Équipe 208

Document d'architecture logicielle

Version 1.0

Historique des révisions

Date	Version	Description	Auteur
2023-03-19	1.0	Cas d'utilisation + diagramme de Paquetage	Hamza et Renel
2023-03-20	1.1	Rédaction de l'introduction, cas d'utilisation et diagramme de séquence	Daniel
2023-03-20	1.2	cas d'utilisation, diagramme de paquetage, diagramme de déploiement	Lounes
2023-03-20	1.3	cas d'utilisation et diagramme de séquence + diagramme de déploiement	Yanis
2023-03-21	1.4	cas d'utilisation et diagramme de sequence	Melvice

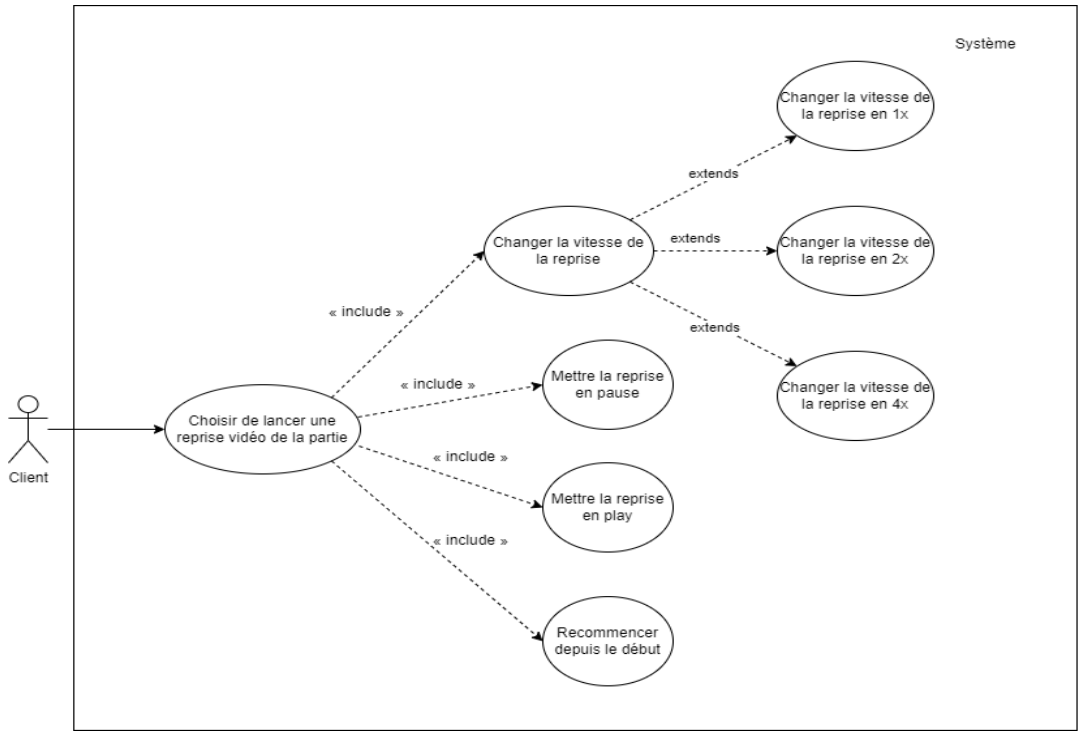
Table des matières

1. Introduction

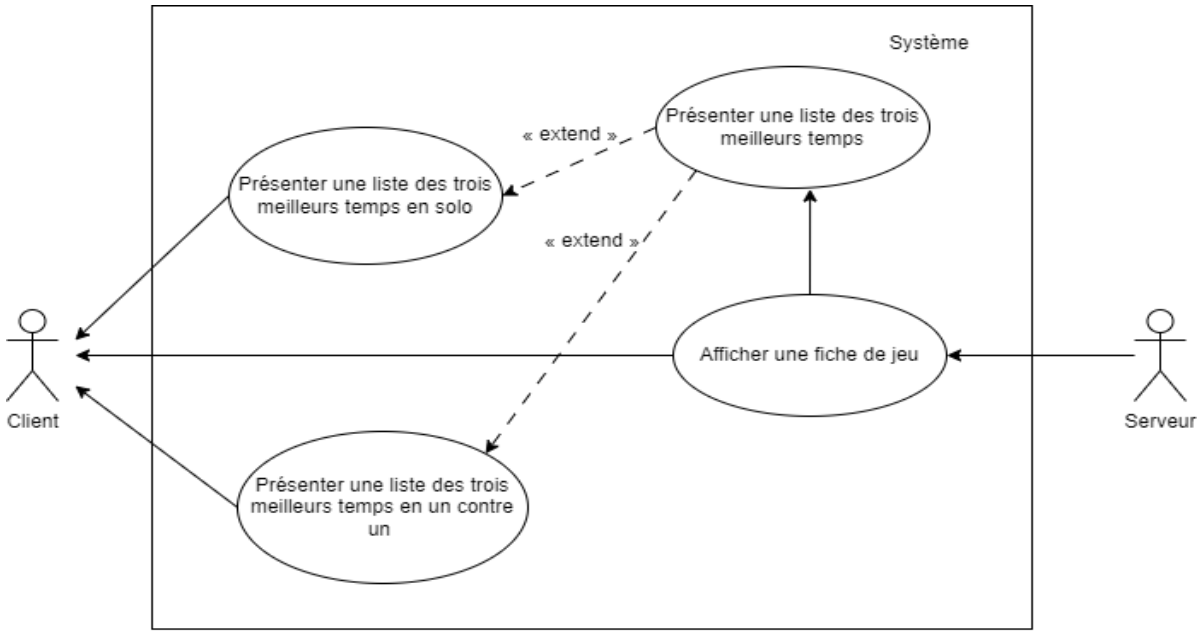
Le document d’architecture logicielle est structuré en quatre sections qui permettent d'appréhender le contenu et l'organisation du projet sous différents angles. La première section propose une vue d'ensemble des cas d'utilisation. La seconde section décrit les processus mis en place dans le cadre du projet. La troisième section présente une vue logique des différentes composantes du système. Enfin, la quatrième section donne une vue d'ensemble de la façon dont le projet sera déployé.

2. Vue des cas d’utilisation

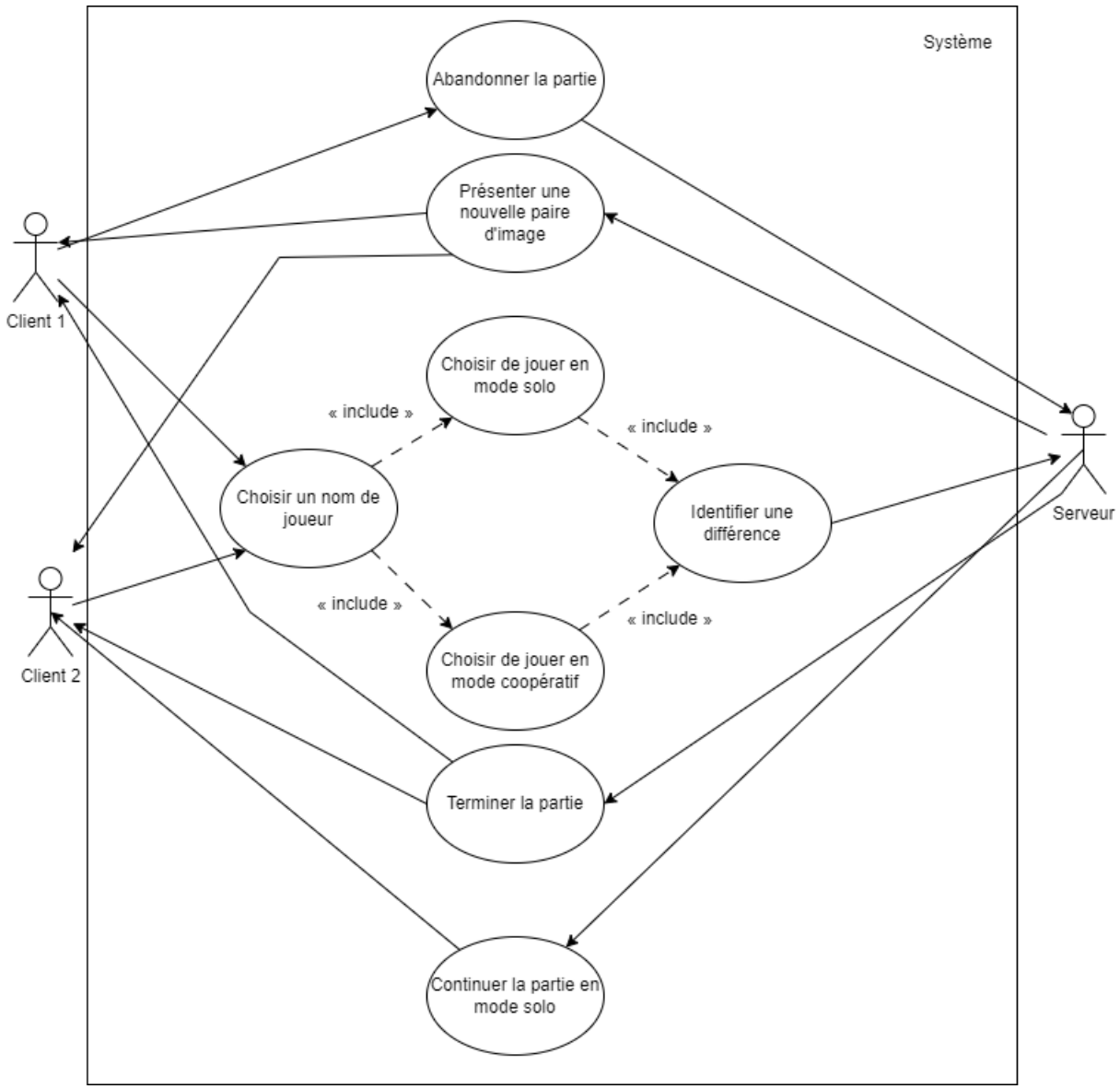
Cas d’utilisation 1 - La reprise vidéo de la partie à la fin du jeu.



