***Projet - Takenoko***

Novembre 2022

V1.0

Philippe Collet

# **Sujet**

Il s'agit de réaliser une version informatisée du jeu "Takenoko" d’Antoine Bauza, en Java, compilable et exécutable avec maven selon le canevas donné à l’initialisation dans votre repository github.



Il faut pour cela réaliser les fonctionnalités suivantes :

* Représentation de jeu (représentation des plateaux, parcelles, des sections de bambou, des irrigations, aménagements, cartes d’objectif, des jetons actions, comptage des points, validité des coups) ;
* Moteur de jeu (mise en place, gestion des pioches, lancement des dés, actions, détermination des points, de la fin de la partie...) ;
* Robots de jeu, en partant d'une version très simple à des stratégies de jeu plus sophistiquées ;
* Simulation de parties entre 2 robots ou plus, comptage des points, des victoires et classement entre vos robots ;
* Visualisation textuelle de l'état du jeu (en fin de partie pour commencer, tout au long de la partie ensuite).

**Points importants sur la portée du projet :**

* Une version graphique **N'EST PAS** demandée, ne sera pas demandée par la suite, et NE SERA PAS valorisée, même en bonus.
* **AUCUNE** saisie sur la ligne de commande n’est permise.
* Une version avec un ou plusieurs joueurs humains **N’EST PAS** demandée, ne sera pas demandée par la suite, et NE SERA PAS valorisée, même en bonus. Il s’agit d’avoir uniquement plusieurs joueurs robots intelligents, avec des stratégies potentiellement différentes.
* La priorité est donnée à une version à 2 joueurs robots intelligents, plutôt qu'à 3 ou 4 robots assez stupides.
* Pour faciliter le feedback du client, chaque nouvel élément intégré dans votre moteur de jeu et dans la représentation doit être utilisé par au moins un robot.

# **Ressources sur le jeu**

* Règles du jeu en PDF : <https://www.trictrac.net/jeu-de-societe/ressource/clique/383f6c35aed2eaace1a85aec076074c1/takenoko>
* Résumé des règles : <https://jeuresume.files.wordpress.com/2013/03/takenoko-1-1.pdf>
* Page du site tric-trac : <https://www.trictrac.net/jeu-de-societe/takenoko>
* Page de ressources externes du site tric-trac (présentations vidéo, etc.) : <https://www.trictrac.net/jeu-de-societe/takenoko/ressources>
* Site de jeu en ligne contre d'autres joueurs : <https://boardgamearena.com/gamepanel?game=takenoko>

# **Nouveaux outils de développement**

En plus de l’utilisation systématique des tests unitaires en JUnit et de Maven pour la construction, dans le cadre de ce second projet, nous vous demandons toujours de piloter le découpage **progressif** des fonctionnalités en slices (une slice définit de la valeur “métier” verticalement), mais aussi, vu la taille plus importante du projet, des livraisons stables et fournissant de la valeur chaque semaine. Pour cela, plus de fichier Excel, mais des outils Agiles, et simples, fournis par GitHub, comme vu lors du dernier cours.

Vous devrez donc :

* Créer un *Projet* sur github : <https://docs.github.com/en/issues/planning-and-tracking-with-projects/creating-projects/creating-a-project>
* Créer une issue par tâche (une tâche technique comme changer le pom.xml de Maven, refactorer, etc.) ou une slice (un ajout de fonctionnalité verticalement dans le projet) :
  + <https://docs.github.com/en/issues/tracking-your-work-with-issues/creating-an-issue>
  + Avec des issues “feature” (pour les slices) et “task” (pour les tâches...). Vous pouvez renommer ou créer des labels : <https://docs.github.com/en/issues/using-labels-and-milestones-to-track-work/managing-labels>
  + Si vous avez besoin d'organiser des sous-tâches ou des todos dans une issue, utilisez les *task lists*, cela crée des cases à cocher très pratiques : <https://docs.github.com/en/get-started/writing-on-github/working-with-advanced-formatting/about-task-lists>
  + Lorsqu’un membre de l’équipe, commence à travailler sur une issue, il la passe de t*odo* à *inprogress.* Il va ensuite faire plusieurs commits pour avancer le développement (voir plus bas) et quand l’issue est terminée, il peut la fermer en la passant à l’état *done.*
* Utiliser la notion de *milestone* pour regrouper toutes les tâches livrées sur une semaine. Il suffit de décrire le milestone en quelques mots (c’est cette description que vous devriez donner en TD pour répondre aux questions : que me montrez-vous cette semaine ? Qu’allez-vous livrer la semaine prochaine ? Il suffit de créer des milestones et d’éditer une issue pour la relier à un milestone. Cela vous donne aussi une roadmap visible en termes d’issues closes et encore ouvertes
* Lier chacun de vos commits à une issue en particulier. Cela veut dire que chaque ajout ou modification de code doit se faire dans le cadre d’une issue (ne faites pas de commit qui concerne plusieurs issues, c’est une mauvaise pratique pour pouvoir tracer les changements dans les grands projets). Pour lier un commit, il suffit de référencer le numéro de l’issue dans le message de commit : <https://docs.github.com/en/get-started/writing-on-github/working-with-advanced-formatting/autolinked-references-and-urls>

Au prochain cours, nous introduirons aussi des notions de « branche » et de « pull request » sur git pour faciliter le développement. Pour l’instant, démarrer le découpage en suivant la procédure qui suit.

# **Démarrage**

Vous devez tous évidemment commencer le travail par comprendre le métier, donc jouer (en ligne) ! Quel que soit votre niveau, si vous êtes déjà dans une équipe, vous pouvez créer l’équipe et le repository en le nommant takenoko-2023-LettredEquipe avec votre lettre d’équipe à la fin (votre lettre est donnée sous la déclaration faite par message sur slack), en suivant le lien suivant (nouveau Classroom GitHub pour ce projet) :

<https://classroom.github.com/a/wb2v21Ep>

Ensuite, il faut pour la prochaine séance (donc à la rentrée) :

* Initialiser le projet GitHub comme expliqué ci-dessus ;
* Faire un premier découpage en milestone (un milestone par semaine jusqu’à la première semaine prochaine) en réfléchissant à de la livraison régulière de fonctionnalités (développer un bot à la fin n’est clairement pas une option). Ces milestones vont sûrement changer dans leur contenu mais l’idée est de pouvoir vous donner du feedback en TD pour mieux les définir et piloter de mieux en mieux en autonomie par la suite.
* Écrire et détailler les premières issues “slices” pour le premier milestone, et au moins avoir les issues sans description pour le second. Tout détailler n’est pas demandée et serait i) contre-productif, ii) sûrement faux, car vous avez besoin d’avancer dans le développement pour avoir une meilleure vision, mieux découper, mieux tester, mieux livrer, etc.
* Idéalement, avec un premier milestone très simple et bien vertical, vous pourriez avoir le code et les tests déjà prêts avec le premier milestone !

**Travail à réaliser et planning**

* Livraison d’une démo cohérente (fonctionnalités nouvelles réalisées de bout en bout, représentation->moteur->robot) et stable (fonctionnalités présentées testées) **chaque semaine**
* Livraison d’une version compilable et exécutable la plus avancée possible (mais stable !) le vendredi 16 décembre à 18h (tag MIDTERM).
* **De nouvelles fonctionnalités** (et la livraison finale avec un rapport) arriveront avec la semaine à temps plein PS5 du 6 février 2023 !