HUNT

Hunt est un projet de simulation d'un système proie-prédateur.

OBJECTIF GENERAL

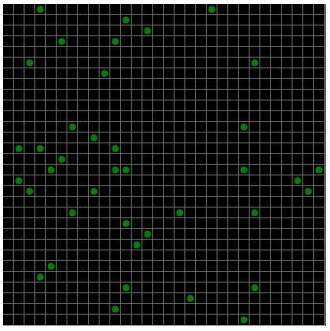
Dans un plan en deux dimensions, vous allez simuler la cohabitation de deux espèces: des proies et leurs prédateurs (vous pouvez imaginer des lapins et des renards, par exemple). En suivant des règles de naissance, de déplacement, de reproduction, de durée de vie et de chasse (pour les prédateurs) équilibrées,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

FONCTIONNEMENT GENERAL

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.

ETAPE 1 : AU PARADIS DES PROIES

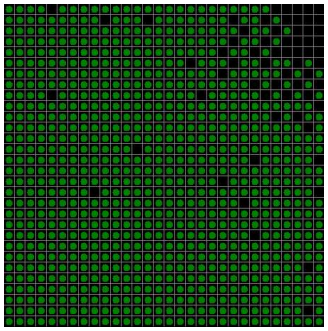


La première étape de ce projet consiste à réussir à faire évoluer des proies seules sur le plan. Cette étape est importante pour votre compréhension du projet. Elle va vous permettre de mettre en place les structures principales et l'organisation générale du projet. Toutefois, le résultat obtenu est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

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

- **NAISSANCE DES PROIES** : les proies naissent à des **positions aléatoires** à une **fréquence FPRO** donnée. (On veillera bien entendu à ne pas faire naître de proie sur une position occupée.) *Exemple: si FPRO=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 DPRO** 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 DPRO=40, les proies nées au commencement du jeu mourront au 40ème tour.*
- **DEPLACEMENT:** à chaque tour, chacune des proies se déplace aléatoirement d'une case dans une des 9 directions possibles de son voisinage (y compris sur sa propre case = pas de déplacement). 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.

ETAPE 2 : PLETHORE 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.

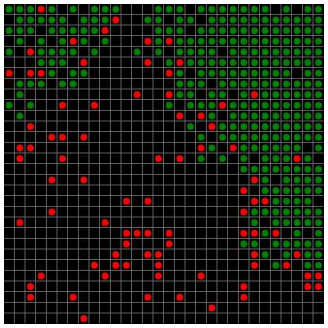
Dans la réalité, ces situations surviennent lorsqu'une espèce n'est plus confrontée à aucun prédateur et dispose de suffisamment de ressources alimentaires. Il s'ensuit nécessairement une explosion de la population et une famine pour cause de destruction des ressources.

Dans notre modèle, les proies ne sont les prédateurs d'aucune autre espèce et ne manquent jamais de nourriture : elles peuvent se reproduire sans contrainte, et naissent à un rythme régulier FPRO indépendant de tout autre paramètre extérieur.

La nouvelle règle que nous introduisons à cette étape 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. S'ensuit en général une explosion des naissances par cascade.

ETAPE 3 : INTRODUCTION DES PREDATEURS



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:

- **PREDATEUR INITIAUX** : au commencement de la partie, un nombre NPRE de prédateurs est positionné aléatoirement sur le terrain.

- **ENERGIE DES PREDATEURS** : à la naissance, chaque prédateur dispose d'une énergie EPRE, qui diminue de 1 à chaque tour. Si elle atteint zéro le prédateur meurt de faim. A la différence de l'âge des proies, l'énergie des prédateurs peut remonter (on ne se soucie pas de l'âge des prédateurs qui sont censés vivre plus longtemps que les proies

tant qu'ils se nourrissent…et risquent de mourir de faim avant de mourir de vieillesse). 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 EPRE=12 augmente cet EPRE de MIAM=5 si il mange une proie.*

- **REPRODUCTION DES PREDATEURS :** les prédateurs ne peuvent se reproduire que s'ils disposent d'un niveau d'énergie EREPRO, qu'on fixera supérieur à EPRE. 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 PREDATEURS :** pour qu'un prédateur ait une chance de manger, il faut qu'il dispose d'un algorithme de chasse efficace. Si vous lui attribuez un déplacement aléatoire similaire à celui des proies, il y a peu de chance qu'il survive. Aussi, vous devez élaborer un algorithme de calcul de la position de la proie la plus proche.

ETAPE 4 : PERFECTIONNEMENT DE LA SIMULATION

Nous vous proposons ici quelques suggestions d'améliorations visant à rapprocher la simulation d'une situation réelle, et à visualiser son évolution.

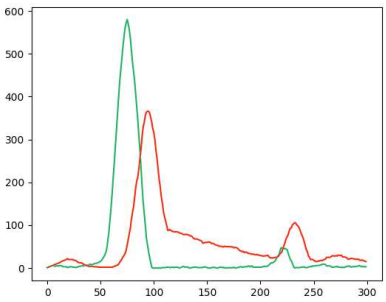
ALGORITHME DE CHASSE

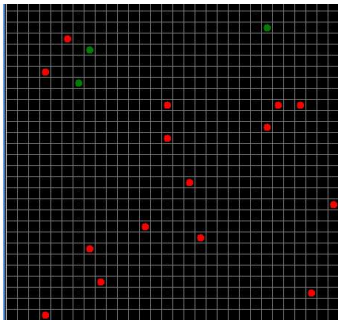
Afin de mieux coller à la réalité, on pourra introduire un paramètre FLAIR qui indique la distance maximale à laquelle un prédateur peut sentir une proie. Au-delà de cette distance, la proie est invisible pour le prédateur. (Niveau: moyen)

Par ailleurs, vous réaliserez rapidement que la simulation consomme beaucoup de ressources lorsque de nombreux prédateurs et proies sont présents simultanément sur le terrain. Vous pourrez tenter d'imaginer une solution algorithmique qui évite de calculer toutes les distances entre un prédateur et l'ensemble des proies du terrain. (Niveau: difficile)

VISUALISATION

Python est extrêmement puissant et facile d'accès lorsqu'il s'agit de visualiser des données. Vous pourrez faire appel au module `pyplot` de la librairie `math`, qui vous permettra de tracer les courbes des effectifs des deux populations.





Le comportement attendu au terme de l'étape 3 est un phénomène de cycles: la population de proies augmente au gré des rencontres et envahit le terrain, les prédateurs disposent alors d'une quantité importante de nourriture qui lui permet de se reproduire en masse. Cela entraîne rapidement une famine pour cette population importante de prédateurs, qui décroit naturellement…jusqu'au prochain cycle.

VOS IDEES !

Vous pouvez laisser libre cours à toute amélioration possible de cette simulation. Votre créativité fait l'objet d'une compétence spécifique (voir dernière section) qui impacte votre note.