

Université Côte d'Azur/DS4H
MIAGE – MASTER 1

Rapport final

Outils d'ingénierie

par

Paul BUREL

Geoffrey LALIC

Axel MAÏSSA

Djade KHALDI

Rémy NGUYEN

Aryamaan SINGH KUNWAR

Monsieur Philippe RENEVIER

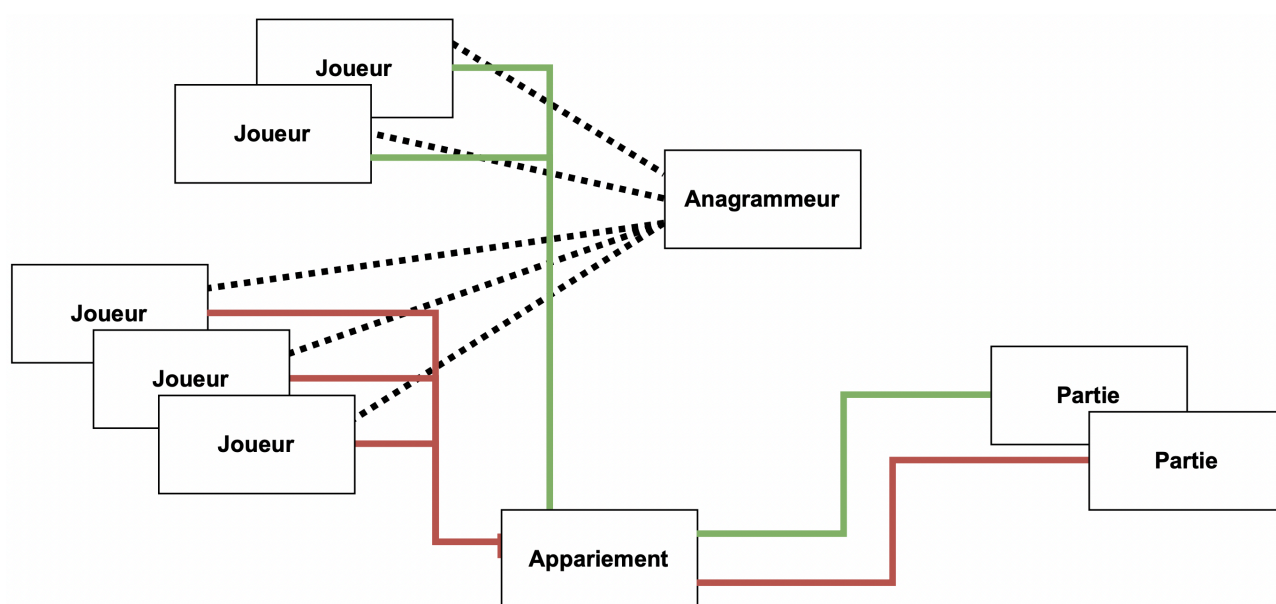
Sommaire

Diagramme général de l'architecture	3
Échanges entre services	5
Service Stateless VS Statefull	6
Tests	6
Démonstration vidéo	6

Digramme général de l'architecture

Quatre types de services sont présents dans notre code :

