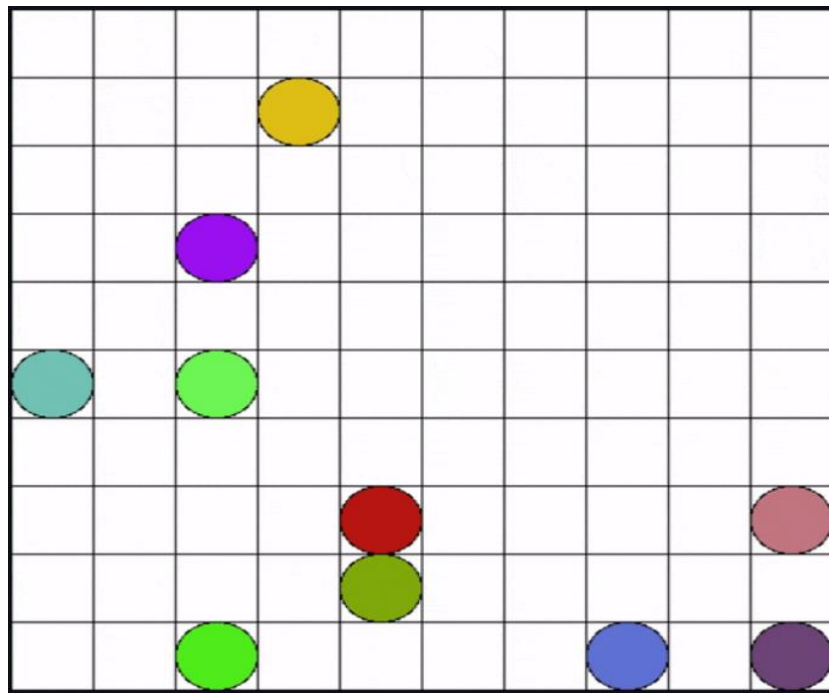


# Master Informatique et Modélisation Des Systèmes Complexes (IMSC)

Département de Mathématique - Informatique

## Compte rendu TP1 : S.M.A Architecture générale classique



Réalisé par :

➤ HAYAR HAMZA

Responsable :

➤ Mr. GADI TAOUFIK

Année universitaire : 2021/2022

## *Table des matières*

I.	Introduction.....	3
II.	Choix de JADE .....	3
III.	ConceptSion.....	4
1	Choix de Méthode .....	4
2	Phase 1 (Analyse) .....	5
➤	Les Objectifs de système : .....	5
➤	Diagramme de Cas d'utilisation .....	6
➤	Diagramme de séquence .....	7
➤	Diagramme d'activité .....	8
➤	Rôles .....	8
3	Phase 2 (Conception).....	9
➤	Représentation d'agent Particule .....	9
➤	Représentation d'agent BoardManager .....	9
➤	Représentation d'agent GUI.....	10
➤	Communication entre les Agents .....	10
➤	Diagramme de class .....	11
IV.	Implémentation.....	11
1	GUI.....	11
2	Les Messages enter les agents (outil sniffer) .....	12

## *Tableau de figures*

Figure 1 Architecture de MaSE.....	5
Figure 2 Diagramme de cas d'utilisation .....	6
Figure 3 Diagramme de séquence.....	7
Figure 4 Diagramme d'activité - déplacer .....	8
Figure 5 Représentation d'agent particule.....	9
Figure 6 Représentation d'agent BoardManager.....	9
Figure 7 Représentation d'agent GUI.....	10
Figure 8 Communication entre BoardManager - Particule .....	10
Figure 9 Communication entre Board manager - AgentGUI.....	10
Figure 10 Diagramme de Class .....	11
Figure 11 GUI.....	11
Figure 12 Sniffer .....	12

# TP1

## I. Introduction

Le but de ce TP était de réaliser un système multi-agents permettant à des agents “particules” d’interagir avec les autres à travers des rebonds.