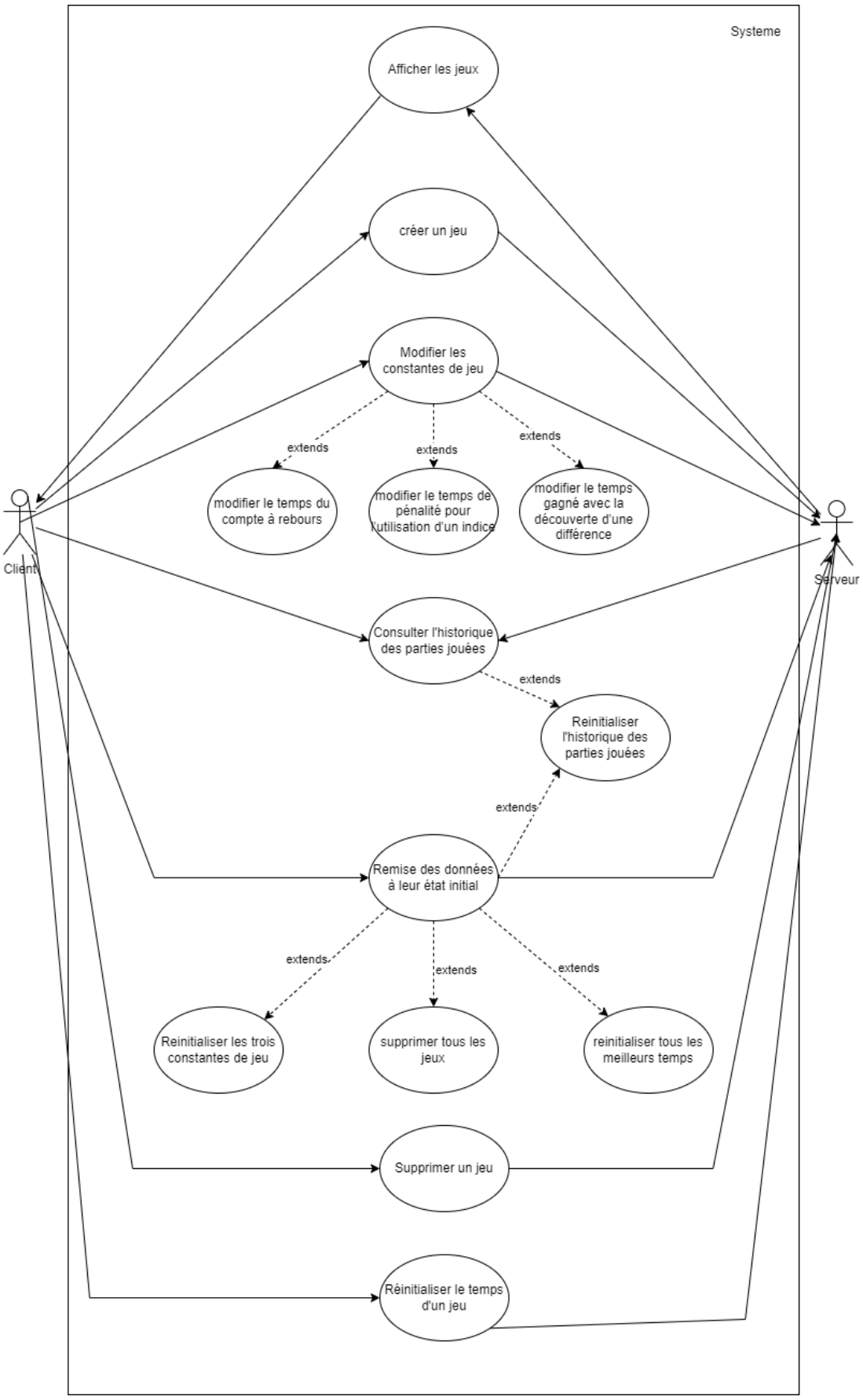
Cas d’utilisation 2: Présenter sur la fiche de chaque jeu les meilleurs temps



