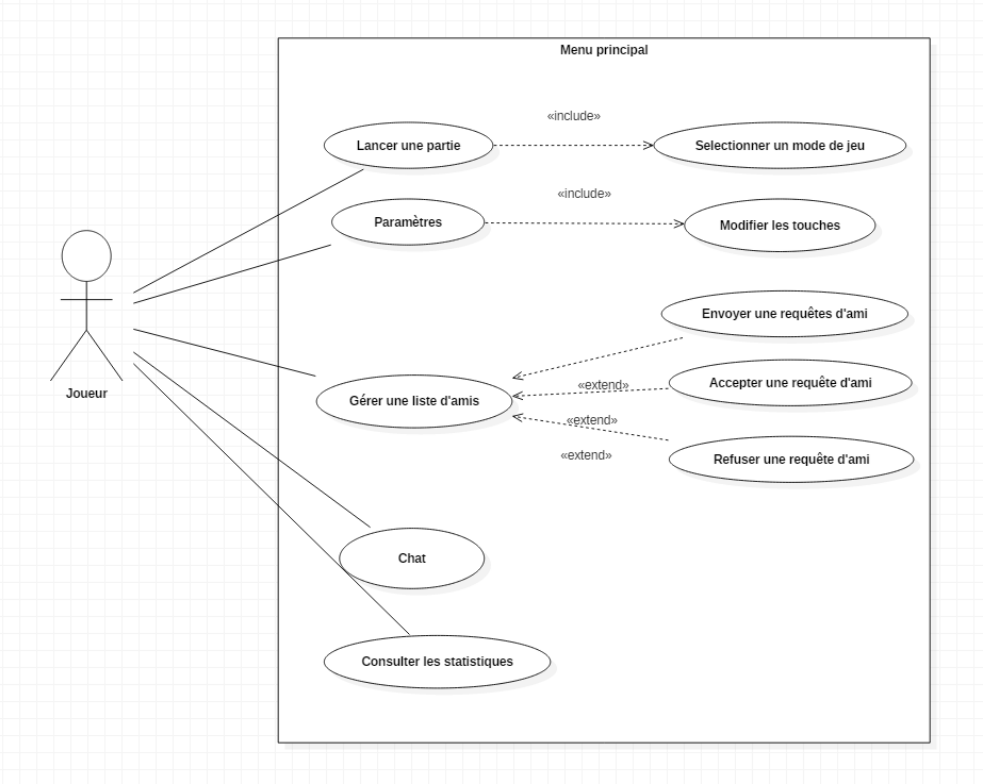
3.1.1 Connexion

- **Acteur(s) :** Joueur.
- **Pré-conditions :** Le joueur doit avoir un compte et le serveur doit être en ligne.
- **Post-conditions :** Le joueur accède au menu principal.
- **Cas particulier :** Le joueur entre de mauvais identifiants. Les champs sont vidés et un message est affiché, l'invitant à entrer des identifiants valides.

3.1.2 Enregistrement

- **Acteur(s) :** Joueur.
- **Pré-conditions :** Le serveur doit être en ligne et l'identifiant n'existe pas encore.
- **Post-conditions :** Le compte utilisateur est crée, le joueur accède à la fenêtre de connexion.
- **Cas particulier :** Le joueur n'entre pas d'identifiant ou de mot de passe ou ceux-ci ne correspondent pas au format demandé.

3.2 Menu principal



3.2.1 Lancer une partie

Le joueur décide à quel mode il jouera.
Acteur(s) : Joueur.
Pré-condition(s) : Le serveur doit être en ligne et un mode de jeu à été choisi au préalable.
Post-condition(s) : Le joueur est dans une partie et joue.
Cas particulier(s) : Néant.

3.2.2 Paramètres

Le joueur peut modifier les touches permettant de déplacer les Tetriminos ainsi que le volume de la musique dans le jeu.
Acteur(s) : Joueur.
Pré-condition(s) : Le joueur doit être connecté.

Post-condition(s) : L'utilisateur peut modifier les touches du clavier.
Cas particulier(s) : Lorsque que l'utilisateur sélectionne une touche déjà utilisée, un message d'avertissement est affiché.

3.2.3 Gérer la liste d'amis

Le joueur peut consulter sa liste d'amis et ajouter ou supprimer des amis.
Acteur(s) : Joueur.
Pré-condition(s) : Néant.
Post-condition(s) : La liste d'amis ainsi que les actions possibles à effectuer s'affichent.
Cas particulier(s) : Néant.

