



SOFTWARE REQUIREMENT DOCUMENT

Capitali\$
(INFO-F-209 : Partie 1)

Loïc Blommaert - Hugo Charels - Yacoub Lahdo - Rémy Ryckeboer
Bao Tran - Hà Uyên Tran - Joachim Violon

6 décembre 2022

Table des matières

1 Introduction :	3
1.1 But du projet	3
1.2 Glossaire	3
1.3 Historique	4
2 Besoins utilisateur : Fonctionnels	5
2.1 Écran de connexion	5
2.1.1 Se connecter à un compte	5
2.1.2 Créer un compte	5
2.2 Écran d'accueil	6
2.2.1 Créer et lancer une partie	6
2.2.2 Rejoindre une partie	6
2.2.3 Gérer une liste d'amis	7
2.3 Avant une partie	7
2.3.1 Salon d'attente	7
2.4 Durant une partie	7
2.4.1 Démarrage de la partie	7
2.4.2 Déroulement d'un tour	7
2.4.3 Déplacement du joueur	8
2.4.4 Achat et vente de propriétés et de bâtiments	9
2.4.5 Mise aux enchères	9
2.4.6 Négociation	9
2.4.7 Hypothéquer une propriété	9
2.4.8 Piocher une carte	9
2.4.9 Abandon d'une partie en cours	9
3 Besoins utilisateur : Non fonctionnels	9
3.1 Interactivité, agréabilité et dynamisme	10
3.1.1 Clics et boutons	10
3.1.2 Limitation des temps de réponses	10
3.1.3 Sons et bruitages	10
4 Besoins Système : Fonctionnels	11
4.1 Lancement du jeu	11
4.1.1 Crédit d'un compte	11
4.2 Gestion des profils	11
4.3 Gestion du chat	11
4.4 Créer un salon de jeu	11
4.5 Détruire un salon de jeu	11
4.6 Gestion d'une partie	11
4.6.1 Gestion de l'argent	11
4.6.2 Gestion des propriétés	12
4.6.3 Gestion de la banque	12
4.6.4 Gestion du temps	12
4.6.5 Gestion des actions	13

5 Besoins Système : Non fonctionnels	13
5.1 Système d'exploitation	13
5.2 Réseau	13
5.3 Disponibilité	13
5.4 Performances	13
5.5 Capacité d'une partie	13
5.6 Sécurité	14
5.7 Versions	14
6 Design et fonctionnement du Système	15
6.1 Diagramme de classe	15
6.1.1 Classes du jeu	15
6.1.2 Classes du serveur	17
6.2 Gestion des comptes	18
6.2.1 Création d'un compte	18
6.2.2 Connexion à un compte	18
6.3 En partie	19
6.3.1 Actions liées au jet de dés	19
6.3.2 Arrivée sur une case carte	20
6.3.3 Utilisation du chat	21
6.3.4 Fonctionnement d'une négociation	22

1 Introduction :

Ce document s'adresse à l'ensemble des membres du projet et a pour vocation de regrouper de manière explicite les critères et les besoins du client pour la conception du jeu Capitali\$.

1.1 But du projet

L'objectif du projet est de réaliser une version électronique du jeu Capitali\$. C'est un jeu de plateau (formé de 40 cases contiguës et constituant un cycle) dans lequel les joueurs, chacun leur tour, sont amenés à s'enrichir et faire tomber en faillite leurs adversaires. Chaque tour, les joueurs peuvent réaliser certaines actions. Celles-ci sont soit obligatoires comme lancer les dés pour avancer, payer le loyer d'une propriété ou piocher une carte, soit facultatif comme acheter des propriétés ou construire des bâtiments. Un capital de départ est alloué à chaque joueur et il existe plusieurs moyens de gagner ou perdre de l'argent durant la partie. Lorsqu'un joueur ne peut plus payer son dû, il est éliminé, et le dernier joueur en lice sera le vainqueur. Pareillement, si la durée fixée de la partie est écoulée, le joueur le plus riche est le vainqueur.

Il existe 3 types de cases achetables : les propriétés (regroupées par couleur), les gares et le compagnies de service public.

Deux modes de jeu seront disponibles :

- Classique : Il n'y a pas de limite de temps, la partie dure jusqu'au moment où il reste un seul joueur.
- Rapide : Il y a une limite de temps et certaines règles spécifiques permettant d'accélérer la partie.

En dehors d'une partie, le joueur a la possibilité de créer un compte et de discuter avec d'autres joueurs possédant un compte.

1.2 Glossaire

Définition des termes, des mots ou des acronymes qui seront rencontré au sein de ce document :

- **Database** : une base de donnée, un ensemble de donnée structurée
- **Faire un double** : avoir les deux dés qui indiquent le même nombre
- **Terrain** : cases du plateau achetables
- **Interactivité** : le taux de communication entre plusieurs utilisateurs, ici les joueurs entre eux, et le programme
- **Agréabilité** : à quel point un objet, ici le programme, possède des caractéristiques rendant l'utilisation plaisante
- **MAJ** : Mise À Jour
- **Pseudo** : Pseudonyme, nom choisi pour désigner un compte par l'utilisateur
- **Salon de jeu** : Sous-serveur gérant les parties des joueurs

1.3 Historique

Version	Modifications	Auteurs	Date
0.1	Diagrammes de classes : classe du jeu	Hugo Charels Hà Uyên Tran Rémy Ryckeboer	12/11/2022
0.2	Use case écran d'accueil et connexion	Rémy Ryckeboer	18/11/2022
0.3	Diagrammes de classes : classe du jeu	Hugo Charels	18/11/2022
0.4	Diagramme de séquences : chat	Loïc Blommaert Yacoub Lahdo	20/11/2022
0.5	Use case : tour d'un joueur	Joachim Violon	21/11/2022
0.6	Besoin utilisateur fonctionnels : Écran de connexion et Écran d'accueil MAJ	Rémy Ryckeboer	22/11/2022
0.7	Diagrammes de classes : classes serveur et client	Hugo Charels	23/11/2022
0.8	Besoins utilisateur fonctionnels + Besoins système	Bao Tran	25/11/2022
0.9	Diagramme de séquences : enregistrement et connexion de l'utilisateur	Loïc Blommaert Yacoub Lahdo	26/11/2022
0.10	Diagrammes de classe : classes du jeu, serveur, client, interface graphique	Hugo Charels	26/11/2022
0.11	Besoins systèmes fonctionnels	Bao Tran	27/11/2022
0.12	Design et fonctionnement du système et déroulement d'un tour	Hà Uyên Tran	27/11/2022
0.13	Use case : déroulement d'un tour MAJ	Joachim Violon	27/11/2022
0.14	Diagrammes de séquences : gestion de comptes et en partie	Loïc Blommaert Yacoub Lahdo	27/11/2022
0.15	Corrections et améliorations des textes	Loïc Blommaert Hà Uyên Tran Yacoub Lahdo	28/11/2022

2 Besoins utilisateur : Fonctionnels

2.1 Écran de connexion

La figure 1 montre comment l'utilisateur va pouvoir se créer un compte et par la suite s'y connecter. L'utilisateur a donc deux choix d'action : soit il se connecte à son compte, soit il en crée un.

2.1.1 Se connecter à un compte

S'il décide de se connecter le programme lui demandera simplement de s'identifier avec un nom d'utilisateur ainsi que son mot de passe, si l'un de ses deux paramètres n'est pas donné l'authentification échouera forcément. De la même manière si le programme ne trouve aucune correspondance entre le nom d'utilisateur rentré et ceux présents dans la database, l'authentification n'aboutira pas. Pour finir si le mot de passe n'est pas correct et ne correspond pas à celui associé au nom d'utilisateur, l'authentification échouera également. Si aucun des 3 cas cité précédemment ne se produit, alors l'utilisateur se connecte à son compte.