Les systèmes qu’on a se compose de board et des particules, chaque particule a une direction et une position unique dans le board, la direction de particule se prend par hasard lors de création de cette particule. Les particules ils se déplacent dans chaque instant (frame), chaque particule suit sa direction, si deux particules au plus sont choquées entre eux alors un des particules prendre la priorité d’occuper cette nouvelle position et les autres vont dévier, ainsi que chacun prendre la direction d’autre

Dans ce système il y’a beaucoup des conflits entre les particules pour cela on doit créer un agent Broker qui se charge de synchroniser les messages des particules entre eux et de traiter tous les cas des choques

Pour visualiser ces interactions on doit créer une interface graphique (GUI) pour afficher le board ainsi chaque particule et leurs mouvements au cours du temps

## II. Choix de JADE

JADE, le Java Agent DEvelopment framework, est un cadre logiciel entièrement implémenté en langage Java. Il est conforme aux spécifications FIPA et peut être distribué sur plusieurs machines, qui n'ont même pas besoin de partager le même OS (système d'exploitation). En outre, l'architecture de communication de JADE est flexible et efficace, avec de nombreux modes disponibles, tels que le blocage, l'interrogation et le délai d'attente.

### III. ConceptSion

Afin de faire une bonne implémentation, on doit fait la conception, cela nous permet de bien comprendre ce système multi-agent, dans cette partie on travaille avec langage de modélisation AUML qui s'adresse au ce genre des systèmes.

#### 1 Choix de Méthode

Dans la conception on doit utiliser une méthodologie (Démarche), pour ce TP on a choisi la méthodologie MaSE.

La méthodologie MaSE (Multi-agent Systems Engineering) est une méthodologie complète pour le développement de systèmes multi-agents hétérogènes. MaSE couvre le cycle de vie complet du système, de l'analyse à la conception en utilisant un certain nombre de modèles basés sur des graphiques (utilise des composants AUML – Agent Unified Modelling Language -).

Les modèles sont transformés en diagrammes afin de décrire les agents du système, leurs communications, et la structure interne de chaque agent détaillé en profondeur.

MaSE est soutenu par un outil de génie logiciel appelé AgentTool. AgentTool permet au concepteur de spécifier formellement tous les modèles de MaSE. Il supporte également les transformations de conception automatisées et la vérification automatique des communications inter-agents. Le cycle de vie de MaSE est itératif. On suppose que l'analyste ou le concepteur passe librement d'une étape à l'autre et d'une phase à l'autre, de sorte que, à chaque étape successive, des détails supplémentaires sont ajoutés et, finalement, une conception complète et cohérente du système est obtenue.

Le but de la phase d'analyse est de produire un ensemble de rôles dont les tâches décrivent ce que le système doit faire pour répondre à ses exigences globales. La phase d'analyse MaSE se compose de trois étapes : Capturer les objectifs, appliquer les cas d'utilisation et affiner les rôles. Rôles.

