

# Compte-rendu du projet de colonisation

Bastien PRUNIER, Bahia SECHI, Clément SERVAT, Matéo RABOTIN

## Introduction

### Répartition des rôles :

- Planète : Bahia Bastien
- Robots : Matéo Clément
- On pense échanger nos rôles de temps à autres pour voir tout le projet

## Description du système

### Planète :

- JavaFX : Pour l'affichage des sprites, de la planète.
- Concepts mis en œuvre sur la planète :
  - Logique floue
  - Système multi-agent (plusieurs agents réactif, attentif à leur entourage pour remarquer une possible modification)
  - Métamorphose :
    - Nous avons décidé d'opérer les métamorphoses sur la planète de façon à ce que les voisins de la case exploitée soient impactés. Ainsi, nous nous servons de la logique floue pour obtenir le pourcentage de métamorphoses et d'une fonction permettant de récupérer les voisins de la case.
    - Quand l'eau d'une case de lac est totalement épuisée, la case se transforme en désert.

**La planète :** L'agent planète est un agent **réactif**. Son but est d'empêcher sa colonisation par les robots en modifiant sa surface. Il suit l'avancée des robots et peut lancer une métamorphose si une exploitation est trop importante. Cet agent n'a connaissance que des cases composant la planète (leur état, (extraction ?) etc ...). Pour savoir quelles cases ont été récemment modifiées, la planète sera notifiée grâce à un système d'évènement et choisira si oui ou non elle doit métamorphoser la région et à quel point. Elle forme un système à part entière : elle est toute seule, les cases ne sont pas des agents.

### Colonisation :

Tous les robots seront des agents **cognitifs**, capable d'analyser le terrain et déduire des zones de danger et de sécurité.

- Centralisateur :
  - Missions : Rester à la base, Surveillance l'exploration, collecte de données, comptabilise les ressources (récolte et consommation), Cartographie des ressources de la planète
  - Comportement -> multi-agent un pour chaque mission et un qui chapote le tout ?
- Agriculteurs :
- 3 Extracteur de minerai : recherche de minerai -> Q-Learning
  - Missions : Mine et rapporte à la base
- 3 Récolteurs de nourriture :
- 3 Constructeurs de pipelines :



**Les robots :** Les robots sont des agents pour l'instant réactif. Ces robots se déplacent dans le but d'explorer la planète et trouver des ressources. Ils connaissent leur environnement en utilisant des capteurs, servant d'adaptateur entre la planète et les robots, et agissent en fonction de leur outil, leur stratégie de mouvement et leur type de capteur. Dans l'état actuel du projet, ils n'ont pas une grande quantité d'intelligence si on peut la quantifier : il se limite à se déplacer vers des ressources, les exploiter et communiquer avec le robot central. Ce robot central sert de "cerveau" pour l'ensemble du système des robots et centralise des données telles que les tableaux de Q-learning (un par type de robot), les tableaux de ressources connues et des statistiques comme l'arrivée d'eau. Il est le seul robot avec qui la colonie interagit.

Les catégories de robots à retenir sont :

- Agriculteur
- Mineur
- Pipelineur
- Ramasseur

Tous les robots sauf le pipelineur vont chercher leur ressource et l'exploiter. Ils reportent la case au robot central et ramènent leurs récoltes à la base. Quant à lui, le pipelineur va chercher de l'eau pour construire un pipeline entre cette case et la base. Après l'avoir créé, il va faire des aller-retours entre la base et la case d'eau.

## Difficultés rencontrées

- Nous avons, dans un premier temps, écrit notre fichier FuzzyLogic (.FLL) mais ce dernier était malformé. Nous avons alors cherché de nombreuses ressources, notamment sur la documentation officielle ainsi que sur le forum de FuzzyLite. En fouillant dans le cours, nous avons trouvé un exemple fourni par M. LEBRET. Nous l'avons donc repris pour obtenir un squelette de fichier et l'avons modifié petit à petit pour y inscrire les valeurs du projet de colonisation.
- Compréhension des algorithmes et fonctionnement de l'IA sans expérience préalable
- Nous avons passé beaucoup de temps à préparer la conception de la simulation et au final quelques fonctionnalités prévues n'ont pas pu être implémenté ce qui donne donc du code non utilisé comme le patron observer dans la planète.

## Améliorations possibles

- Pouvoir laisser l'utilisateur modifier les fonctions d'appartenance à sa guise. L'idéal serait de faire une deuxième fenêtre en JavaFX et de pouvoir entrer les valeurs qui seraient récupérées et modifiées par rapport au fichier FLL par défaut.
- Créer un bordereau permettant de visualiser le nombre de cases exploitées, métamorphosées, le nombre de robots en marche ou cassés. En bref, une sorte de tableau de bord utile pour suivre les différentes données disponibles.
- Implémenter la dynamique de pipeline pour les robots puissent rapatrier les ressources prélevées à leur base.
- Régler les valeurs que nous avons inscrites arbitrairement pour que la simulation soit bien homogène et fonctionne bien.
- Finir la JavaDoc pour les fonctions un peu plus secondaires, nous nous sommes concentrés sur l'essentiel. Nous pouvons également la générer une fois cela fait. Attention au petit souci entre Lombok et JavaDoc.

## Aides extérieures

- Cours d'Intelligence Artificielle – 2A ENSICAEN – Alain LEBRET
- L'Intelligence Artificielle pour les développeurs - Virginie MATHIEVET
- Forum [FuzzyLite](#)