2.1.2 Créer un compte

S'il décide de créer un compte, il devra impérativement rentrer un nom d'utilisateur unique ainsi qu'un mot de passe. Si le nom d'utilisateur apparaît déjà dans la database, la création du compte ne se fera pas, et l'utilisateur sera amené à trouver un autre nom d'utilisateur. Dans le cas où le nom n'existe pas encore dans la database, la création du compte aura bien lieu, et la connexion à celui-ci se fera automatiquement.

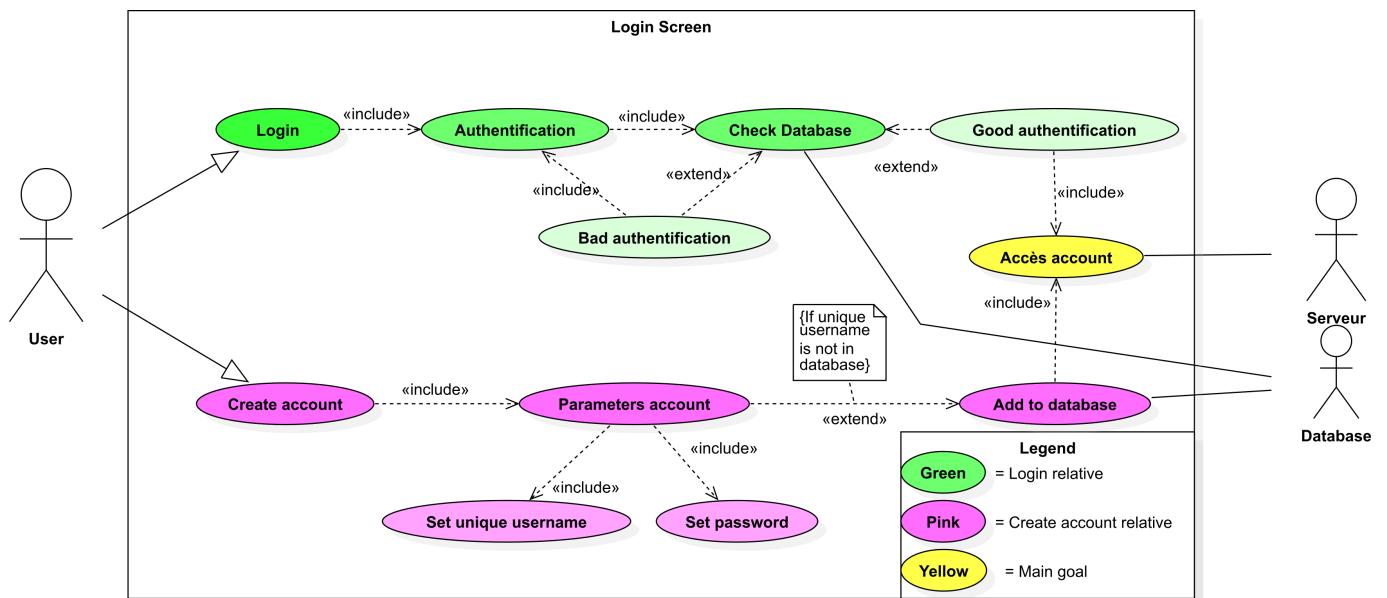


FIGURE 1 – Use Case : Connexion

2.2 Écran d'accueil

La figure 2 montre ce que l'utilisateur va pouvoir faire sur le menu d'accueil du programme. Trois grandes catégories se démarquent : la création d'une partie, la connexion à une partie déjà créée et l'accès à sa liste d'amis.

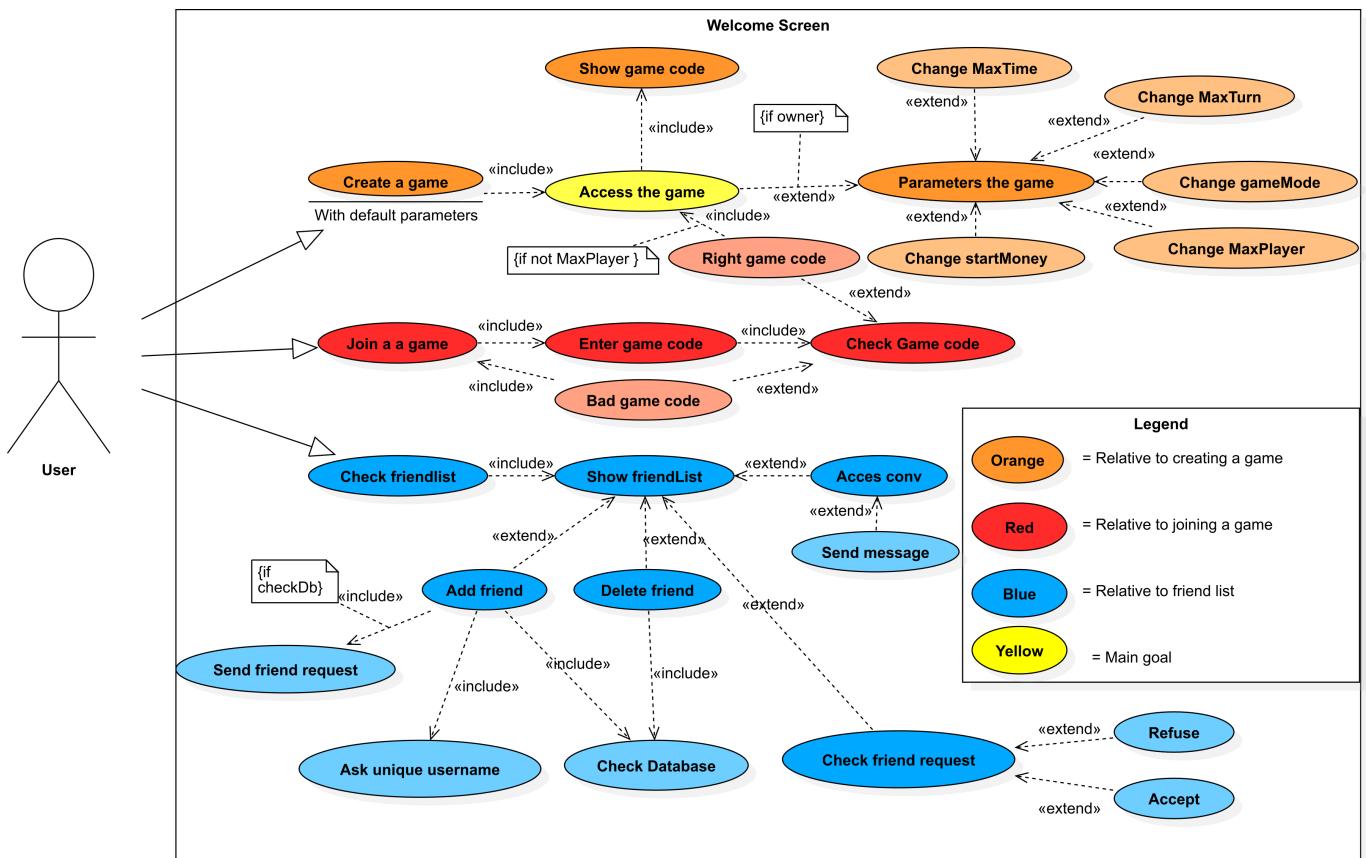


FIGURE 2 – Use Case : Menu Principale

2.2.1 Créer et lancer une partie

Le joueur peut créer une partie en interagissant avec le programme, une fois créée, le joueur se retrouve dans un salon d'attente où il a la possibilité de modifier les paramètres de la partie. Une partie aura toujours des paramètres par défaut modifiables, tant que celle-ci n'a pas commencée. Lorsque la partie est créée, elle est aussi associée à un port visible à l'écran. Si au moins deux joueurs sont présents dans la partie, Le créateur de la partie peut lancer la partie.