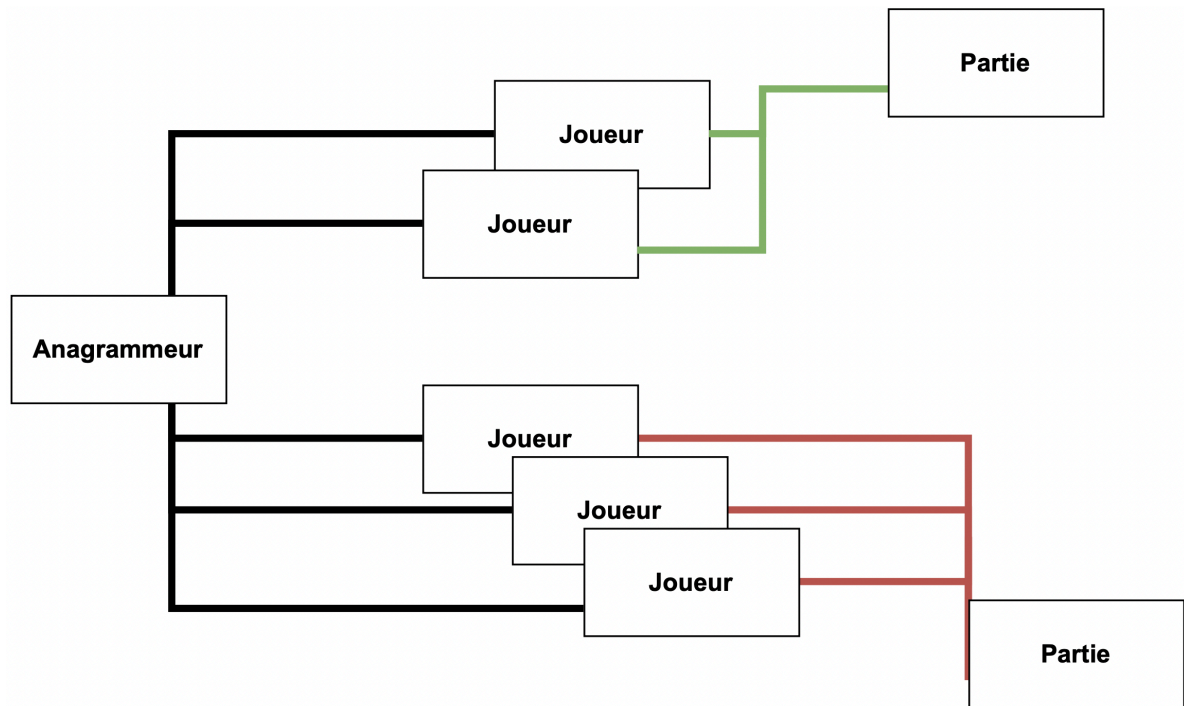
- **Joueur**, il peut y en avoir plusieurs, tous ne jouant pas à la même partie. A chaque tour, partie lui demandera de faire une proposition de mot à placer sur le plateau ou de passer son tour et donc d'échanger une lettre.
- **Partie**, il est également possible d'en avoir plusieurs. Son but est de garantir l'intégrité du plateau, de gérer la pioche et de faire la gestion de tour.
- **Anagrammeur**, une seule instance de ce service existe. Un joueur possède des lettres et souhaite savoir quel mot il peut placer en fonction de celles-ci mais également celles disposées sur le plateau, le joueur fera donc un appel à l'anagrammeur pour qu'il lui retourne une liste de mots qui conviennent.
- **Appariement**, une seule instance existe. Son but est de faire le lien entre des joueurs cherchant une partie et des parties qui manquent de joueur pour commencer.



Chacun des modules est un webservice, toutes les liaisons présentes sur ce digramme sont des des POST ou des GET.

Il peut y avoir autant de joueur et de partie que l'on souhaite mais il y aura une seule instance d'Appariement et d'Anagrammeur.

Appariement se tient prêt à recevoir une demande de la part d'un joueur qui souhaiterait intégrer une partie et une partie laisse savoir à appariement qu'elle a de la place pour des joueurs. Lorsque Appariement a informé Partie des joueurs qu'elle lui a attribué, ainsi que de leur moyen d'être appelés (IP, Port), le travail d'appariement est terminé et la partie peut commencer. Sur le diagramme plus haut, on a matérialisé une première partie avec des liens **rouges** et une seconde avec les **verts**.



Ce diagramme représente deux parties se jouant simultanément. À chaque tour, partie demande à ses joueurs de lui renvoyer des lettres à placer sur des cases. Pour faire des propositions de mots qui existent, les joueurs envoient leur lettres et le plateau (qui contient également des lettres).

Échanges entre services

Une fois les web services Partie, Anagrammeur et Joueur lancés, ils envoient un message à Appariement.

