



PS5 - Sprint final

Ph. Collet - lundi 06/02/2023

Planning du sprint de la semaine

- **Lundi 6 matin** : publication du sujet / préparation du sprint (issues dans 1 ou plusieurs milestones)
- **Lundi 6 après-midi** : passage dans toutes les équipes
- **Mardi 7** : Passage dans toutes les équipes + support à la demande
- **Mercredi 8** : Support à la demande
- **Jeudi 9 matin** : Autonomie
- **Jeudi 9 après-midi** : Entretien technique (planning à venir)
- **Vendredi 10 matin** : Entretien technique (planning à venir)
- **Vendredi 10 après-midi** : Fin du sprint en quasi-autonomie + support exceptionnel à la demande
- **Vendredi 10 février 22h** : Livraison finale tag FINAL (code + rapport)

Fonctionnalités additionnelles 1

- Simulation de 1000 parties de votre meilleur bot contre le second (avec d'autres bots pour compléter)
- Simulation de 1000 parties de votre meilleur bot contre lui-même (ou autant de clone de lui-même que de joueurs)
- L'affichage sur la sortie standard doit comprendre le nombre et pourcentage de parties gagnées/perdues/nulles, et le score moyen de chaque bot, et **aucun autre affichage** (il faut donc supprimer pour la livraison finale toute trace de partie, tout en les conservant pour une démo sur 1 partie). Utiliser du logging est la meilleure solution.
- Logging en Java : <https://www.vogella.com/tutorials/Logging/article.html>

Fonctionnalités additionnelles 1

- Le passage en mode démo 2 x 1000 parties ou une seule partie avec des logs (pour la soutenance) se fait par options sur la ligne de commande

- 2 x 1000 parties (utilisées par le script de livraison) :

```
mvn exec:java -Dexec.args="--2thousands"
```

- Mode démo d'une seule partie avec log complet, pour la soutenance :

```
mvn exec:java -Dexec.args="--demo"
```

- Vous **devez** vous appuyer sur la librairie JCommander

- https://jcommander.org/#_overview

Fonctionnalités additionnelles 2

- En complément, un fichier .csv regroupant les statistiques des bots devra être produit à l'emplacement "stats/gamestats.csv".
- Dans un second temps, si la simulation est relancée, et si ce fichier est présent, il sera relu et les nouvelles statistiques seront agrégées, puis de nouveau sauvegardées.
- Vous **devez** vous appuyer sur la librairie **OpenCSV**
 - <https://howtodoinjava.com/java/library/parse-read-write-csv-opencsv/>
- Références pour gérer les chemins indépendamment du système d'exploitation :
 - <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/nio/file/Path.html>
 - <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/nio/file/Paths.html>

Fonctionnalités additionnelles 2

- Donc, ajout d'une troisième option en ligne de commande à implémenter avec JCommander :
 - Lancement d'une simulation à plusieurs parties (pas forcément 1000) avec relecture de "stats/gamestats.csv" s'il existe et ajout des nouvelles statistiques

```
mvn exec:java -Dexec.args="--csv"
```

- A démontrer en soutenance aussi !

Fonctionnalités additionnelles 3

Implémentation (et **tests** assurant que le comportement est conforme) d'un bot avec le comportement suivant :

- Il récupère un maximum de bambous, même s'il n'a pas de cartes avec la couleur correspondante.
- Il essaie d'avoir 5 cartes objectif en main tout le temps. Les deux premiers mouvements du bot devraient donc être de prendre une carte objectif et de prendre un canal d'irrigation.
- Quand il tire une la météo « ? » dans les premiers tours, il prend une irrigation.
- Il essaie de se focaliser sur plusieurs (pas toutes) cartes à la fois. S'il a deux cartes Panda, il va se focaliser sur les deux en même temps.
- Il surveille les mouvements de ses adversaires (ou de l'adversaire potentiellement en tête), et s'il détecte qu'il essaie de réaliser un objectif particulier, il essaie de saboter ou de ralentir la réalisation.

