

# Rapport de projet : Takenoko

Équipe G : Maxime Bouis, Corentin Artaud, Soufiane Aouri, Hugo Croenne

## Synthèse du projet

### Fonctionnalités réalisées :

Dans le moteur du jeu nous avons mis en place la possibilité de poser des parcelles en respectant les règles du jeu (adjacente à l'étang ou à au moins deux autres parcelles). Nous avons implémenté les entités Panda et Jardinier qui vont pouvoir se déplacer en ligne droite. Le Panda mangera un bambou sur sa case d'arrivée si elle en contient et le Jardinier fera pousser un bambou sur sa parcelle d'arrivée ainsi que sur ses voisines. Nous avons mis en place une pioche pour les parcelles et des pioches pour les objectifs Panda, Parcelle et Jardinier qui peuvent être validés. Nous avons implémenté l'empereur qui sera affecté au joueur qui fini la partie.

Nous avons développé deux robots pouvant jouer avec le moteur de jeu, décrit ci-dessus, au travers de la façade Joueur. Les robots observent l'état du jeu par le Joueur associé, font un choix d'action et leur Joueur l'effectue. Le premier robot, le moins intelligent, fonctionne de manière séquentielle sur chaque action possible et la réalisera aléatoirement sauf pour le Panda qu'il essaie de déplacer sur la parcelle où se trouve le jardinier. Le second robot, plus intelligent, va se focaliser sur un seul objectif de sa main et tenter de le réaliser en utilisant un algorithme cherchant à optimiser le nombre de coups à jouer. De plus, il est possible de lui donner différentes stratégies à suivre comme se focaliser plus sur un type d'objectif que sur les autres.

### Fonctionnalités non réalisées :

Nous n'avons pas implémenté les aménagements, la météo et les irrigations. De plus, nous objectifs contenant plusieurs couleurs ne sont pas dans la liste des objectifs à réaliser car ils sont plus complexes à gérer par les robots que les objectifs unicolores.

### Niveau de confiance :

Nous sommes confiants sur la qualité du moteur de jeu car il est couvert par de nombreux tests. Cependant nous sommes moins confiants quant à la qualité des robots pour lesquels il manque des tests.

### Parties de bonne qualité :

La façade Joueur et le moteur du jeu sont de bonne qualité ainsi que les tests écrit pour les vérifier. De plus, les algorithmes de résolution des objectifs du robot intelligent sont performants.

### Parties de mauvaise qualité :

Les robots ne sont presque pas testés car nous avons eu du mal à comprendre comment écrire des tests pour ces parties sachant que nous ignorons quel sera l'état du terrain sur lequel ils effectueront leurs actions.

# Pourquoi c'est un bon projet ?

## Organisation :

Nous avons découpé le projet en releases fines, ayant ainsi à la première release une version avec deux joueurs et un robot qui pose aléatoirement sur le plateau des parcelles incolores contenant un bambou chacune par défaut et fait déplacer le panda (Fonctionnalités : robot aléatoire, parcelles incolores, déplacement du panda).

## Code :

L'architecture du projet est solide, le code est propre, avec de bons choix concernant l'utilisation de différentes notions de la programmation orientée objet, et les structures de données les plus adaptées aux différents problèmes et besoins.

Le moteur du jeu est testé sur deux temps : des tests unitaires pour chaque classe, et des tests utilisant un robot programmable pour simuler une partie.

Une couverture de 63 % sur l'ensemble du projet.

## Démonstration :

Nous avons deux robots dont un plus intelligent que l'autre. Le moins intelligent est équilibré car il gagne à 50 % contre lui même. Quant au second, le plus intelligent, il gagne à 99 % contre le robot moins intelligent. Le plus intelligent contre lui même gagne à 50 % cependant lorsqu'on change ses stratégies on peut observer de légères différences : un robot se focalisant plus sur les objectifs Panda fasse au même se focalisant plus sur les objectifs Jardinier va gagner à 55 % ce qui montre que certains objectifs sont plus rentable que d'autres à réaliser.

# Pourquoi c'est un mauvais projet ?

## Organisation :

Nous avons encore des difficultés pour communiquer, nous n'étions pas toujours au courant des dernières modifications effectuées ainsi que des problèmes rencontrés par les autres membres du groupe.

## Code :

Nous avons commencé le développement du robot intelligent un peu trop tard. C'est pourquoi il n'a pas évolué parallèlement au développement du moteur du jeu, ainsi nous avons pas eu le temps d'ajouter quelques fonctionnalités comme les irrigations étant implémentées mais pas utilisées par le robot ou comme les aménagements que nous avons choisi d'abandonner car ils n'auraient jamais été utilisés.

## Démonstration :

Toutes les fonctionnalités n'ont pas été implémentées comme les aménagements, la météo ou seulement partiellement comme les irrigations n'étant pas utilisées par les robots.

## Rétrospective

Nous devons continuer à effectuer des réunions régulières (au moins une par semaine) afin de faire un point sur l'avancement de chacun et de l'ensemble du projet tout en énumérant les nouvelles tâches à réaliser et en les affectant à chacun. Nous devons aussi continuer à nous entraider et à conserver une bonne entente et écoute au sein du groupe.

Nous devons arrêter définitivement de procrastiner car nous avons accumuler du retard à cause de ce défaut lorsqu'il y avait des problèmes à résoudre.