

SCI – Simulations Schelling et Wa-Tor

Description du projet

Dans le cadre du cours de SCI, il nous a été demandé, sur base du moteur de la "boîte à particule" développé précédemment, d'implémenter des simulations des phénomènes de Schelling (la ségrégation) et de Wa-Tor (proies-prédateurs). Pour ce faire, la première étape fut la refonte de l'application développée afin d'en augmenter la modularité et l'extensibilité.

Réalisation et choix de conception

Le projet a été réalisé en Java, utilise Swing pour l'interface, et produit le cas échéant des fichiers de log au format CSV.

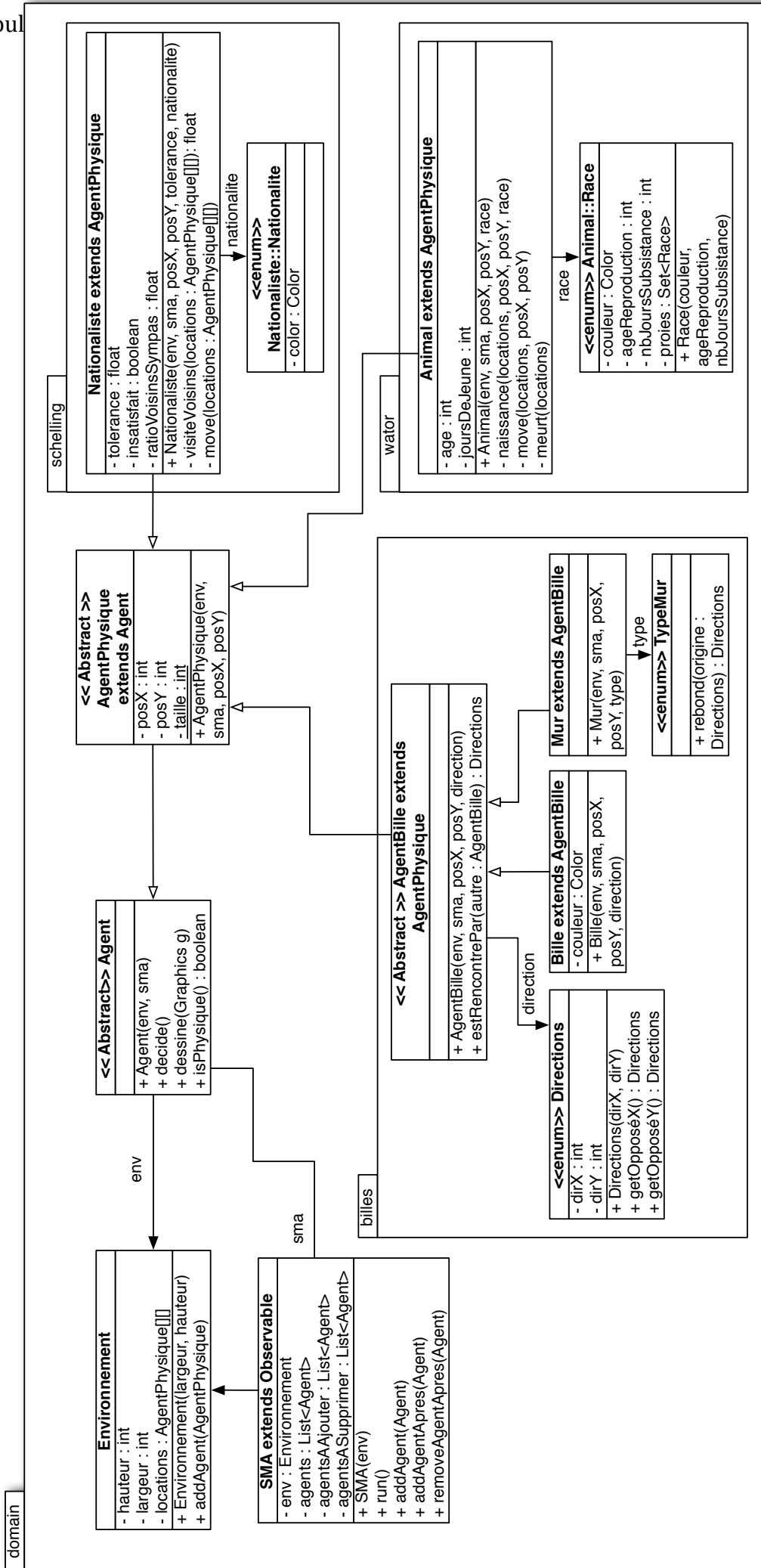
Voici à la page suivante le diagramme de classe du package *domain* contenant le moteur et les différents packages des différents modèles de simulation.