3.2.4 Chat

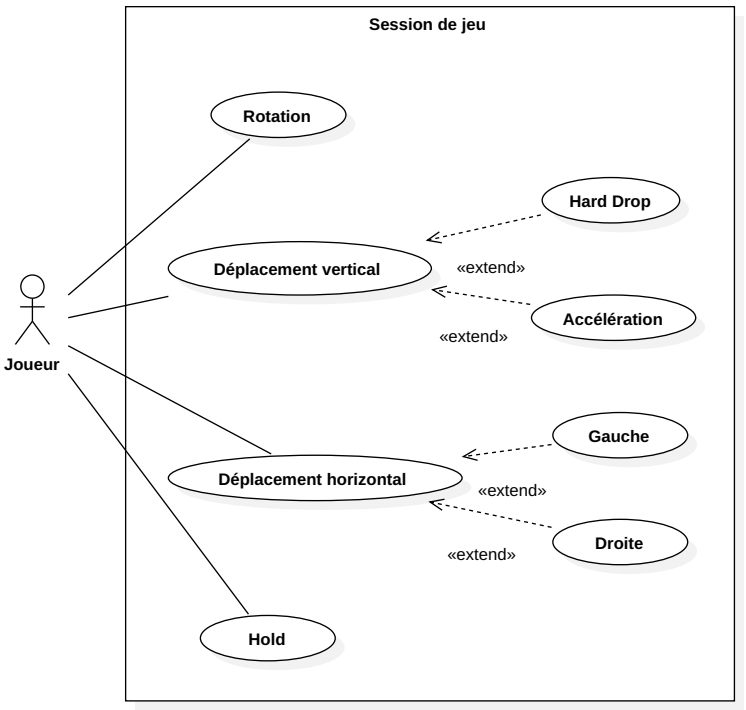
Depuis le menu principal, on peut accéder à un salon de chat. Là, on peut discuter avec tous les joueurs qui sont également en ligne.
Acteur(s) : Joueur.
Pré-condition(s) : Le joueur doit être connecté.
Post-condition(s) : Le joueur entre dans le lobby
Cas particulier(s) : Néant.

3.2.5 Consulter les statistiques

Le joueur peut consulter ses statistiques personnelles ainsi que celles de ses amis.
Acteur(s) : Joueur.
Pré-condition(s) : Néant.
Post-condition(s) : Les statistiques du joueur sont affichées.
Cas particulier(s) : Néant.

3.3 Session de jeu

Dans une session de jeu classique, l'utilisateur peut déplacer horizontalement les tétriminos ou les tourner dans le sens souhaité. Tous ces déplacements se mesurent en unité. Les déplacements simples se mesurent par une unité quel que soit le sens ou le type de mouvement. Ainsi, indépendamment du mode de jeu, le tétriminos peut chuter directement vers le bas sans que le joueur ne puisse le redéplacer.



3.3.1 Rotation

Le joueur a la possibilité de faire tourner les pièces.

- **Acteur :** Joueur.
- **Pré-condition(s) :** Le tetriminos doit être en train de chuter.
- **Post-condition(s) :** Le tetriminos a tourné de 90° vers la droite ou la gauche.
- **Cas particulier(s) :** Néant.

3.3.2 Déplacement horizontal

Le joueur a la possibilité de déplacer horizontalement les pièces.

- **Acteur :** Joueur.
- **Pré-condition(s) :** Le tetriminos doit être en train de chuter.
- **Post-condition(s) :** Le tetriminos s'est déplacé d'une unité (bloc) vers la droite ou la gauche.
- **Cas particulier(s) :** Néant.

3.3.3 Déplacement vertical

Le joueur a la possibilité d'influencer le déplacement vertical des pièces.

- **Acteur :** Joueur.
- **Pré-condition(s) :** Le tetriminos doit être en train de chuter.
- **Post-condition(s) :** Le tetriminos s'est déplacé d'une unité (bloc) ou plus vers le bas.
- **Cas particulier(s) :**
  - **Hard Drop :** Le tetriminos est déplacé instantanément le plus bas possible.
  - **Accélération :** La chute du tetriminos est accélérée.

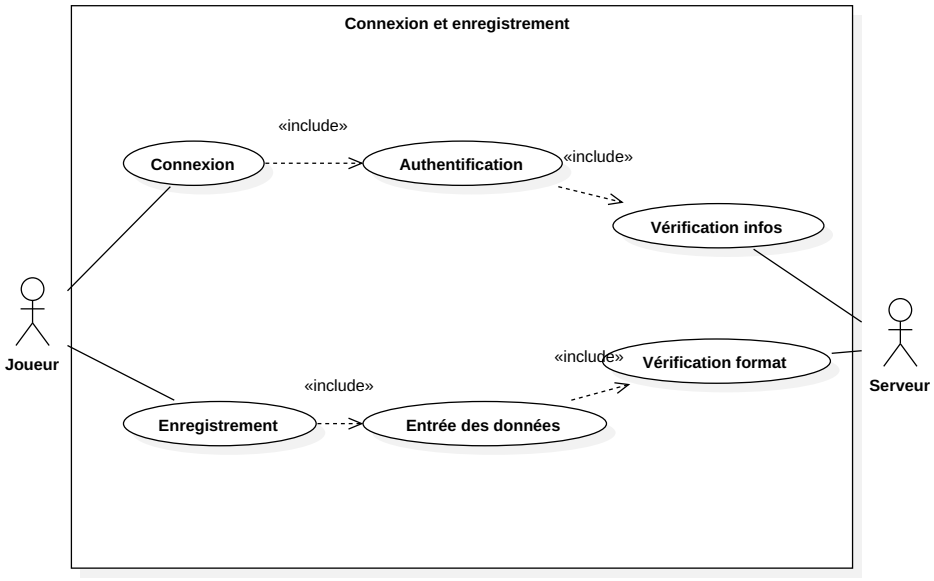
3.3.4 Hold

Le joueur décide de sauvegarder la pièce pour l'utiliser plus tard.