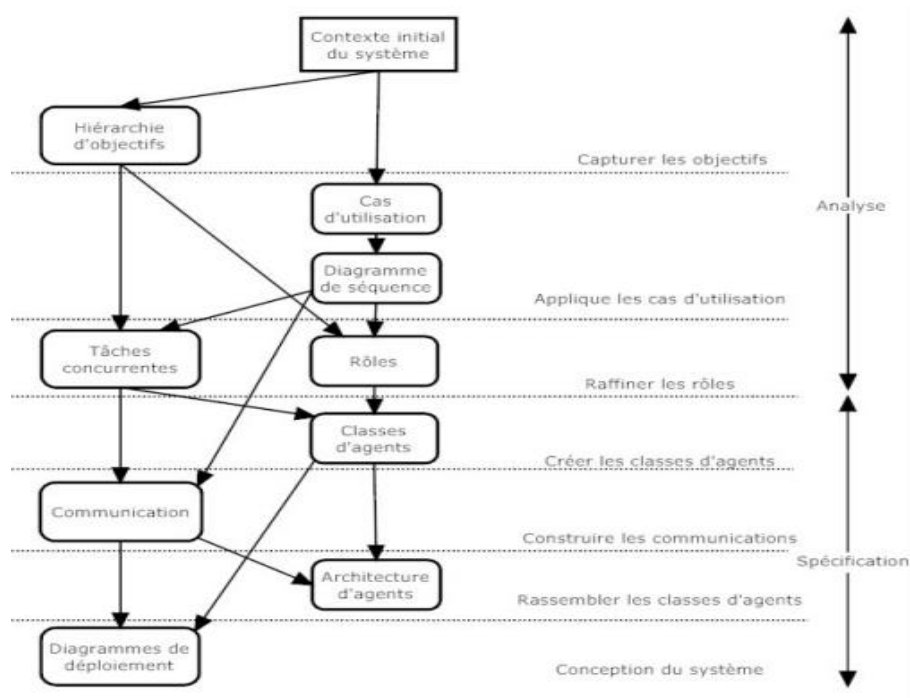


Figure 1 Architecture de MaSE

## 2 Phase 1 (Analyse)

### ➤ Les Objectifs de système :

- Un nombre des particules qui se déplacent dans un espace 2D
- Chaque particule possède une direction et des coordonnées (x,y)
- Chaque particule prendre sa direction initiale de manière aléatoire
- Les particules peuvent se choquer entre eux
- Les Particules qu'ils se choquent doivent changer leurs directions
- Afficher ces interactions dans une interface graphique
- Chaque particule se représente par une couleur distincte

## ➤ Diagramme de Cas d'utilisation

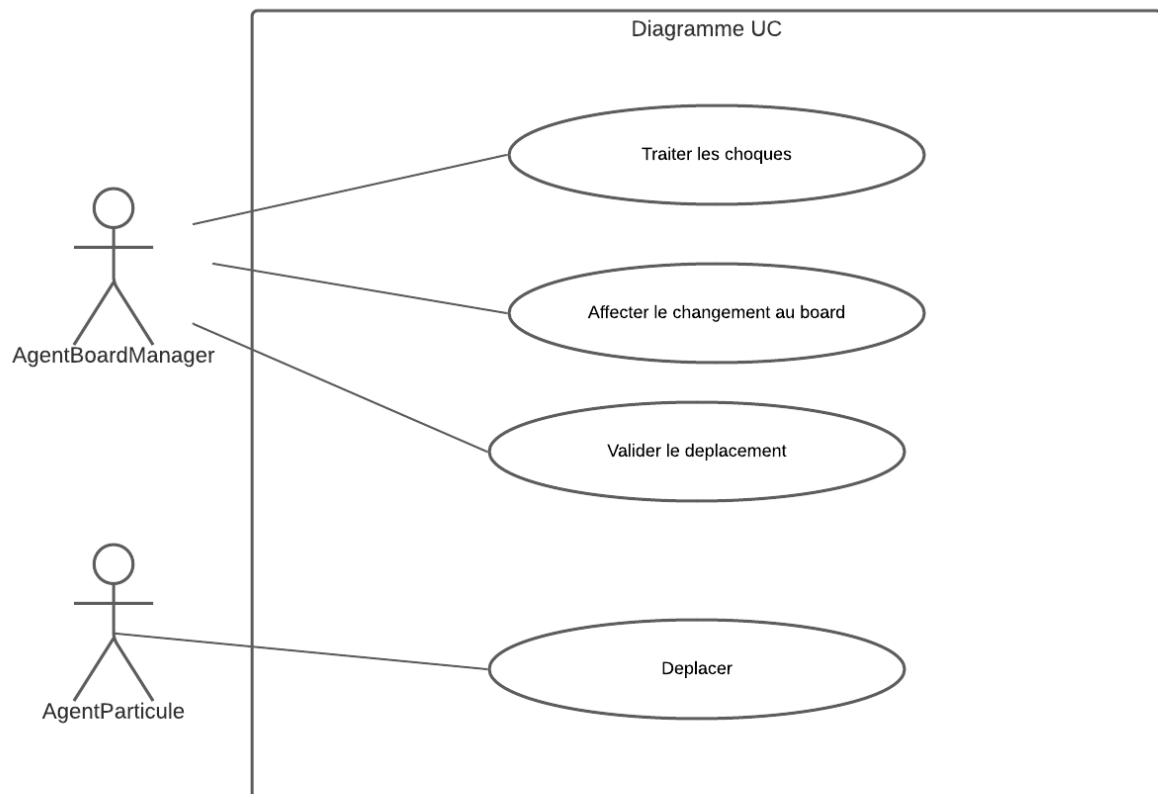


Figure 2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation de notre système est simple, on a seulement de deux acteurs qui interagissent Particule et BoardManager, Particule peut déplacer à chaque instant, et BoardManager fait le traitement des chocs, valide les déplacements des particules et affecter les changements des positions de toutes les particules à l'interface.