2.2.2 Rejoindre une partie

Pour rejoindre une partie, il faut rentrer un identifiant de partie. Si celui-ci est correct alors l'utilisateur sera automatiquement connecté à la partie, dans un salon d'attente, si

1. le nombre maximum de joueur de celle-ci n'est pas encore atteint.

2. la partie n'a pas été lancée

Si le code est incorrect, un message d'erreur s'affiche à l'écran, et l'utilisateur peut rentrer un nouveau code.

2.2.3 Gérer une liste d'amis

Accéder à sa liste d'amis permet de voir quels utilisateurs sont amis avec nous, et de consulter leur profil. A partir de là on peut décider de gérer sa liste d'amis en ajoutant un ami via son nom d'utilisateur, ou de supprimer un ami en consultant son profil. Il est possible d'accéder à une conversation privée avec ses amis.

2.3 Avant une partie

2.3.1 Salon d'attente

A la création d'une partie, les joueurs peuvent rejoindre un salon d'attente où chaque joueur peut sélectionner une icône permettant de le représenter sur le plateau, pour la version avec interface graphique. Le créateur de la partie a aussi la possibilité de modifier les paramètres modifiables de la partie.

2.4 Durant une partie

2.4.1 Démarrage de la partie

Au démarrage, l'ordre dans lequel les joueurs jouent est déterminée au lancement du jeu, de manière aléatoire, et chaque joueur possède la somme d'argent définie précédemment.

2.4.2 Déroulement d'un tour

La figure 3 explicite les différentes possibilités d'interaction qu'un joueur peut effectuer durant son tour de jeu. Si le joueur est en prison durant son tour, il doit essayer de s'en retirer. Si le joueur est libre, il a le choix de faire des échanges, ou faire des transactions avec ses propriétés. Ensuite, il jette des dés pour avancer et il subit l'action de sa case d'arrivée.

Les adversaires (qui constituent des joueurs) sont aussi amenés à interagir à certains événements provoqués en dehors de leur tour, soit aux enchères, soit en cas d'endettement. Le jeu lui propose de faire des ventes de ses propriétés pour éviter cela.



FIGURE 3 – Use Case : Déroulement d'un tour pour un joueur

2.4.3 Déplacement du joueur

Le joueur doit obligatoirement se déplacer chaque fois que c'est son tour. Pour ce faire, il interagira avec le programme pour lancer les dés, dont la somme indiquera au joueur le nombre de case qu'il va avancer. Si le joueur est en prison, le déplacement ne s'applique que si le joueur fait un double, ou dans tous les cas au quatrième tour en prison.

Si le joueur fait un double au lancé de dé, il peut relancer les dés après le déplacement du premier lancé de dé. Au troisième double consécutif, le joueur est envoyé en prison.

2.4.4 Achat et vente de propriétés et de bâtiments

Chaque fois qu'un joueur tombe sur une case propriété, gare ou compagnie de service public, si celle-ci n'est pas possédée par un autre joueur, et que le joueur possède assez d'argent, il a la possibilité d'acheter la case. S'il n'achète pas le terrain, celui-ci est mis aux enchères.

Si le joueur tombe sur une case propriété, il peut faire construire des bâtiments dessus si :

- il possède le terrain et les autres terrains qui ont la même couleur
- il a assez d'argent

2.4.5 Mise aux enchères

Lorsqu'un terrain est mis aux enchères, durant maximum 30 secondes, les joueurs adverses pourront proposer des prix (qui doivent obligatoirement augmenter après chaque mise) auxquels ils souhaiteraient racheter le terrain. Une fois le temps écoulé, le joueur ayant misé la plus grande somme possédera désormais le terrain.

Si un joueur ne souhaite pas ou plus participer aux enchères, il a la possibilité de le faire savoir. Si tous les joueurs présents initialement aux enchères sauf un se retirent des enchères, celles-ci s'arrêtent automatiquement, peut-importe le temps qu'il reste

2.4.6 Négociation

Tout au long de la partie, un joueur aura la possibilité de lancer une négociation avec un autre joueur pendant son tour, pour potentiellement racheter un de ses terrains.

2.4.7 Hypothéquer une propriété

Lorsqu'un joueur n'a plus assez d'argent pour s'acquitter d'une dette, celui-ci à la possibilité de sélectionner des terrains qu'il voudrait hypothéquer, sinon il pourra ignorer l'action.

2.4.8 Piocher une carte

Chaque fois, qu'un joueur tombe sur une case carte, il doit piocher une carte.

Il y a deux types de cartes :

- Les cartes chance
- Les cartes communauté

Les cartes sont utilisées instantanément sauf la carte sortie de prison dont l'utilisation sera proposée à chaque tour de prison. Les cartes possèdent toutes des effets différents, et lorsqu'une carte est utilisée, celle-ci est directement remise en dessous de la pile de carte correspondante.

2.4.9 Abandon d'une partie en cours

Lors d'une partie en cours, si le joueur quitte la partie (ou perd sa connexion), il a une minute pour se reconnecter. Dans le cas où il ne se reconnecte pas pendant ce laps de temps, il sera considéré alors comme étant en faillite.

3 Besoins utilisateur : Non fonctionnels

Le jeu doit proposer une interface agréable et intuitive permettant au joueur d'interagir facilement avec les différents outils du jeu. Le joueur ne doit pas avoir un temps d'attente trop long à chaque action qu'il fait en partie (lancer les dés, acheter un bâtiment, ...) ou en dehors de la partie (créer un compte, créer une partie, ...).

3.1 Interactivité, agréabilité et dynamisme

3.1.1 Clics et boutons

Les utilisateurs interagissent plus facilement en cliquant sur des boutons avec une souris qu'en écrivant du texte sur leur clavier. Pour la version avec interface graphique, la présence de bouton sera fortement employée pour rendre le jeu plus interactif, agréable et intuitif.

3.1.2 Limitation des temps de réponses

Pour que le jeu soit dynamique et qu'un joueur ne bloque pas la partie en étant inactif, des quanta de temps limités pour chaque action seront accordées avant que celles-ci ne soient exécutées automatiquement.

3.1.3 Sons et bruitages

Pour une expérience plus agréable, la présence de sons et bruitages pourrait-être souhaitable

4 Besoins Système : Fonctionnels

4.1 Lancement du jeu

Au lancement du jeu, le joueur peut soit créer un compte, soit se connecter pour pouvoir accéder aux fonctionnalités du jeu. S'il a déjà un compte, le serveur doit détecter si le pseudo entré par le joueur pour s'y connecter existe et que le mot de passe introduit correspond au pseudo mis.

4.1.1 Crédation d'un compte

Au lancement, le jeu doit permettre aux joueurs n'ayant pas de compte d'en créer un. Pour créer un compte, le joueur doit insérer un nom d'utilisateur et un mot de passe. Le système doit détecter si le pseudo inséré par le joueur existe déjà ou pas. Si c'est le cas, le système doit afficher un message d'erreur indiquant que le pseudo est déjà utilisé par quelqu'un d'autre.

Si l'identifiant n'est pas encore utilisé, le serveur crée un compte et connecte le joueur.

4.2 Gestion des profils

Le serveur doit permettre aux utilisateurs de changer son pseudo et le mot de passe de son compte. Il faut aussi qu'il puisse gérer sa liste d'amis.

4.3 Gestion du chat

Le serveur doit permettre aux utilisateurs de communiquer, en parallèle du jeu, à tout moment de la partie.

4.4 Créer un salon de jeu

Le serveur doit gérer la création de salons, et permettre au joueur responsable de sa création de paramétriser la partie. Il peut changer la limite de temps, choisir le mode de jeu, le nombre de joueur, etc... Après cette phase, chaque joueur peut choisir une icône pour le représenter sur le plateau.

