

Université Côte d'Azur/DS4H  
MIAGE – MASTER 1

# Rapport final

## Suivi de projet

par

Paul BUREL

Geoffrey LALIC

Axel MAÏSSA

Djade KHALDI

Rémy NGUYEN

Aryamaan SINGH KUNWAR

Monsieur Olivier OUDOT

# Sommaire

<b>Introduction</b>	<b>3</b>
Objectifs du projet .....	3
Liste des membres de l'équipe avec leur rôle respectif .....	3
<b>Contenu technique</b>	<b>5</b>
Sujet du projet.....	5
Vue statique .....	5
<b>Planning</b>	<b>7</b>
Liste des tâches.....	7
Diagramme de Gantt.....	9
<b>Suivi de projet</b>	<b>10</b>
3 KPI .....	10
Autres modifications ajoutées au tableur .....	10
Programme Python.....	10
<b>Bilan</b>	<b>11</b>
Bilan Global sur le Suivi de Projet.....	11
Apprentissage, Problème et Améliorations.....	11

# Introduction

---

## Objectifs du projet

L'objectif de ce projet est d'appliquer le plus fidèlement possible les méthodes, techniques et outils de suivi de projet telles qu'on pourrait le faire lors de la réalisation d'un projet en entreprise.

Ce suivi est appliqué à un projet de développement Java : la réalisation d'un scrabble sous forme de webservice.

Les étapes d'analyse, de conception et de réalisation se sont succédées afin d'effectuer quatre livraisons sur une période de huit semaines. Une date de rendu a été définie, laquelle n'étant pas extensible.

Les six membres de l'équipe se sont divisés en trois binômes et se sont repartis des modules différents.

Chaque semaine l'avancement a été reporté sur un tableur afin de suivre le consommé de chaque tâche et de remarquer les dépassements éventuels.

---

## Liste des membres de l'équipe avec leur rôle respectif

Dans l'ordre alphabétique,

### **Paul BUREL**

Binôme de Ayramaan

S'occupe du développement de la classe Partie, accès sur les tests fonctionnels sur l'ensemble des classes et implémentation spring, tests des classes communes.

### **Geoffrey LALIC**

Binôme de Rémy

Développement de la classe Anagramme, mise en place de Docker, test de la classe anagramme, commun, conception algo joueur. Ecriture du programme Python permettant de générer le suivi de projet.

### **Axel MAÏSSA**

Binôme de Djade

Développement de la classe Joueur et de classes communes. Suivi de projet, rédaction du rapport final.

**Djade KHALDI**

Développement de la classe Joueur et de classes communes. Intégration de spring boot, test d'intégration de Joueur.

**Rémy NGUYEN**

Développement de la classe Anagramme, mise en place de Docker, test de la classe anagramme, commun, conception algo joueur. Suivi de projet.

**Aryamaan SINGH KUNWAR**

Développement de la classe Partie, classes communes, architecture et implémentation spring boot.

# Contenu technique

---

## Sujet du projet

Le but du projet de développement est de créer un scrabble sous forme de webservice. Chaque service (un joueur, la partie, ...) est un conteneur docker qui expose un webservice. Une partie est jouée sans humain.