## ➤ Diagramme de séquence

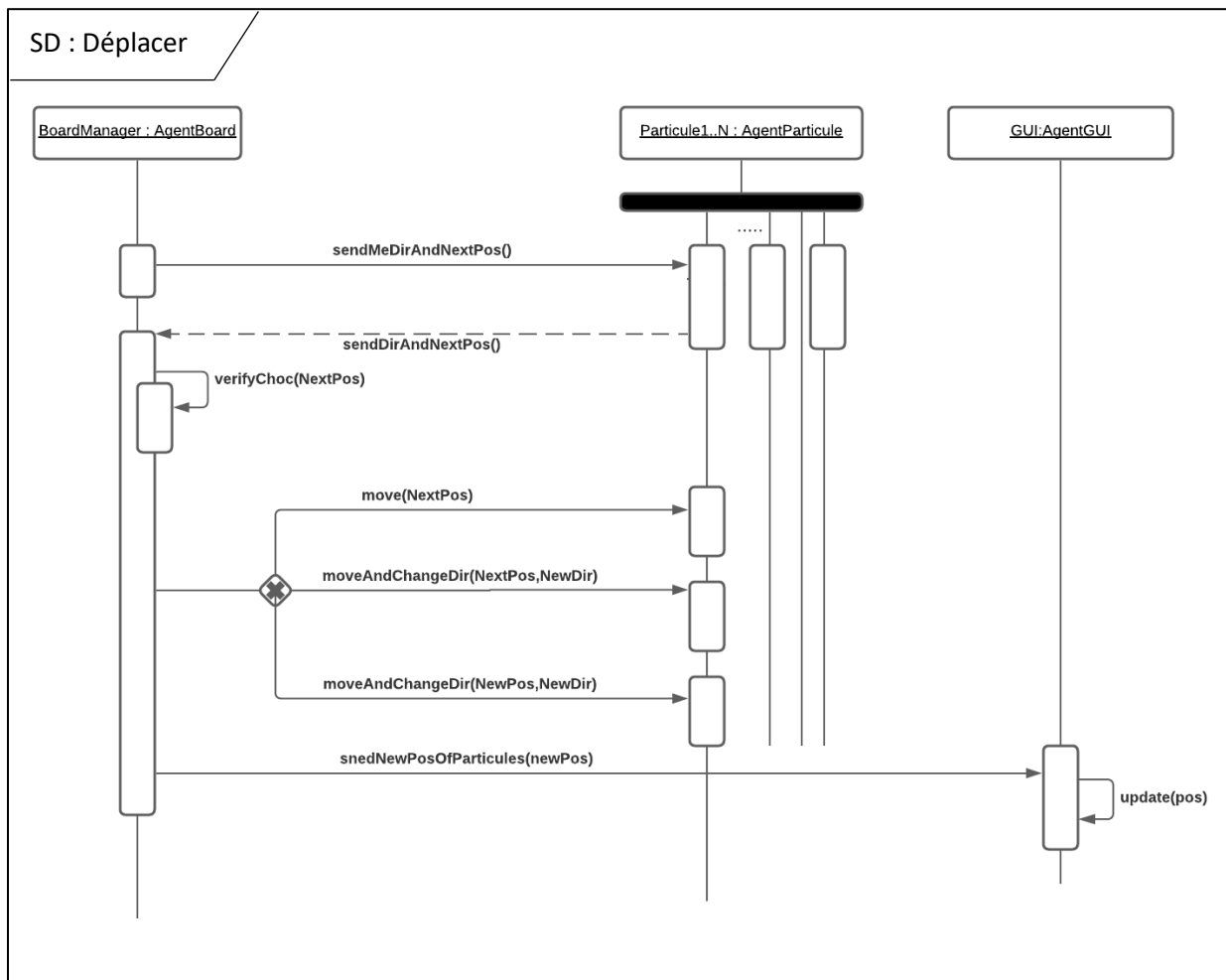


Figure 3 Diagramme de séquence

Dans diagramme de séquence on remarque que l'agent boardManager collecte les nouvelles positions des tous les particule (Les autre agent particules il interagissent par la même façon alors on les remplacent par un seul agent dans le diagramme), après la collection des nouvelle position, l'agent BoardManager vérifier s'il existe des choques et il le traite, puis il envoyer message à tous les agent particule pour déplacer.