4.5 Détruire un salon de jeu

Le serveur doit supprimer tout salon de jeu qui ne contiendrait plus de joueur mais aussi fermer correctement tous les salons en cours si on vient à arrêter le serveur.

4.6 Gestion d'une partie

4.6.1 Gestion de l'argent

Chaque joueur reçoit en début de partie une certaine somme d'argent en fonction du mode de jeu. Le joueur peut gagner ou perdre de l'argent.

Les différents moyens de gagner de l'argent

- Le joueur gagne de l'argent chaque fois qu'il passe sur la case départ
- Le joueur gagne de l'argent chaque fois qu'un autre joueur passe sur ses propriétés. Il gagne de l'argent en fonction du rang de la propriété possédée.
- Le joueur peut gagner de l'argent grâce aux cartes.

Les différents moyens de perdre de l'argent

- Le joueur perd de l'argent (20\$) à chaque tour en partie rapide.
- Le joueur perd de l'argent s'il tombe sur une case taxe.
- Le joueur perd de l'argent s'il tombe sur la propriété d'un autre joueur. L'argent qu'il doit payer varie en fonction du rang et du nombre de bâtiments adverses.
- Le joueur peut perdre de l'argent à cause des cartes.
- Le joueur a le choix de payer s'il veut sortir de prison ou s'il veut acheter un terrain ou un bâtiment.

4.6.2 Gestion des propriétés

Il y a 3 types de cases achetables. Chaque fois qu'un joueur passe sur une de ces cases, il a la possibilité de l'acheter si celle-ci est encore libre.

Les cases propriétés : Il y a 22 cases terrains formant, par groupe de 2 ou de 3, des ensembles de couleur sur le plateau. Le prix d'achat de ces cases sont croissants par rapport à la case départ. Le jeu doit permettre aux joueurs de placer des bâtiments sur une case seulement si toutes les cases d'une même couleur sont possédées par ce même joueur. En début de partie rapide, il faut que le jeu distribue aléatoirement deux terrains à chaque joueur, ceux-ci doivent directement payer le prix des deux terrains.

Les cases gares : Il y a 4 cases gares se trouvant au milieu de chaque côté du plateau. Elles ont toutes le même prix. Le loyer de ces cases est proportionnel au nombre de gares possédées par le joueur.

Les cases compagnies de service public : Il y a 2 de ces cases sur le plateau. Pour savoir le prix du loyer, il faut que le joueur, qui doit payer, lance les dés. Le prix du loyer est le résultat du lancer, multiplié par 5 si le propriétaire de la case n'en possède qu'une et par 12 sinon.

4.6.3 Gestion de la banque

La banque est une entité indépendante ayant comme rôle de :

- gérer la totalité de l'argent et les propriétés qui ne sont pas encore achetés par les joueurs.
- verser le salaire et les bonus des cartes.
- encaisser l'argent des taxes, des amendes, des achats des terrains non achetés et des achats des bâtiments.
- prêter de l'argent sur des propriétés hypothéquées.
- encaisser de l'argent venant des joueurs (20\$) à chaque tour dans le mode rapide.

Celle-ci possède de l'argent à l'infini, elle ne pourra donc jamais tomber en faillite

4.6.4 Gestion du temps

Dans une partie classique, il n'y pas de limite de temps sur la durée de la partie contrairement à la partie rapide, où il y a une limite de temps après laquelle le joueur le plus riche gagne. Pour calculer la fortune d'un joueur, la valeur de revente de ses bâtiments et la valeur d'hypothèque de ses terrains non hypothéqués doivent être rajoutés à son argent.

Il y a une limite de temps à chaque tour d'un joueur, celle-ci peut être définie par le joueur à la création de la partie. Le chronomètre indiquant le temps restant au joueur se déclenche après les déplacements du joueur. Le joueur peut décider d'arrêter son tour et permettre au joueur suivant de jouer avant la fin du chronomètre. Si le joueur est en train de faire une action à la fin du temps imparti, celle-ci se fait interrompre. Les actions qui peuvent nécessiter de l'utilisateur pour un but uniquement interactif, seront aussi soumises à une limite de temps. Si celles-ci ne sont pas effectuées avant la fin du temps, le système effectuera automatiquement l'action pour le joueur (exemple : cliquer pour lancer les dés)

4.6.5 Gestion des actions

A chaque tour, le jeu ne doit permettre qu'au joueur dont c'est le tour d'interagir avec le système (sauf dans les situations exceptionnelles décrites en section 2, comme les négociations ou les enchères). Le chat n'est pas considéré par ce point.

En plus du lancer de dés et du déplacement, le joueur peut faire plusieurs actions comme :

- négocier avec les autres joueurs.
- poser des bâtiments sur ses terrains (voir 4.6.2)
- vendre ses bâtiments ou hypothéquer.

Le seul moment où le jeu ne doit pas permettre au joueur de faire ces actions, c'est quand celui-ci est en prison. Dans cette situation, le jeu doit donner la possibilité au joueur d'utiliser une carte pour sortir de prison si celui-ci en possède une.

5 Besoins Système : Non fonctionnels

5.1 Système d'exploitation

Le jeu doit être exécutable sur Linux

5.2 Réseau

Le joueur doit pouvoir jouer en multijoueur sur un réseau local.

5.3 Disponibilité

Le jeu est accessible tant que la connexion entre le client et le serveur est assurée. Le serveur doit être manuellement lancé par un utilisateur, qui sera responsable de son exécution.

5.4 Performances

Le jeu doit être fluide avec un temps de réponse sous l'ordre des millisecondes en conditions normales.

5.5 Capacité d'une partie

Il y a 6 joueurs maximum par partie.

5.6 Sécurité

Pour pouvoir accéder à une partie, le joueur doit se connecter à un compte grâce à un mot de passe. Il faut que le joueur ne puisse pas tricher dans une partie, le joueur ne doit pas pouvoir faire une action qui enfreignent les règles du jeu.

5.7 Versions

Une première version du jeu sera jouable uniquement dans le terminal. Une seconde version mettra à jour le logiciel pour lui donner une interface graphique. Les deux versions devront-être compatible

6 Design et fonctionnement du Système

Ce point reprendra des diagrammes de classe et de séquence développant le fonctionnement du système durant l'utilisation du joueur

Note : La communication entre le serveur et le client, pour l'affichage d'informations à l'écran dans les diagrammes de séquences, sont implicitement considérés. Lorsque le serveur envoie un message à un utilisateur, il est sous-entendu que l'utilisateur interagira avec le programme.

6.1 Diagramme de classe

6.1.1 Classes du jeu

La figure 4 est le diagramme de classe de la partie jeu du système, c'est à dire celles qui concernent la gestion d'une partie, le plateau, les joueurs. Le jeu Capitalist contient les Player et le Board. Il y a une classe Obtainable qui généralise les éléments pouvant être obtenus par le joueur, et faisant des actions sur les joueurs. La classe Property (représentant les propriétés) hérite donc de Obtainable tout comme la class Card (représentant les cartes).

