

IN200: simulation d'un système proie-prédateur

Mars 2022

Objectif général

Dans un plan en deux dimensions, vous allez simuler la cohabitation de deux espèces : des proies et leurs prédateurs (vous pouvez imaginer des lapins et des renards, par exemple). En suivant des règles de naissance, de déplacement, de reproduction, de durée de vie et de chasse (pour les prédateurs) données, les populations de ces deux espèces vont évoluer de façon cyclique et coordonnée.

Fonctionnement général

Sur un plan de dimension 30x30 (paramétrable) se trouvent des proies et des prédateurs. À chaque **tour de jeu**, vous allez appliquer des **règles de vie** à chacun des individus du plan. Au terme d'un tour, les individus auront donc évolué : ils se seront déplacés, seront morts de vieillesse ou de faim, ou auront été mangés (pour les proies).

Un tour de jeu consiste donc à aller visiter chacun des individus afin de lui appliquer les règles de vie propres à son espèce, proie ou prédateur.

La suite du document détaille en 4 étapes le sujet. Le choix de l'interface graphique est laissé libre. Votre interface doit permettre au minimum de démarrer une simulation, de l'interrompre puis de la redémarrer, de sauvegarder son état dans un fichier et de la recharger.

Étape 1 : au paradis des proies

La première étape de ce projet consiste à faire évoluer des proies seules sur le plan. Cette étape est importante pour votre compréhension du projet. Elle va vous permettre de mettre en place les structures principales et l'organisation générale du projet. Toutefois, le résultat obtenu est connu d'avance : en l'absence de prédateurs et de reproduction des proies, une population stable de proies évoluera au gré des naissances et des morts.

Voici les règles de vie à appliquer à chaque tour de jeu :

- **proies initiales** : au commencement de la simulation, un nombre N_{pro} de proies est positionné aléatoirement sur le terrain.
- **naissance des proies** : les proies naissent à des positions aléatoires à une fréquence F_{pro} donnée. (On veillera bien entendu à ne pas faire naître de proie sur une position occupée.) Exemple : si $F_{pro} = 3$, cela indique que 3 proies naissent à chaque tour de jeu.
- **âge des proies** : à la naissance, chaque proie a une durée de vie A_{pro} donnée, exprimée en nombre de tours. À chaque tour, chacune des proies perd une unité de durée de vie, et meurt si elle atteint 0. Exemple : si $A_{pro} = 5$, les proies nées au commencement du jeu mourront au 5ème tour.
- **déplacement** : à chaque tour, chacune des proies se déplace d'une case dans une des 8 directions possibles de son voisinage. Bien entendu, certaines positions du voisinage d'une proie peuvent être inaccessibles, si la case est occupée par une autre proie ou si le bord du plateau est atteint. En l'absence de possibilité, une proie peut rester sur sa position.

Étape 2 : pléthore de proies

À cette étape, vous allez simplement ajouter une capacité de reproduction aux proies, qui vont alors se multiplier. Le résultat obtenu est également facile à anticiper : la multiplication des proies, non encore soumises à la pression de prédateurs, va entraîner un envahissement du terrain.

Dans la réalité, ces situations surviennent lorsqu'une espèce n'est plus confrontée à aucun prédateur et dispose de suffisamment de ressources alimentaires. Il s'ensuit nécessairement une explosion de la population et une famine pour cause de destruction des ressources.

La nouvelle règle que nous introduisons à cette étape et qui remplace la règle de naissance des proies est la suivante :

- **reproduction** : lorsque, au gré des hasards des naissances et des déplacements aléatoires des proies, deux proies se retrouvent côte à côte, elles donnent naissance à une nouvelle proie qu'on placera à une position aléatoire autour des deux proies concernées.

Étape 3 : introduction des prédateurs

Les lapins ont assez gambadé, il est temps d'introduire des prédateurs dans cette histoire. Nous allons considérer que ces prédateurs n'ont pas de prédateurs, mais qu'ils ont besoin de se nourrir pour survivre. On introduira un certain nombre de prédateurs sur le terrain au commencement du jeu, qui devront donc chasser pour survivre.

Voici les nouvelles règles que nous introduisons :

- **prédateurs initiaux** : au commencement de la simulation, un nombre N_{pre} de prédateurs est positionné aléatoirement sur le terrain.
- **âge des prédateurs** : à la naissance, chaque prédateur a une durée de vie A_{pre} qui, comme pour les proies, diminue d'une unité à chaque tour. Quand cet âge atteint zéro, le prédateur meurt. Afin de se rapprocher de la réalité, on choisira $A_{pre} \geq A_{pro}$.
- **énergie des prédateurs** : à la naissance, chaque prédateur dispose d'une énergie E_{pre} , avec $E_{pre} < A_{pre}$, qui diminue de 1 à chaque tour. Si elle atteint zéro le prédateur meurt de faim. Pour remonter son niveau d'énergie, un prédateur doit manger une proie. On introduira également un paramètre $MIAM$ qui indique le niveau d'énergie gagné lorsqu'un prédateur mange une proie. Exemple : un prédateur disposant d'un $E_{pre} = 12$ augmente cet E_{pre} de $MIAM = 5$ s'il mange une proie.
- **reproduction des prédateurs** : les prédateurs ne peuvent se reproduire que s'ils disposent d'un niveau d'énergie E_{repro} , qu'on fixera supérieur à E_{pre} . Ils doivent donc manger suffisamment pour atteindre ce niveau de reproduction. Une fois ce niveau atteint, on fera apparaître aléatoirement un nouveau prédateur sur le terrain.
- **chasse des prédateurs** : un prédateur mange une proie quand il se retrouve sur la même case qu'elle. Dans un premier temps, les prédateurs se déplacent de manière aléatoire.

Étape 4 : perfectionnement de l'algorithme de chasse

Afin de mieux coller à la réalité, on pourra introduire un paramètre $FLAIR$ qui indique la distance maximale à laquelle un prédateur peut sentir une proie. Au-delà de cette distance, la proie est invisible pour le prédateur. S'il sent une proie, le prédateur se déplace d'une case vers la proie la plus proche, sinon il se déplace au hasard.

De la même manière, vous pouvez programmer une stratégie de fuite pour les proies.