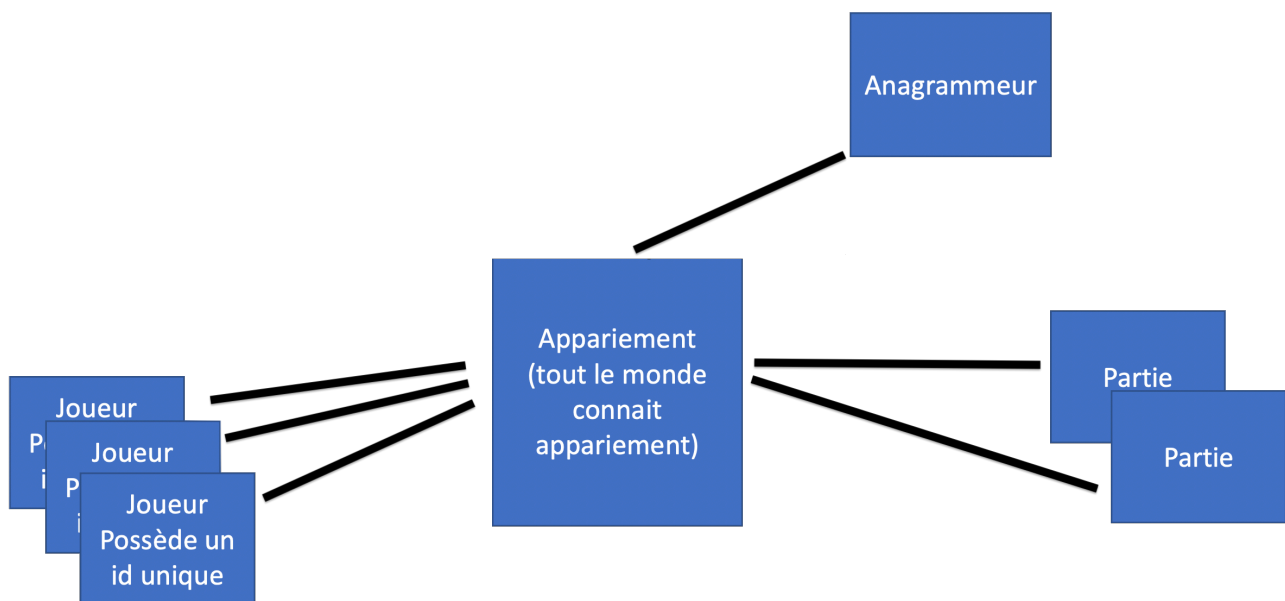
La partie distribue 7 lettres à chaque joueur, elle leur demande de jouer, les joueurs propose une liste des « Propositions » (Objet contenant une lettre, et la position à laquelle elle doit être posée) ou, s'il ne peut pas jouer, il défausse une lettre et la partie lui en donne une nouvelle en échange.

La partie reçoit cette Proposition, vérifie qu'elle est valide (vrai mot, position autorisée, ...) et l'inscrit sûr le plateau.

---

## Vue statique

Une architecture générale a été donnée dans le sujet avec un conteneur par joueur, une classe appariement, un anagramme et un conteneur par partie.



Le déroulement est organisé par la classe partie. La classe Appariement est chargée de faire le lien entre un joueur qui souhaite jouer et une partie. L'anagrammeur est chargée de trouver des mots lorsqu'on lui donne un array de lettre.

Sont imposés des langages, outils et technologies, voici la liste :

- Java / Java Spring

- Architecture Web Service
- Versionnage avec Github
- Intégration continue sur Travis CI
- Test d'intégration
- Orchestration des conteneurs avec docker-compose

# Planning

---

## Liste des tâches

### **Analyse sujet (AS) :**

Cette tâche consiste à cerner le sujet ainsi que la logique et les règles du jeu de Scrabble.

### **Conception jeu scrabble (CS) :**

Le but de cette tâche est de concevoir le jeu, sous forme de code, donc distinguer les différentes parties à coder.

### **Heure de TD dédié au projet (TD) :**

Heure de travaux dirigés par M. RENEVIER dans le cadre de l'UE Outils pour Ing. Logicielle et M. OUDOT pour l'UE Suivi de projet. De façon générale, l'équipe ne pointera pas sur cette tâche car pendant les heures de TD, l'équipe a travaillé sur d'autres tâches (développement, suivi de projet).

### **Réunion Team (R) :**

Comptabilisation des réunions de l'équipe.

### **Project Tracking (PT):**

Tâche qui regroupe l'activité de suivi de projet (inscription workload), le script python permettant de générer de KPI et la rédaction du rapport final.

## **MODULE JOUEUR (DJ)**

### **Architecture de la classe Joueur (DJC):**

Mise en place d'un squelette de Joueur.

### **Développement fonctions Joueur (DJF):**

Développement des méthodes de joueur : algorithme permettant de choisir des lettres à poser ainsi que leur position.

### **Développement WebServices Joueur (DJWS):**

Développement du webservice joueur afin qu'il puisse être utilisé par Partie.

## **MODULE PARTIE (DP)**

**Architecture de la classe Partie (DPC):**

Mise en place d'un squelette de la classe Partie.

**Développement fonctions Partie (DPF):**

Développement des méthodes de Partie, gestion des tours, des points, de la pioche et du plateau.

**Développement WebServices Partie (DPWS):**

Développement du webservice Partie.

**MODULE APPARIEMENT (DA)****Architecture de la classe Appariement (DAC):**

Mise en place d'un squelette de la classe Appariement.

**Développement fonctions Appariement (DAF):**

Développement des méthodes de la classe Appariement.

**Développement WebServices Appariement (DAWS):**

Développement du webservice Appariement.

