

Simulation TP5

Cahier des charges Système Multi-Agents

Arquillière Mathieu - Zangla Jérémy

12 décembre 2019

Table des matières

1	Introduction	2
2	Modélisation	2
2.1	Agents	2
2.1.1	Récolteurs	2
2.1.2	Mangeurs	2
2.2	Environnement	3
2.2.1	Elements de l'environnement utiles au agents	3
2.3	Diagramme de classe	3
3	Interface	4

Table des figures

1	Diagramme état-transition de l'agent <i>récolteur</i>	2
2	Diagramme état-transition de l'agent <i>mangeur</i>	3
3	Diagramme de classe du système multi-agents	4
4	Diagramme cas d'utilisation	4

1 Introduction

L'objectif ici est de réaliser un système multi-agents simple, afin d'en comprendre et d'expérimenter les difficultés et les intérêts.

Notre problématique est donc :

« Quels agents créer et comment les intégrer dans un système afin qu'ils interagissent entre eux et avec l'environnement ».

Cette problématique implique de comprendre comment un ensemble d'agents doit agir en simultané dans un même espace (en 2 dimensions dans notre cas).

2 Modélisation

2.1 Agents

Notre système se composera de 2 types d'agents :

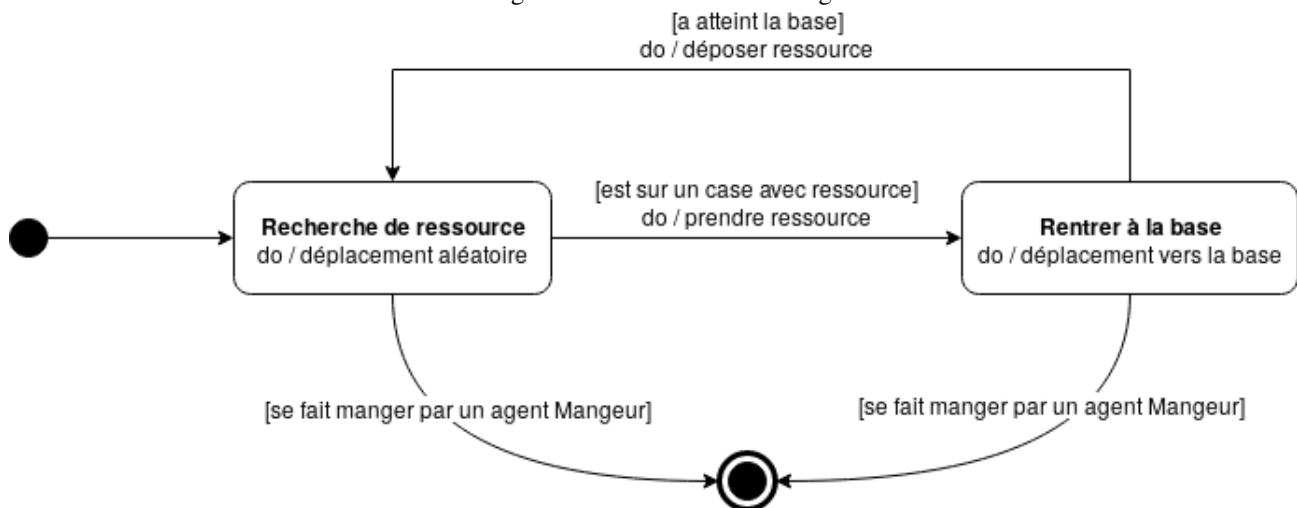
- les *récolteurs*
- les *mangeurs*

2.1.1 Récolteurs

L'objectif de cet agent est récolter des ressources et de les ramener à sa base. Chaque agent de ce type a été créé par une *case particulière* de l'environnement qu'on appellera sa *base*. Un *récolteur* bouge aléatoirement dans l'espace en 2 dimensions. Lorsqu'il trouve une case avec des ressources, il la prend et se déplace alors vers leur base (ce qui implique que ces agents savent à tous moments l'endroit de leur base). Une fois dessus, il dépose ses ressources et se remet en recherche.

Les *récolteurs* naissent grâce à une base. Lorsqu'une base possède assez de ressources, elle les consomme et crée un nouvel agent *récolteur*.