Finalement l'agent BoardManager envoie les nouvelles positions des tous les particules à l'agent GUI pour que ce dernier affichera à l'interface graphique



## ➤ Diagramme d'activité

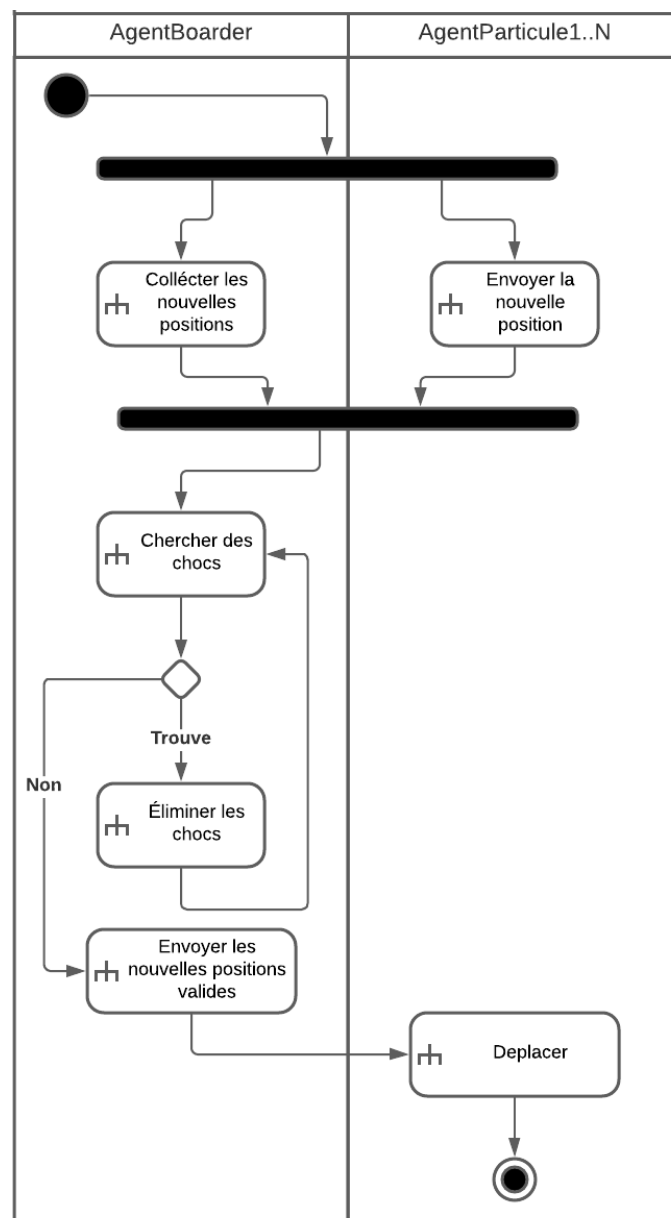


Figure 4 Diagramme d'activité - déplacer

Dans le diagramme activité on remarque que l'agent BoardManager traiter les chocs avec un boucle, et la condition d'arrêt c'est s'il ne trouve plus de choque.