Cas d'utilisation 3: Jouer en coopératif au mode temps limitée

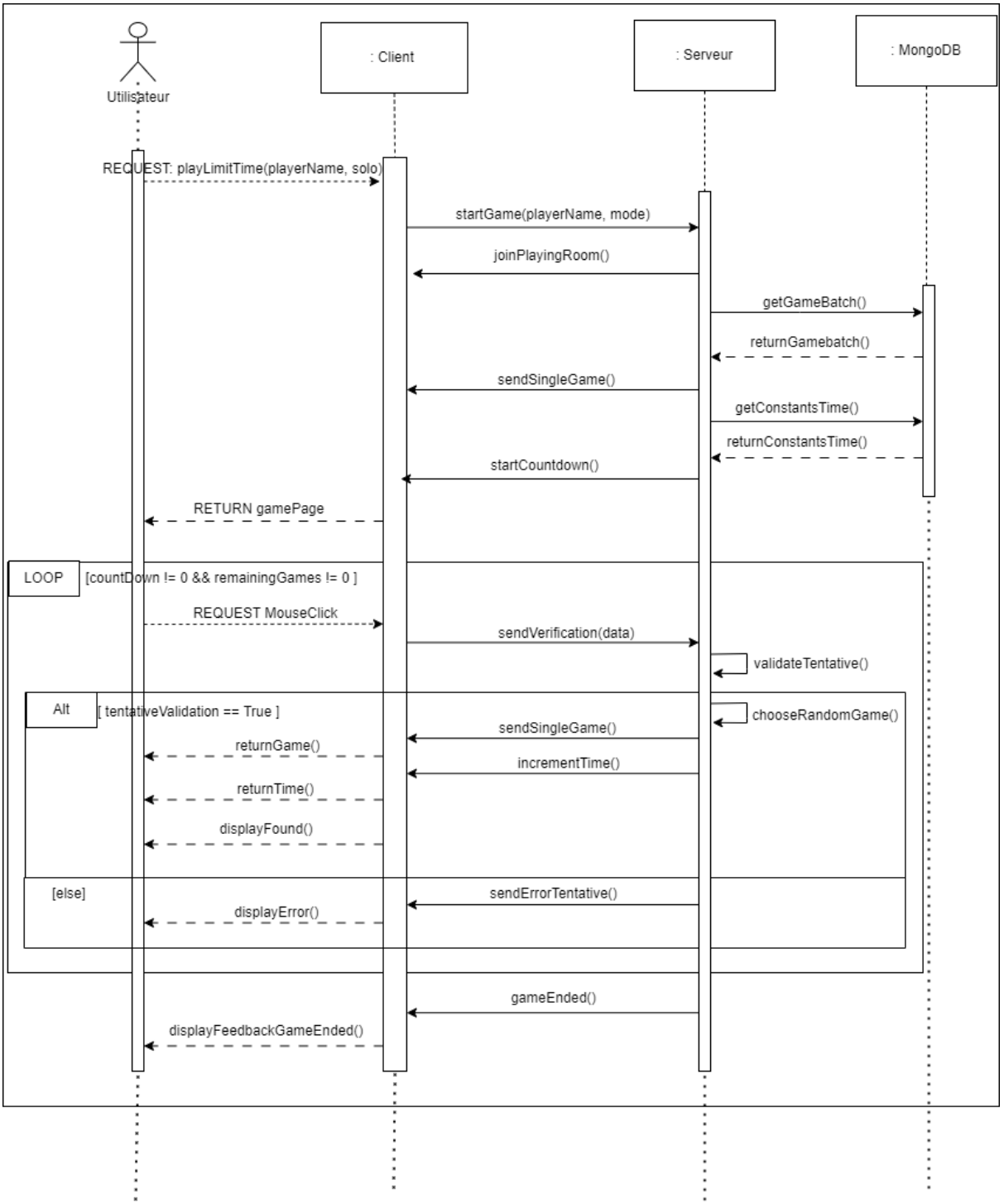


Cas d’utilisation 4: La page configuration

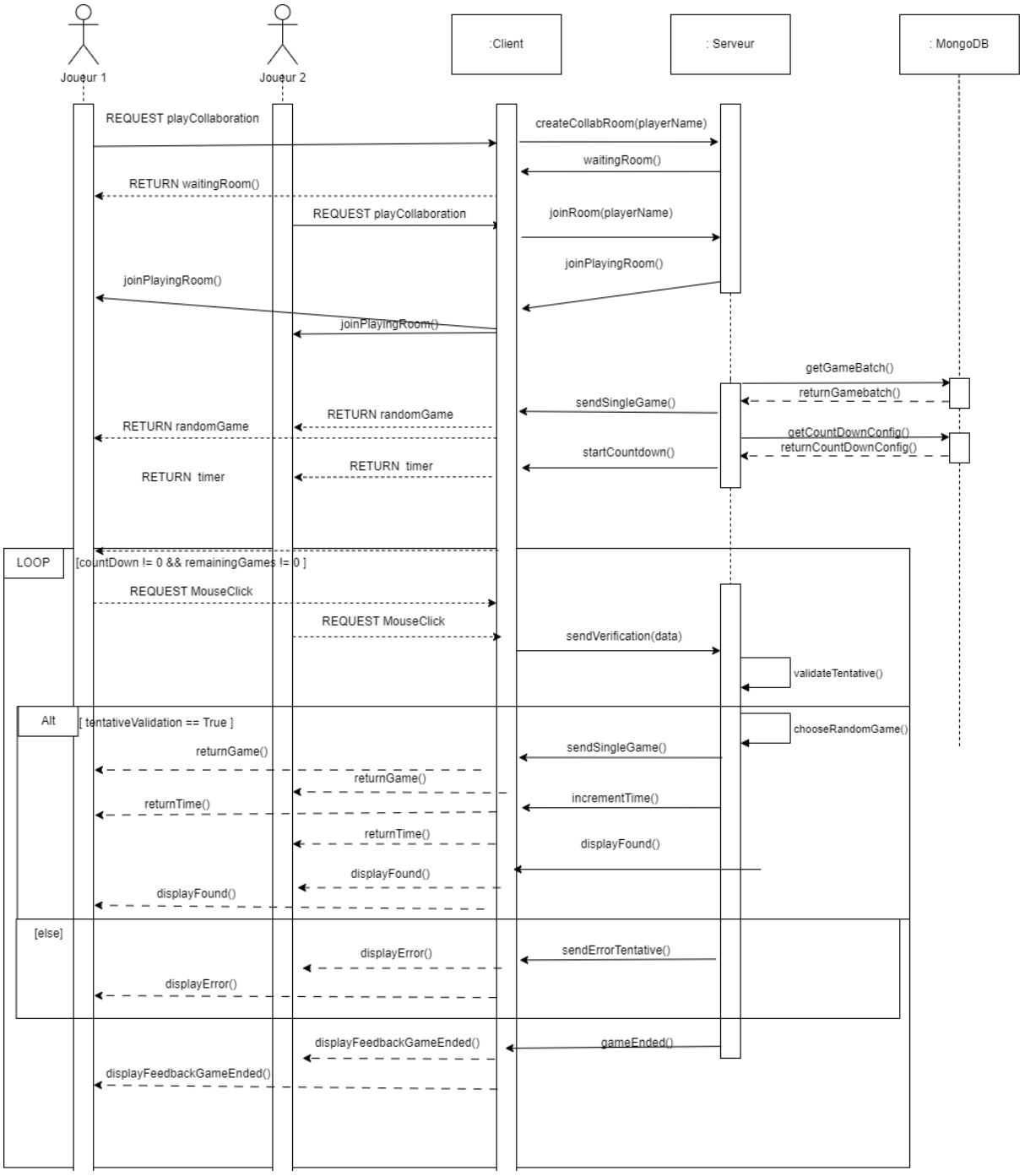


3. Vue des processus

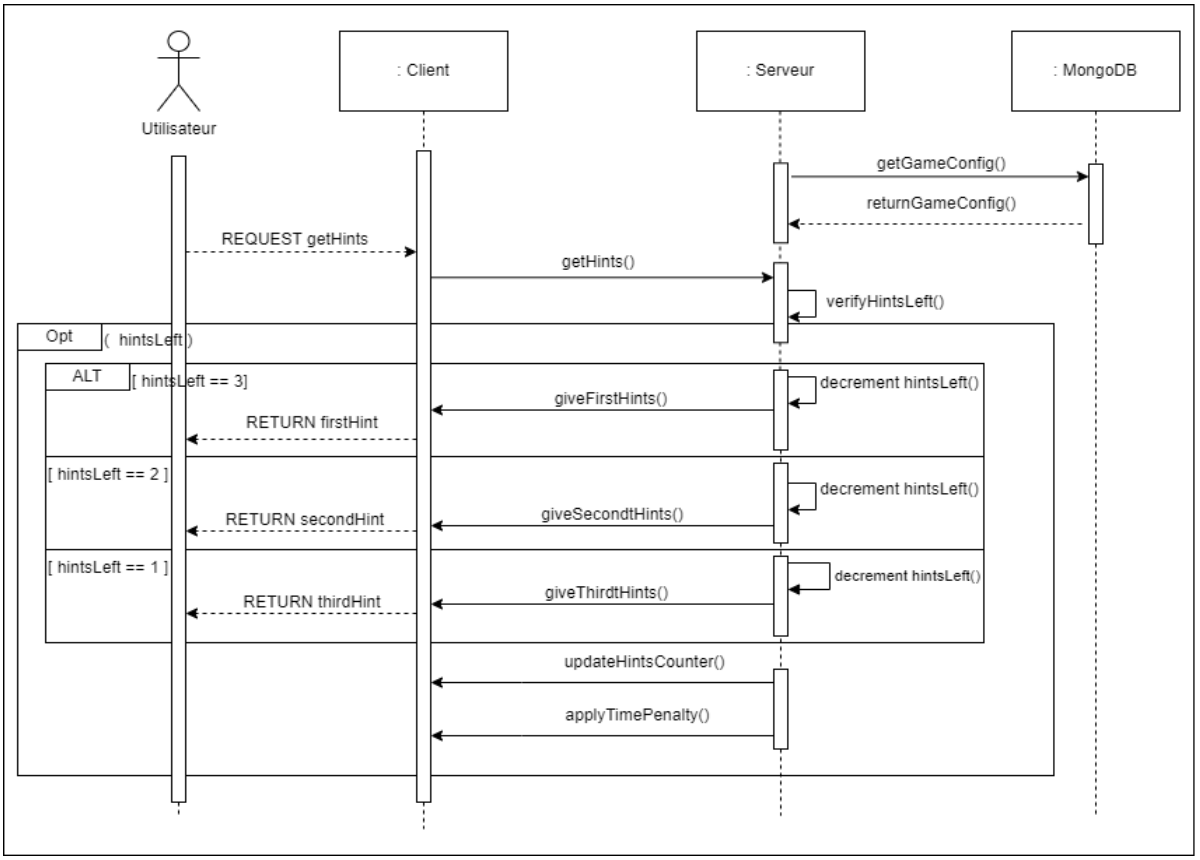
Processus 1: Jouer en temps limité en mode solo