J'ai choisi de travailler avec de l'héritage pour ajouter des spécificités à chaque niveau de mon arbre d'héritage, ainsi, chaque élément dessiné est un agent, mais je laisse la porte ouverte à une extension pour laquelle il y aurait des agents non dessinés (non physiques). Au sein de chaque package de modèle, se retrouve une classe générique (sous-classée à son tour ou non) qui ajoute les méthodes et attributs spécifiques au modèle donné et donne le comportement des agents.

Cette structure offre une grande souplesse car les classe communes restent très générique et on descend la spécificité le plus proche possible de l'agent final, de ce fait, chaque comportement peut être précisément implémenté sans affecter les modèles et les classes non concernées. Une piste d'amélioration serait de reprendre la logique utilisée pour Schelling et Wa-Tor pour unifier les classes du package billes.

La bibliothèque JCommand a été utilisée pour aider à la modularité et à la facilité de mise en place des menus des commandes de lancement. Celle-ci s'appuie sur le pattern command et permet une grande modularité pour lancer des applications très différentes telles que les modèles différents utilisés ici, et ce en gérant une liste de paramètre unique pour chacun d'entre eux. Ainsi chaque mode de simulation possède sa classe `Simulation***Command` qui en définit les arguments et lance l'application en utilisant le modèle demandé.

Ces différentes classes de simulations utilisent des analyseurs, situés dans le package éponyme pour récupérer et agréger les données à sortir et les envoyer dans un fichier CSV.



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: 1380057889804

-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: 1380057889802
  -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

