

# Software Requirements Document: White House Defense

Groupe 4

December 19, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	But du projet . . . . .	3
1.2	Glossaire . . . . .	4
1.3	Historique . . . . .	4
<b>2</b>	<b>Besoins utilisateur: Fonctionnels</b>	<b>5</b>
2.1	Connexion . . . . .	5
2.2	Menu principal . . . . .	5
2.2.1	Lancement d'une partie . . . . .	6
2.2.2	Gérer son profil . . . . .	6
2.2.3	Consulter le profil d'un autre utilisateur . . . . .	6
2.2.4	Consulter le classement . . . . .	6
2.2.5	Supporter un autre joueur (Mode spectateur) . . . . .	6
2.3	Gérer une liste d'amis . . . . .	6
2.4	Chat . . . . .	6
2.5	En partie . . . . .	7
2.5.1	Achat/vente des tours . . . . .	7
2.5.2	Placer des tours . . . . .	7
2.5.3	Abandon d'une partie en cours . . . . .	7
<b>3</b>	<b>Besoins utilisateur: Non fonctionnels</b>	<b>7</b>
<b>4</b>	<b>Besoin système: Fonctionnels</b>	<b>8</b>
4.1	Connexion . . . . .	8
4.2	Gestion des comptes . . . . .	8
4.2.1	Création d'un compte . . . . .	8
4.2.2	Suppression d'un compte . . . . .	8
4.3	Gestion des profils . . . . .	8
4.3.1	Affichage des profils . . . . .	8
4.3.2	Modifier son profil . . . . .	9
4.3.3	Consulter les classements . . . . .	9
4.4	Créer une partie . . . . .	9
4.4.1	Matchmaking . . . . .	9
4.5	Gestion d'une partie: . . . . .	10
4.5.1	Gestion des pnj . . . . .	10
4.5.2	Gestion de l'argent . . . . .	11
4.5.3	Gestion du score . . . . .	11
4.5.4	Gestion de la difficulté . . . . .	12
4.5.5	Choix du vainqueur de la partie . . . . .	12
<b>5</b>	<b>Besoins système: Non fonctionnels</b>	<b>12</b>
5.1	Système d'exploitation . . . . .	12
5.2	Réseau . . . . .	12
5.3	Disponibilité . . . . .	12
5.4	Performances . . . . .	12
5.5	Capacité . . . . .	12
5.6	Sécurité . . . . .	12
5.7	Robustesse . . . . .	13
<b>6</b>	<b>Design et fonctionnement du Système</b>	<b>13</b>
6.1	Design du système . . . . .	13
6.1.1	Explications complémentaires . . . . .	13
6.2	Fonctionnement du système . . . . .	15
6.2.1	Inscription . . . . .	15
6.2.2	Connexion . . . . .	16
6.2.3	Chat . . . . .	16

6.2.4	Matchmaking: Client . . . . .	17
6.2.5	Matchmaking: Serveur . . . . .	17
6.2.6	Diagramme d'activité : Matchmaking . . . . .	18
6.2.7	Boucle du jeu: Client . . . . .	18
6.2.8	Boucle du jeu: Serveur . . . . .	19

# 1 Introduction

## 1.1 But du projet

L'objectif de ce projet est de réaliser un jeu de type "Tower Defense". L'objectif d'un jeu de ce type est simple : Un joueur doit défendre sa base face à des attaques successives de vagues de pnj. Dans le cadre de ce projet il nous est demandé de réaliser une version multi-joueurs. Plusieurs joueurs s'affronteront donc en même temps sur la même carte.

Le thème de notre jeu est Donald Trump, président des États-Unis. Notre objectif sera donc de rappeler au travers du jeu différents éléments qui ont marqué sa campagne dans une optique humoristique. Les pnj seront donc représentés notamment par des mexicains, des journalistes et des terroristes. La base d'un joueur sera représentée par la maison blanche.

Pour pouvoir jouer, l'utilisateur doit se connecter à l'aide d'un pseudonyme. Il faut donc s'être inscrit au préalable afin de pouvoir jouer à *White House Defense*.

Trois modes de jeu sont disponibles : classique, par équipe et contre la montre. Un mode "extra" supporter est également présent. Dans ce mode, l'utilisateur observe une partie tout en faisant profiter un joueur de plus d'argent que les autres.

En dehors d'une partie, un joueur a la possibilité de discuter avec ses amis, de personnaliser son profil et de consulter le profil d'un autre joueur.

Au cours d'une partie, un joueur doit placer des tours de manière stratégique sur la carte pour vaincre les pnj attaquant sa base. Pour acquérir des tours, le joueur a à sa disposition de l'argent qu'il dépense dans une boutique virtuelle. Tous les modes de jeu se jouent à quatre joueurs, la carte sera coupée en 4, chaque partie étant identique.

## 1.2 Glossaire

- PNJ: personnage non jouable.
- Vague: ensemble de PNJ.
- Matchmaking: système qui met en relation des joueurs qui souhaitent jouer au même moment, de façon à créer une partie et lancer un jeu.
- Pseudo: nom d'emprunt utilisé par les utilisateurs.
- Frame: interval de temps pendant lequel l'état du jeu est mis à jour et une nouvelle image est dessinée.

## 1.3 Historique

Numéro de version	Nom	Modifications	Date
0.1	David Engelman/Benjamin Engelman	Besoins utilisateur	3/12/16
0.2	Gregory Bonaert	Besoins systeme	4/12/16
0.3	David Engelman	Gestion pnj + gestion argent	4/12/16
0.4	Benjamin Engelman	Gestion fin de partie et score	8/12/16
0.5	Benjamin Engelman/Edouard Gavrioloia	Besoin utilisateur: Non fonctionnels	11/12/16
0.6	David Engelman	But du projet + détail besoin utilisateur	11/12/16
0.7	Benjamin Engelman/David Engelman	Ajout use case diagram : utilisateur + système (connexion, matchmaking)	17/12/16
0.8	Jurgen Gjini	Ajout use case diagram : besoin systeme	17/12/16
0.9	Gregory Bonaert	Ajout du glossaire	18/12/16
0.10	Gregory Bonaert	Gestion de la difficulté + Classements + Revente des tours	18/12/16
0.11	David Engelman	Inclusion diagrammes séquences	18/12/16
0.12	Benjamin Engelman	commentaires diagrammes séquences	18/12/16
0.13	David Engelman	ajouts commentaires diagrammes séquences	18/12/16
0.14	David Engelman/Benjamin Engelman	design système	18/12/16
0.15	Arthur Pierrot	activity diagram matchmaking	19/12/16
0.16	Edouard Gavrioloia/Alexandre van Hauwaert	besoin système: non-fonctionnels	19/12/16

