

Vie Artificielle

Agents, environnement

Dernière mise à jour: février 2019

Mots clés: *Modélisation individu-centrée, interactions agents/environnement.*

Dans le cadre de ce TP, on s'intéresse à la modélisation individu-centrée. Il s'agit de modéliser des "agents" pouvant se déplacer dans et influençant leur environnement. Par exemple, on peut souhaiter modéliser le comportement d'une fourmi, ou encore le rapport complexe entre proies et prédateurs. Téléchargez l'archive disponible sur le site, vous y trouverez un point de départ pour programmer des agents.

La fourmi de Langton

[sur ordinateur]

Pour commencer, récupérez, installez et exécutez le code source disponible sur le site Web. Vous devrez lancer la classe *MyEcosystem_ants* pour exécuter le programme. En l'état, il s'agit d'un seul fourmi qui se promène au hasard en laissant une trace multi-couleur (le comportement est visible dans *LangtonAnt.java*).

Modifier le code de *LangtonAnt.java* pour programmer un comportement d'une fourmi de Langton tel que:

- l'environnement connaît deux états: gris ou blanc
- l'agent se déplace tel que:
 - si la case courante est grise, tourne à 90° vers la droite, avance
 - si la case courante est blanche, tourne à 90° vers la gauche, avance
 - dans les deux cas, inverse l'état de la case avant de bouger

Faites tourner cette simulation pendant 11000 pas au moins.

Astuce: vous pouvez changer la vitesse d'affichage (variable `delai`)

Proies-prédateurs

[sur ordinateur]

Vous devrez cette fois lancer la classe *MyEcosystem_predprey* pour exécuter le programme. En l'état, il s'agit de deux populations d'agents qui bougent au hasard dans l'environnement en laissant une trace qui dépend de leur type. Avant de commencer, il est important que vous passiez un peu de temps pour étudier le fonctionnement de ces classes. Toutes ne sont pas utiles, et il y a de nombreux commentaires pour vous aider à comprendre.

Question 1.

En plus des deux groupes d'agents, on souhaite maintenant obtenir le résultat suivant:

- des agents "proies" (des moutons) qui bougent au hasard
- des agents "prédateurs" (des loups) qui bougent au hasard
- les prédateurs "mangent" les proies s'ils se trouvent sur la même case. Une proie mangée disparaît.

Il s'agit dans un premier temps d'implémenter l'action de cette règle lorsqu'un agent «prédateur» et un agent «proie» se trouvent sur la même case.

Implémentez ce modèle. Vous devrez fixer la taille des populations.

Question 2.

En utilisant le script de génération de graphe vu lors du dernier TME, vous devrez créer un graphe montrant l'évolution du nombre de proies (abscisses) au cours du temps (ordonnées). Générez 4 graphes pour 4 exécutions différentes.

Question 3.

On souhaite ensuite introduire la notion de reproduction:

- toutes les N itérations, une proie se "reproduit". C'est à dire qu'une nouvelle proie apparaît au même endroit.
- toutes les M itérations, un prédateur se reproduit.
- un prédateur disparaît ("meurt") si il n'a pas "mangé" de proie pendant les dernières L itérations. $L < M$.

Implémentez ces options. Réglez les paramètres N, M et L tel qu'on observe pas d'extinctions totales de proies et/ou de prédateurs. Qu'observe t'on sur la variation de la taille des populations au court du temps?

Question 4.

En utilisant le script de génération de graphe, vous devrez créer deux graphes montrant (1) l'évolution du nombre de proies au cours du temps et (2) l'évolution du nombre de prédateurs. Générez 2x4 graphes pour 4 exécutions différentes, en utilisant un choix de paramètres (cf. question 3) ou on observe pas d'extinctions.

Indice: on cherche à mettre en évidence l'oscillation du nombre de proies et de prédateurs au cours du temps.

Question 5.

On souhaite maintenant étendre les comportements des proies et des prédateurs:

- les proies se déplacent au hasard, sauf si un prédateur est proche (voisinage de Von Neumann). Dans ce cas, une proie "fuit" dans la direction opposée à celle du prédateur.
- les prédateurs se déplacent au hasard, sauf si une proie est proche. Dans ce cas, le prédateur "poursuit" la proie.

Implémentez ce nouveau modèle.

Question 6.

Enfin, on souhaite introduire la notion de nourriture dans l'environnement. Pour cela on va considérer que chaque case peut ou non contenir de l'herbe. L'herbe repousse avec une probabilité P_{herbe} et l'herbe est mangé par les proies.

On considère qu'un mouton meurt s'il n'a pas mangé d'herbe après O itérations. $O < N$.

Implémentez l'herbe. réglez L,M,N,O afin qu'il n'y ait pas d'extinction.

On souhaite maintenant intégrer un environnement. A cette fin, on reprendra les états de l'automate cellulaire feu de forêt. Programmez un environnement dans lequel on trouve des cases occupées par un arbre ou non. Dans cet environnement, vous commencerez par ajouter une population de fourmi de Langton, puis d'agents proies et prédateurs.

Une fois cette intégration faite, vous serez prêt à travailler sur votre projet.