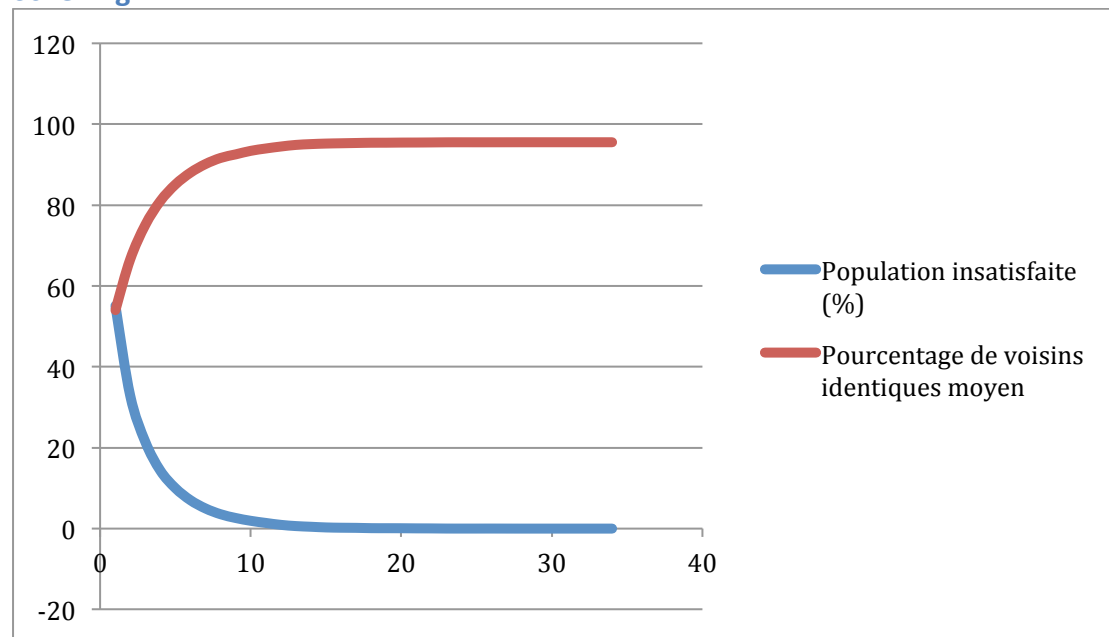
wator      Lance une simulation avec des proies et des
           prédateurs.
Usage: wator [options]
Options:
  -ageReproductionPoissons
    Age de reproduction des poisson.
    Default: 4
  -ageReproductionRequins
    Age de reproduction des requins.
    Default: 7
  -hauteur
    Hauteur.
    Default: 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: 1380057889806
  -tempsArret
    Temps avant l'arrêt en secondes (0 pour l'infini).
    Default: 60
  -tempsAttente
    Temps d'attente entre deux tours en millisecondes.
    Default: 120
```

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

Étude des résultats

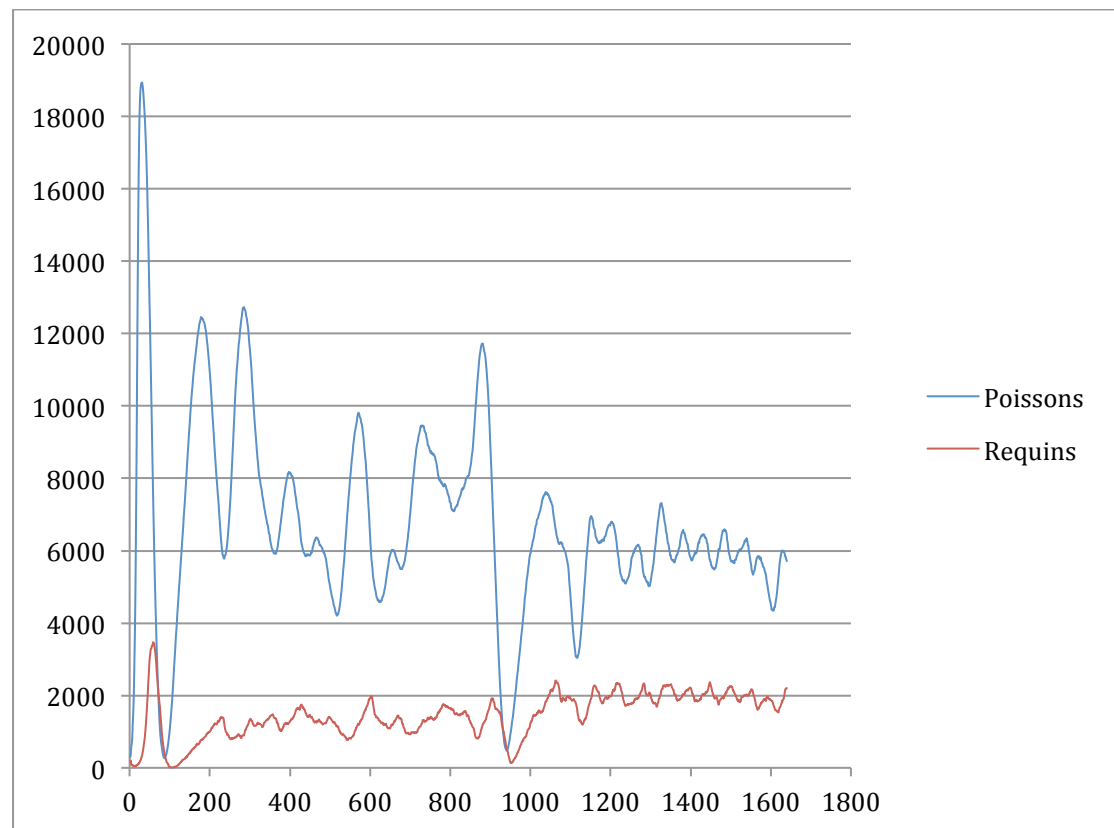
Lors de mes tests, j'ai extrait les données de la simulation à chaque étape afin de pouvoir tirer une analyse des résultats. Voici les graphiques synthétisant ces résultats.

Schelling

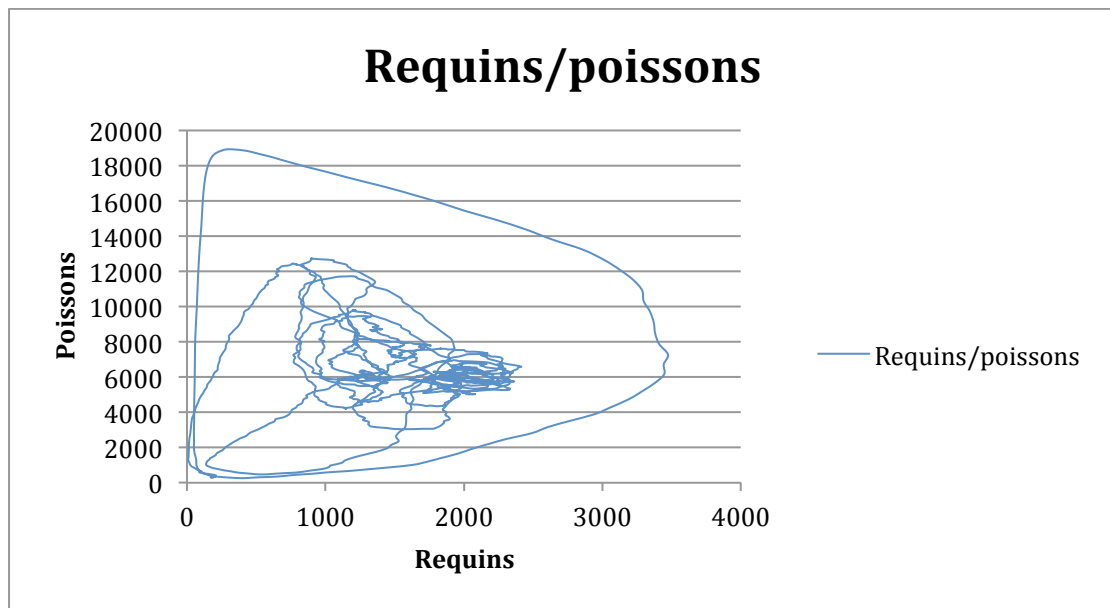


Comme on le constate sur ce graphique la population nationaliste tend à atteindre un état de stabilité dans lequel chacun est satisfait, le taux d'insatisfaction tend alors vers 0 et le pourcentage de voisin de même nationalité plafonne à une valeur dépendant de la tolérance des individus. Celle-ci était placée à 60% de voisins de même nationalité, dans le cas de ce graphique. On voit bien le phénomène de ghetto-isation et de ségrégation apparaître. Comme nous l'avons observé en cours, le pourcentage moyen de voisins de même nationalité excède de loin celui requis par la tolérance (95% contre 60%).

Wa-Tor



Ce premier graphique montre l'évolution de la population de requins et de poissons à chaque tour. On peut constater que les deux populations évoluent en parallèle, celles des requins restant moins nombreuse et légèrement en retard sur celle des poissons : le nombre de poissons augmente, cela entraine une augmentation du nombre de requins qui peuvent se nourrir mieux, ce qui entraine une diminution du nombre de poissons, mangés par les requins, ce qui entraine une diminution du nombre de requins, faute de nourriture, ce qui entraine une augmentation du nombre de poissons car ils redeviennent peu chassés, et la boucle est bouclée.



Ce second graphique montre les évolutions relatives des deux populations. On y retrouve globalement l'ellipse prévue par la théorie.