## 2 Besoins utilisateur: Fonctionnels

### 2.1 Connexion

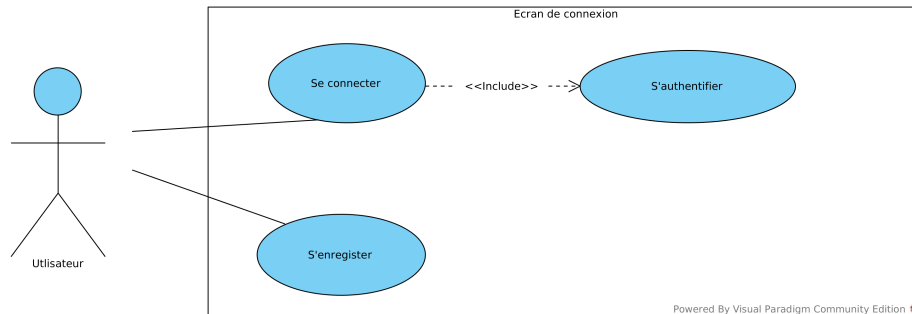


Figure 1: Use case : Connexion

Lors du lancement du jeu, une fenêtre de connexion s'ouvre pour permettre à l'utilisateur de se connecter à son compte. Si celui-ci ne possède pas encore de compte, il a la possibilité de s'en créer un. Si l'utilisateur entre des données erronées le programme le lui signale. Une fois l'authentification réussie l'utilisateur arrive dans le menu principal.

### 2.2 Menu principal

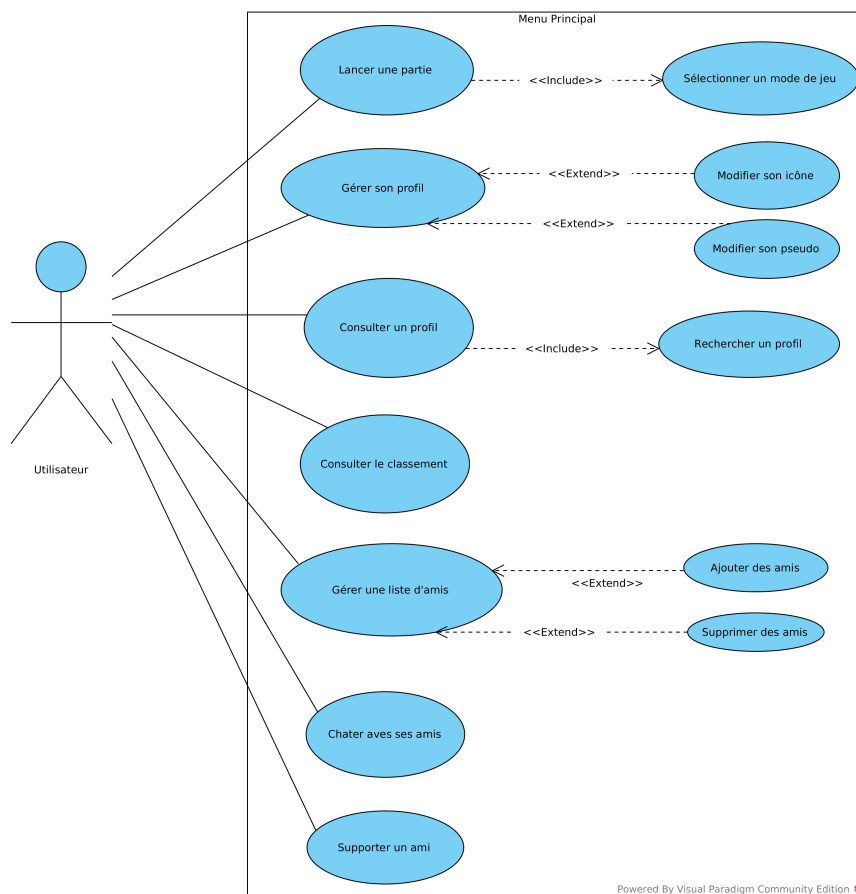


Figure 2: Use case : Menu principal

### 2.2.1 Lancement d'une partie

Le joueur a la possibilité de choisir parmi plusieurs modes de jeu:

- Mode classique
- Mode contre la montre
- Mode par équipe

Une fois un mode de jeu sélectionné, le joueur est placé dans une file d'attente en attendant que suffisamment de joueurs soient disponibles. Quand une partie est trouvée le programme invite le joueur à accepter la partie. Si le joueur décline il n'est pas replacé en file d'attente.

### 2.2.2 Gérer son profil

Chaque utilisateur possède un profil qu'il peut personnaliser. Il a la possibilité de:

- Modifier son pseudo : le joueur est invité à saisir un nouveau pseudo. Si celui-ci est disponible son pseudo est modifié. Si il est indisponible le programme le lui signale et le joueur est invité à recommencer.
- Modifier son icône : le joueur choisit une icône parmi une collection proposée.

Le profil d'un joueur contient également le nombre de parties jouées et le nombre de victoires dans les différents modes de jeu.

### 2.2.3 Consulter le profil d'un autre utilisateur

Le joueur peut consulter un profil en recherchant un pseudo. Le profil du joueur recherché sera alors affiché si il est trouvé. Sinon un message d'erreur sera affiché

### 2.2.4 Consulter le classement

Le joueur peut consulter le classement des différents modes de jeu. Trois classements différents seront donc consultables:

1. Le classement du mode classique
2. Le classement du mode contre la montre
3. Le classement du mode par équipe

Après sélection d'un classement, celui-ci est affiché. Dans chacune de ces catégories, les joueurs sont classés selon leur nombre de victoires.

### 2.2.5 Supporter un autre joueur (Mode spectateur)

Le joueur a la possibilité de se connecter à une partie en cours en tant que spectateur et de choisir un joueur qu'il supporte.

## 2.3 Gérer une liste d'amis

Dans le menu, le joueur peut consulter sa liste d'amis. Des amis sont des autres joueurs ayant été ajoutés à cette liste. Il a donc la possibilité d'ajouter et de supprimer des amis.

## 2.4 Chat

Si le joueur possède des amis en ligne dans sa liste d'amis il peut discuter avec eux au travers d'un chat. Si le joueur est déconnecté d'internet, ses messages ne seront plus envoyés et le programme lui affichera un message pour le lui signaler.

