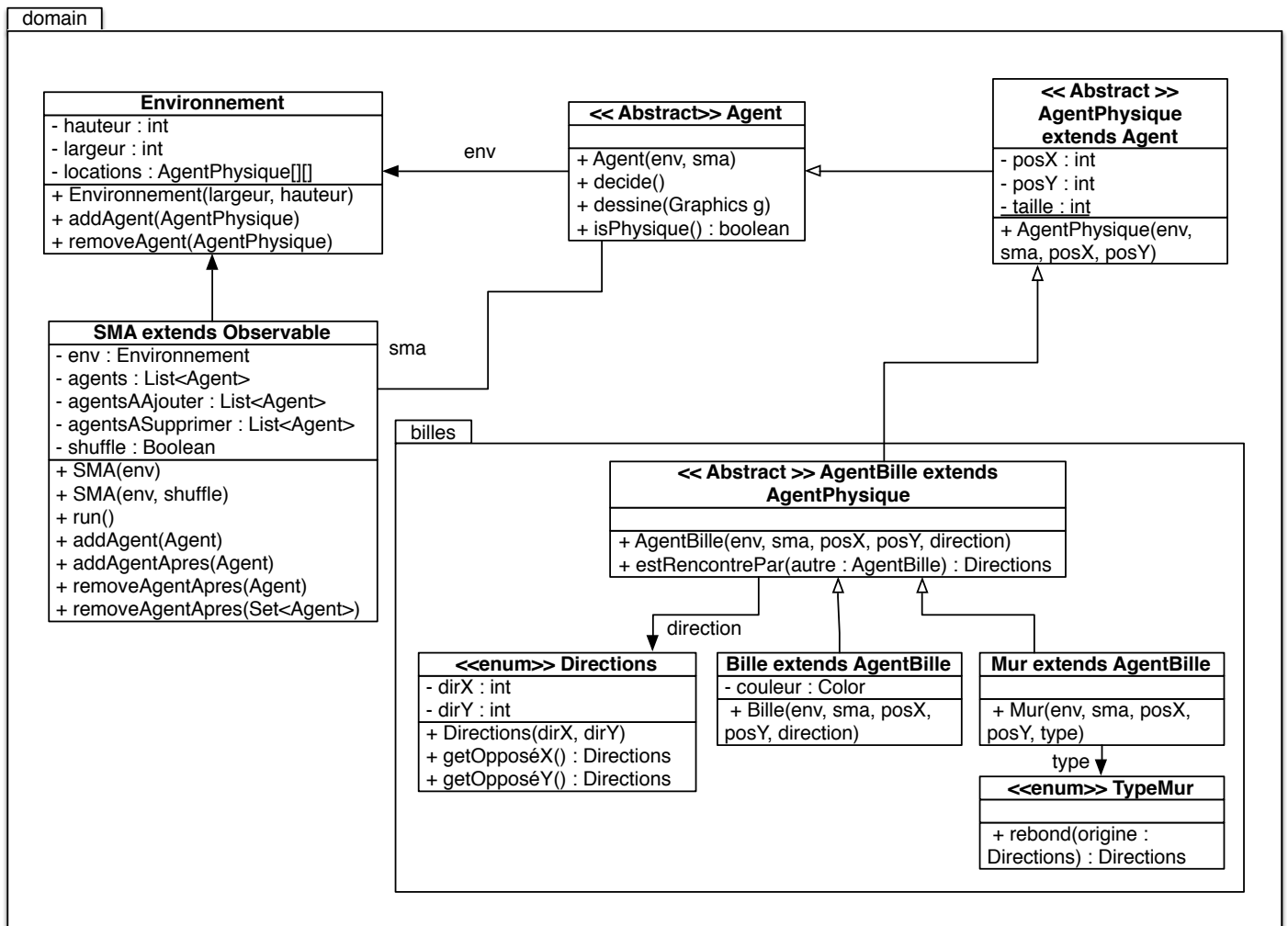


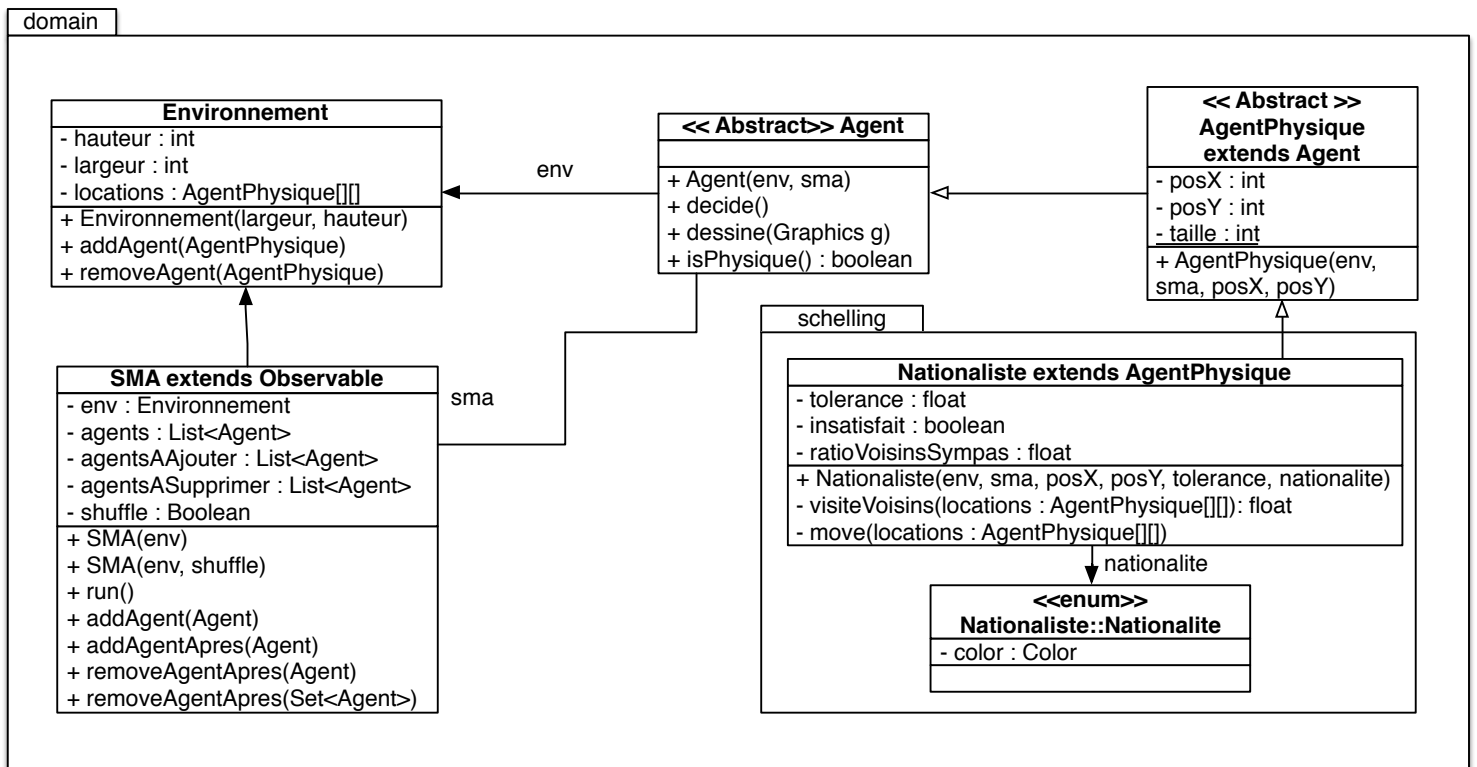
Comme on le voit, le moteur des simulations et du jeu est unique et commun.