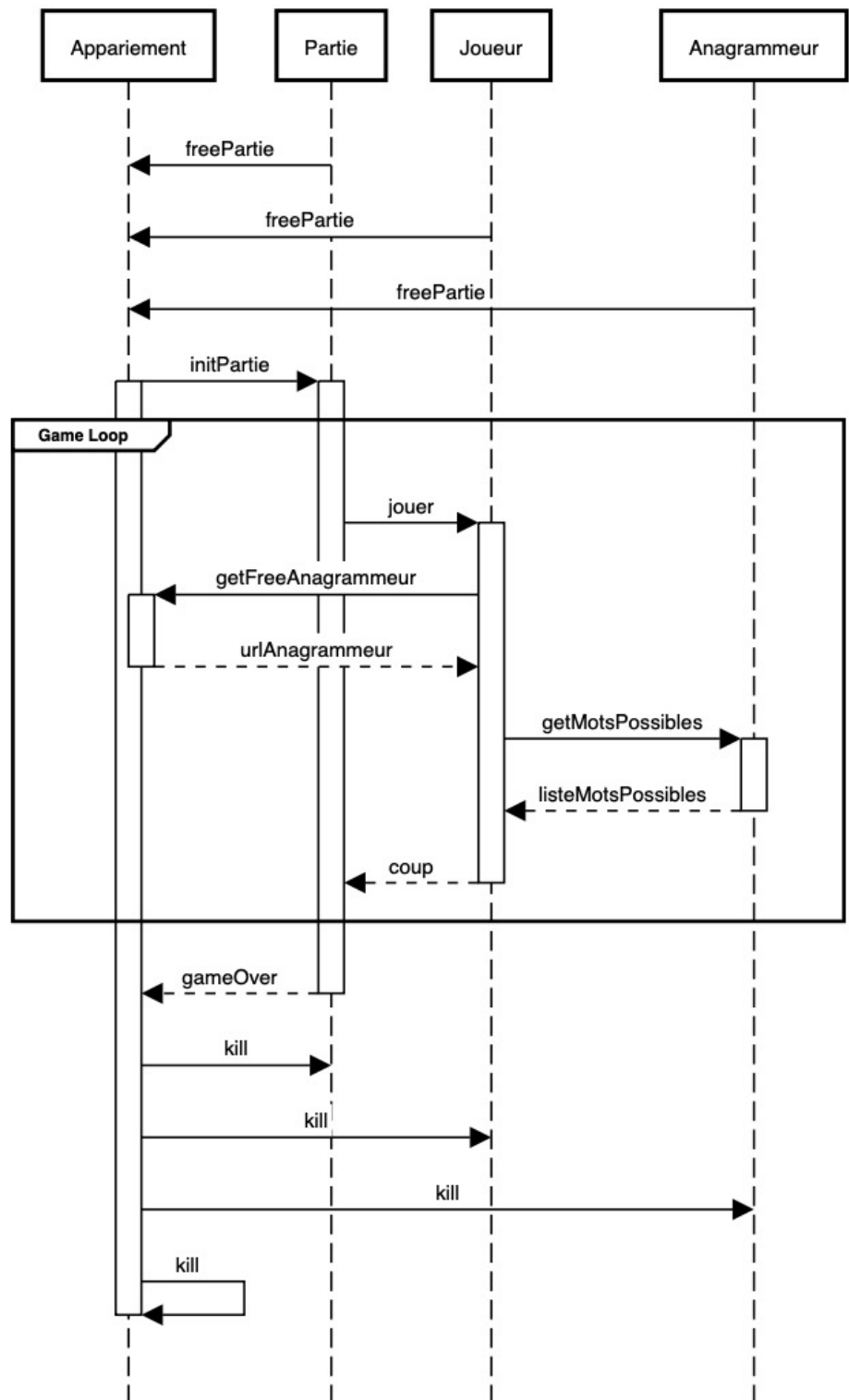
Quand ce dernier a une Partie et quatre Joueurs il envoie un message à Partie pour lancer la partie.

Partie commence le jeu, il y a plusieurs tours qui ont lieu jusqu'à ce qu'il y ait un gagnant.

Chaque tour, Partie demande à chaque Joueur de jouer, le message contient la main du joueur et le plateau actuel.

Lorsque le Joueur est invité à jouer, il demande d'abord à Anagrammeur tous les mots possibles avec sa main. Avec la réponse de ce dernier, il choisit la meilleure option et répond à Partie avec le mot et sa position sur le plateau.

Dès qu'il y a un gagnant, Partie répond à Anagrammeur. Anagrammeur tue alors tous les web services y compris lui-même. Ceci est fait pour avoir un build réussie dans Travis.



Service Stateless VS Statefull

Anagrammeur est stateless, car ce service renvoie les mots correspondant aux lettres mis en paramètres et ne tient pas compte du contexte de la partie et des joueurs.

Appariement est statefull, car ce service vérifie le nombre de parties libres, le nombre de joueurs libres et le nombre d'anagrammeur libres avant de créer une partie et d'y ajouter les joueurs.

Partie est statefull, car ce service va mettre à jour l'état de chaque joueur au niveau de leurs mains et leurs points respectifs, et du plateau courant. Ainsi, il gardera en mémoire l'état des joueurs et du plateau.

Joueur est stateless, car ce service se contentera de jouer lorsqu'il en aura la possibilité en fonction de la main attribuée par la partie et le plateau.

Tests

Pour chaque modules il existe un fichier test associé.

Pour les module Anagrammeur, Appariement, Joueur et Partie nous avons créé des fichiers nommés *TestNomdumodule.java*. Dans ces fichiers des tests unitaires sont réalisés afin de s'assurer du bon fonctionnement des méthodes codées.

Dans le module Partie, se trouve le fichier de test *TestPartieController.java*, c'est un test Spring et d'intégration qui simule le lancement et le déroulement d'une partie de façon simplifiée.

Démonstration vidéo

La vidéo est disponible en **cliquant ici** (ou à cette url : <https://www.youtube.com/watch?v=vpwwS-k3Ss>).