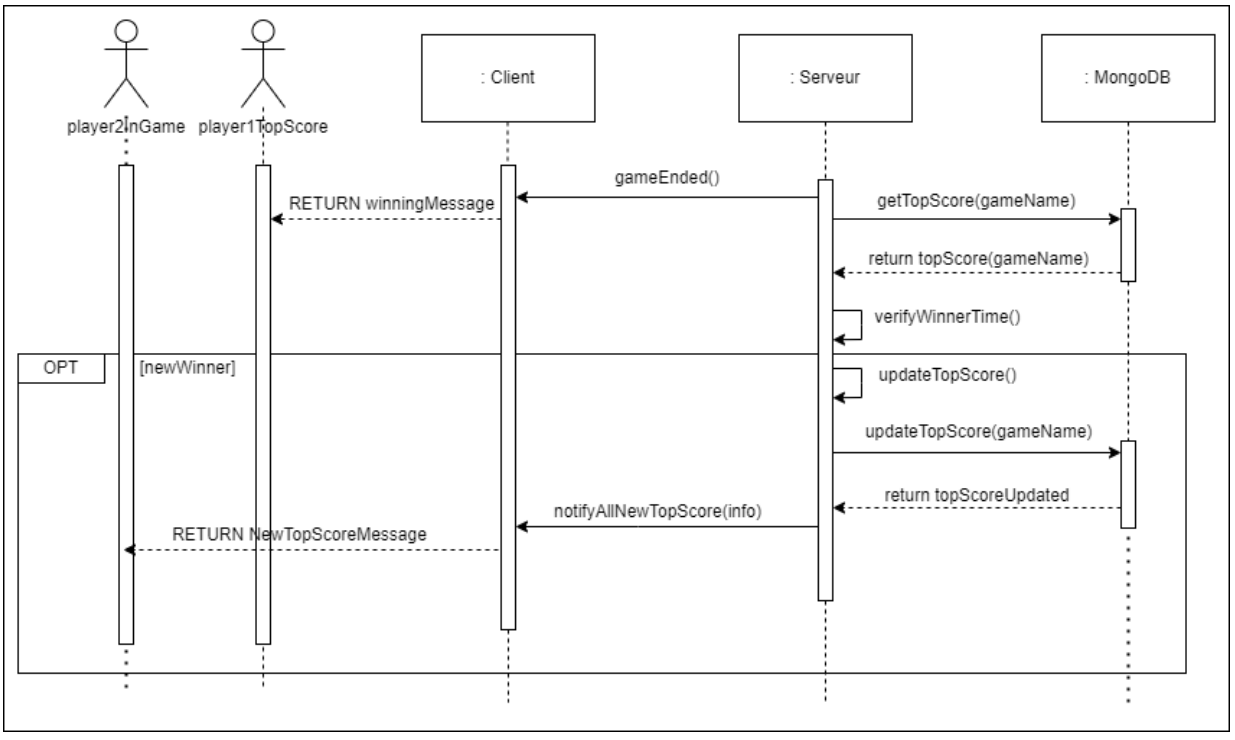
Processus 2: Jouer en temps limité en mode coopératif



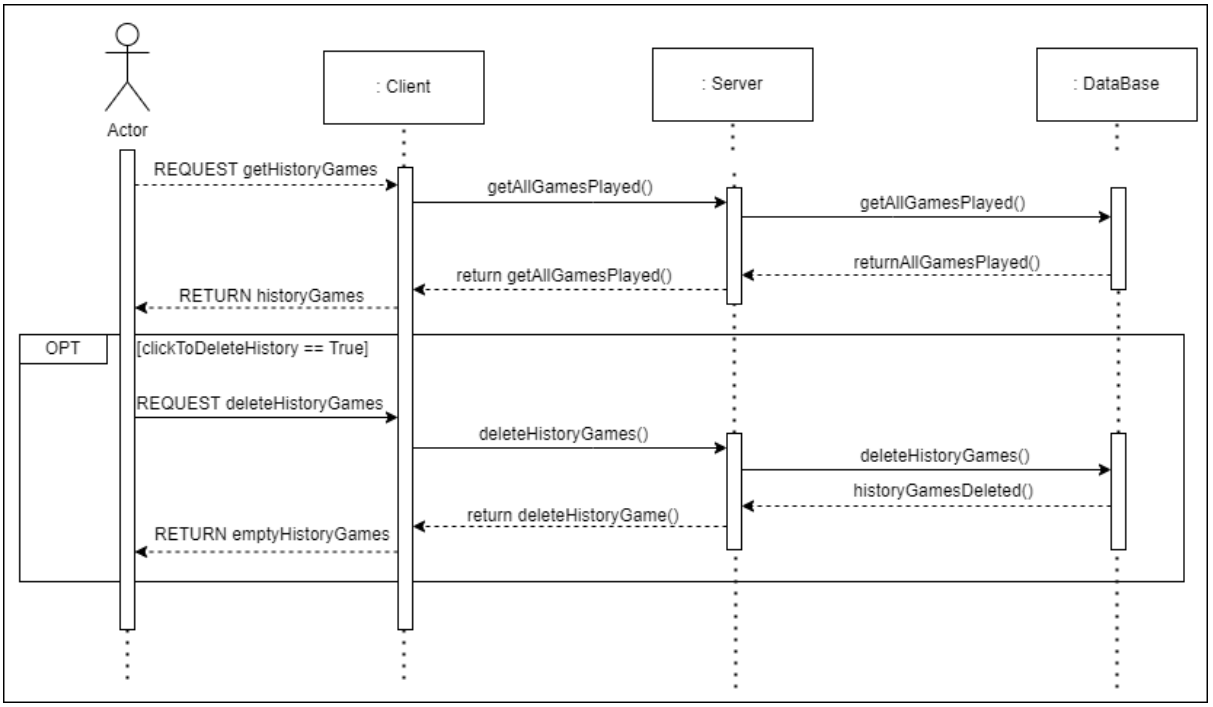
Processus 3: Obtenir un indice



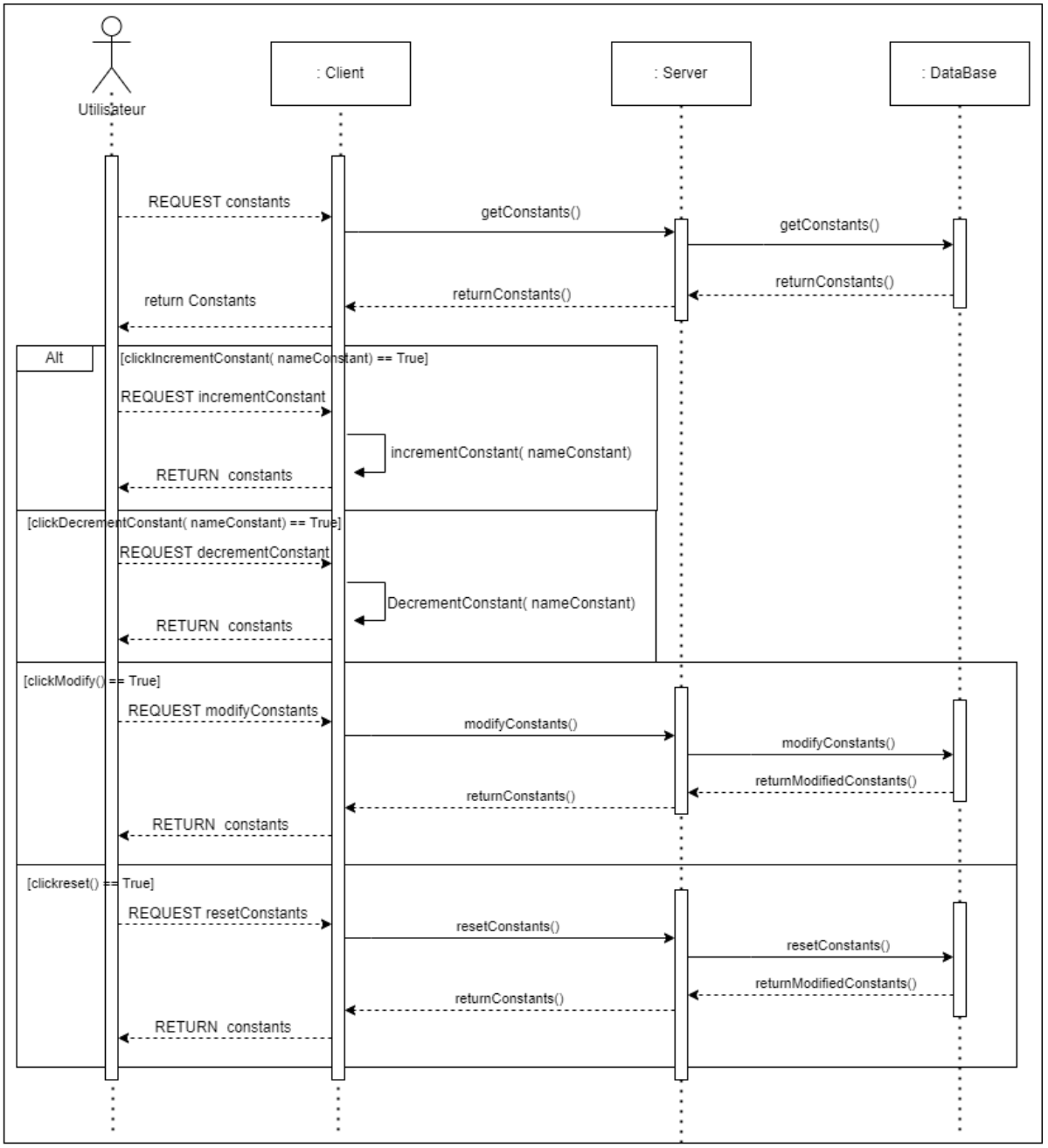
Processus 4: Envoyer des messages de partie (global)