Voici pour mémoire les diagramme spécifique aux trois premières réalisations :

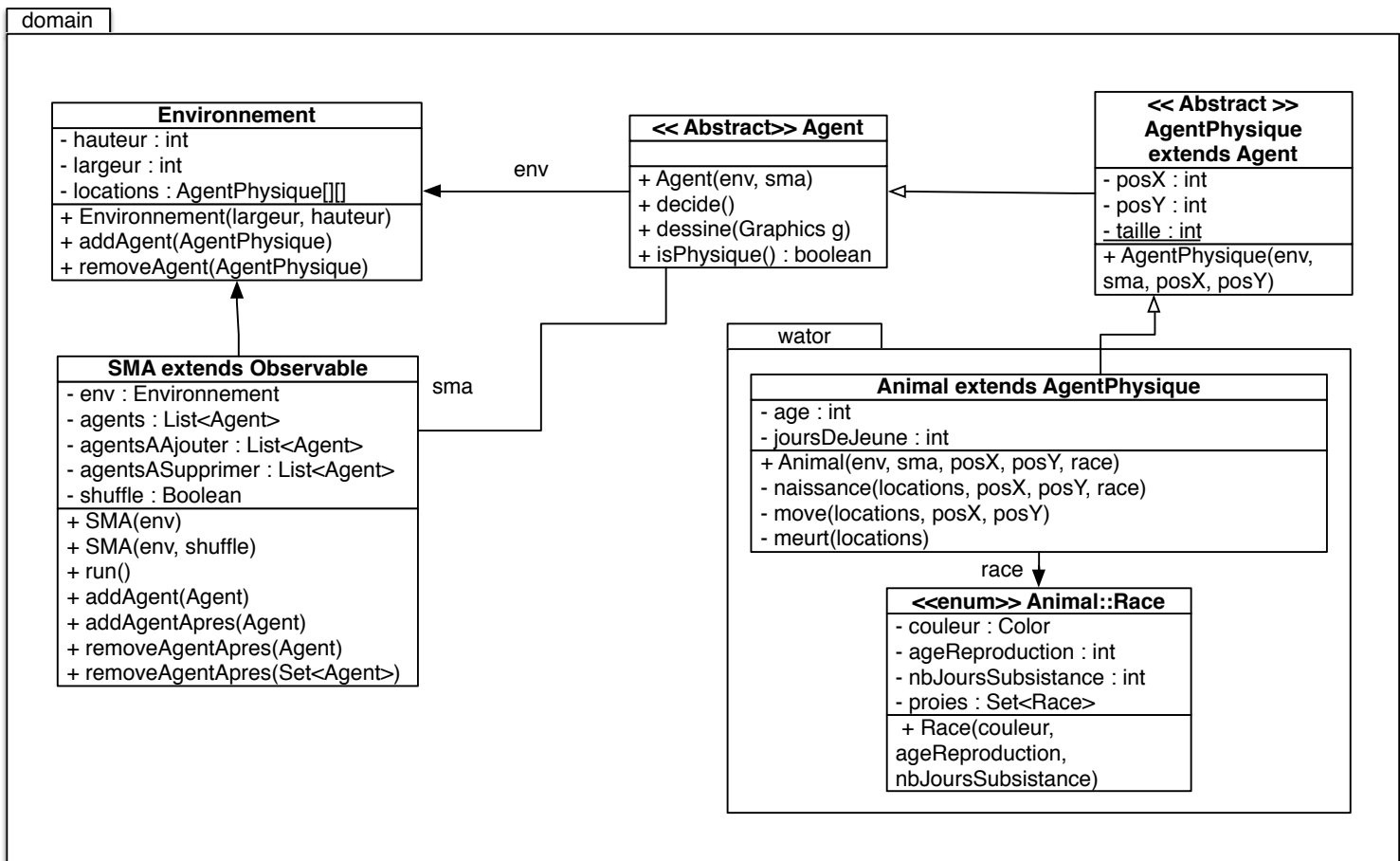
La boîte à particule :