## 2.5 En partie

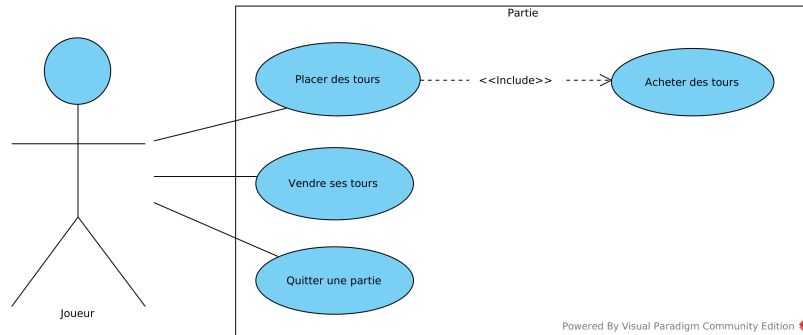


Figure 3: Use case : En partie

### 2.5.1 Achat/vente des tours

Si le joueur possède l'argent nécessaire, celui-ci peut acheter différents types de tours et les placer sur la carte, l'argent dépensé lui est alors décompté. Si le joueur possède des tours sur la carte, celui-ci peut également revendre des tours déjà placées, la somme correspondante au prix de la tour lui est alors reversée.

### 2.5.2 Placer des tours

Si le joueur a acheté des tours il peut les placer sur des emplacements de la carte réservés à cet effet.

### 2.5.3 Abandon d'une partie en cours

Lors d'une partie en cours, le joueur a la possibilité de quitter une partie en cours. Dans ce cas une défaite lui sera attribuée.

## 3 Besoins utilisateur: Non fonctionnels

Le jeu doit proposer une interface agréable et facile d'utilisation. Les parties doivent être rapides. En effet, le temps d'une partie ne devrait pas dépasser 10 minutes. De plus quand le joueur recherche une partie, celui-ci ne devrait pas devoir attendre plus de 5 minutes dans la file d'attente (sauf cas exceptionnel). Bien entendu, l'accès au jeu est gratuit.



## 4 Besoin système: Fonctionnels

### 4.1 Connexion

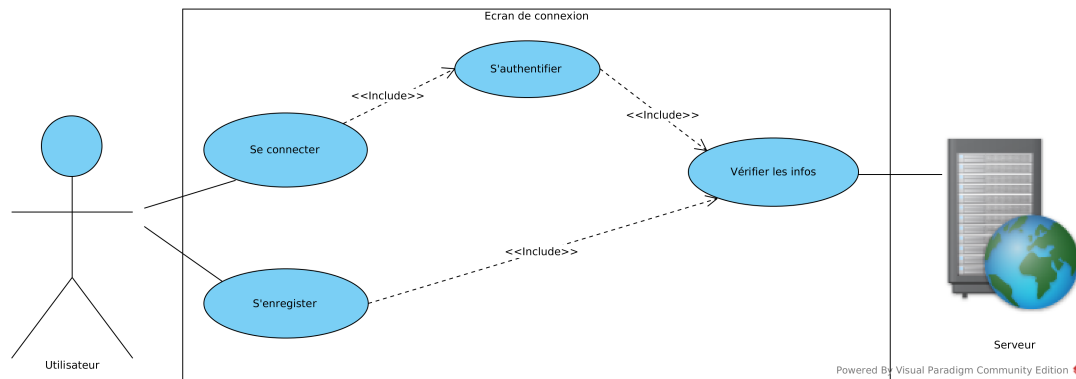


Figure 4: Use case : Connexion system

Au démarrage du jeu, l'utilisateur peut s'authentifier via un formulaire qui lui demande son nom d'utilisateur et son mot de passe. Les données entrées sont envoyées au serveur pour que celui ci effectue des vérifications.

Si il introduit des données erronées, un message d'erreur renvoyé par le serveur est affiché. Sinon, le joueur est connecté à son compte et arrive dans le menu principal du jeu.

### 4.2 Gestion des comptes

#### 4.2.1 Création d'un compte

Au démarrage du jeu, le joueur peut créer un nouveau compte. Un formulaire sera affiché qui lui demande un nom d'utilisateur et un mot de passe.

Le client envoie le pseudo désiré et le mot de passe au serveur. Si le pseudo n'est pas disponible, le serveur renvoie un code d'erreur, qui permet au client d'afficher message d'erreur à l'utilisateur.

Si le pseudo est disponible et correct, le serveur crée le compte et l'indique au client, sinon le programme renverra une message d'erreur. Si tout c'est bien passé, le joueur peut maintenant se connecter à son compte comme expliqué à la section 3.1.

#### 4.2.2 Suppression d'un compte

Sur son profil, le joueur a la possibilité de supprimer son compte. Avant d'envoyer un message au serveur pour que celui ci supprime le compte, le client affiche un message de confirmation pour être certain que l'utilisateur veut bien supprimer son compte. Si l'utilisateur confirme, un message est envoyé au serveur pour lui demander de supprimer ce joueur de la base de données. Le compte est alors supprimé. Le joueur est déconnecté.

### 4.3 Gestion des profils

#### 4.3.1 Affichage des profils

Le joueur peut voir les informations sur des profils (le sien ou celui d'autres utilisateurs). Le client envoie alors une requête au serveur pour que celui ci lui fournisse les informations du profil recherché par l'utilisateur. Le serveur renvoie les informations suivantes:

- Son nom d'utilisateur/pseudo
- Son icône

- Le nombre de parties jouées
- Le nombre de victoires pour différents modes de jeu

Si l'utilisateur veut afficher le profil d'un joueur qui n'existe pas, le serveur renvoie un message d'erreur qui est affiché par le client.

#### 4.3.2 Modifier son profil

Le joueur a la possibilité de modifier certaines informations de son profil:

- Son icône
- Son nom d'utilisateur/pseudo

Pour vérifier la disponibilité du pseudo, le client envoie le pseudo désiré au serveur, qui lui indique si le nom est disponible. Si il n'est pas disponible, le serveur renvoie un message d'erreur que le client affiche à l'utilisateur. Celui ci est alors invité à proposer un autre pseudo.

#### 4.3.3 Consulter les classements

Le joueur peut voir le classement de meilleurs joueurs. Le client envoie alors une requête au serveur, qui lui renvoie les informations suivantes:

- Le pseudo du joueur
- Son score dans les différents modes de jeu
- Le nombre de parties jouées

### 4.4 Créer une partie

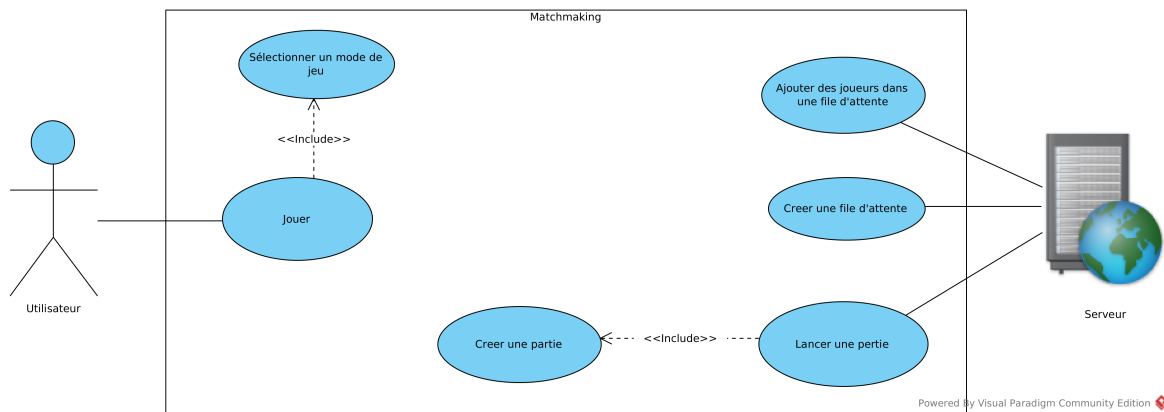


Figure 5: Use case : Matchmaking

#### 4.4.1 Matchmaking

Lorsqu'un joueur demande de se connecter à un partie, le serveur sera responsable du match-making.

