

# SCI – Boite à particule

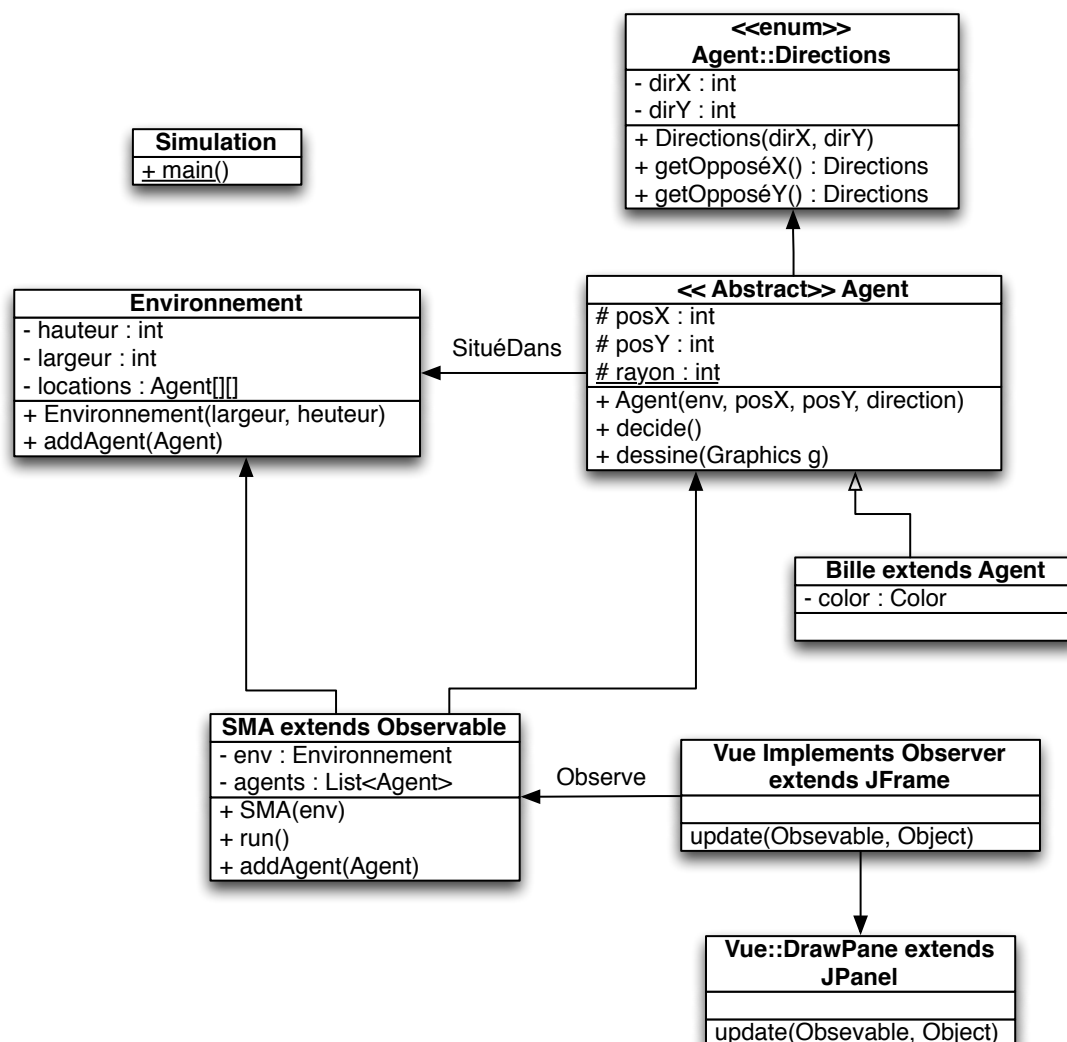
## Description du projet

Dans le cadre du cours de SCI, il nous a été demandé de réaliser une "boite à particule" en utilisant le pattern MVC et une architecture de système multi agents.

## Réalisation et choix de conception

Le projet a été réalisé en Java, et utilise Swing pour l'interface.

La découpe en classes est la suivante :



J'ai choisi de ne considérer que des agents à coordonnées entières se déplaçant dans un espace à coordonnées entières également.

La classe Agent est abstraite afin d'en permettre plusieurs implémentations selon les besoins, ici seules les Billes en héritent mais on peut très facilement envisager d'ajouter des murs, ou des rochers qui seraient immobiles, par exemple.

Les déplacements sont dictés par des directions contenues dans une Enum, ce qui simplifie le traitement en abstrayant totalement les calculs de changements de direction : quand une bille touche un mur, elle demande à sa direction quelle est la direction opposée en X ou en Y et ne doit pas connaître plus de détails que cela.

Le dessin des Agents est géré par ceux-ci afin qu'un type d'agent particulier puisse le cas échéant se dessiner d'une manière différente d'un autre (un rocher ne sera pas forcément rond, un mur non plus).

La mise à jour de la vue se fait après avoir donné un tour de décision à chaque Agent de sorte que l'on ne rafraichisse l'affichage que le minimum de fois.

### Lancement de l'application et options

La commande pour lancer l'application est la suivante :

```
java -jar ...[votre path]/SCI_fondements/target/SCI_fondements-0.0.1-SNAPSHOT-jar-with-dependencies.jar
```

Elle peut être suivie d'une ou plusieurs options, dans l'ordre :

- Hauteur : la hauteur de la zone dessinée
- Largeur : la largeur de la zone dessinée
- NbAgents : le nombre de Billes
- RayonAgents : la taille des billes en pixels
- TempsAttente : le temps d'attente entre chaque tours
- TempsArrêt : le temps avant l'arrêt de la simulation (si 0, pas d'arrêt)
- Seed : la seed qui peut être utilisée pour générer une même séquence aléatoire

Le zip fourni contient le projet Maven généré par Eclipse, ainsi que les deux JAR : un standalone (with dependencies) et un non.

## Étude des performances

Lors de mes tests, j'ai mesuré le temps moyen d'un tour d'animation en fonction du nombre de billes et j'ai observé une corrélation linéaire entre ces deux valeurs, comme le montre le graphique suivant de ce temps en fonction du nombre de billes :

