

Ecosystème

Travaux Pratiques 07 du module 05 - Les variables complexes

Avant de démarrer ce TP, il convient d'avoir suivi les vidéos des modules 1 à 6 de ce cours.

Durée estimée

Environ 2 heures

Énoncé

Objectif

Dans ce TP, nous allons plonger au cœur d'un écosystème. Nous allons créer une simulation d'un écosystème en utilisant un modèle simple basé sur une chaîne alimentaire composée de trois animaux. L'objectif de ce TP est de simuler un écosystème en partant sur un schéma simplifié de chaîne alimentaire de trois animaux.

Introduction

Imaginons un écosystème avec trois animaux numérotés de 1 à 3. La chaîne alimentaire est simple : l'animal 1 se nourrit de l'animal 2, l'animal 2 se nourrit de l'animal 3 et l'animal 3 se nourrit de l'animal 1. Le schéma ainsi respecté est le même que pierre – feuille – ciseaux. Le TP se déroulera comme suit.

- Nous allons commencer par créer un micro-écosystème en plaçant neuf animaux dans un carré 3x3. L'animal situé au centre sera mangé et remplacé par un prédateur s'il est entouré d'une majorité de prédateur.
- Nous créerons ensuite un macro-écosystème en plaçant des animaux dans un carré NxN. Nous regarderons ensuite chaque animal pour savoir s'il est entouré de prédateurs et s'il est mangé et remplacé.
- Nous observerons comment l'écosystème évolue au fil du temps.

Niveau 1 – To eat or not to eat ? - Difficulté ++

La première étape est de créer le plus petit écosystème qui soit. Nous allons répartir les animaux dans un carré 3x3 en respectant le schéma ci-dessous.

1	3	2
1	2	1
3	2	1

La case centrale représente la proie et devra être modifiée en fonction du nombre de prédateurs autour.

1. S'il y a plus de prédateurs alors la proie sera remplacée.
2. Si au contraire ils sont en minorité alors il n'y a pas de changement.
3. S'il y a en majorité à la fois des prédateurs et des proies alors il y a une chance sur deux que la proie soit remplacée par le prédateur.
4. S'il y a en majorité à la fois les prédateurs et des animaux neutres, alors la proie est remplacée par le prédateur.

La vérification se fera en console pour le moment en affichant la première liste puis la deuxième pour voir si la proie a bien été modifiée en fonction des conditions.

Exemple :

```

----- Avant -----
      1 2 3
      1 2 3
      2 3 1
----- Après -----
      1 2 3
      1 1 3
      2 3 1
-----

```

Niveau 2 – La loi de la jungle - Difficulté +++

Il va falloir maintenant créer une méthode pour mettre à jour un macro écosystème simulé par une grille de taille $N \times N$.

Imaginez une grille multi-dimensionnelle de taille $N \times N$ représentée par une liste de taille N contenant des listes de taille N . Chaque case de cette grille contient un entier compris entre 1 et 3, représentant un animal.

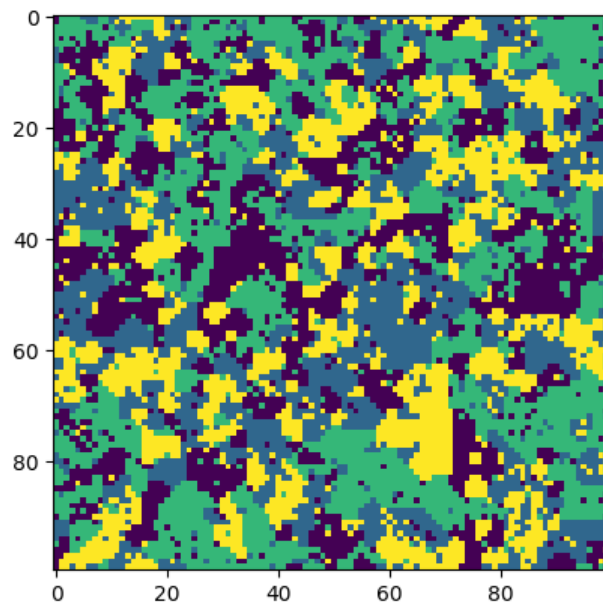
Dans ce niveau 2, notre objectif est de mettre à jour chaque animal en fonction de ses voisins. Autrement dit, nous allons appliquer l'algorithme du niveau 1 à toutes les cases de cette grille du niveau 2.

Niveau 3 – Affichage dynamique - Difficulté +

Pour observer l'évolution de l'écosystème dans le temps, nous allons utiliser la fonction d'affichage fournie dans les ressources.

Nous mettrons en place une boucle qui effectuera un certain nombre de pas pour afficher en temps réel les changements de l'écosystème sur l'écran. Vous pourrez ainsi observer comment l'écosystème évolue au fur et à mesure du temps.

Le résultat attendu de cette simulation est représenté ci-dessous, vous pourrez ainsi visualiser les changements qui se produisent dans l'écosystème au fil du temps.



Niveau 4 – Have fun

Augmentez le nombre d'espèces animales, modifiez la taille de l'écosystème, changez les conditions de prédation ou le nombre de pas. Amusez-vous à voir quelles sont les différences. Marchez dans les pas de Darwin : Faites en sorte qu'un animal ait un pourcentage de chance d'être muté en un autre animal et faites évoluer cette probabilité.

Conseils

Procéder étapes par étapes, et écrire toute la structure de l'algorithme avec des méthodes vides.

Utiliser des constantes pour simplifier l'écriture de votre code. Ne cherchez pas à optimiser dès le début, et décrire votre algorithme avec un scénario de saisies optimistes et objectives (sans chercher à faire tomber votre programme en erreur).

Objectif / Niveau

1. Essentiel : Arriver à créer l'algorithme du niveau 1.
2. Attendu : Réussir à mettre à jour l'écosystème dans le niveau 2.
3. Avancé : Afficher l'écosystème dans le temps et jouer avec les différents paramètres d'entrée.

Solution

Des propositions de solution pour ce TP sont placées dans les éléments en téléchargement liés à ce module.

Astuces

Utiliser la fonction "randint" du package "random" pour obtenir une valeur aléatoire comprise entre zéro et un maximum (inclus). Vous pouvez aussi utiliser la fonction "choice" du même package "random" pour choisir une valeur aléatoire d'une liste. Quelques exemples ci-dessous :

```
from random import randint, choice

# Retourne un nombre entier compris entre 0 et 9
print("randint(9) = ", randint(0, 9))

ANIMAUX = [1, 2, 3]

# Retourne une valeur aléatoire de la liste
animal_aleatoire = choice(ANIMAUX)
```