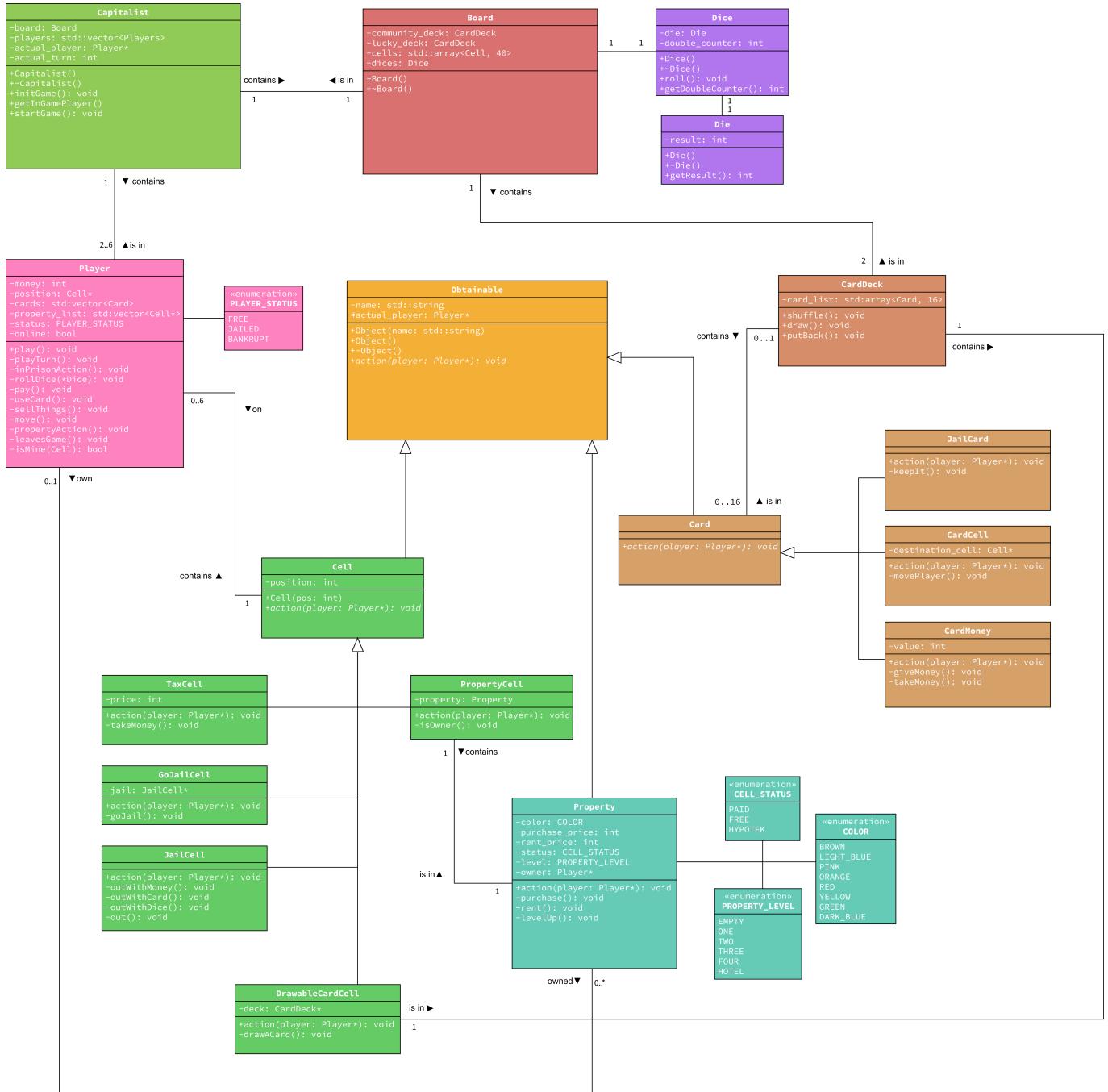


FIGURE 4 – Diagramme de classe

6.1.2 Classes du serveur

La figure 5 est le diagramme de classe de la partie serveur et client du système qui est divisé en six parties.

- L'interface de connexion (bleu menthe)
- La base de donnée (mauve)
- La classe du Jeu vu dans la figure 4 (vert)
- Le serveur (bleu clair)
- Le système de messagerie (bleu)
- Le client (rouge)

En bas, il y a MainUi correspondant à l'interface du jeu, Game.

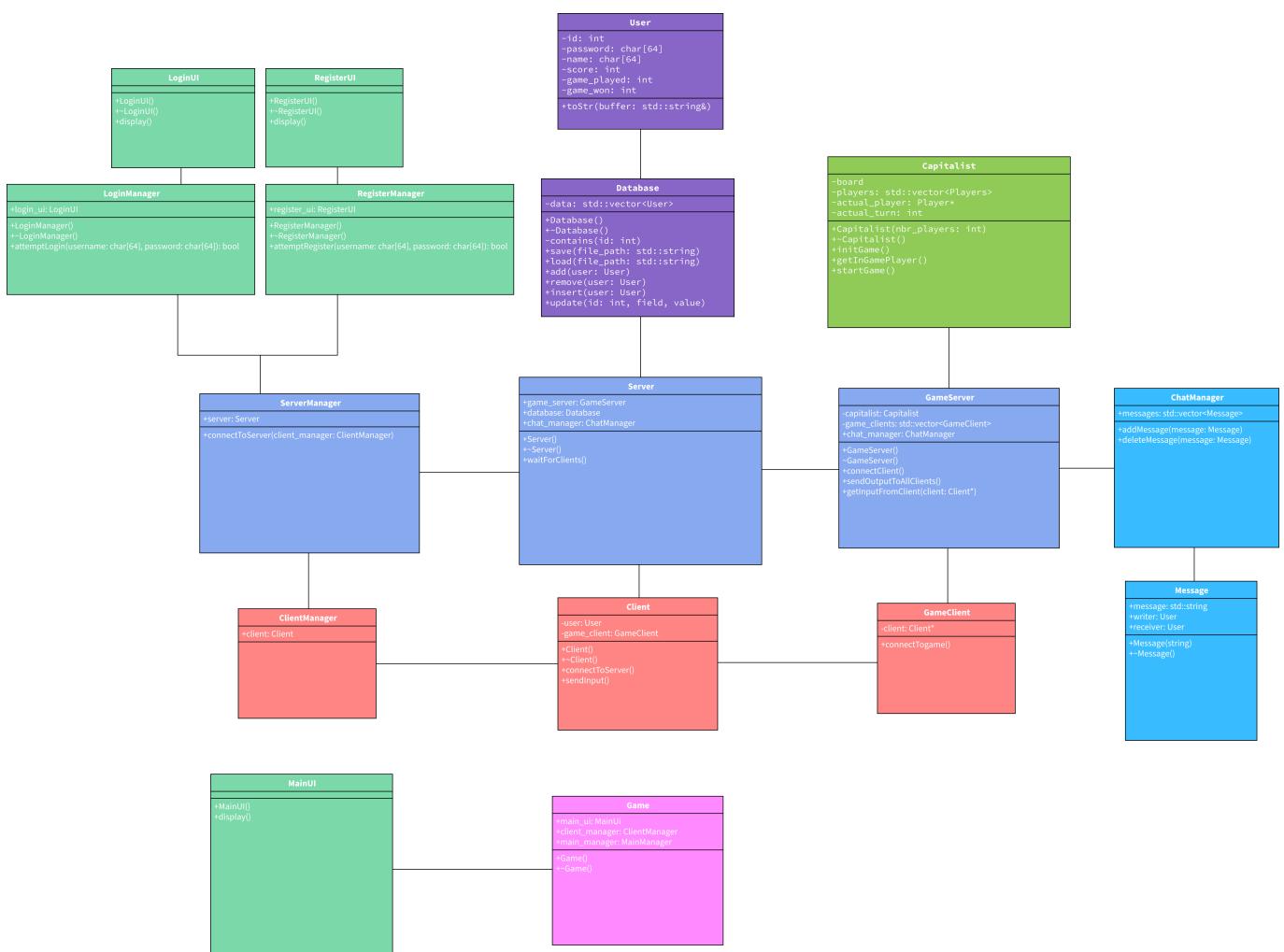


FIGURE 5 – Diagramme de classe du serveur

6.2 Gestion des comptes

6.2.1 Crédation d'un compte

Ce diagramme nous montre la séquence d'action qui est réalisée lorsqu'un utilisateur voudrait créer un compte. En premier lieu il y aura une vérification des données envoyées par l'utilisateur par le SignInManager (vérifier les caractères spéciaux, champs vide, ...). Si un compte au même nom existe déjà, un message sera envoyé à l'utilisateur pour lui avertir, sinon un compte sera créé et l'utilisateur aura accès au jeu.