- Acteur(s) : Joueur
- Pré-condition(s) : La pièce doit être en train de chuter.
- Post-condition(s) : La pièce est sauvegardée.
- Condition(s) particulière(s) : Il y a déjà un tetriminos sauvergardé, et donc il sera échangé avec le tetriminos courant.

4 Besoins fonctionnels système

4.1 Connexion et enregistrement

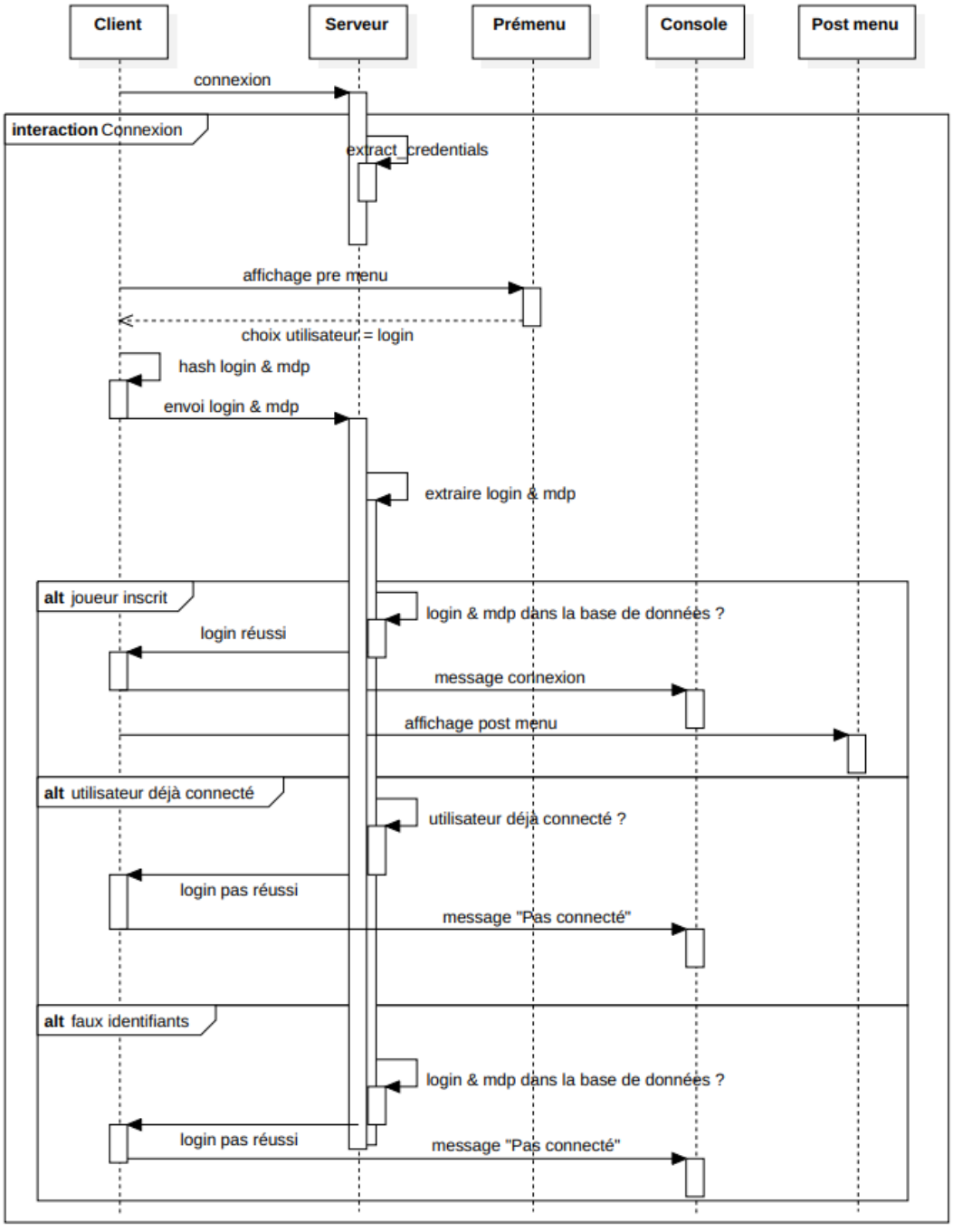


4.1.1 Vérification infos

- Acteur(s) : Serveur
- Pré-condition(s) : Le serveur doit être en ligne.
- Post-condition(s) : Si l'authentification est correcte, le menu principal s'affiche chez le joueur.
- Condition(s) particulière(s) : Si l'authentification est incorrecte, un message d'erreur est affiché et l'utilisateur peut réessayer de se connecter.