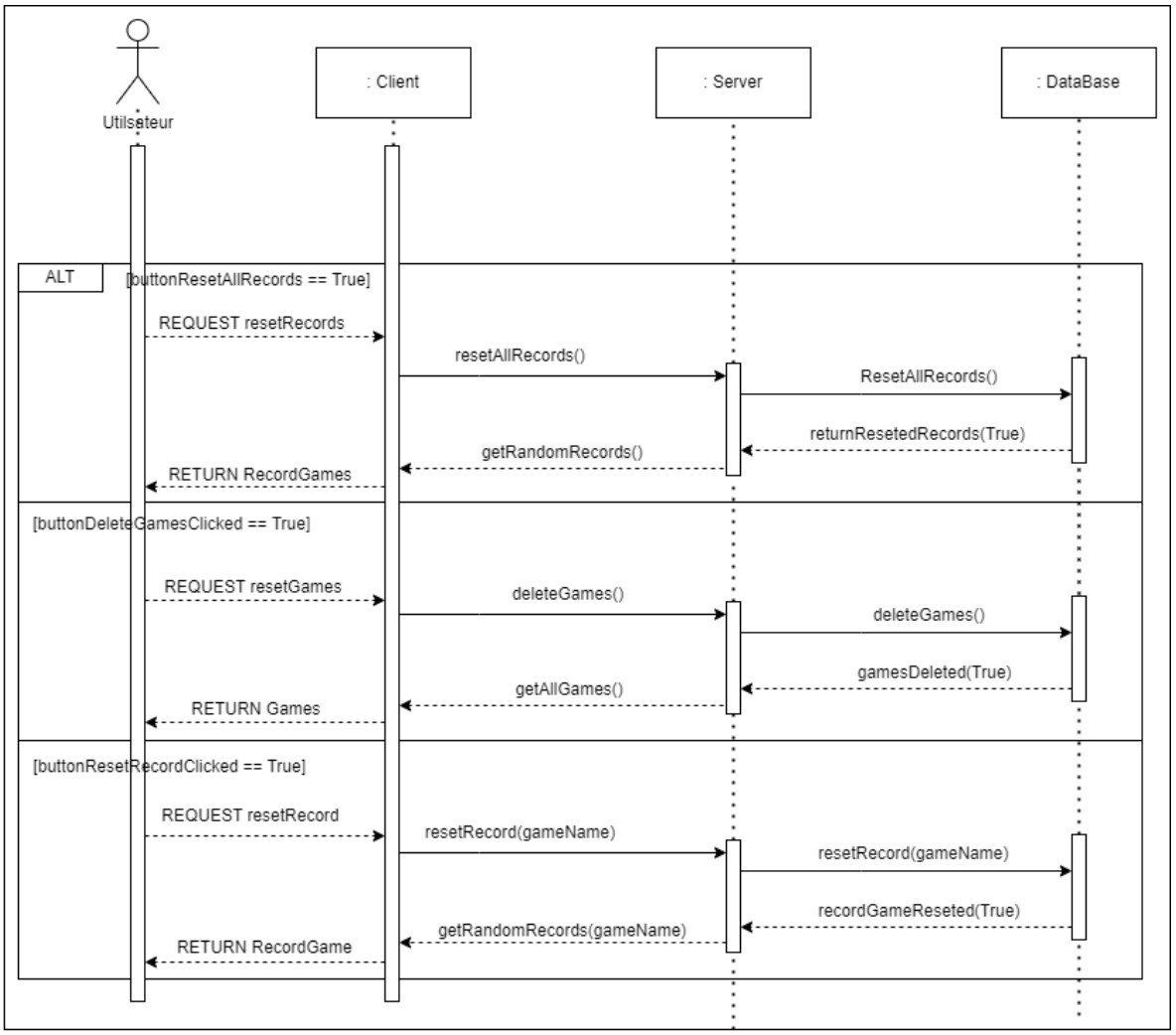
Processus 5: Voir l’historique des parties jouées



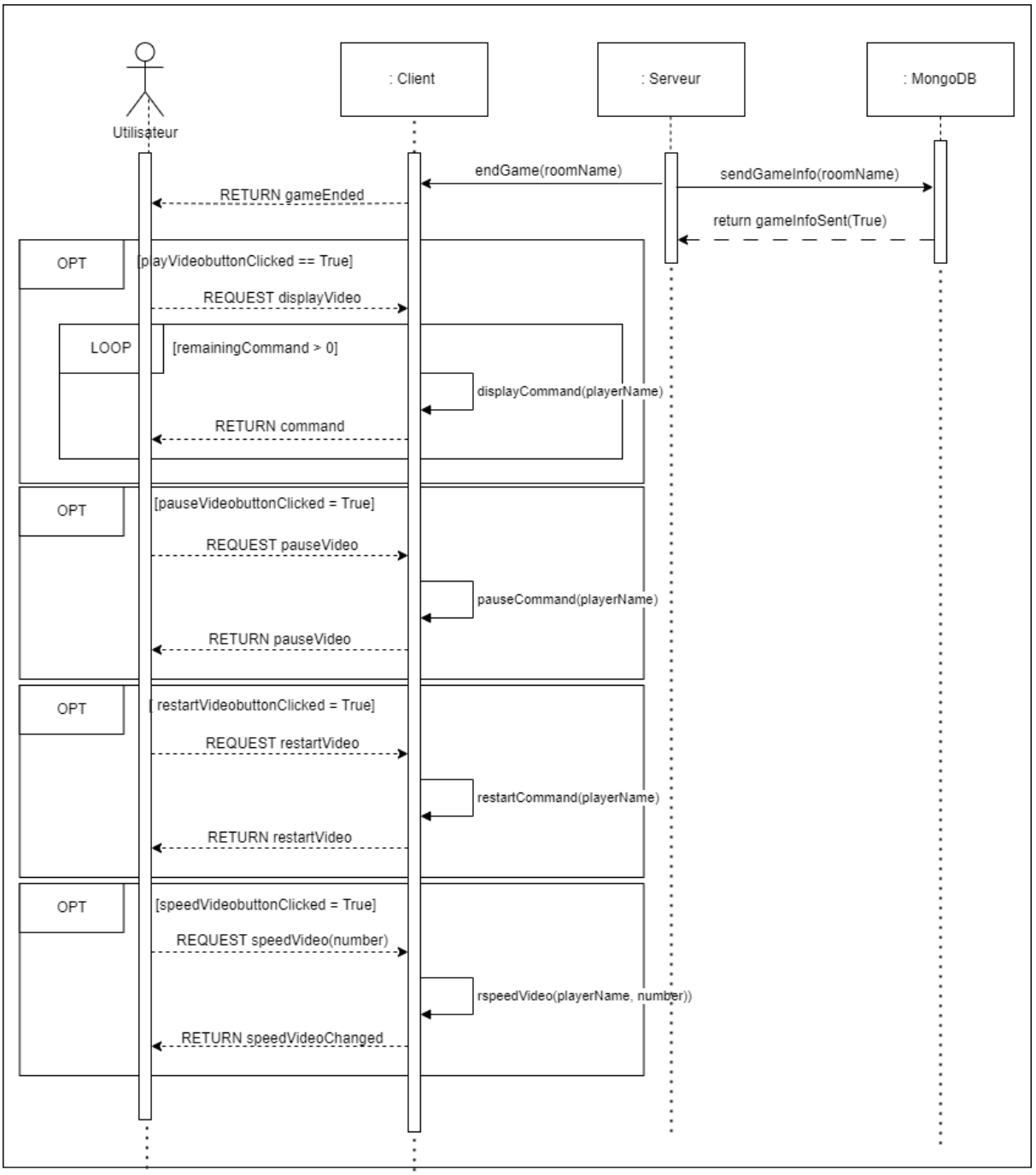
Processus 6: Configurer certaines constantes de jeu dans la vue de configuration.