1. Si il n'y a aucune partie en attente de joueurs, le serveur crée un nouvelle partie et ajoute le joueur à cette partie
2. Si il y a une partie en attente de joueurs, le serveur rajoute le joueur à cette partie. Au cas où il y a assez de joueurs, la partie est lancée et est retirée de la liste des parties en attente.

## 4.5 Gestion d'une partie:

### 4.5.1 Gestion des pnj

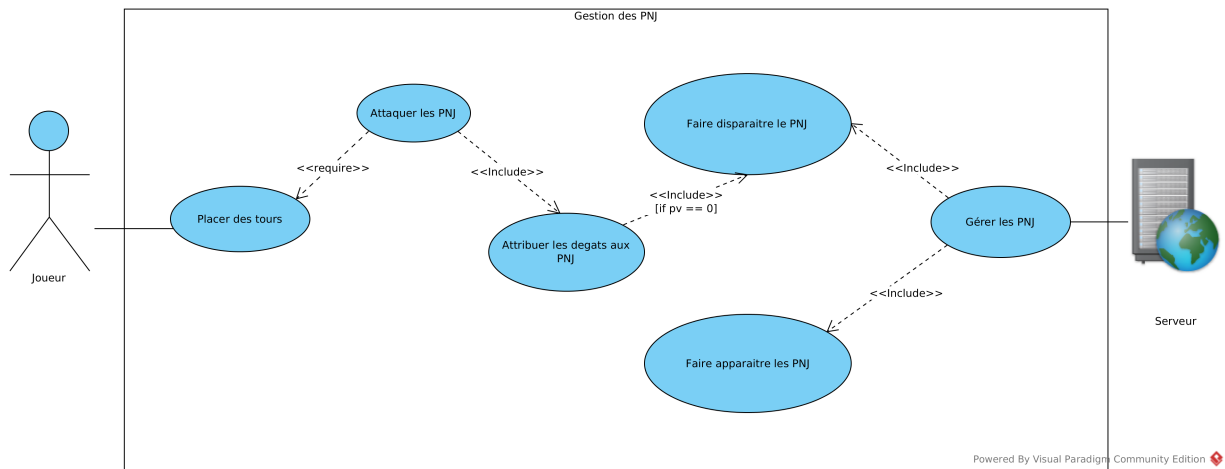


Figure 6: Use case : Gestion des PNJ

### Apparitions des pnj

A chaque début de vague, le serveur fait apparaître les pnj au centre de la carte. Les vagues de tous les joueurs sont composés du même nombre de pnj. Tous les joueurs ont donc à faire à des vagues de niveau identique.

### Vie des pnj

Les pnj possèdent un total de points de vie, celui ci étant initialement égal à 5. Chaque fois qu'un pnj est touché par une tourelle celui-ci se voit infligé des dégâts réduisant son nombre de points de vie. Le nombre de dégâts qui lui sont infligés dépend du type de la tours l'ayant touché. Une fois qu'un pnj a perdu tous ses points de vie celui disparaît de la carte.

### Capacités des pnj

Un pnj peut tout simplement avancer sur la carte. Son objectif est de se rendre dans la base du joueur. Un pnj ne possède pas de moyen de faire des dégâts.

### 4.5.2 Gestion de l'argent

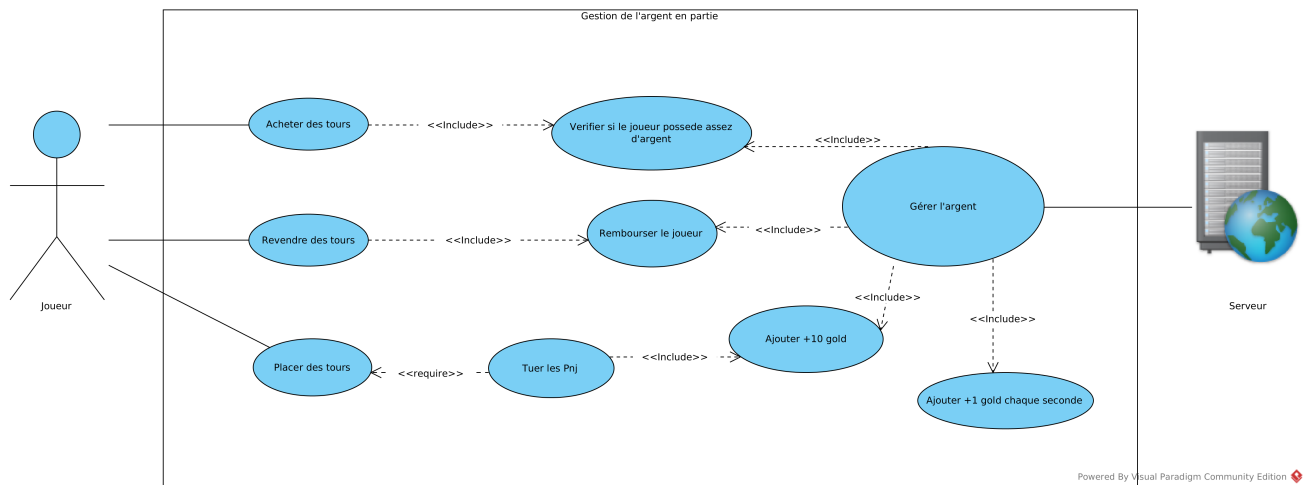


Figure 7: Use case : Gestion de l'argent

#### Gain d'argent

Chaque joueur possède en début de partie une certaine somme d'argent. Le joueur gagne de l'argent de 2 manières différentes:

1. Toutes les 1 seconde le joueur gagne +1 d'argent. Entre les différents vagues le joueur ne gagne pas d'argent
2. Le joueur gagne +10 d'argent pour chaque pnj tué

#### Dépenser de l'argent

Le joueur à la capacité de dépenser son argent pour acheter des tours et donc améliorer son système de défense. Ces achats se font au travers d'une boutique.