4.1.2 Vérification format

- Acteur(s) : Serveur
- Pré-condition(s) : Le serveur doit être en ligne.
- Post-condition(s) : Si la vérification est concluante, le joueur est inscrit dans le fichier répertoriant les login et mots de passes et le menu de connexion s'affiche.
- Cas particulier(s) : Si la vérification n'est pas concluante, un message d'erreur est affiché et l'utilisateur peut réessayer de s'enregistrer.


Client
Serveur
Prémenu
Console
Post menu
connexion
interaction Connexion
extract_credentials
affichage pre menu
choix utilisateur = login
hash login & mdp
envoi login & mdp
extraire login & mdp
alt joueur inscrit
login & mdp dans la base de données ?
login réussi
message connexion
affichage post menu
alt utilisateur déjà connecté
utilisateur déjà connecté ?
login pas réussi
message "Pas connecté"
alt faux identifiants
login & mdp dans la base de données ?
login pas réussi
message "Pas connecté"

4.2 Gestion des comptes

4.2.1 Création d'un compte

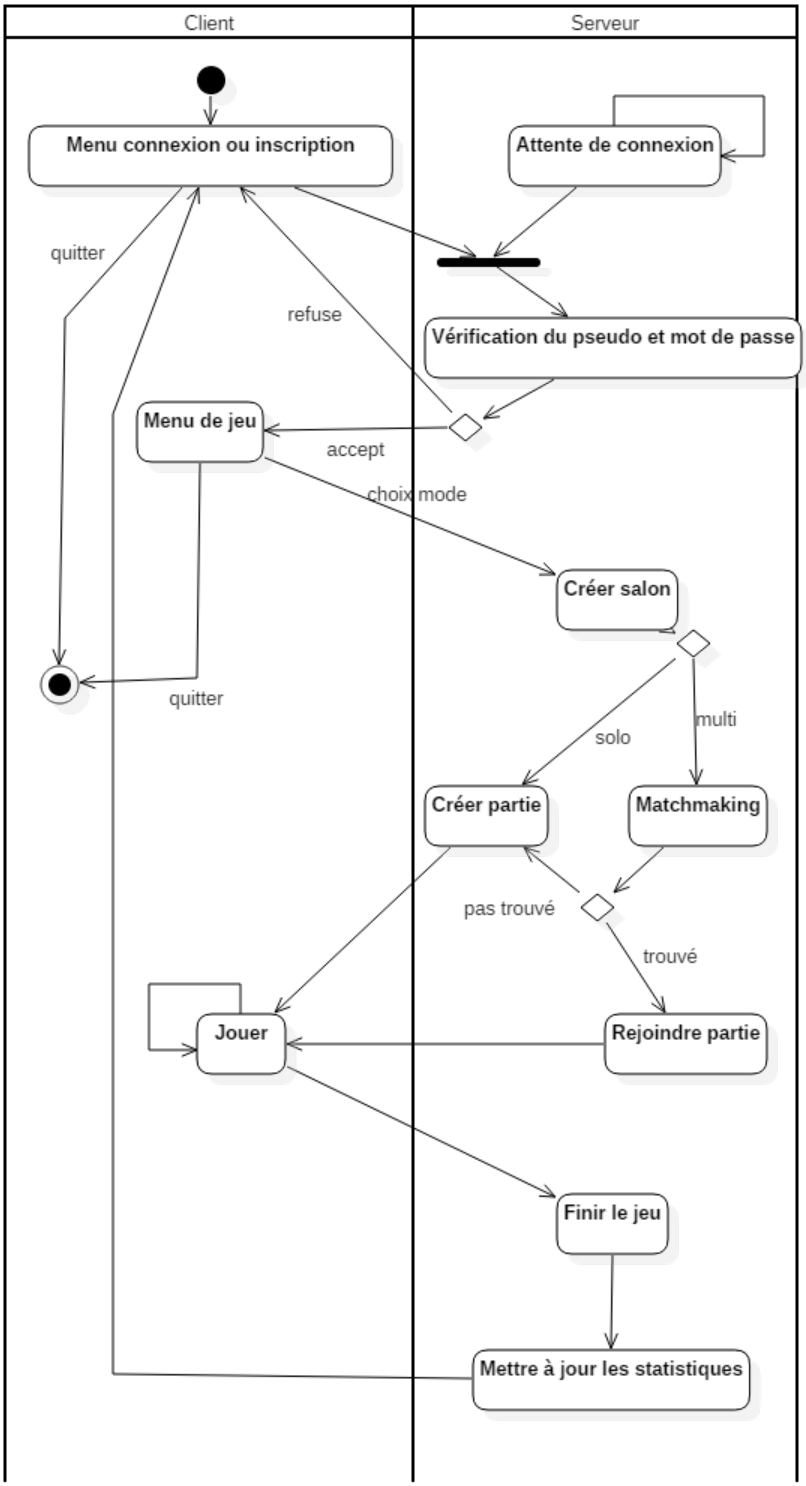
Lorsqu'il lance le jeu pour la première fois, le joueur a la possibilité de se créer un compte. Pour cela, une fenêtre est affichée et il doit choisir un identifiant et un mot de passe. Si ceux-ci sont valides, un compte sera créé et le joueur pourra l'utiliser.