Processus 7: Remettre certaines données du système à leur état initial dans la vue de configuration



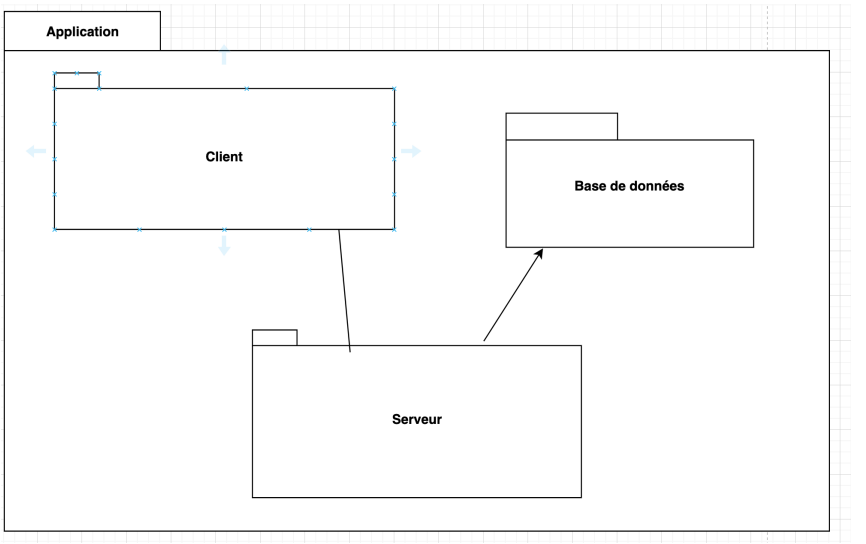
Processus 8: Revoir la partie en mode *Classique* rejouée en entier après la fin de la partie.



4. Vue logique

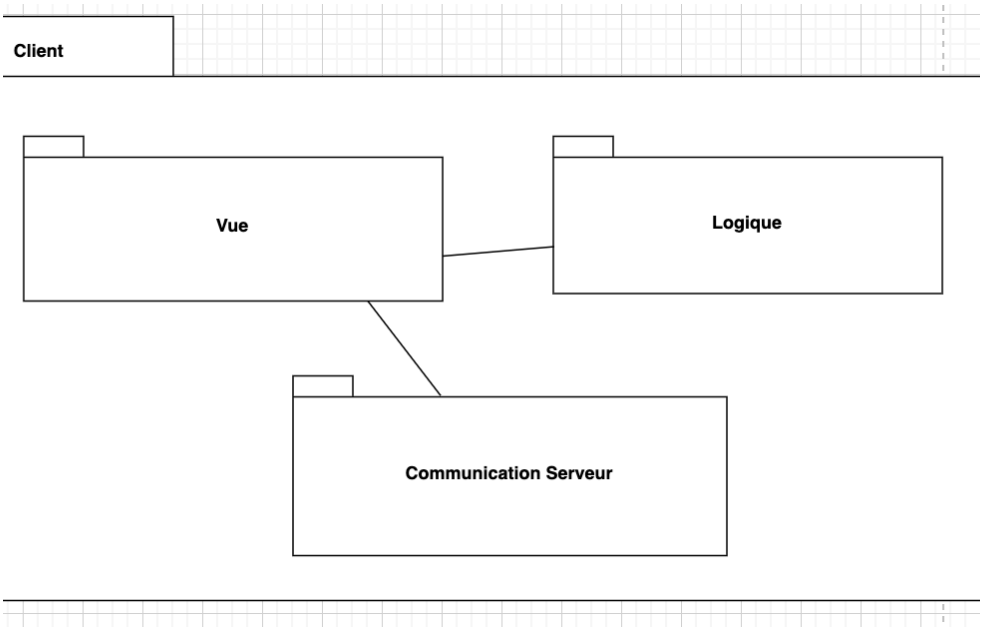
Application

Le paquetage application est notre paquetage complet du jeu, celui-ci regroupe 3 sous-paquetages qui sont Client, Service et Base de données



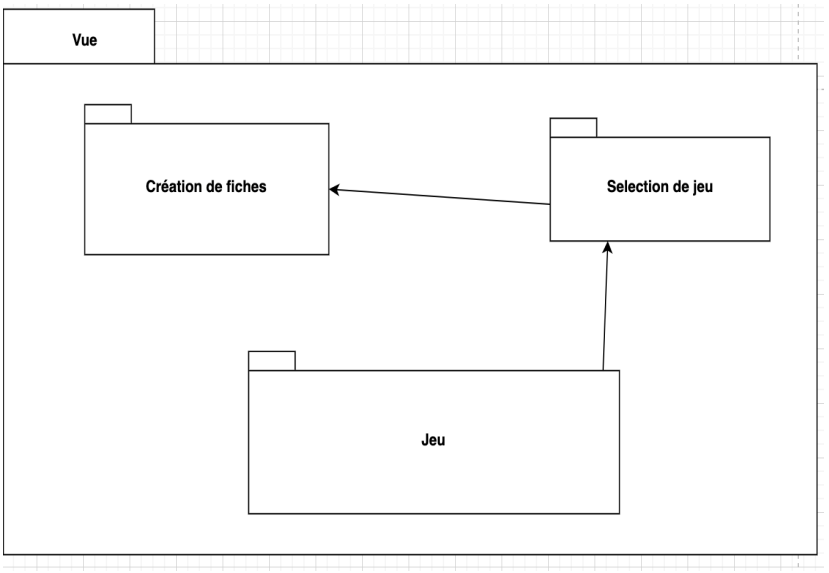
Client

Le paquetage client regroupe donc tout ce qui lie le jeu et le client, celui-ci regroupe la vue, la logique et la communication avec le serveur (à travers des sockets).



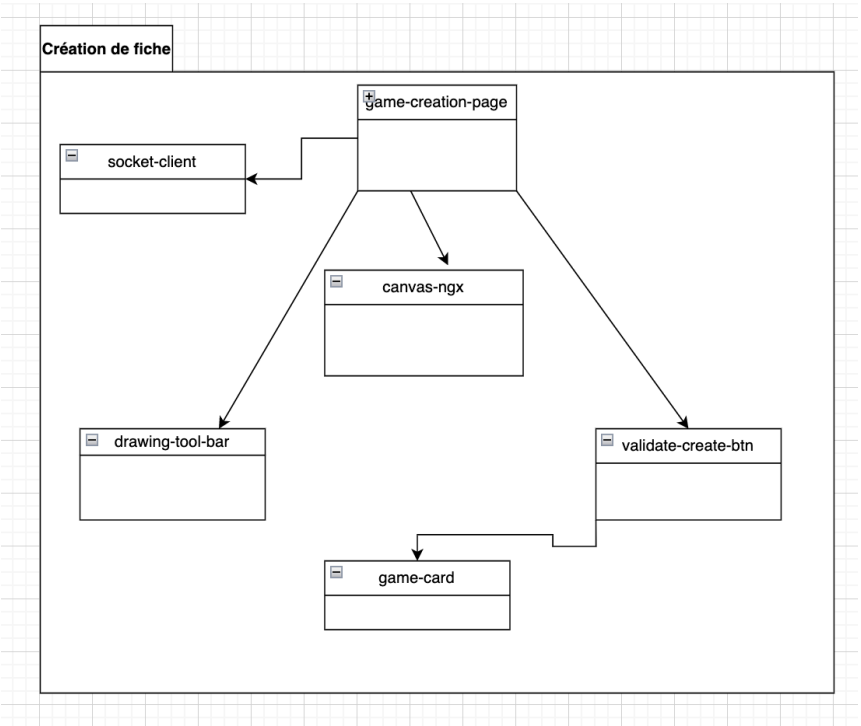
Vue

Le paquetage vue regroupe 3 sous-paquetages, la créations de fiches, la sélection de jeu et le jeu. Le paquetage vue représente donc tout ce que le client voit sur le jeu, que ce soit les fiches, le jeu principal…