## ➤ Rôles

Agent	Particule1..N	BoardManager	GUI
Rôle	Particule	Board Manager	Interface_graphique

Tableau 1 Tableau des rôles

### 3 Phase 2 (Conception)

#### ➤ Représentation d'agent Particule

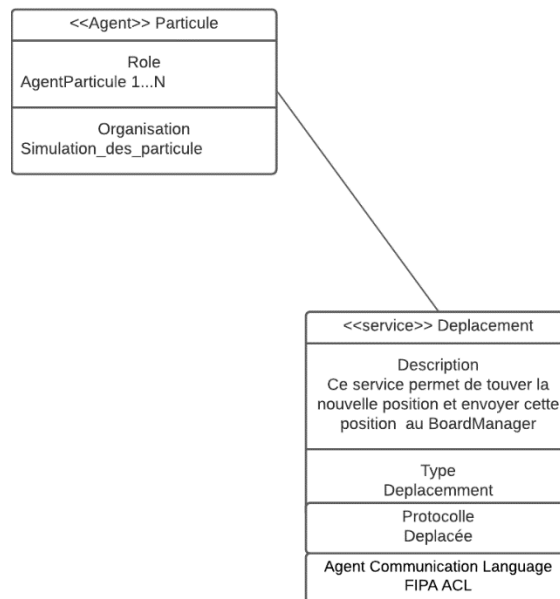


Figure 5 Représentation d'agent particule

Représentation de l'agent particule et le service de déplacement à l'aide de l'AUML

#### ➤ Représentation d'agent BoardManager

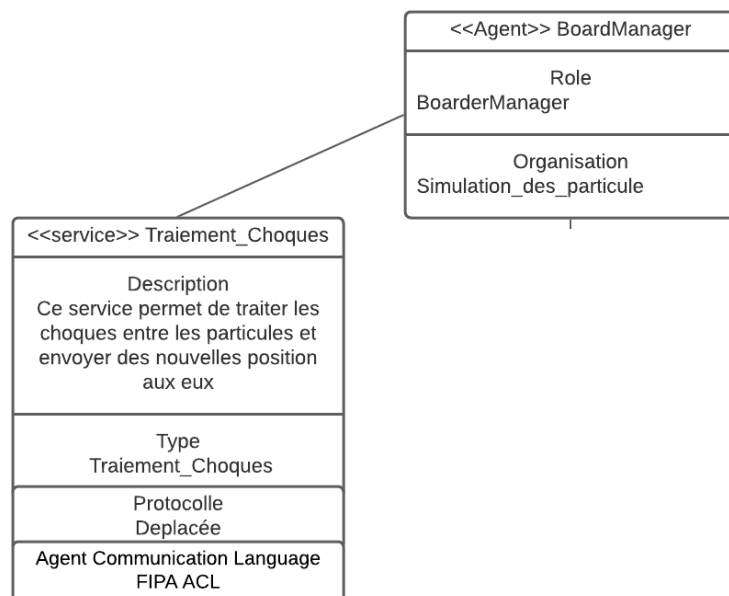


Figure 6 Représentation d

## ➤ Représentation d'agent GUI

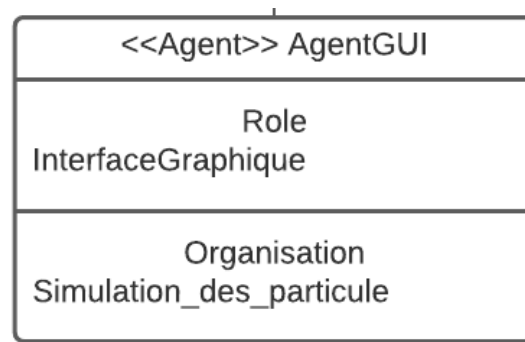


Figure 7 Représentation d'agent GUI

## ➤ Communication entre les Agents

Entre Particule et BoardManager

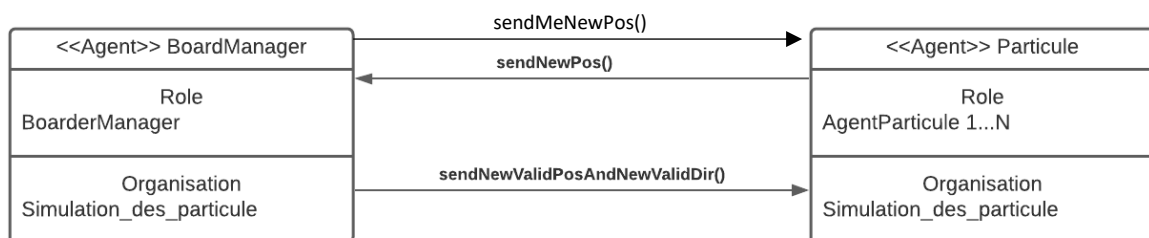


Figure 8 Communication entre BoardManager - Particule

La communication entre les agents boardmanager et particule se fait dans manière suivant