FIGURE 1 – Diagramme état-transition de l'agent *récolteur*

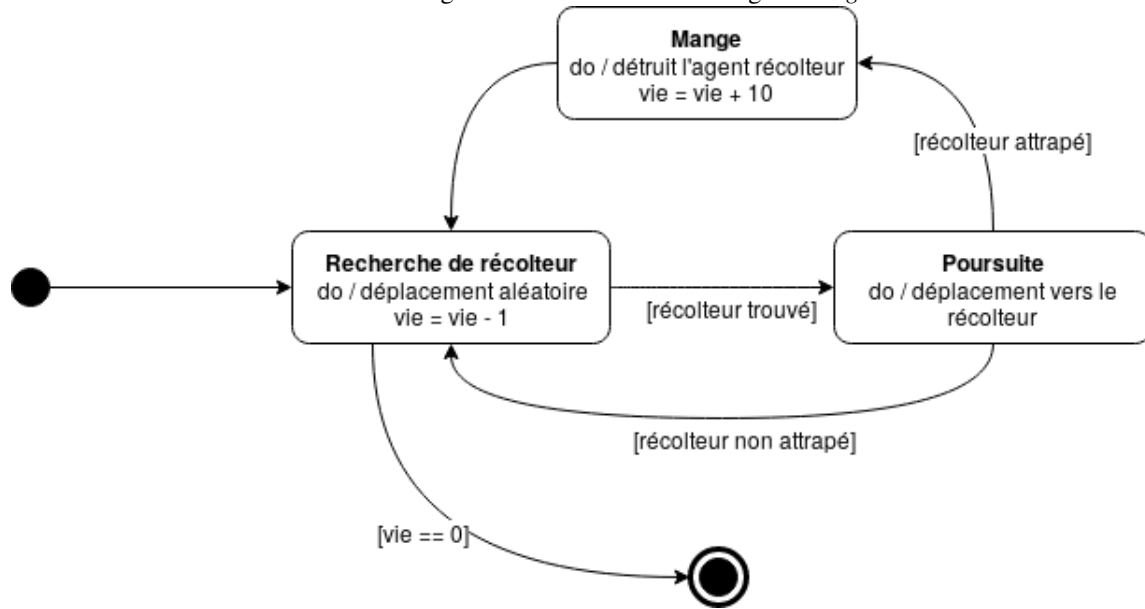


2.1.2 Mangeurs

L'objectif de cet agent est de manger des agents *récolteurs* afin de survivre. Il se déplace aléatoirement jusqu'à ce qu'il trouve dans un voisinage de Moore d'ordre 3 un agent *récolteur*. Dès lors, il se déplace vers celui-ci (ses déplacements se font dans un voisinage de Moore d'ordre 2). Si il atteint un *récolteur* (être sur la même case de l'espace 2D) alors il le détruit.

Le *mangeur* possède une "barre de vie" qui diminue à chaque nouvel état du système. Manger un *récolteur* permet de regagner de la vie. Si il en mange un alors que sa vie était assez haute (exemple : > 90%) alors il crée un nouvel agent *mangeur*.

FIGURE 2 – Diagramme état-transition de l'agent *mangeur*



2.2 Environnement

L'environnement sera une matrice (20x20 à la base). Une case de cette matrice peut :

- être vide
- contenir un agent
- contenir une ressource
- contenir une base

2.2.1 Elements de l'environnement utiles au agents

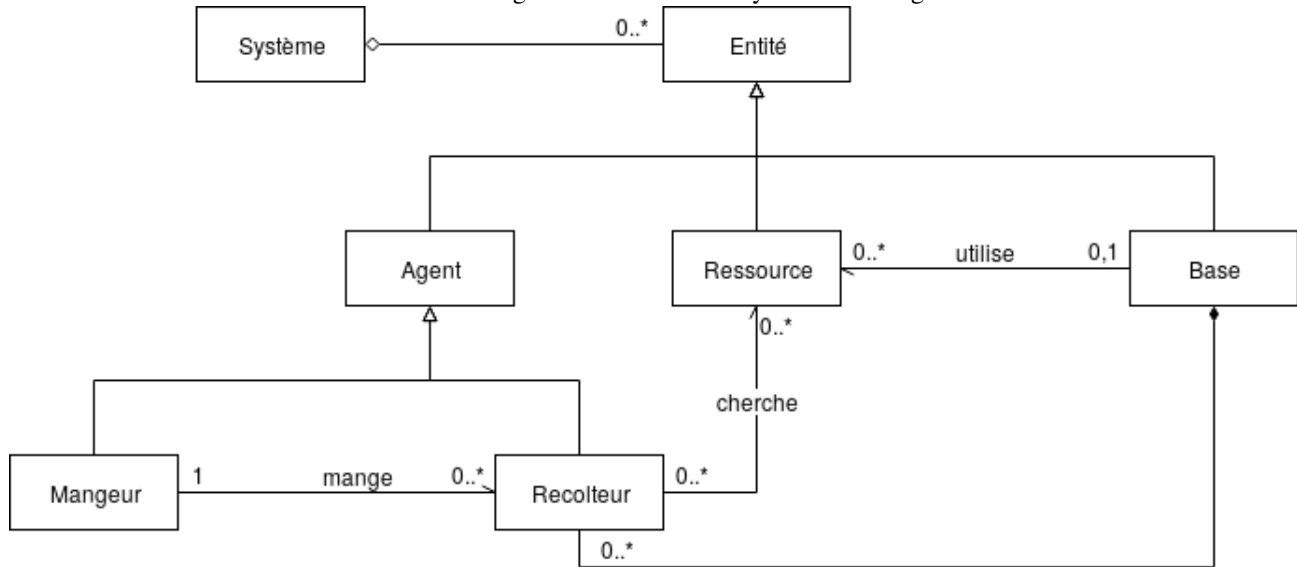
Une ressource a une position dans l'espace 2D et a un type : faible, moyen ou fort et correspond à une quantité pour les bases. Tous les certains temps du système, un certain nombre de ressources apparaissent aléatoirement dans l'espace.

Une base interagit avec des agents *récolteurs*. Elle possède une position fixe dans l'espace 2D et s'occupe de créer des *récolteurs*. En effet ceux-ci rapporte à la base des ressources qui, au dessus d'un certain seuil, sont consommées afin de générer un nouvel agent *récolteur*.

2.3 Diagramme de classe

Afin de mieux comprendre les relations entre les agents et leur environnement, on réalise un diagramme de classe non détaillé, qui sera ensuite complété et modifié lors de l'implémentation.

FIGURE 3 – Diagramme de classe du système multi-agents



3 Interface

Le système multi-agents pourra être utilisé de manière très simple. C'est-à-dire que l'interface consistera à pouvoir lancer une simulation en réglant des paramètres comme l'endroit d'apparition des agents / bases / ressources et de leurs nombres. On pourra également modifier la vie des agents *mangeurs* et les probabilités des types de ressources.

Une autre partie importante est de pouvoir récupérer des informations, des statistiques sur les systèmes qu'on observe. Ainsi notre système enregistrera certaines données au fur et à mesure de son évolution et pourra les exporter (sous forme de fichier par exemple).

FIGURE 4 – Diagramme cas d'utilisation

