

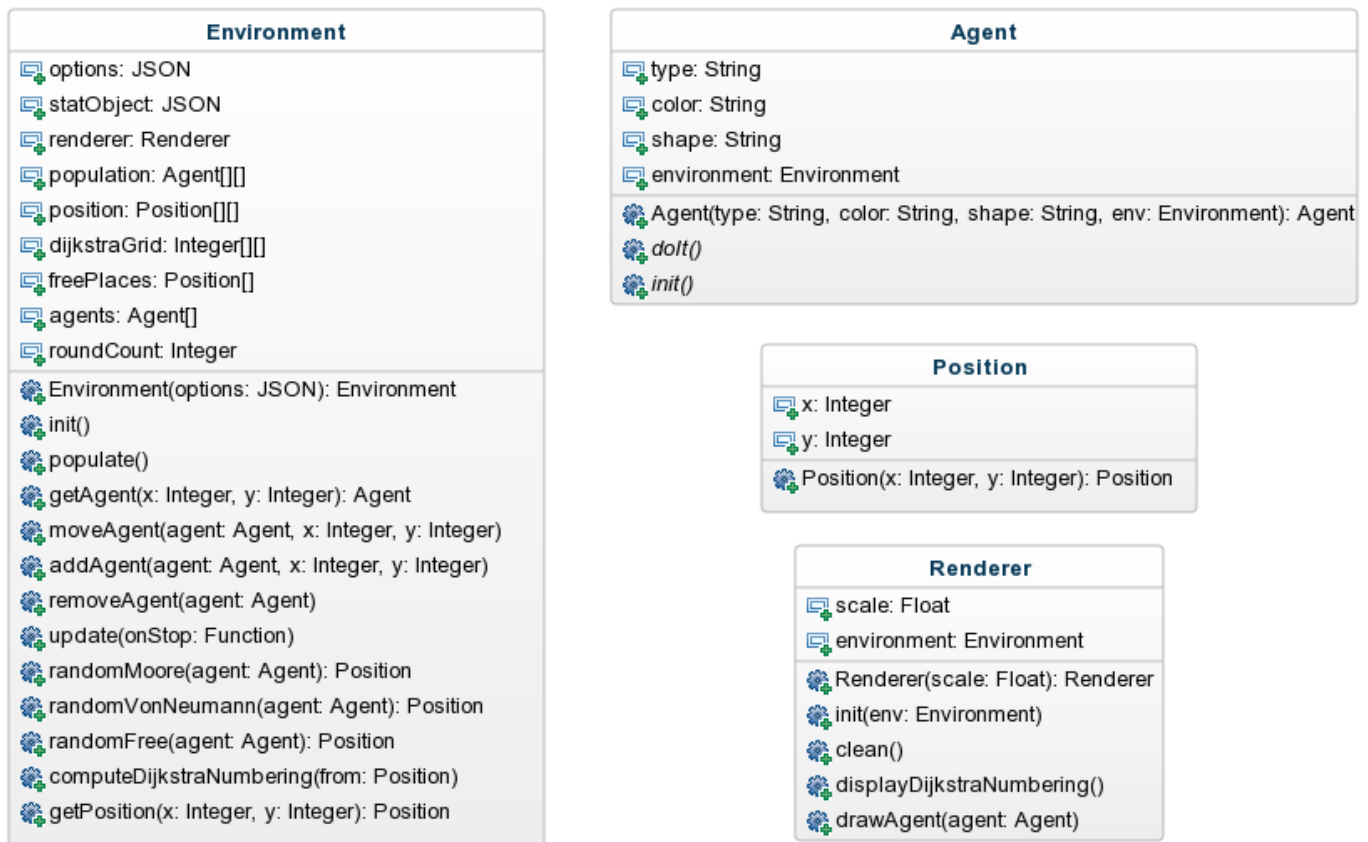
Systèmes Multi-Agent

Général

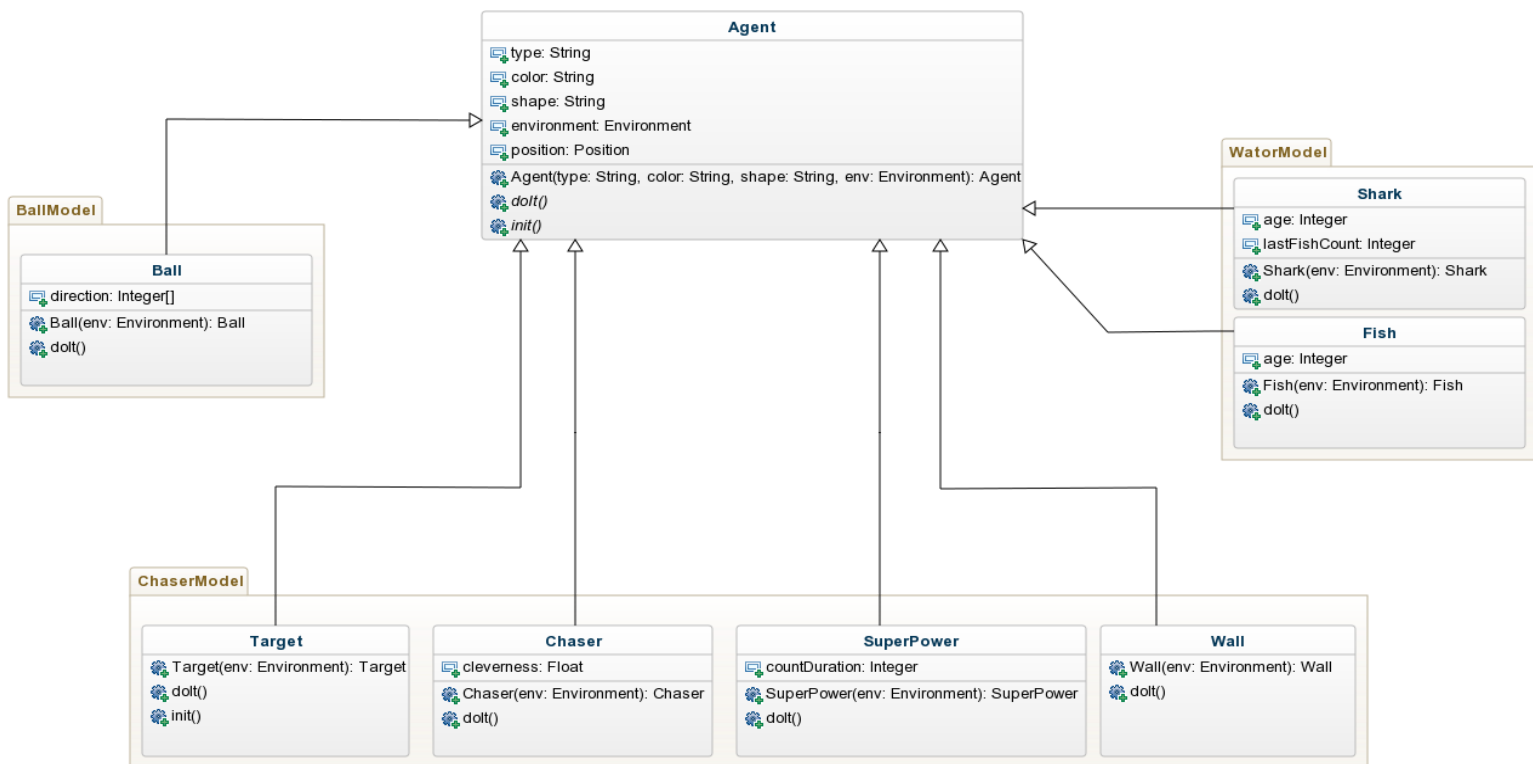
Technologie

Le langage de programme utilisé est Javascript. Ce choix est dirigé par la simplicité de développement et il n'a pas besoin d'être compilé. De plus, il permet de mettre en place à l'aide de HTML des interfaces utilisateurs conviviales et simple d'utilisation. Par contre, l'utilisation de Javascript ne permet pas une forte montée en charge, et par corrélation de gérer un grand nombre d'agents. Cependant les rendus restent fluides, et les convergences désirées restent observables.