Une boutique est une fenêtre comportant un listing des différentes tours disponibles à l'achat ainsi que leurs prix. Le joueur peut donc choisir parmi cette liste ce qu'il souhaite acheter. Lors de l'achat d'un objet un message est envoyé au serveur pour que celui puisse décompter la somme dépensée de l'argent possédé par le joueur. Si un joueur tente d'acheter un objet sans posséder l'argent nécessaire un message d'erreur lui est renvoyé par le serveur.

#### Récupérer de l'argent

Le joueur peut revendre des tours qu'il a déjà placées, récupérant son argent qu'il peut réutiliser pour construire des tours ailleurs. Il récupère exactement la somme qu'il a dépensé lors de l'achat de la tour.

### 4.5.3 Gestion du score

#### Dans le mode contre la montre

Le but du mode de jeu contre la montre est de tuer le plus de pnj dans un temps imparti. Le score d'un joueur correspond donc au nombre de pnj tué.

#### Dans le mode classique et par équipe

Dans ces deux modes de jeu, le score est également le nombre de pnj tués mais le score n'intervient pas dans le choix du vainqueur.

#### **4.5.4 Gestion de la difficulté**

Au fil de temps, la difficulté de chaque vague augmente progressivement. Le serveur du jeu contrôle cette difficulté de 2 façons différentes:

1. La quantité de PNJ par vague
2. Les types de PNJ présents dans la vague: certains PNJ ont plus de points de vie, les rendant plus résistants aux attaques des tours et augmentant la difficulté de la vague

Pour que la partie ne dure pas trop longtemps, la difficulté augmente relativement rapidement.

#### **4.5.5 Choix du vainqueur de la partie**

Lorsqu'une partie se termine, le serveur doit choisir un gagnant. Ce choix dépend du mode de jeu.

##### **Choix du vainqueur dans le mode contre la montre**

C'est le joueur avec le plus grand score qui est désigné comme vainqueur, tous les autres joueurs sont alors désignés comme perdants.

##### **Choix du vainqueur dans le mode de jeu classique**

Le dernier joueur encore en vie est désigné comme vainqueur, tous les autres sont alors désignés comme perdant.

##### **Choix du vainqueur dans le mode de jeu par équipe**

Les joueurs de la dernière équipe encore en vie sont tous désignés comme vainqueur. Les joueurs de toutes les autres équipes sont tous désignés comme perdants.

## **5 Besoins système: Non fonctionnels**

### **5.1 Système d'exploitation**

Le jeu doit être exécutable sur le système d'exploitation «Linux».

### **5.2 Réseau**

Le jeu se joue en réseau, une connexion internet est requise.

### **5.3 Disponibilité**

Pour se connecter, le serveur doit être en ligne. S'il est en ligne, il est accessible sans autres conditions.

### **5.4 Performances**

Les temps de rafraîchissement et les temps de réponses sont de l'ordre des millisecondes.

### **5.5 Capacité**

Le jeu peut gérer quatre joueurs actifs par partie ainsi qu'un nombre indéfini de spectateurs.

L'espace de stockage nécessaire pour une partie est de l'ordre des Megabytes.

### **5.6 Sécurité**

Pour participer à une partie, un joueur doit se connecter à son compte. Pour cela, il doit s'authentifier en introduisant un pseudo et un mot de passe corrects.

## 5.7 Robustesse

Si un joueur se déconnecte du serveur, le jeu est mis en pause un certain temps avant de déclarer le joueur en question perdant.

# 6 Design et fonctionnement du Système

## 6.1 Design du système

Le Système est divisé en 8 entités:

1. Connexion et Inscription (en rose)
2. Gestion des comptes (en gris)
3. Menu principal (en brun)
4. Profil (en jaune)
5. Classement (en mauve)
6. Matchmaking (en beige)
7. Chat (en vert)
8. Jeu (en bleu)

### 6.1.1 Explications complémentaires

Globalement, chaque entité du système est composée de 3 parties principales:

1. L'interface: Elle sert uniquement à afficher les informations et à recevoir les actions de l'utilisateur qu'elle communique au manager.
2. Le manager: Il gère les requêtes envoyées par l'interface et communique avec le serveur pour échanger des informations.
3. Le serveur: Il se charge d'envoyer les informations demandées par le manager et permet également de mettre en relation plusieurs utilisateurs. En partie, le serveur est le maître du jeu.

En partie, chaque client possède son propre gameState. Le serveur en possède un également, c'est le gameState de référence. Le gameManager est chargé de communiquer avec le GameServer pour mettre à jour les gameStates des clients.