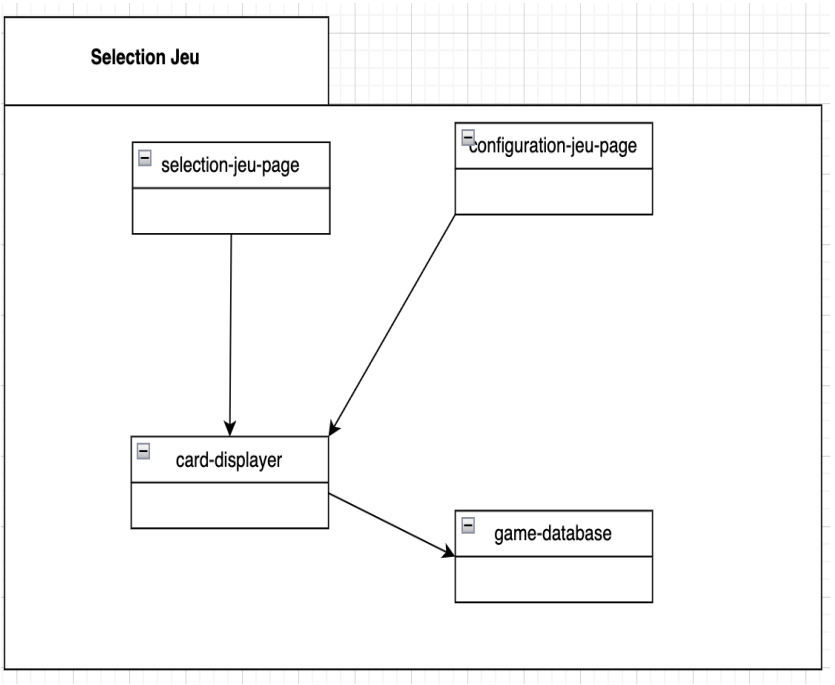
Création de fiche

Notre fiche commence en créant une game dans la vue de configuration, qui nous permet donc d'insérer des canvas et dessiner. On valide le canvas puis on a donc la création d'une game card.



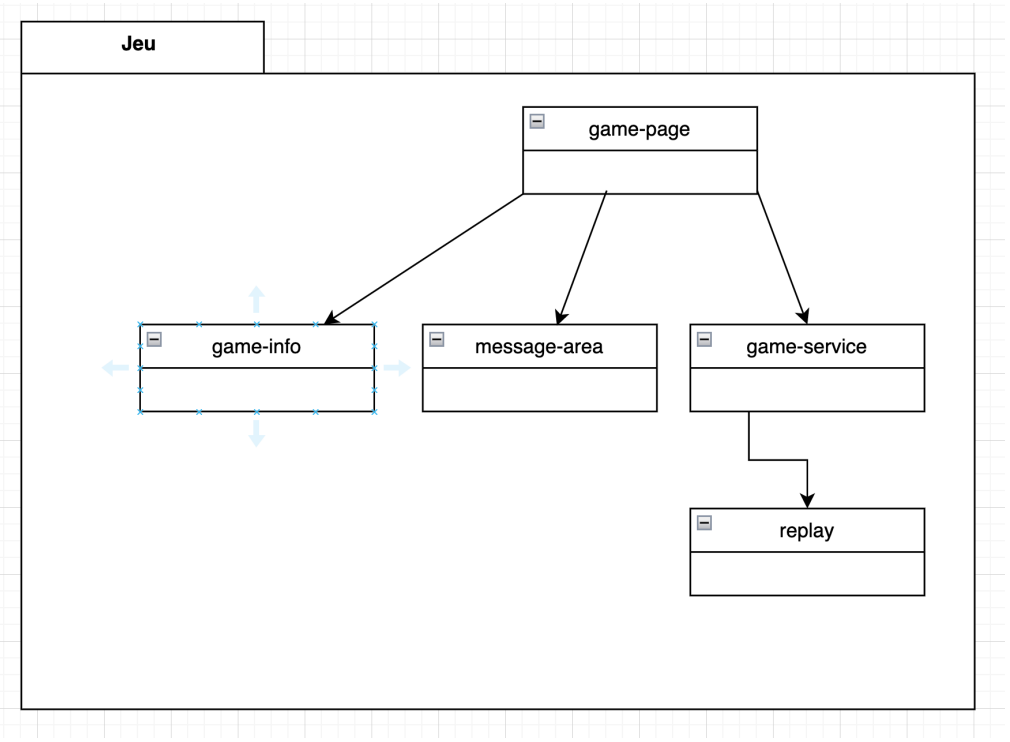
Sélection Jeu

Le paquet sélection jeu implique 2 pages qui vont nous permettre de créer nos cartes et donc d'afficher notre carrousel de game card. Ces game card prennent leur données de game-database.

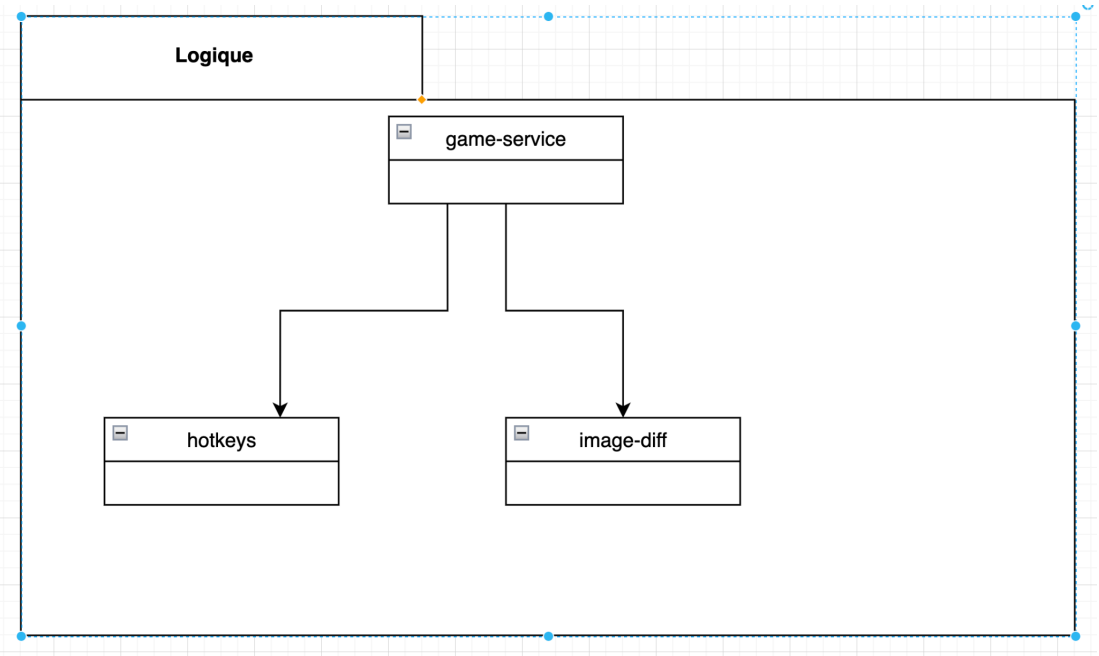


Jeu

Notre paquetage jeu représente le jeu en tant que tel, le gabarit de la page de jeu est donc créer dans game-page, qui lui utilise game (un service) pour effectuer le traitement dans les variables du jeu, obtenir les informations des joueurs… Message-area représente notre messagerie. gameinfo représente les informations de notre jeu. Nous comptons faire pour le sprint 3 un service replay qui se chargera de faire le replay de la partie du côté client.

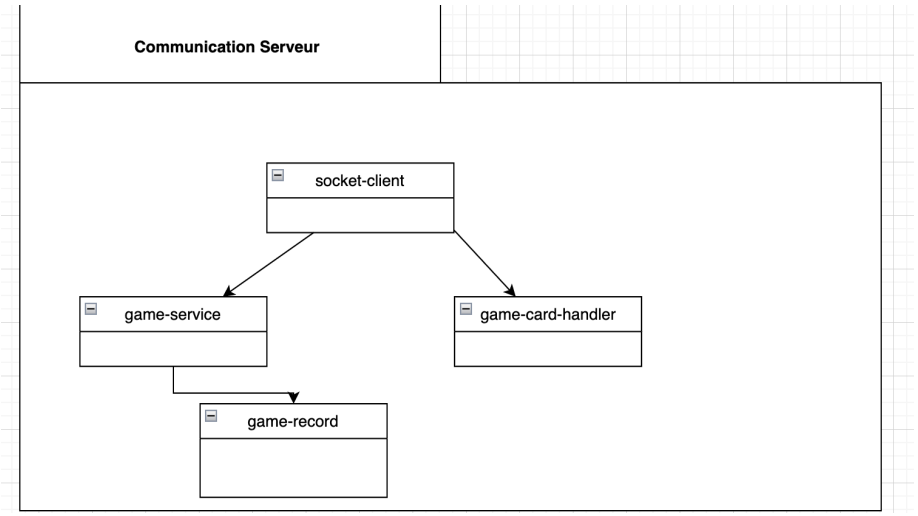


Logique
Ici on parle de toute la logique du jeu. image-diff est un service qui traite la détection de différence entre 2 images. Hotkeys détecte les différentes touches que l'on fait sur le clavier, on utilise cela notamment pour le mode triche. Et game met tout cela ensemble ainsi que gère les différentes données des utilisateurs, gestions des clics, gestions des messages d'erreurs, messages quand on a trouvé une différence…



