

# Vie Artificielle

## Agents et comportements

Dernière mise à jour: janvier 2026

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

Les ressources nécessaires pour cette UE (lien vers les sources, les sujets, etc.) sont disponibles sur Moodle. Pour chaque question, conservez le résultat (code et/ou graphe) afin de pouvoir le montrer.

#### Evaluation 2021-2022

Ce travail est à faire en **binôme**. Vous devez conserver un fichier par question afin de pouvoir le montrer rapidement. Pour les questions 2, 3 et 4, le répertoire devra contenir un (ou deux) graphe(s) au format PNG. Pour les questions 1, 3, 5, 6 le répertoire devra contenir un code fonctionnel.

*La partie "premiers pas vers le projet" est en bonus*

#### Proies-prédateurs

[sur ordinateur]

Exécuter le programme. En l'état, il s'agit de deux populations d'agents qui se déplace au hasard dans l'environnement en laissant une trace, dont la couleur dépend du type de l'agent. 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.

#### QUESTION 1.

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

- des agents "proies" (des moutons) qui se déplacent au hasard
- des agents "prédateurs" (des loups) qui se déplacent 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.

Quelques pistes:

- observez la liste contenant les agents, les comportements des agents sont définis dans Predator et Prey.
- Vous pouvez tout à fait créer deux listes selon le type d'agents. Cependant, il est plus facile de gérer la "mort" d'une proie en ne l'affichant pas (cf. la propriété "running"). Cela permet de ne pas payer le cout d'une déletion dans la liste. Si vous décidez néanmoins de supprimer un élément de la liste, n'oubliez pas que la taille de la liste va changer (ce qui peut poser problème si vous êtes en train de la parcourir).

#### QUESTION 2.

En utilisant le script de génération de graphes vu lors du précédent TME (ou un tableau si vous préférez), vous devrez créer un graphe montrant l'évolution du nombre de proies (ordonnées) au cours du temps (abscisses). Vous devrez montrer un des graphes que vous aurez générés.

#### QUESTION 3.

On souhaite ensuite introduire la notion de reproduction:

- A chaque itération, une proie peut se "reproduire" avec une probabilité  $P_{reproduction\_proie}$ . C'est à dire qu'une nouvelle proie apparaît au même endroit. Cette probabilité est généralement très faible.
- A chaque itération, un prédateur peut se "reproduire" avec une probabilité  $P_{reproduction\_prédateur}$ .
- un prédateur disparaît ("meurt") si il n'a pas "mangé" de proie pendant les dernières  $R_{famine}$  itérations (délai de famine).

Implémentez ces options. Réglez les paramètres  $P_{reproduction\_proie}$ ,  $P_{reproduction\_prédateur}$  et  $R_{famine}$  tel qu'on observe pas d'extinctions totales de proies et/ou de prédateurs, ou *a minima* que cette extinction arrive le plus tard possible, sans pour autant saturer le système (ie. pas de croissance exponentielle). On souhaite pouvoir observer que les nombres de proies et de prédateurs oscillent au cours du temps.

#### QUESTION 4.

En utilisant le script de génération de graphe (ou un tableau), 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 ces deux graphes en utilisant un choix de paramètres (cf. question 3) ou l'on n'observe pas (ou peu) d'extinctions. On s'attend à ce que la taille de la population de chaque type d'individu oscille au cours du temps. Vous aurez probablement besoin de plusieurs essais.

#### QUESTION 5.

On souhaite maintenant implémenter 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. Le prédateur "poursuit" alors la proie.
- Pour donner une chance aux prédateurs, introduisez une probabilité de déplacement tel que les prédateurs ont plus de chance de bouger que les proies. P.ex.  $P_{mouvement\_prédateur} = 1.0$  et  $P_{mouvement\_proie} = 0.75$

#### 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 mangée 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 les différents paramètres (probabilité et délai de famine) afin qu'il n'y ait pas d'extinction.

#### Ecologie et agents: un modèle simple

[sur ordinateur]

#### PREMIER PAS VERS LE PROJET

On souhaite maintenant intégrer un environnement. A cette fin, on reprendra les états de l'automate cellulaire feu de forêt que l'on a vu dans la séance précédente. Programmez un environnement dans lequel on trouve des cases occupées par un arbre ou non. Dans cet environnement, vous commencerez par ajouter une population d'agents proies et d'agents prédateurs. *Une fois cette intégration faite, vous aurez une version minimale à partir de laquelle vous pouvez commencer le projet.*