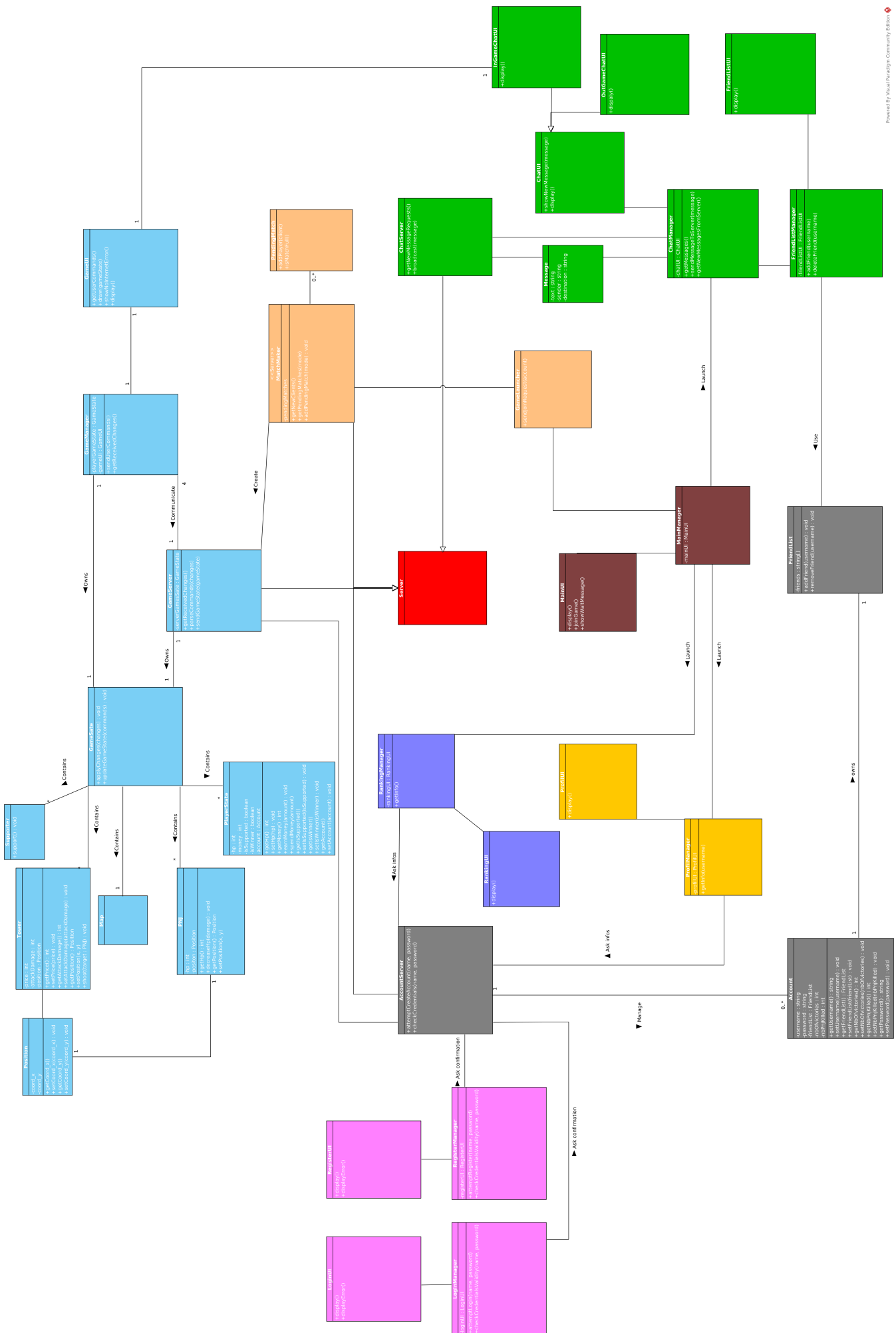


Figure 8: Diagramme de classe

## 6.2 Fonctionnement du système

Voici les diagrammes de séquence illustrant le fonctionnement de différentes parties du système.

### 6.2.1 Inscription

Ce diagramme décrit la séquence d'action réalisées lorsqu'un utilisateur veut se créer un compte. Les données entrées par l'utilisateur seront une première fois vérifiées par le registerManager (vérification de champs vide, nombre de caractères ...) avant des les envoyer au serveur. Celui vérifiera que le nom d'utilisateur est disponible.

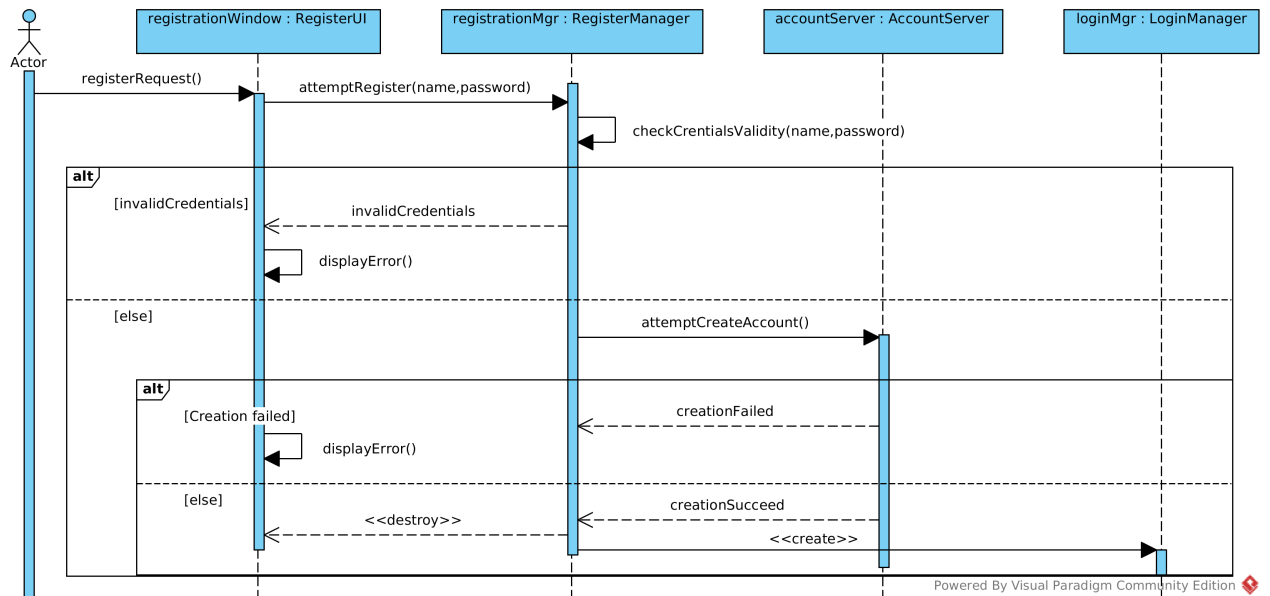


Figure 9: Sequence : Inscription



### 6.2.2 Connexion

Ce diagramme décrit la séquence d'actions réalisées lorsqu'un utilisateur veut se connecter à son compte utilisateur. Comme pour l'inscription, les données sont d'abord vérifiées en interne avant de les envoyer au serveur. Celui-ci vérifiera cette fois-ci que le nom d'utilisateur et le mot de passe concordent.