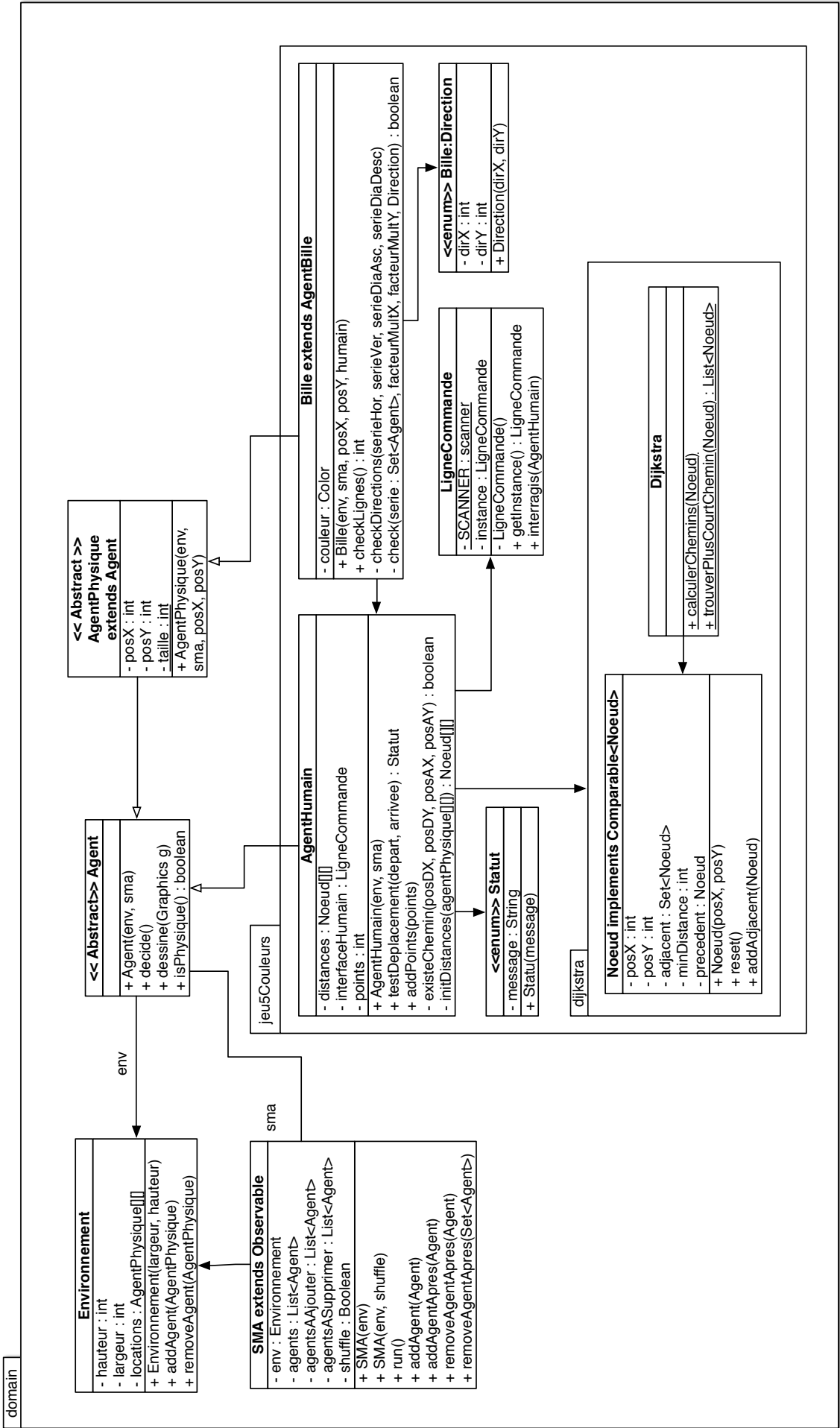
La simulation de Schelling :



La simulation de proie-prédateur :



Et voici le diagramme reprenant le jeu des 5 couleurs :



Deux types d'agents se partagent ici la parole, et il fut nécessaire de modifier le SMA très légèrement afin de les faire parler dans un ordre prédéfini (les bille, puis l'humain). J'ai donc ajouté l'option dans le SMA permettant de choisir s'il faut oui ou non mélanger les agents avant chaque tour.