Conception



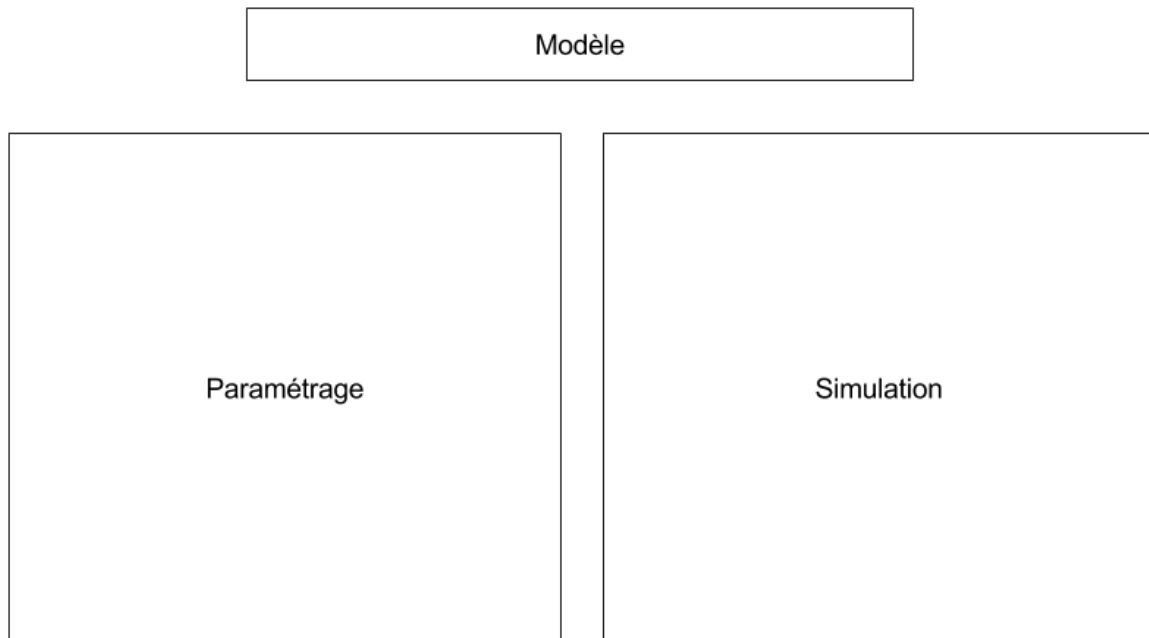
Le coeur de l'application est représenté par la classe *Environment*. Elle permet de gérer les différents agents dans le temps. L'Environment contient également un objet *Renderer*, permettant de dessiner les agents à l'utilisateur. Une classe abstraites *Agent* est définie, chaque type d'agent devra l'étendre.



Chaque simulation présentée est contenue dans un package. Tous les agents étendent la classe abstraite *Agent*, et définissent leurs comportement dans la méthode *dolt*. Une méthode *init* peut également être implémentée dans les agents.

Interface utilisateur

Pour chacun des modèles présentés, une page HTML lui est dédiée. Cette page permet la configuration de l'environnement avant de lancer une simulation. Ces interfaces sont intuitives et permette au plus novice de démarrer une simulation. Certains éléments de configuration de l'environnement sont dynamique, et permette de modifier "à chaud" l'environnement en cours d'exécution. D'autre sont "statique", et prendront effet uniquement lors de la prochaine exécution.



Billes (index_bille.html)

Paramétrage

Paramètre	Définition	Type
Delay	Le temps entre deux actions	Dynamique
Balls	Le nombre de bille	Statique
Size	La largeur / longueur de l'environnement	Statique
Ball Shape	La forme d'une bille	Statique
TorWorld	L'environnement est-il torique ?	Statique
Display grid	Afficher les cases	Statique
Fairness	Appeler <i>dolt</i> dans le même ordre	Statique
Seed	Utiliser une seed pour reproduire une simulation	Statique

Wator (index_wator.html)

Paramétrage

Paramètre	Définition	Type
Delay	Le temps entre deux actions	Dynamique
Agents	Le nombre d'agent total	Statique
Fishes	Le pourcentage de poisson	Statique
Sharks	Le pourcentage de requin	Statique
Fish breed	Le nombre de tour avant qu'un poisson ne se reproduise	Statique
Shark breed	Le nombre de tour avant d'un requin ne se reproduise	Statique
Starving time	Le le nombre de tour sans manger pour un requin avant de mourir	Statique
Size	La largeur / longueur de l'environnement	Statique
TorWorld	L'environnement est-il torique ?	Statique
Display grid	Afficher les cases	Dynamique
Fairness	Appeler <i>do/it</i> dans le même ordres	Statique
Random Seed	Utiliser une seed pour reproduire une simulation	Statique
Graphics	Afficher les graphiques	Dynamique

Chaser (index_chaser.html)

Paramétrage

Paramètre	Définition	Type
Delay	Le temps entre deux actions	Dynamique
Chaser	Le nombre de poursuivant	Statique
Wall	Le nombre de mur	Statique
Super Power	Le nombre case permettant d'inverser les rôles poursuivant/poursuiveur	Statique
Super Power Count	Le nombre de count lors de l'inversement des rôles	Statique
IA Level	L'intelligence d'un poursuiveur lorsqu'il fuit	Statique
Chaser handicap	Ralentir les poursuiveurs en nombre de tour	Statique
Size	La largeur / longueur de l'environnement	Statique
Dijkstra numbering	Affichage des distances de dijkstra par rapport à la cible	Statique
TorWorld	L'environnement est-il torique ?	Statique
Display grid	Afficher les cases	Dynamique
Fairness	Appeler <i>doIt</i> dans le même ordres	Statique
Random Seed	Utiliser une seed pour reproduire une simulation	Statique