Approche à suivre

Réaliser au moins un MVP (Minimum Viable Product) pour chacune des 3 fonctionnalités

= code + test

Faire des choix pour progresser sur les fonctionnalités

En livrant toujours quelque chose testé et fonctionnel (pas de tunnel)

Le travail est décomposable facilement

Les 3 fonctionnalités sont quasiment indépendantes

Si vous êtes en retard dans le périmètre du jeu et des bots

Livrer forcément les 2 premières fonctionnalités (1000 runs + CSV)

**Livrer un jeu et vos bots implémentés dans un tout cohérent
+ un MVP du bot demandé**

Soutenance / Entretien technique

25 minutes avec 2 évaluateurs dont 10 minutes de présentation

démonstration des fonctionnalités réalisées : 1000 parties, une partie avec log, la version csv, un mélange avec une option d'exécution qui vous est spécifique. Cela reste complètement ouvert mais attention au temps !

explications sur comment la version finale est conçue, comment elle est testée + bilan qualité Sonar discuté à gros grain (qu'avez-vous corrigé, quelle dette avez-vous laissé, etc.)

explications sur comment la version finale a été réalisée (issues, milestones, git branching strategy...), quels bons principes vous avez appliqués, quels mauvais principes vous avez détectés

explications sur qui a fait quoi dans l'équipe, et son organisation

Slides pas obligatoires mais ça peut aider...

Livraison finale

Vendredi 10 février 22h

Tag FINAL

```
git clone url_de_votre_rep id_projet  
cd id_projet  
git checkout FINAL  
mvn clean package  
mvn exec:java -Dexec.args="--2thousands"
```

Attention, tous les tests doivent passer sinon le script le détectera et n'effectuera pas l'exécution

Rappels

Vous devez lier les commits aux issues

Pas de commit sans issue liée !

Les issues terminées sont fermées

Les milestones terminés sont fermés

Une issue (tâche) est majoritairement fonctionnelle

**Appartient à 1 seul milestone
ou est dans le backlog en attente**

Peut être non fonctionnelle (refactoring, setup mockito...)

**Une issue non terminée dans un milestone doit
changer de milestone d'appartenance**

Pas d'issue récurrente ou multi-milestones

Scenario industriel

Votre équipe va partir sur un autre projet et faites comme si votre projet allait être repris par une autre équipe.

Vous faites une rétrospective de comment s'est déroulé ce projet.

Mercredi après-midi au plus tard : Ayez une discussion entre vous pour identifier ce que vous avez réussi ou raté.

Pour le prochain projet, que conserveriez-vous et que changeriez-vous ?

Vous écrivez une page GitHub (format markdown, suffixe .md) qui sera lue par la nouvelle équipe qui reprend la suite pour l'aider à se familiariser avec votre projet.

« Rapport » - Knowledge Exchange (1/2)

1. Point d'avancement

- un résumé des fonctionnalités réalisées sur tout le jeu (couverture des règles du jeu indépendamment de ce qui était demandé cette semaine), et éventuellement la liste de ce qui n'a pas été fait.
- un résumé de ce qui a été fait pour les logs (en quelques lignes max, quels choix ont été faits pour les réaliser)
- un résumé de ce qui a été fait pour les statistiques en CSV (en quelques lignes max, quels choix ont été faits pour les réaliser)
- un résumé de ce qui a été fait pour le bot spécifique demandé, et éventuellement une comparaison avec votre meilleur bot et une analyse de pourquoi celui qui gagne est le meilleur

« Rapport » - Knowledge Exchange (2/2)

2. Architecture et qualité

- Comment est faite l'architecture du projet ? Et quels choix vous ont amené à la réaliser ainsi ?
- Où trouver les infos (de la java doc, de la doc sur les points et les classes importants ?)
- Etat de la base de code : quelles parties sont bien faites ? Quelles parties sont à refactor et pourquoi ? Comment la sortie SONAR le montre-elle (ou pas) ?

3. Processus

- Qui est responsable de quoi / qui a fait quoi ?
- Quel est le process de l'équipe (comment git est utilisé, branching strategy)

Livraison du « rapport »

Sous le tag FINAL

**Dans un répertoire nommé doc, une page au format
markdown (.md) nommée**

RAPPORT-X.md

où X est la lettre de votre équipe (A, B, ...)

**Réfléchir au contenu du rapport mercredi vous aide
à extraire ce qui est le plus important à montrer et
à dire en soutenance technique...**







Dans la semaine ?



slack

#si3-ps5