4.3 Gestion du classement

La gestion du classement se fait grâce à une base de données dans laquelle on enregistre la somme des scores et le timer d'une partie.

4.4 Création d'une partie

Ce diagramme montre l'ensemble d'actions effectuées de la création d'une partie jusqu'à sa fin.

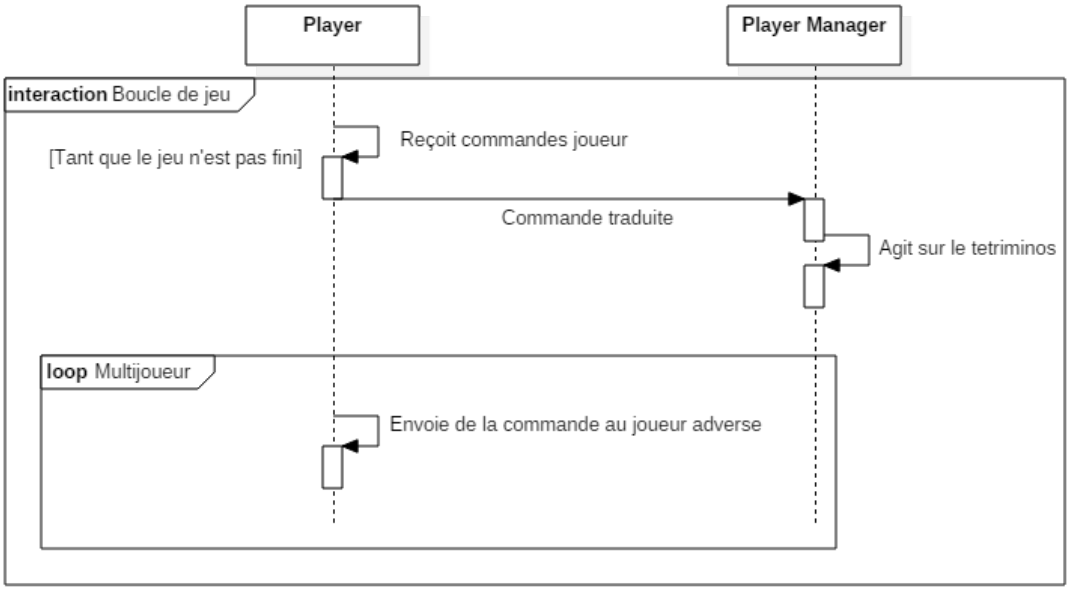

Client
Serveur
Menu connexion ou inscription
Attente de connexion
quitter
refuse
Vérification du pseudo et mot de passe
Menu de jeu
accept
choix mode
Créer salon
quitter
solo
multi
Créer partie
Matchmaking
pas trouvé
trouvé
Jouer
Rejoindre partie
Finir le jeu
Mettre à jour les statistiques

4.5 Session de jeu

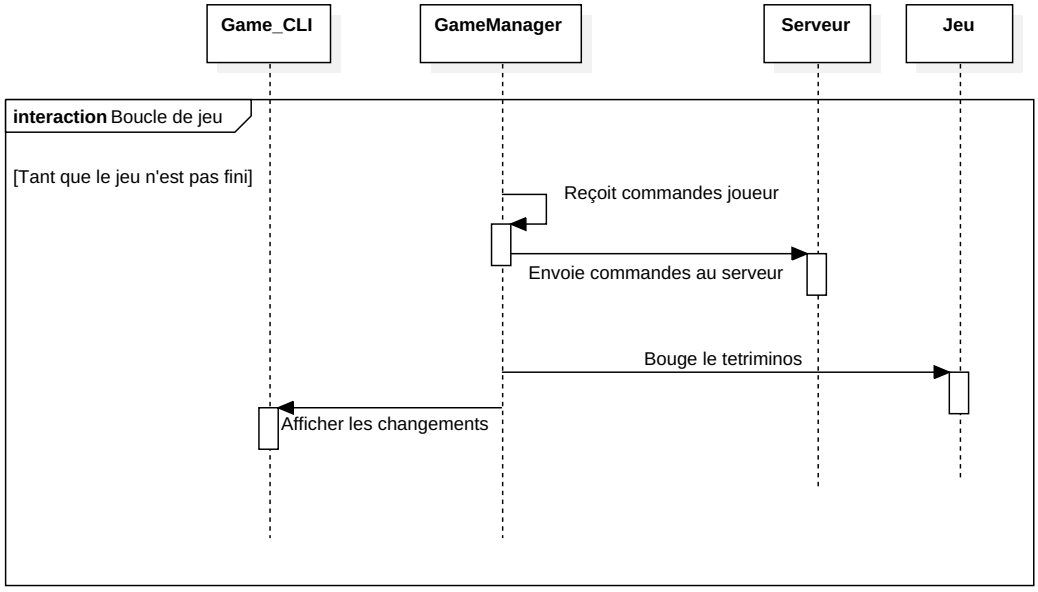
Cette section décrit l'ensemble d'actions possibles que le serveur peut devoir gérer pendant la partie.

