

# Simulation Multi-Agent – Analyse et conception

Pierre CHEVALIER, Pierre-Loup PISSAVY

décembre 2015

## 1 Présentation

L'énoncé nous demande de créer une simulation multi-agent dont le cadre d'évolution est une matrice  $20 \times 20$ .

Le sujet que nous avons choisi consiste à simuler la conquête de planètes par des factions et chaque case est un agent planète.

### 1.1 Déroulement et objectifs

L'unité de temps par itération pour la simulation est l'UTI (Unité de Temps Interstellaire). Le but des agents factions est de détruire la faction adverse, et l'objectif des planètes contrôlées est d'exécuter la volonté de leur faction.

### 1.2 Règles

Le monde est composé de planètes et de factions. Pour exister, chaque faction doit posséder une planète mère. La faction développe des colonies qui pourront à partir de ressources récoltées et générées par la planète mère, mener des actions d'extension de la faction. Les ressources de la faction ne sont pas infinies et elle choisit de les attribuer ou non.

Pour conquérir une planète limitrophe, il faut dépenser une certaine quantité de ressources, et lorsque la planète est déjà occupée par une autre faction, il faut ajouter une valeur de défense. Une fois conquise, la planète apporte des ressources supplémentaires à la banque centrale de la faction.

La défense des colonies est croissante dans le temps, lorsqu'elles ne sont pas attaquées. Si une faction voit sa planète mère conquise par une faction adverse, elle disparaît et devient alors une colonie de l'adversaire, les planètes alors occupées par ses ex-colonies deviennent libres.

## 2 Analyse

### 2.1 Classes d'agents

- Factions (groupe d'agents planètes),
- Planètes,
  - Planètes mères,
  - Planètes colonies.

## 2.2 Factions

Nos agents appartiendront à la catégorie des agents cognitifs.

La faction par exemple aura connaissance seulement des planètes limitrophes à sa frontière, et aura un désir d'expansion et de destruction des factions ennemies. On peut même penser à implémenter un système de rancune envers une faction adverse qui a été plus agressive qu'une autre : cela aura pour effet de influencer les choix de la faction concernant les attaques.

Les planètes de la faction, quant à elles, recevront des ordres de ladite faction mais auront également une volonté propre, avec un système de rancune parallèle. Toutefois, aucune mutinerie n'est prévue à ce stade.

## 2.3 Structures de donnees

La grille de la simulation sera un tableau de tableaux, car on a besoin d'accès rapides pour répartir l'activation des agents.

Les factions sont composées d'un pointeur vers la liste des planètes possédées. Nous avons choisi une liste car il n'est pas possible de savoir à l'avance combien la faction possèdera de planètes. Les planètes possèdent également un pointeur vers la faction à laquelle elles appartiennent.

Schéma UML

# 3 Conception

## 3.1 Dimension aléatoire

Si le besoin s'en fait ressentir, nous nous baserons sur des probabilités pré-existantes dans les jeux vidéos ou recensées dans des films de science-fiction, par exemple  $\frac{\text{nombre de planètes détruites}}{\text{nombre de planètes attaquées}}$ .

Diagramme de Gantt

Selon le temps qui nous restera, nous pensons implémenter d'autres règles comme des planètes spéciales, des technologies, de la spécialisation dans les factions etc.

Dans le cadre de l'aspect cognitif de nos agents, nous pouvons penser à développer un système de négociation (cessez-le-feu, aide d'une planète alliée, etc.) et également un système d'espionnage.