Ensuite, il m'a fallu implémenter une interface en ligne de commande pour contrôler le jeu, laquelle est appelée par l'agent humain du système, demande à l'humain une entrée et la vérifie avant de passer la main à l'agent humain qui, après quelques tests supplémentaires, effectue l'action demandée.

Une des vérifications à effectuer impliquait d'implémenter l'algorithme de Dijkstra pour vérifier l'existence d'un chemin entre la position de bas et la position finale de la bille. Bien qu'il eut été possible d'implémenter une version très simplifiée de Dijkstra, par exemple en donnant un même numéro à toutes les cases de chaque "zone" dont les cases sont atteignables, mais l'exercice me semblait plus intéressant, et réutilisable, en implémentant une version normale, ne demandant pas plus de ressources, de l'algorithme.

Une autre vérification consistait à repérer les lignes formées par 5 billes ou plus. Cette vérification, par choix, est effectuée par chaque bille, lors de son tour de jeu, et contrôle dans les 4 directions, dans chaque sens en partant de la bille courante, si une série de 5 ou plus existe. Il se peut que plusieurs séries soient formées et donc repérées par une seule bille.

A chaque tour, avant de rendre la main, l'agent humain dispose 3 nouvelles billes sur le plateau, lesquelles peuvent créer des lignes et donc disparaître directement.

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 [command] [command options]
```

Commands:
help Displays the help about a specific command or about the software.

Usage: help [options]

Options:

-command

The command for which the help is requested.

schelling Lance une simulation avec des personnes nationalistes.

Usage: schelling [options]

Options:

-densite

Densité en pourcent.

Default: 70

-hauteur

Hauteur.

Default: 700

-largeur

Largeur.

```
        Default: 1200
    -seed
        Seed pour "reproduire un hasard".
        Default: 1381075253934
    -tempsArret
        Temps avant l'arrêt en secondes (0 pour l'infini).
        Default: 25
    -tempsAttente
        Temps d'attente entre deux tours en millisecondes.
        Default: 700
    -tolerance
        Tolérance des individus en pourcent.
        Default: 60

billes      Lance une simulation avec des billes.
Usage: billes [options]
Options:
    -hauteur
        Hauteur.
        Default: 700
    -largeur
        Largeur.
        Default: 1200
    -nbAgents
        Nombre d'agent.
        Default: 1000
    -seed
        Seed pour "reproduire un hasard".
        Default: 1381075253931
    -tempsArret
        Temps avant l'arrêt en secondes (0 pour l'infini).
        Default: 15
    -tempsAttente
        Temps d'attente entre deux tours en millisecondes.
        Default: 120
    -test
        Test.
        Default: false

wator      Lance une simulation avec des proies et des
prédateurs.
Usage: wator [options]
Options:
    -ageReproductionPoissons
        Age de reproduction des poisson.
        Default: 4
    -ageReproductionRequins
        Age de reproduction des requins.
        Default: 7
    -hauteur
        Hauteur.
        Default: 700
    -largeur
        Largeur.
        Default: 700
    -nbJoursSubsistanceRequins
        Nombre de jours de subsistance des requins.
        Default: 3
    -nbPoissons
        Nombre de poissons.
        Default: 300
    -nbRequins
        Nombre de requins.
```

```
    Default: 150
  -seed
    Seed pour "reproduire un hasard".
    Default: 1381075253936
  -tempsArret
    Temps avant l'arrêt en secondes (0 pour l'infini).
    Default: 270
  -tempsAttente
    Temps d'attente entre deux tours en millisecondes.
    Default: 120

5couleurs      Lance une simulation avec des billes.
Usage: 5couleurs [options]
Options:
  -dimension
    Dimension de l'environnement (carré).
    Default: 400
  -nbBilles
    Nombre de billes de départ.
    Default: 5
  -seed
    Seed pour "reproduire un hasard".
    Default: 1381075253938
```

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

Pour jouer, il suffit d'introduire les numéros de lignes et de colonnes de départ et d'arrivée comme indiqué par le jeu.