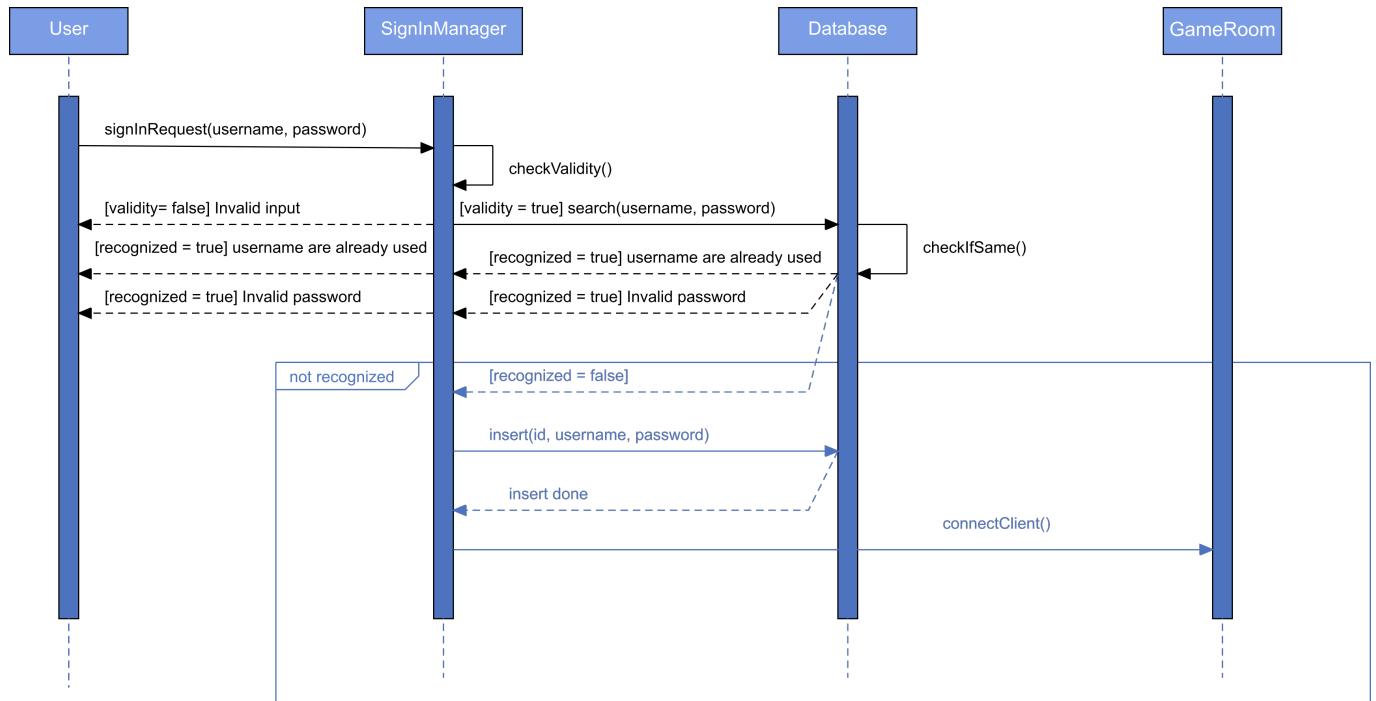


FIGURE 6 – Diagramme de séquence : Crédation d'un compte

6.2.2 Connexion à un compte

Ce diagramme nous montre la séquence d'actions qui est réalisée lorsqu'un utilisateur voudrait se connecter à son compte. Le LoginManager s'occupera de vérifier les informations reçues par le joueur. Si toutes les informations reçues sont correctes, le joueur aura accès au jeu.

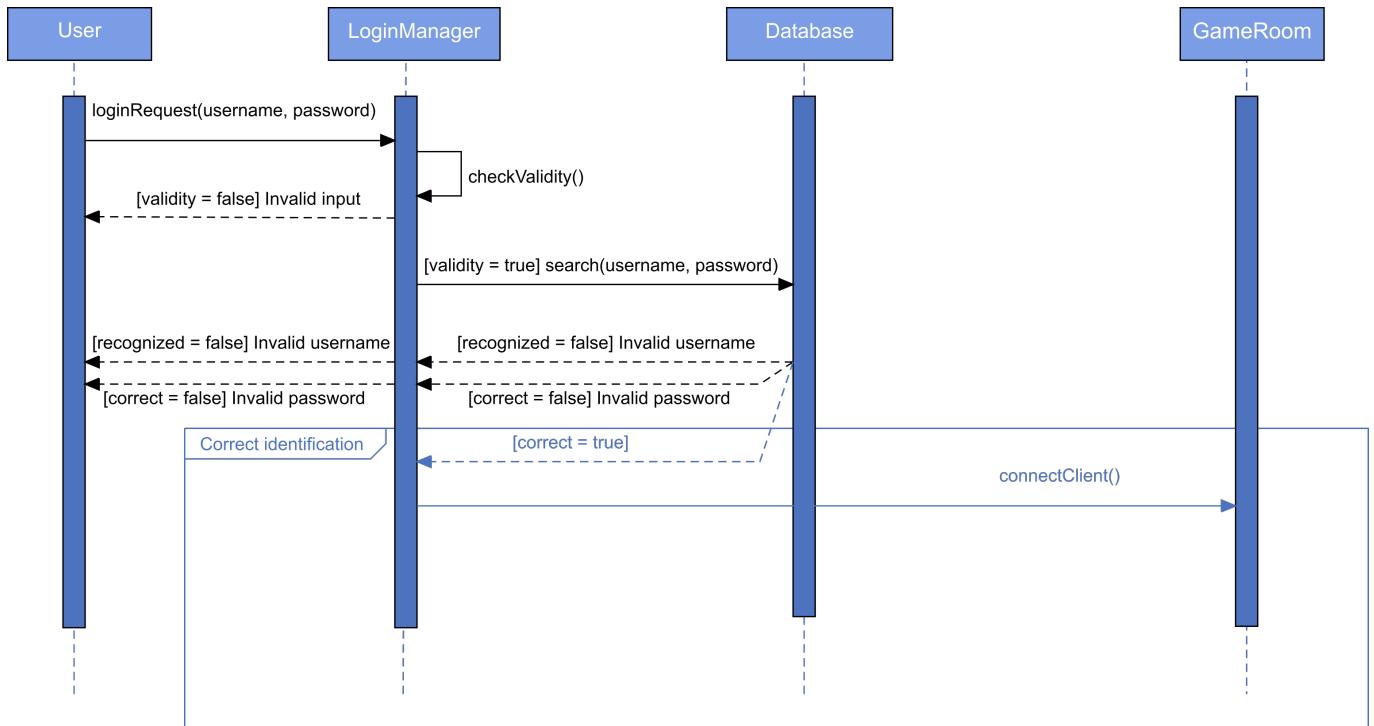


FIGURE 7 – Diagramme de séquence : Login de l'utilisateur dans une partie de jeu

6.3 En partie

6.3.1 Actions liées au jet de dés

Ce diagramme nous montre la séquence d'actions provoquée par un lancer de dés, débouchant à un déplacement du joueur. Le joueur arrivera sur une case selon la valeur des dés, et subira son action. Si c'est une case propriété libre et que le joueur ne décide pas de l'acheter, des enchères sur cette propriété sont lancées pour le reste des joueurs.