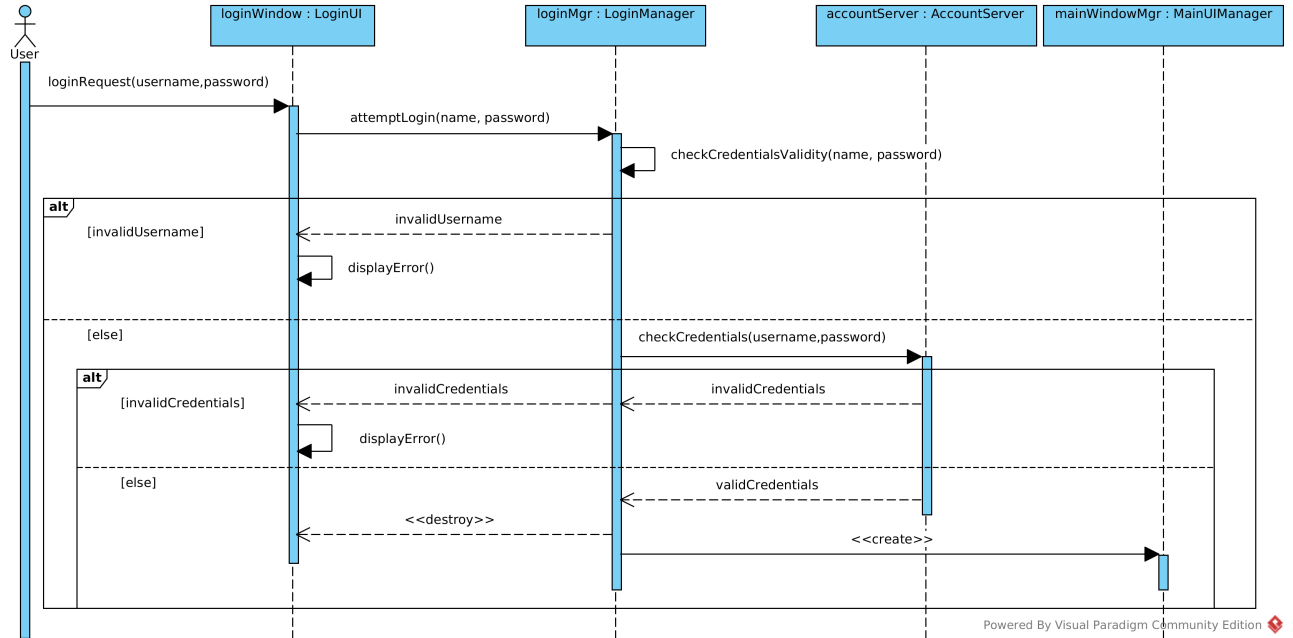


Figure 10: Sequence : Connexion

### 6.2.3 Chat

Ce diagramme décrit la séquence d'action réalisées afin d'envoyer et recevoir des messages. Quand l'utilisateur tape sur la touche entrée pour envoyer un message, celui-ci est placé dans un buffer contenu dans le ChatManager. La méthode `getMessages()` du `chatManager` consiste en fait à aller chercher tous les messages que l'utilisateur a placé dans le buffer.

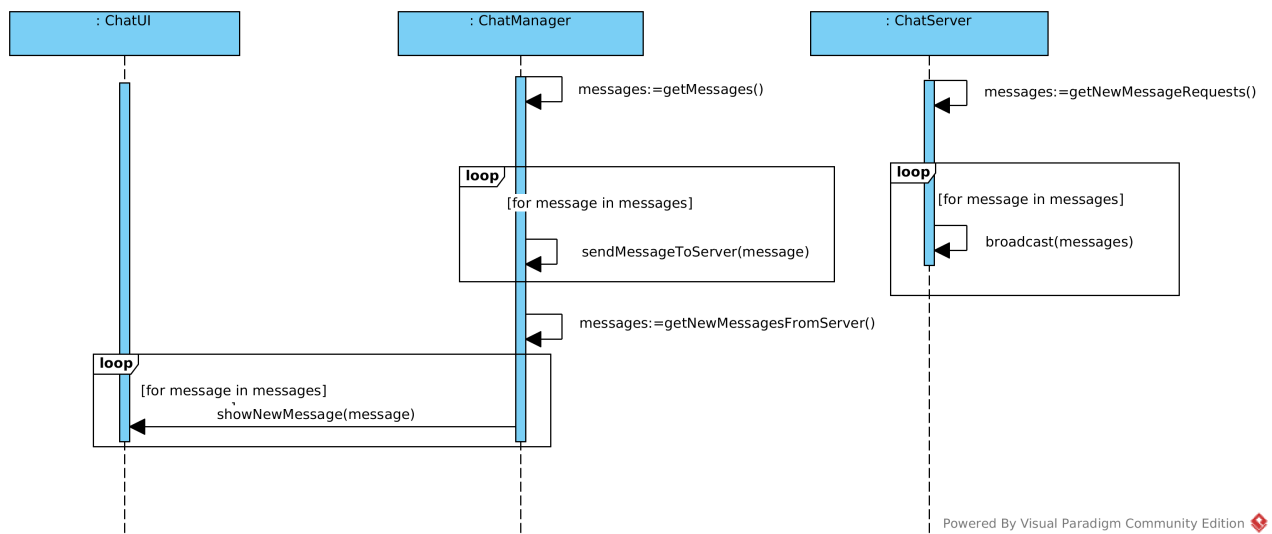
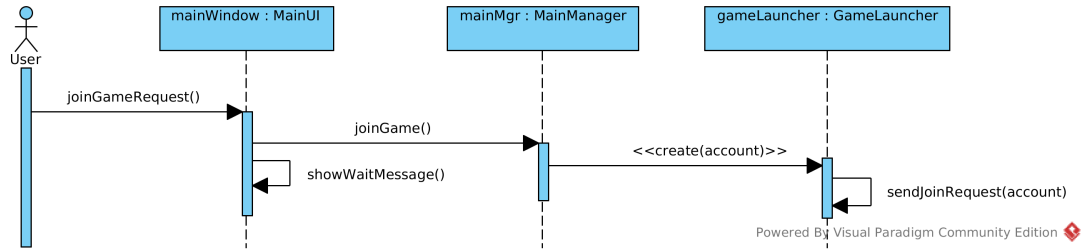


Figure 11: Sequence : Chat

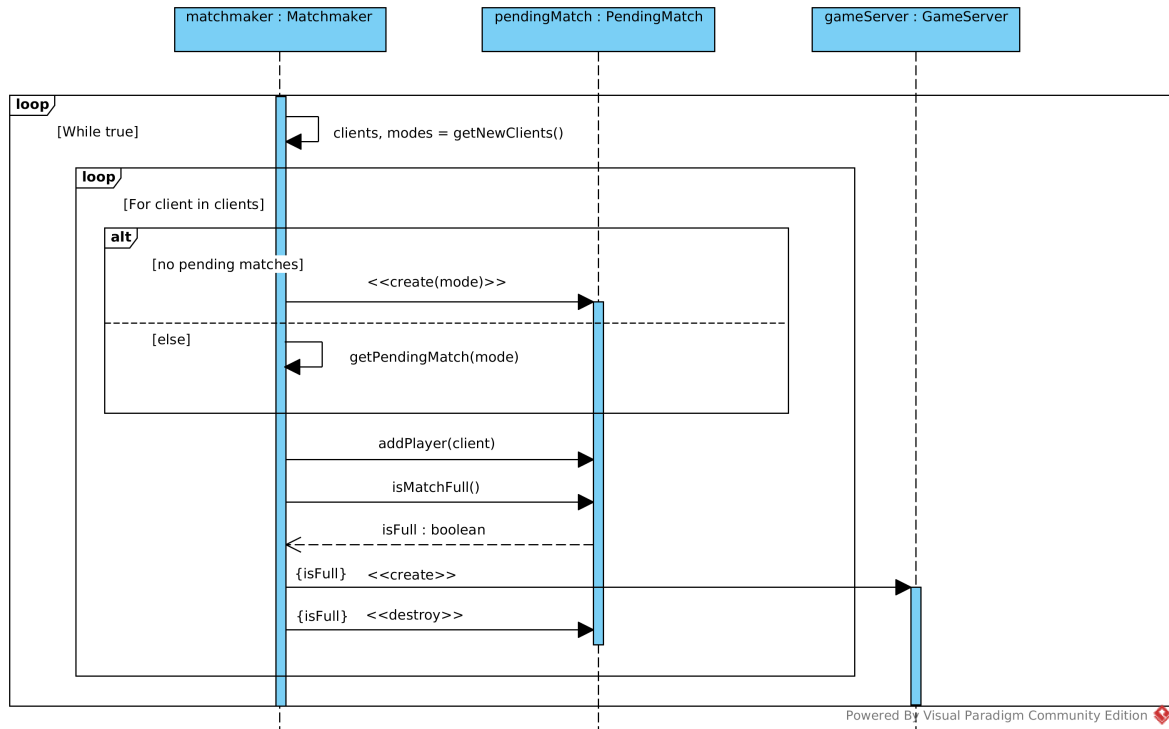
### 6.2.4 Matchmaking: Client

Ce diagramme décrit la séquence d'actions réalisées quand un utilisateur décide de rejoindre une partie (coté client).



