



PS5 - Sprint final

Ph. Collet - lundi 03/01/2022



2022



TODO (if not already done)

Chaque groupe crée un canal privé slack avec le format suivant:

si3-ps5-21-22-lettre

(où lettre est la lettre de votre groupe, de A à U)







en y ajoutant @Anne-Marie Dery

@Mathias Cousté

@Johann Mortara

et moi (@Philippe Collet)

Planning du sprint de la semaine

- **Lundi 3 matin** : publication du sujet / préparation du sprint (issues dans 1 milestone)
- **Lundi 3 après-midi** : passage dans les équipes - discussion sur le rapport 
- **Mardi 4** : Daily meeting  + support à la demande 
- **Mercredi 5** : Daily meeting  + support à la demande 
- **Jeudi 6** : Entretien technique (planning à venir) 
- **Vendredi 7** : Fin du sprint en autonomie
- **Samedi 8 janvier 20h** : Livraison finale tag FINAL

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 2

- En complément, un fichier .csv regroupant les statistiques des bots devra être produit à l'emplacement "resources/save/results.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 pouvez vous appuyez sur la librairie OpenCSV si vous la souhaitez et une référence ci-dessous vous explique comment lire dans un fichier placé dans le répertoire "resources", standard dans une organisation maven.
- Lire depuis le répertoire resources: <https://howtodoinjava.com/java/io/read-file-from-resources-folder/>
- OpenCSV : <https://howtodoinjava.com/java/library/parse-read-write-csv-opencsv/>

Fonctionnalités additionnelles 3

- Implémentation (et tests assurant que le comportement est conforme) d'un bot appliquant les deux séries de conseils de débutants de cette conversation de forum : <https://www.trictrac.net/forum/sujet/citadelles-charte-citadelles-de-base>



Richard Publié le 14 mai 2003 20:44:39

Ce texte est (une ébauche) destiné à orienter les joueurs novices.
A vous de le compléter ou de le modifier si besoin.

Deux objectifs ;

- éviter la frustration des nouveaux joueurs face à des situations basiques qui leur semble pourtant trop complexes et les amènent à prendre une décision uniquement basée sur le hasard
- éviter la frustration des joueurs avertis qui jouent avec des joueurs novices et qui pourraient leur reprocher des erreurs d'analyse sur des situations basiques.

Ce texte n'a pas pour but d'être exhaustif en terme de stratégie, ce qui me semble à la limite de l'impossible.

Stratégie de base

1/

Assassin

- Je ne tue pas le Voleur sauf :

Fonctionnalités additionnelles 3



Alphonse Nonyme402 Publié le 29 oct. 2005 19:32:47

C'est un guide de fin de partie un peu bizarre. Je le complèterais bien avec des conseils plus généraux pour débutants.

Je joue le plus souvent à 4 ou 5, avec les personnage de base (je n'aime pas le côté "autiste" de la plupart des autre spersonnages), et en inversant l'ordre de choix a chaque tour. Le roi joue toujours en premier, mais le scrates personnage tournent un tour sur 2 dans le sens des aiguilles d'une montre et un tour sur 2 en sens inverse. Cela permet d'éviter qu'un joueur ne se fasse pourrir toute la partie en choissant trop souvent en dernier. Cette stratégie enlève un tout petit peu d'intérêt au roi

Citadelle est un jeu de construction et d'accumulation de ressources. En clair : a la fin, c'est celui qui a pu dépenser le plsu d'argent qui gagne souvent, on peut se faire plaisir à tuer tout le monde avec l'assassin, mais ca ne fait pas gagner à la fin.

En gros

Assassin :

+

Il est peu vulnérable, sauf au condottiere et au magicien

Il peut ralentir un autre joueur

-

Il ne rapporte RIEN. Un joueur qui prend trop fréquemment l'assassin gagne rarement

Quand vous avez beaucoup de cartes et que les autres ont des mains vides, prenez l'assassin pour assassiner le magicien

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

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

Entretien technique

25 minutes avec 2 évaluateurs dont 10 minutes de présentation

démonstration des fonctionnalités réalisées : cela peut être différent de l'exécution de livraison qui tourne 1000 parties, avec une seule exécution de partie avec des logs précis de ce qui se déroule. Cela reste complètement ouvert.

explications sur comment la version finale est conçue, comment elle est testée

explications sur comment la version finale a été réalisée (issues, milestones sur GitHub), quels bons principes de votre rapport vous avez appliqués

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

Slides pas obligatoires mais ça peut aider...

Livraison finale

Samedi 8 janvier 20h

Tag FINAL

```
git clone url_de_votre_rep id_projet  
cd id_projet  
git checkout FINAL  
mvn clean package  
mvn exec:java
```

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