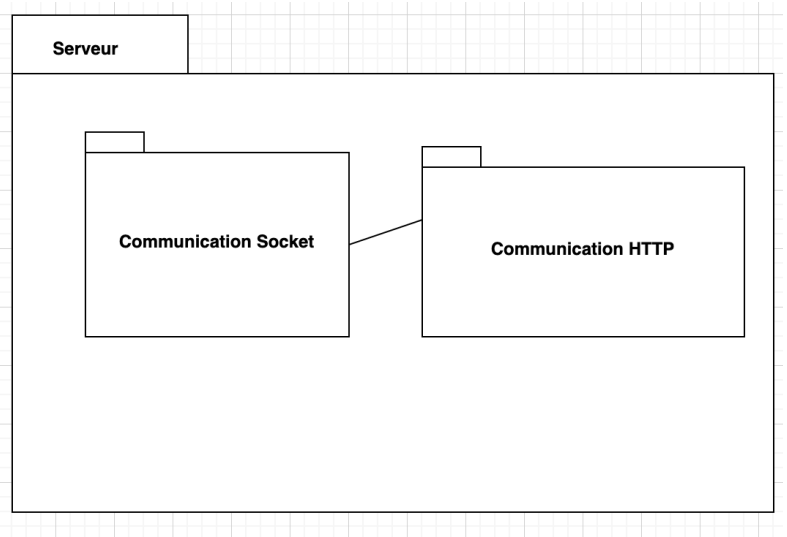
Communication Serveur

Le paquetage communication serveur représente la communication entre le client et le serveur. On utilise principalement socket-client qui lui se charge des connexions aux room, l'envoie des sockets, la gestion…



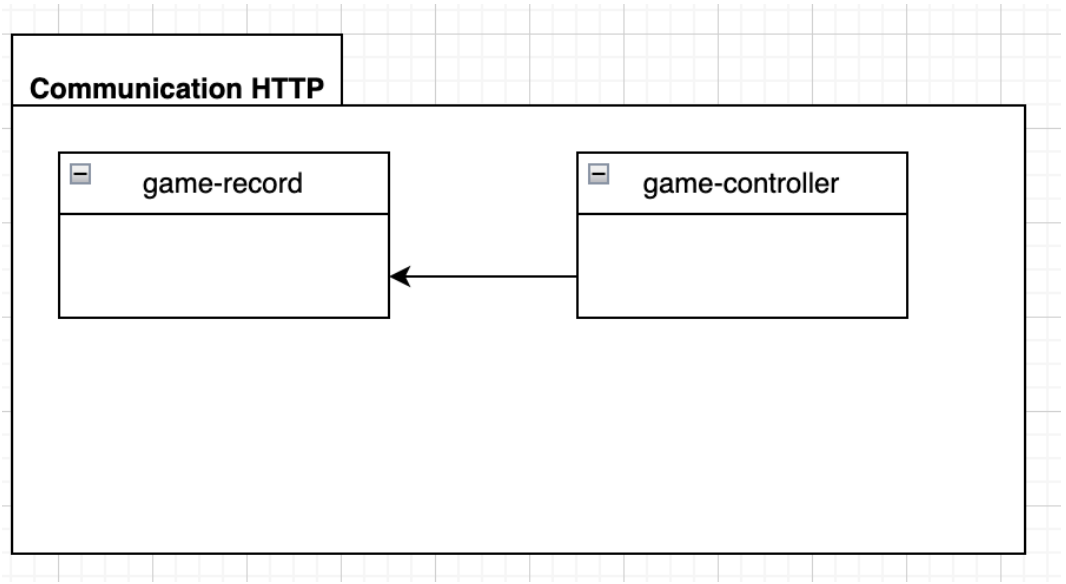
Serveur

Le paquetage serveur représente les deux types de communications, socket et http.



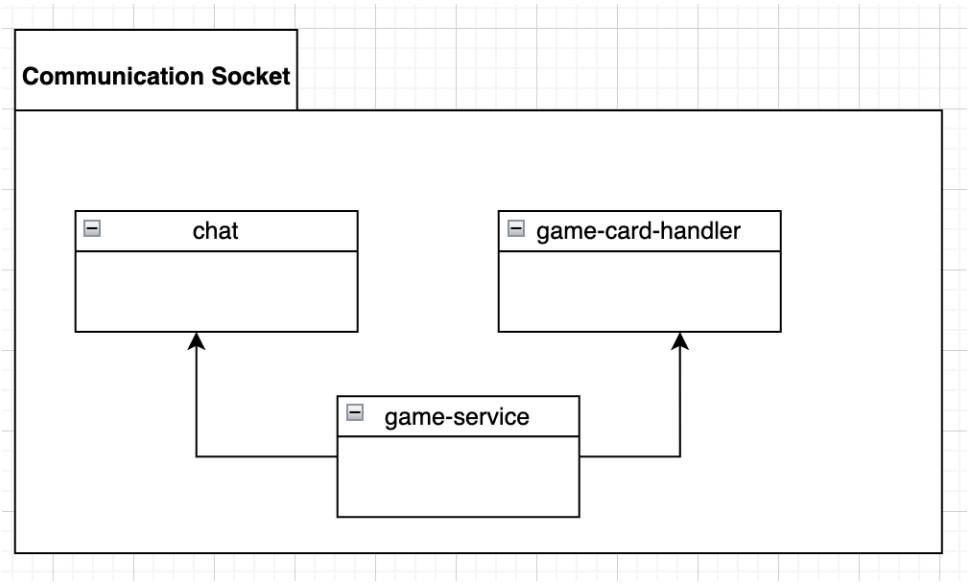
Communication HTTP

Le paquetage communication HTTP ici utilise principalement 2 fichiers. Ce sont 2 controllers, game-controller et game-record appellent l'API getGames afin de fetch tout les games présent dans le serveur avec leur meilleur 3 record, et permettent d'actualiser les games, notamment les constantes.

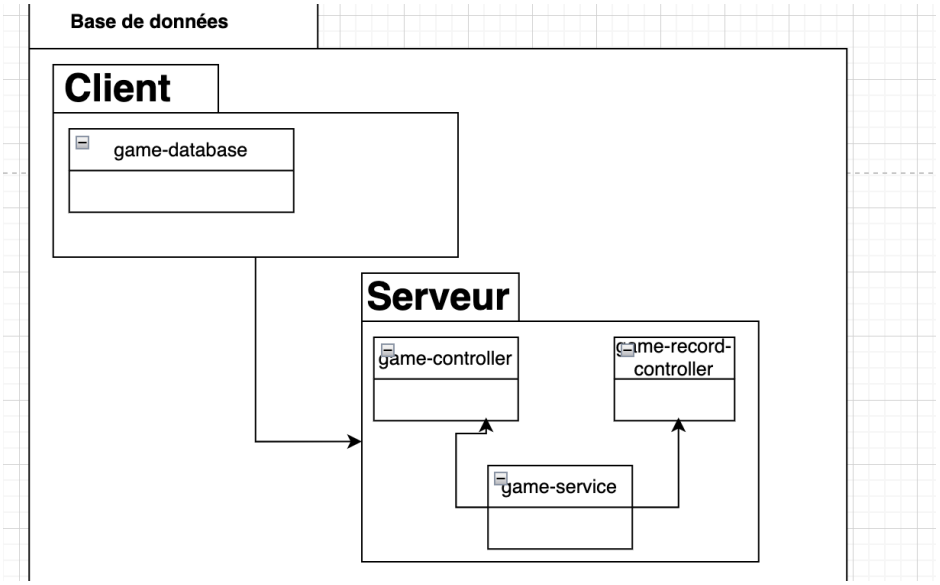


Communication Socket

Le paquetage communication socket, utilise 2 gateways, chat et game-card. Les gateway reçoivent des évènement du client, il les écoutent et effectue le traitement correspondant



Base de données
Le paquetage base de données utilise game-database qui fait la connexion avec le serveur pour lui transmettre les jeux créer, fetch les jeux existant et enregistrer les record.



5. Vue de déploiement