### 6.2.5 Matchmaking: Serveur

Ce diagramme décrit la séquence d'actions réalisées quand un utilisateur décide de rejoindre une partie (coté serveur). Les pendingMatches sont des matchs en construction (tous les 4 joueurs n'ont pas encore été trouvés).



## 6.2.6 Diagramme d'activité : Matchmaking

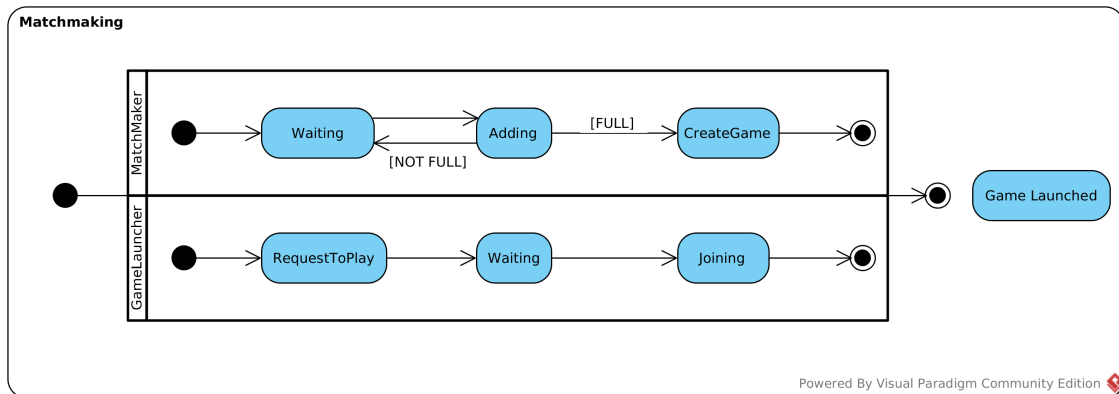


Figure 14: Activité : Matchmaking

## 6.2.7 Boucle du jeu: Client

Ce diagramme décrit la séquence d'actions réalisées lors d'une "frame". En effet cette séquence d'actions se répète un nombre important de fois par seconde tant que la partie n'est pas terminée.

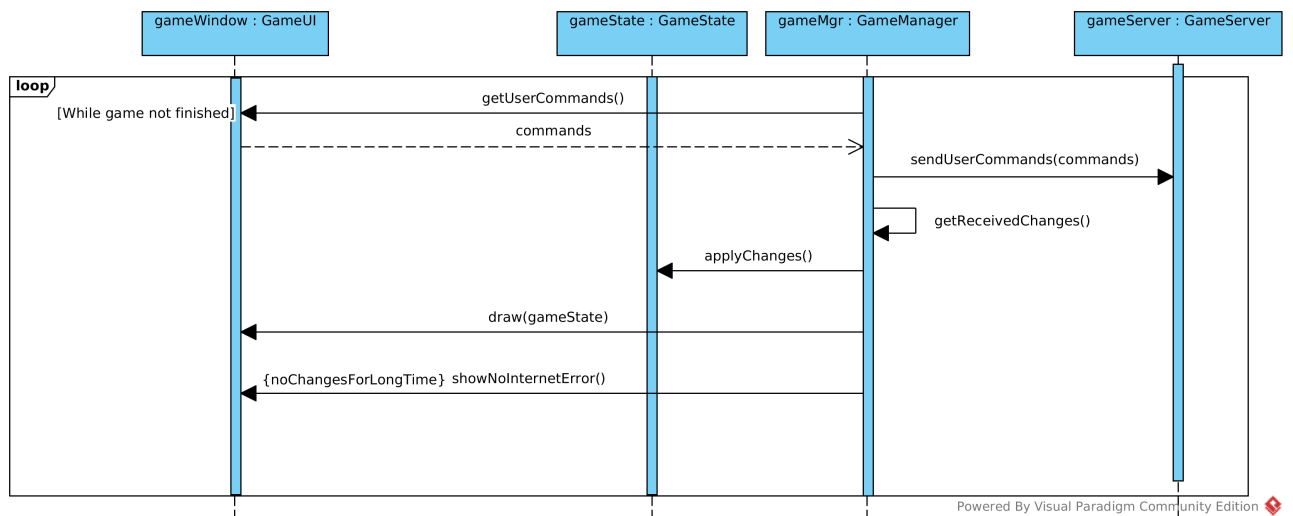


Figure 15: Sequence : Boucle du jeu client

### 6.2.8 Boucle du jeu: Serveur

Ce diagramme décrit la séquence d'actions réalisées par le gameServer, comme pour le diagramme précédent, cette séquence d'actions se répète un très grand nombre de fois par seconde.

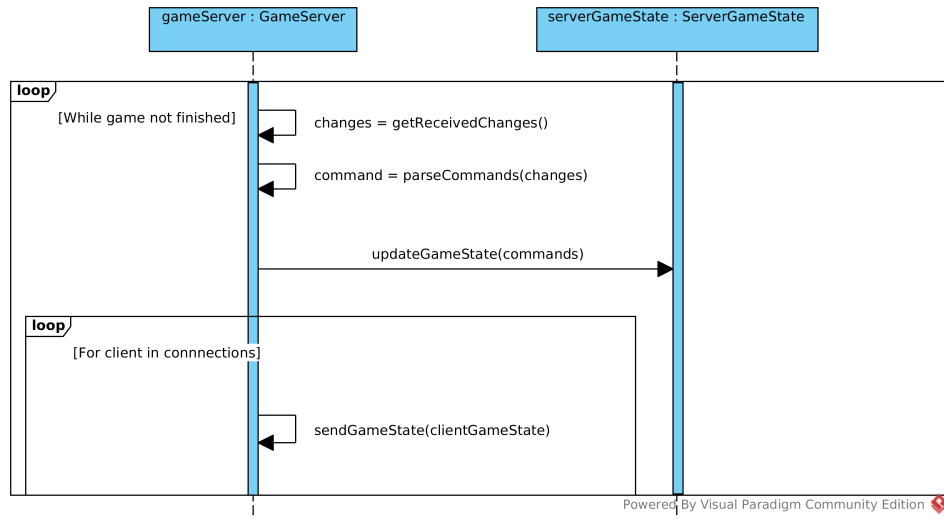


Figure 16: Sequence : Boucle du jeu serveur