4.5.1 Boucle de jeu

Voici l'ensemble d'actions qu'exécute le serveur pendant une partie.



Voici l'ensemble d'actions qu'exécute le client pendant une partie.



4.5.2 Gestion des tetriminos

4.5.3 Apparition des tetriminos

Les tetriminos sont générés aléatoirement. Le tetriminos courant est généré en même temps que le suivant.

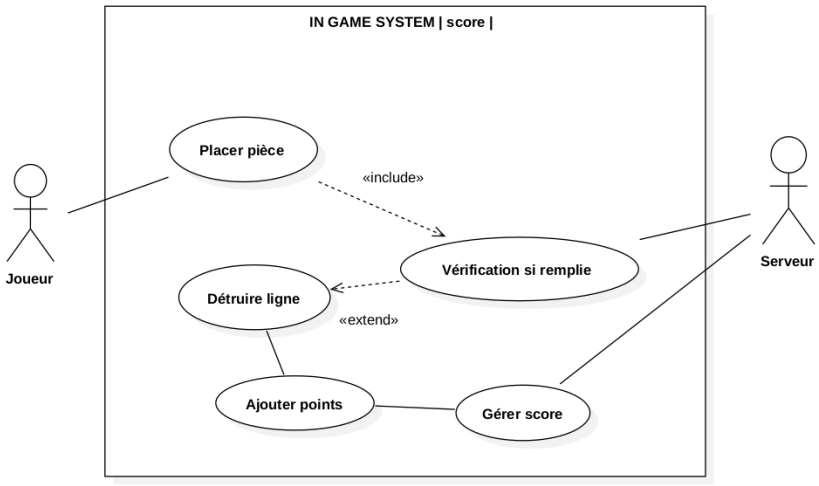
4.5.4 Disparition des tetriminos

Quand une ligne est complète, elle doit être supprimée. Pour cela, on "vide" les blocs de la ligne puis on fait descendre celle qui se trouvait au-dessus.

4.5.5 Difficulté du jeu

La difficulté du jeu est la vitesse à laquelle les tetriminos apparaissent et tombent. Elle reste constante dans tous les modes de jeu sauf le mode marathon. Dans celui-ci, elle augmente chaque fois que le joueur complète 10 lignes.

4.5.6 Score du joueur



- **Placer pièce**
  - **Acteur(s) :** Joueur

- **Pré-condition(s) :** Il reste une place de la taille et de la forme de la pièce dans la grille.
- **Post-condition(s) :** La pièce est placée dans la grille.
- **Cas particulier(s) :** Néant.

- **Vérification ligne remplie**
  - **Acteur(s) :** Serveur.
  - **Pré-condition(s) :** Plusieurs pièces doivent avoir été placées dans la grille.
  - **Post-condition(s) :** Néant.
  - **Cas particulier(s) :** Néant.

- **Ajouter points**
  - **Acteur(s) :** Serveur.
  - **Pré-condition(s) :** Une ligne doit être complète.
  - **Post-condition(s) :** Le score du joueur a augmenté.
  - **Cas particulier(s) :** Néant.

  La formule pour calculer les points ajoutés à un joueur est : $score = score + 50 \times 2^{nbre\ lignes\ complètes} \times (level\ du\ joueur + 1)$.

4.6 Modes de jeu

L'utilisateur a la possibilité de choisir entre 5 modes de jeu différents.

4.6.1 Mode classique

Le mode classique est un tetris traditionnel où le joueur place des tétriminos tant qu'il n'a pas dépassé la grille de jeu.

4.6.2 Mode marathon

Dans le mode marathon, le but du joueur est de terminer 200 lignes au plus vite. Chaque 10 lignes, la gravité s'intensifie et les tetriminos chuteront donc plus vite. Dans ce mode, il n'y a pas que le score qui compte, le temps joue également.

4.6.3 Mode sprint

Le mode sprint suit les mêmes règles que le mode classique, le but étant de terminer au plus vite 40 lignes.

4.6.4 Mode VS

Dans le mode VS, deux joueurs s'affrontent à Tetris. Chaque joueur a deux grilles sur son écran : la sienne et celle de son adversaire. Ce mode de jeu suit les mêmes règles que le mode classique. Le premier joueur qui place un tétriminos dépassant sa grille a perdu et l'autre est donc déclaré gagnant.