- Agent BoardManager envoie un message où il demande la nouvelle position
- Agent Particule envoie leur nouvelle position au BoardManager
- Agent BoardManager envoie les nouvelle position valide et nouvelle direction valide pour cette particule

Entre Agent BoardManager et Agent GUI



Figure 9 Communication entre Board manager - AgentGUI

La communication entre l'agent BoardManager et GUI est très simple, Agent BoardManager envoie les nouvelles positions pour toutes les particules à chaque instant (frame), et GUI reste toujours dans l'attente des messages du BoardManager

#### ➤ Diagramme de class

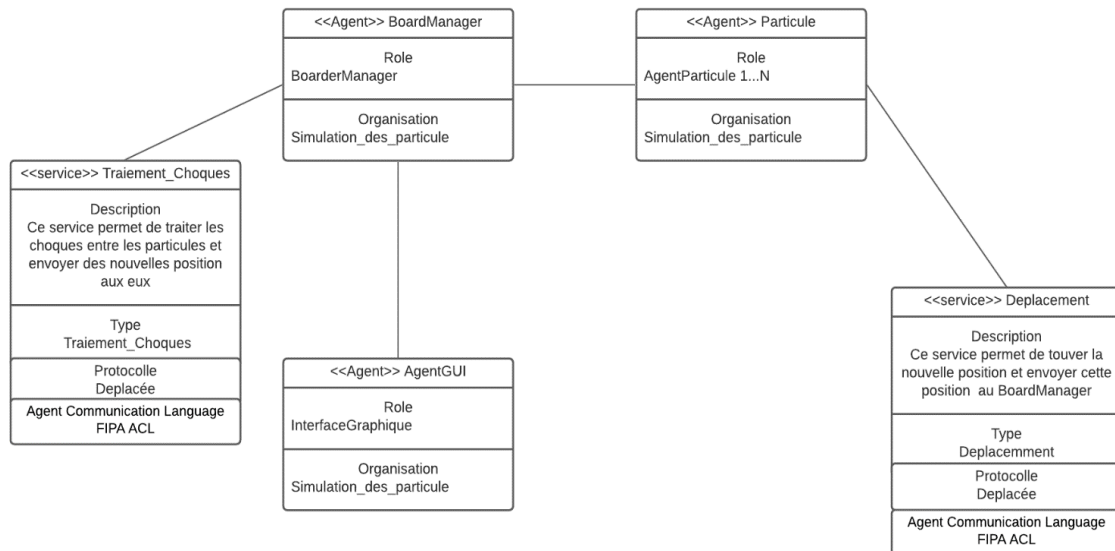


Figure 10 Diagramme de Class

## IV. Implémentation

### 1 GUI

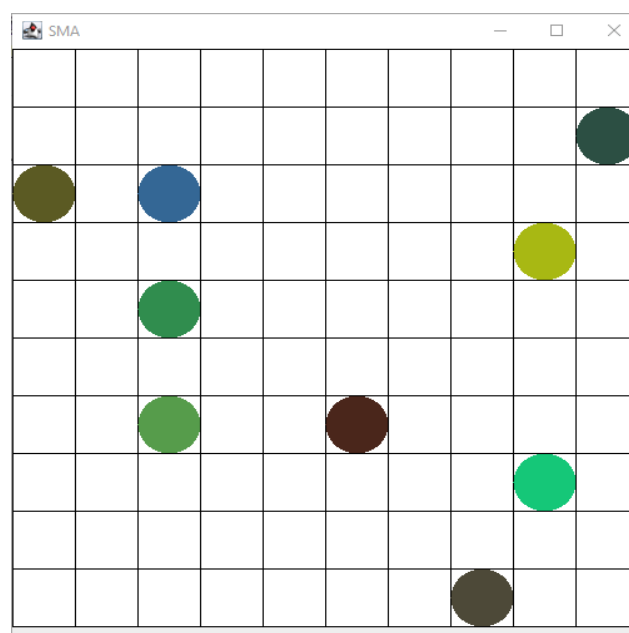


Figure 11 GUI

## 2 Les Messages entre les agents (outil sniffer)

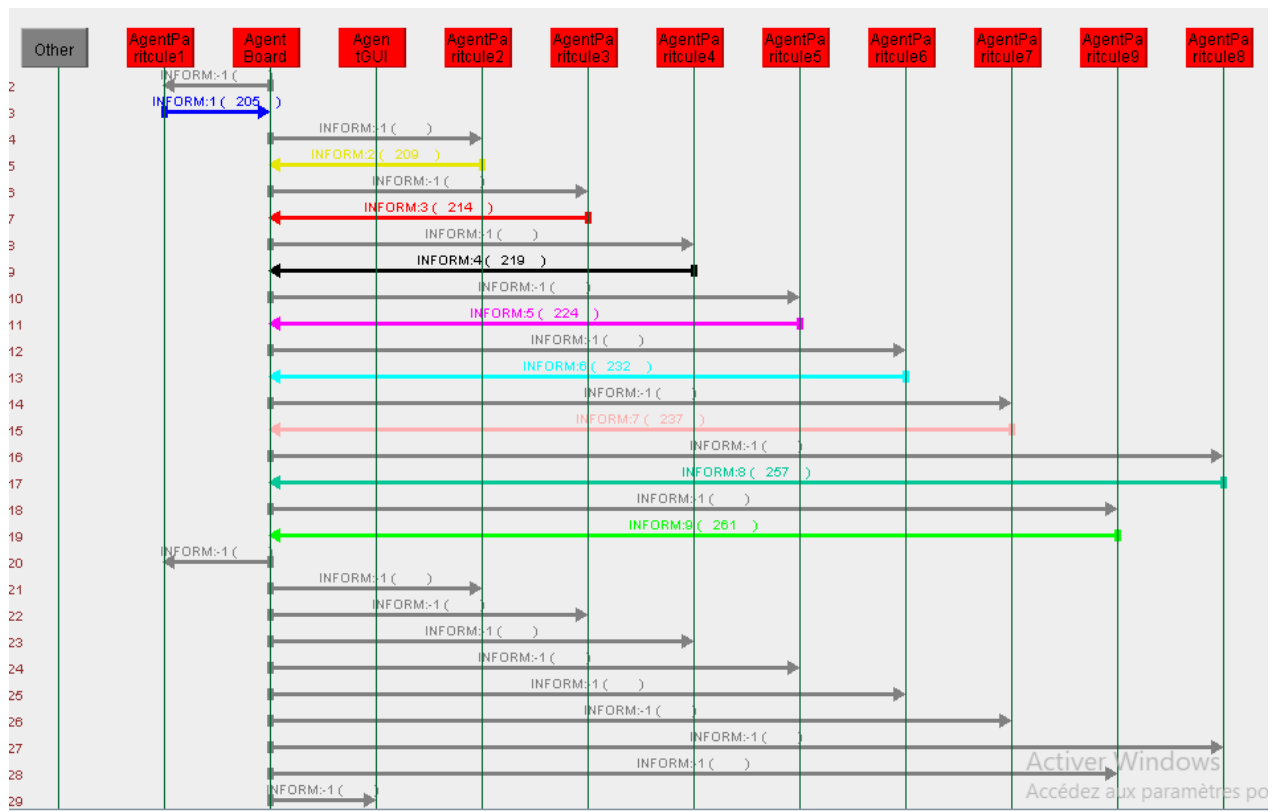


Figure 12 Sniffer

Dans la dernière figue on remarque que l'agent BoardManager (Broker) fait la synchronisation de tout le système.

Un frame se commence par le message entre Agent BoardManager et Agent Particule1 et se termine par le message entre Agent BoardManager et GUI.