**MODULE ANAGRAMMEUR (DAN)****Architecture de la classe Anagrammeur (DANC):**

Mise en place d'un squelette de la classe Anagrammeur.

**Développement fonctions Anagrammeur (DANF):**

Développement des méthodes de la classe Anagrammeur avec algorithme qui permet de choisir des mots en fonction d'une liste de caractères.

**Développement WebServices Anagrammeur (DANWS):**

Développement du webservice Anagrammeur.

**MODULE COMMUN (DC)****Architecture des classes Commun (DCC):**

Mise en place d'un squelette des classes communes (Plateau, Proposition, Case, ...).

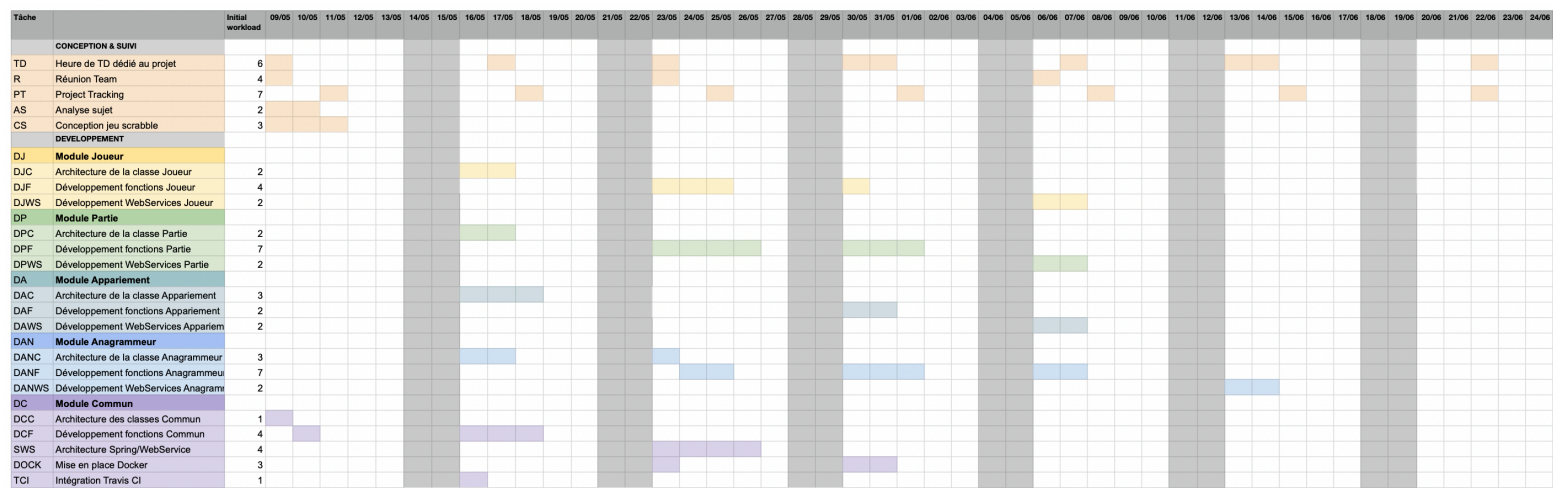


Développement des méthodes des classes communes. Plateau, Plateau factory, case, proposition.

## Architecture générale des Webservices avec spring.

## Dockerisation des webservices.

## Intégration des pipelines Travis qui seront exécutées lors d'un commit + push.



# Suivi de projet

---

## 3 KPI

### ESTIMATION INITIALE VS ACTUEL

Afin de voir si nous avons bien estimé notre initial workload, nous avons créé un indicateur simple qui fait la différence entre l'estimation initiale et actuel. Si ce chiffre est positif, c'est que nous avons surestimé la tâche (ou l'ensemble des tâches), s'il est négatif, c'est que nous l'avons sous-estimé. Enfin, s'il est nul, c'est que l'estimation était parfaite.

Formule : initial workload - actual workload

### JOURS DE TRAVAIL RESTANT

### JOURS DE TRAVAIL MANQUANT

---

## Autres modifications ajoutées au tableur

Décrire les autres modifications que vous avez apportées aux tableur de base, comme des macros par exemple

---

## Programme Python

Décrire le programme Python réalisé, comment il fonctionne et les fonctions qu'il implémente, en particulier les états de sortie

# Bilan

---

Bilan Global sur le Suivi de Projet

---

Apprentissage, Problème et Améliorations

Ce que vous avez appris, ce qui vous a ennuyé, ce qui pourrait être amélioré, etc