4.6.5 Mode Power Up

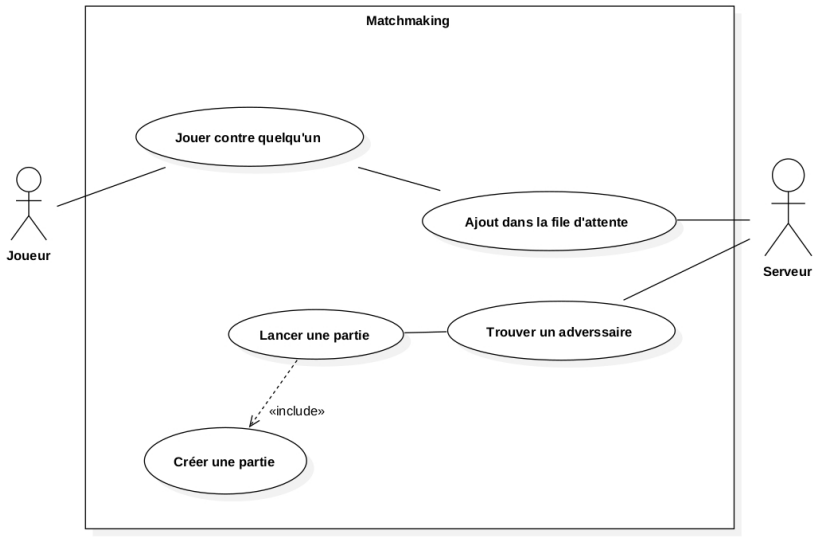
Ce mode se joue également à deux. Il suit les mêmes règles que le mode VS, la différence étant l'apparition de bonus dans la grille dans certains tetriminos. Voici une description de ces bonus :

1. Switch screen : Les stacks des deux adversaires sont échangés.
2. Ghost stack : Le stack de l'adversaire devient invisible pendant un certain temps.
3. Switch keys : Les touches gauches et droites de l'adversaire sont inversées pendant un certain temps.
4. Bomb : Détruit certains blocs du stack.

4.7 Salon de jeu

Le joueur a également la possibilité de jouer en mode multijoueurs. Le serveur va donc remplir les mêmes fonctions que dans une partie classique. Quand un joueur veut faire une partie en multijoueur, le serveur cherche un salon avec une place de libre. Dès qu'il en trouve un, il ajoute le joueur dedans et la partie commence. S'il n'y en a pas de libre, le serveur crée un salon et le joueur doit alors attendre qu'un autre joueur le rejoigne.

4.7.1 Création d'une partie multijoueur



- **Ajout dans la file d'attente**
  - **Acteur(s) :** Serveur.
  - **Pré-condition(s) :** Serveur doit être connecté. La liste d'attente ne doit pas être pleine.
  - **Post-condition(s) :** Le joueur est ajouté à la liste d'attente.
  - **Cas particulière(s) :** Néant.
- **Trouver un adversaire**
  - **Acteur(s) :** Serveur.
  - **Pré-condition(s) :** On doit avoir deux joueurs de même niveau dans la file d'attente.
  - **Post-condition(s) :** On a bien deux joueurs pour la partie multijoueur.
  - **Cas particulière(s) :** Si on ne trouve pas d'adversaire, on retourne au menu principal.

4.7.2 Choix du vainqueur

Le vainqueur est le joueur ayant remporté le plus de points et donc rempli le plus de lignes de la grille.

5 Besoins non fonctionnels

5.1 Utilisateur

Il a des besoins essentiels qui permettent une expérience utilisateur des plus agréables tout au long de sa période de jeu.

- L'utilisateur est déconnecté après une inactivité trop étendue.
- Le jeu en ligne se veut réactif en évitant le moins de latence possible.
- L'interface se veut épurée rendant l'utilisation très interactive.

5.2 Système

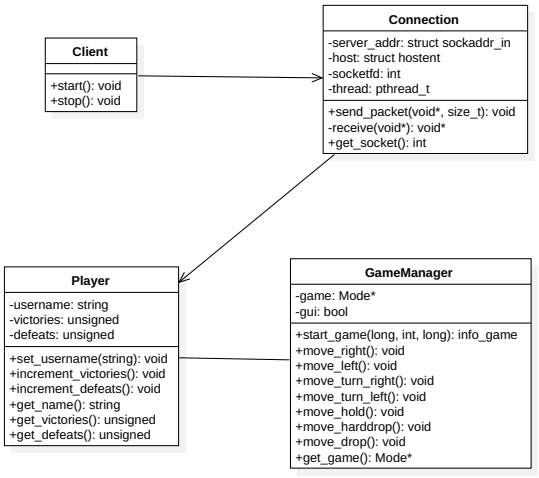
- Le code se veut facilement maintenable.
- Le projet sera codé dans le langage C++, un langage orienté objet qui a été exigé par les assistants.
- Pour le stockage de données, nous avons privilégié le langage mysql.
- La gestion des données sensibles des utilisateurs doivent être pour la plupart, cryptées afin d'assurer une sécurité permanente.
- Le jeu peut s'exécuter sur plusieurs types de machines de façon à garder une expérience minimale acceptable.

6 Design du système

Le système comprend un client et un serveur.