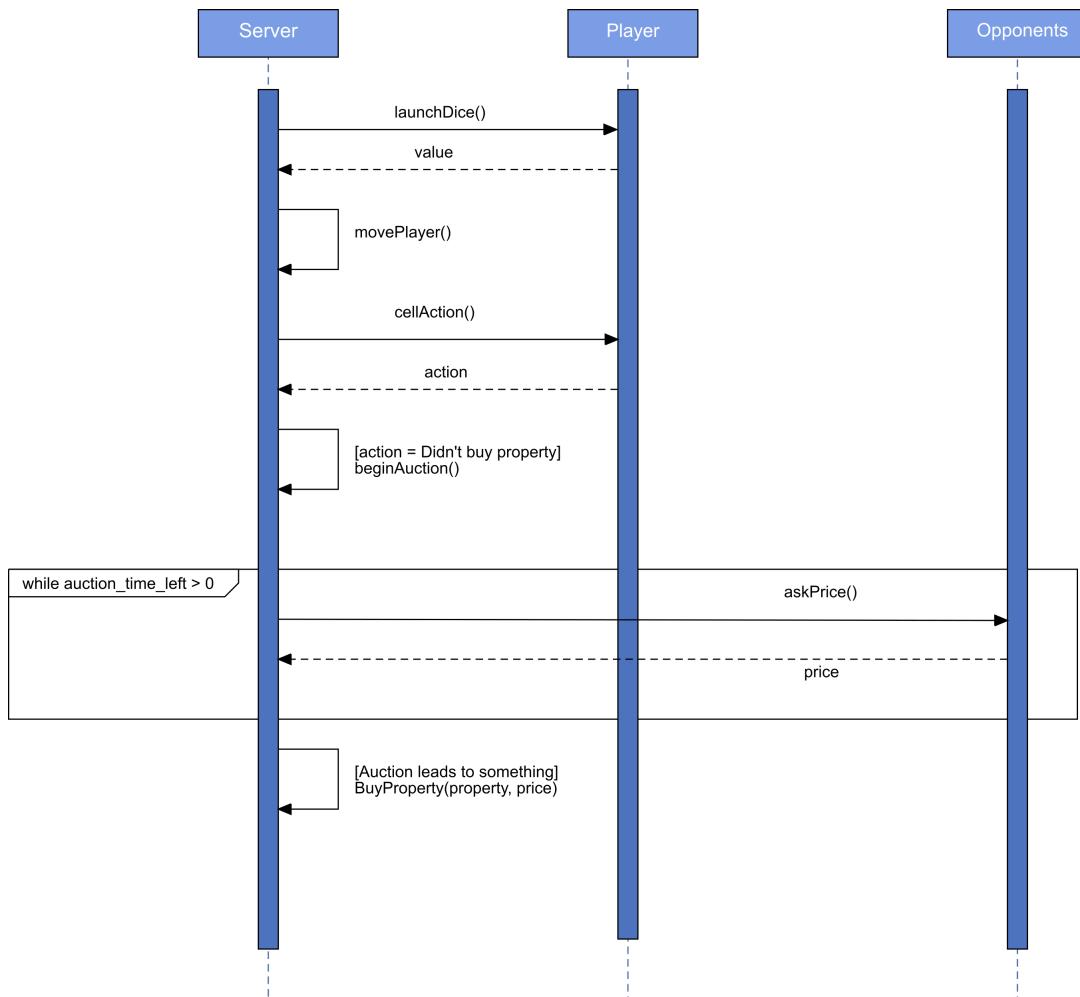


FIGURE 8 – Diagramme de séquence : Actions liées au jet de dés

6.3.2 Arrivée sur une case carte

Ce diagramme nous montre la séquence d'actions se réalisant lorsque le serveur se charge d'un utilisateur qui pioche une carte. Si à l'issue de l'action de la carte, l'utilisateur n'a plus assez d'argent, il devra revendre une ou plusieurs propriétés. S'il n'a toujours pas assez d'argent, il tombera en faillite et tous les autres joueurs seront prévenus.

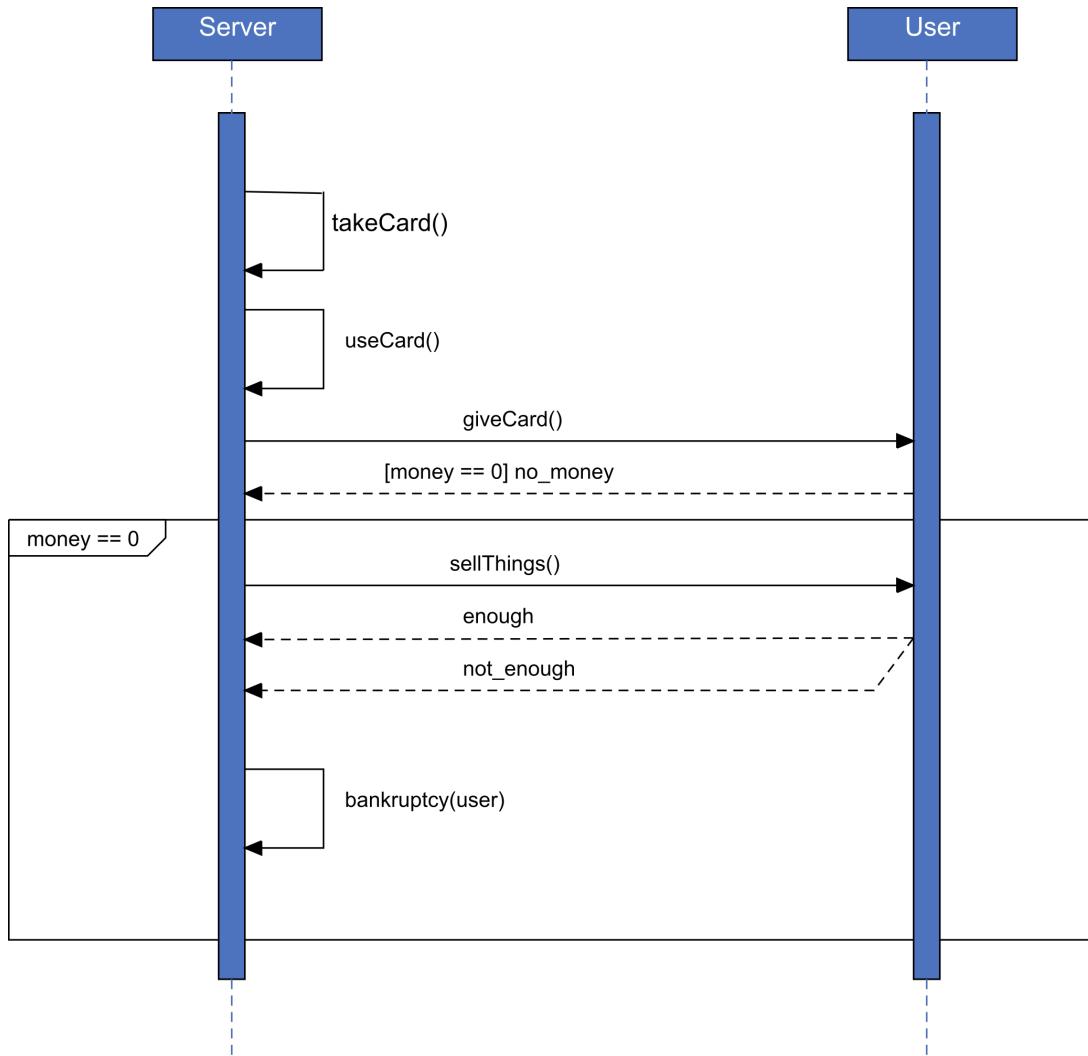


FIGURE 9 – Diagramme de séquence : Piocher une carte

6.3.3 Utilisation du chat

Ce diagramme nous montre la séquence d'actions se réalisant lorsque le serveur réagit quand un message est envoyé dans le chat de la partie. Le ChatManager s'occupe de recevoir les messages envoyés et de les filtrer. Après ce filtre, il envoie une notification au serveur pour lui prévenir qu'un message a été envoyé par un utilisateur. Le serveur récupère le message et l'affiche à tous les autres joueurs dans la partie.