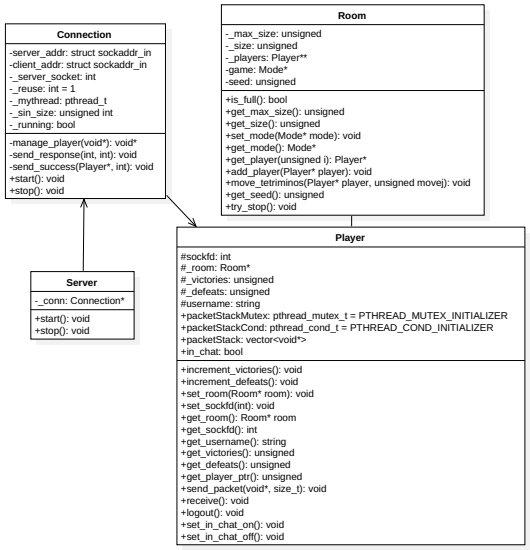
6.1 Client

Voici le diagramme de classes du client. Il représente la structure uniquement. Le serveur contient également les fonctionnalités se rapportant au système d'amis, une partie du chat et les paramètres des touches de jeu. Ces fonctionnalités ne faisant pas partie vraiment de la structure, elles ne sont pas représentées dans ce diagramme.



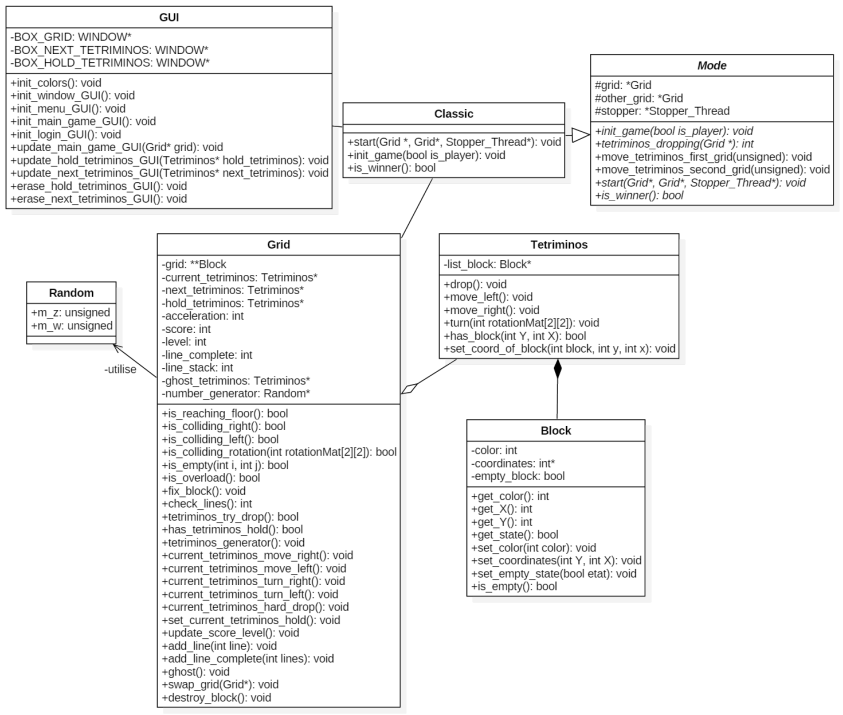
6.2 Serveur

Voici le diagramme de classes du serveur.



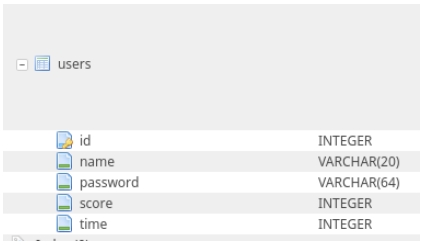
6.3 Jeu

Voici un diagramme de classes représentant le jeu. Le mode classique peut être remplacé par n'importe quel mode. Les getters et setters ont parfois été omis dans les classes par soucis de clarté du schéma.

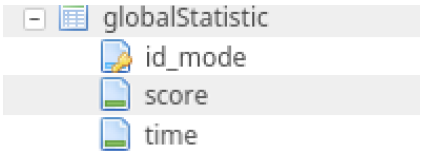


6.4 Base de données

Les données sont stockées dans une base de données SQLite qui contient différentes tables dont la description va suivre. Le choix de SQLite a été fait car il permet d'utiliser le langage SQL. Celui-ci permet de manipuler simplement les données en C++.
Tout d'abord, une table principale users qui stocke tous les noms d'utilisateurs et les mots de passe associés. Les mots de passes sont hachés. Elle contient également le score et le temps



Ensuite, une table globalStatistic qui contient le nombre de victoires et de défaites pour les 4 modes de jeu différents.



Enfin, pour chaque utilisateur, on crée une autre table où seront stockés les amis des utilisateurs ainsi que les demandes d'amis en attente. Un utilisateur peut envoyer une demande d'amis à n'importe quel joueur. Dans la base de données, l'utilisateur qui envoie une demande verra dans sa table une nouvelle ligne nommée "friends" où on trouvera le nom de la personne qui l'a demandé en ami et la ligne "status" sera égale à 0 (demande d'amis). La personne que l'on a demandé en ami aura une nouvelle ligne dans sa table avec le nom de la personne qui l'a demandé en ami et la ligne "status" sera égale à 2. Une fois la demande d'amis acceptée, le "status" de la personne qui a fait la demande d'amis passe à 1.

6.5 Interface

6.5.1 Interface console

Le jeu peut se jouer en mode console. La librairie utilisée est ncurses. C'est la classe GUI qui gère l'affichage des différents menus et du jeu.

6.5.2 Interface graphique