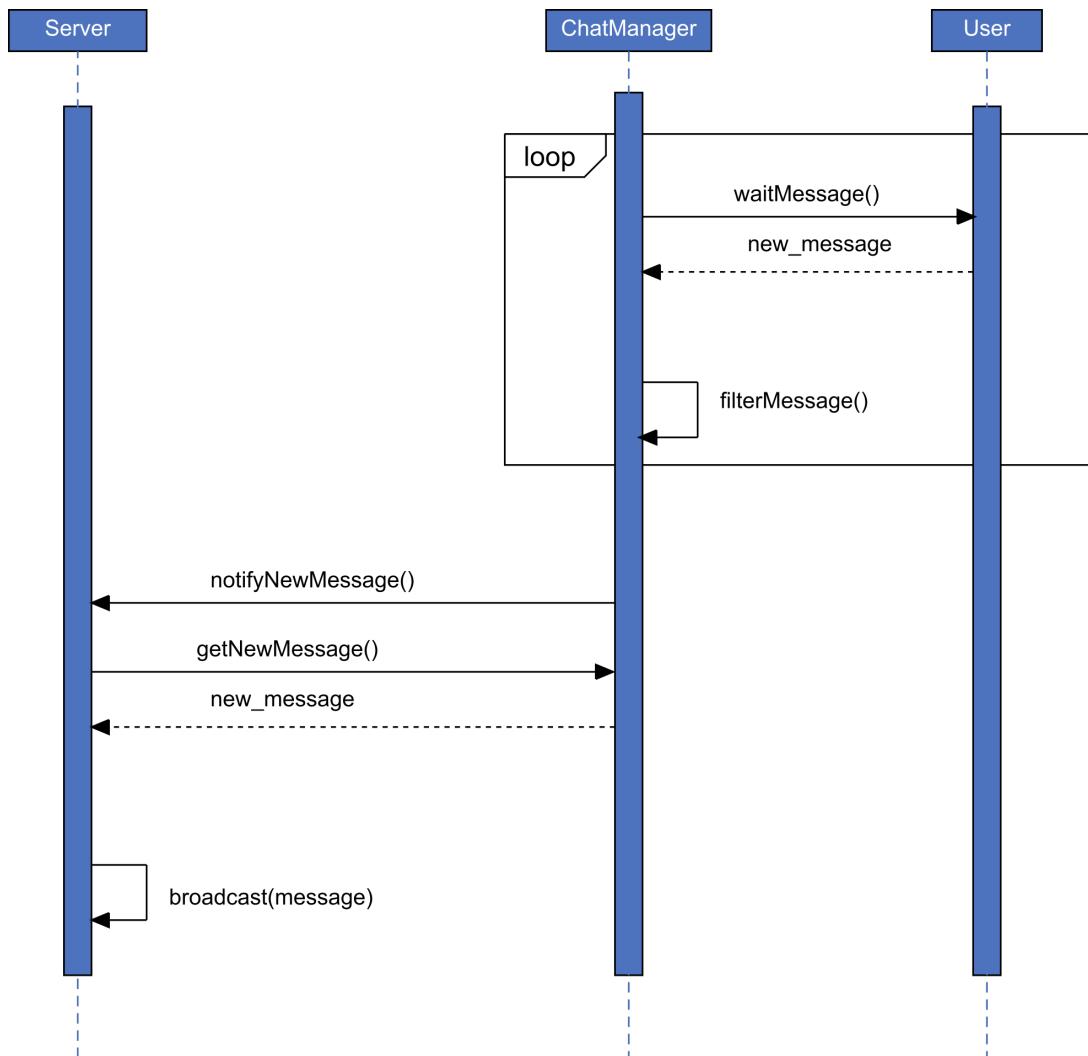


FIGURE 10 – Diagramme de séquence : Communication dans le chat (miaou)

6.3.4 Fonctionnement d'une négociation

Ce diagramme nous montre la séquence d'actions lorsqu'une négociation est lancée. Le joueur responsable de la négociation doit d'abord choisir un joueur avec qui négocier, suivit de la propriété à négocier et le prix de rachat proposé. Ces choix sont confirmés au serveur et envoyés au joueur adverse, qui peut accepter ou refuser l'offre.

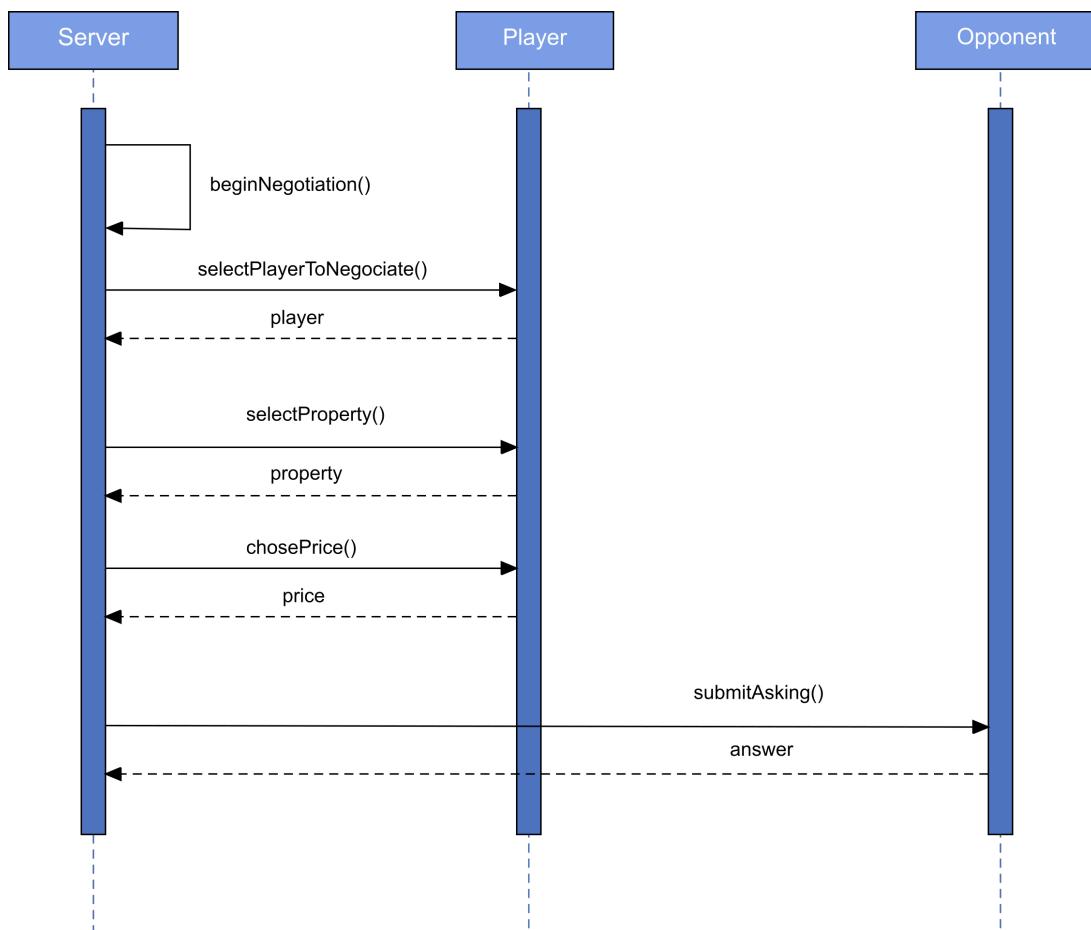


FIGURE 11 – Diagramme de séquence : Fonctionnement d'une négociation