

Projet 1: preys and predators

https://github.com/corentin-pujol/prey_predator

PUJOL Corentin, GAUTIER Alexandre

Ce projet consistait à concevoir un modèle preys and predators (avec en option une troisième catégorie passive de nourriture) en python avec mesa. Nous avons utilisé une CanvasGrid et trois types d'agents se déplaçant au hasard sur cette grille :

Conception des agents et de leur mouvement :

1. Loups

- Mouvement sur la grille
- Si le loup a faim (énergie inférieure à 50% de l'énergie initiale maximale), et qu'au moins un mouton se trouve sur sa cellule, il choisit un de ces moutons au hasard et le mange (gain d'énergie)
- Perte d'énergie
- Si l'agent n'a plus d'énergie, il meurt
- Reproduction de l'agent si celui ci est encore vivant (gestion à l'aide du paramètre taux de reproduction)

2. Moutons

- Mouvement sur la grille
- Perte d'énergie si l'herbe est activée
- Si l'herbe est activée et si il y'a de l'herbe mangeable sur la case et si le mouton a faim (énergie inférieure à 50% de l'énergie initiale maximale), alors l'agent mouton se nourrit (gain d'énergie)
- Si l'agent n'a plus d'énergie, il meurt
- Reproduction de l'agent, si celui-ci est encore vivant à la fin de l'étape (gestion à l'aide du paramètre taux de reproduction)

3. Herbe

- Si l'herbe est tondue, elle repousse (sous une contrainte temporelle)
- Si l'herbe est complètement repoussée, elle est mangeable
- Si l'herbe se fait manger, elle devient tondue et initialise un timer de repousse

Liste des paramètres gérant l'état initial de l'environnement et descriptions :

sheep_initial_energy : Variable réglable grâce à un slider déterminant l'énergie initiale maximale des moutons

wolf_initial_energy : Variable réglable grâce à un slider déterminant l'énergie initiale maximale des loups

grass : Détermine si l'herbe peut être mangée

show_energy : Variable booléenne permettant d'afficher sur la page l'énergie des loups en temps réel.

initial_sheep : Variable réglable avec un slider déterminant le nombre initial de moutons

initial_wolves : Variable identique à initial_sheep, permettant la gestion du nombre initial des loups

sheep_reproduce : Variable réglable grâce à un slider déterminant la probabilité de reproduction d'un mouton

wolf_reproduce : Variable réglable grâce à un slider déterminant la probabilité de reproduction d'un loup

wolf_gain_from_food : Variable réglable grâce à un slider déterminant le gain d'énergie d'un loup après avoir mangé un mouton

sheep_gain_from_food : Variable réglable grâce à un slider déterminant le gain d'énergie d'un mouton après avoir mangé de l'herbe

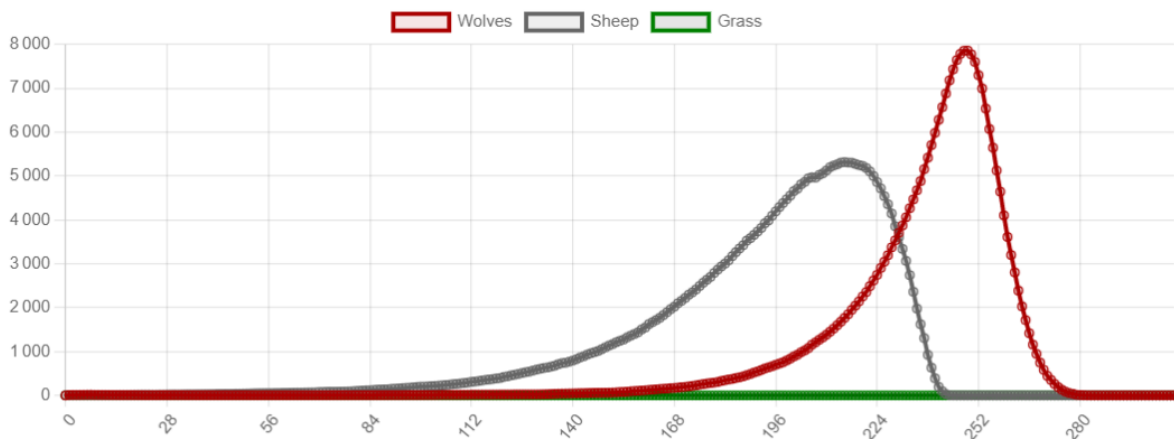
grass_regrowth_time : Temps de repousse de l'herbe, avant qu'elle soit à nouveau comestible

Choix des paramètres initiaux :

Nos paramètres initiaux étant réglables sur notre interface, nous avons pu les choisir par essais successifs afin d'obtenir les différents scénarios suivant :

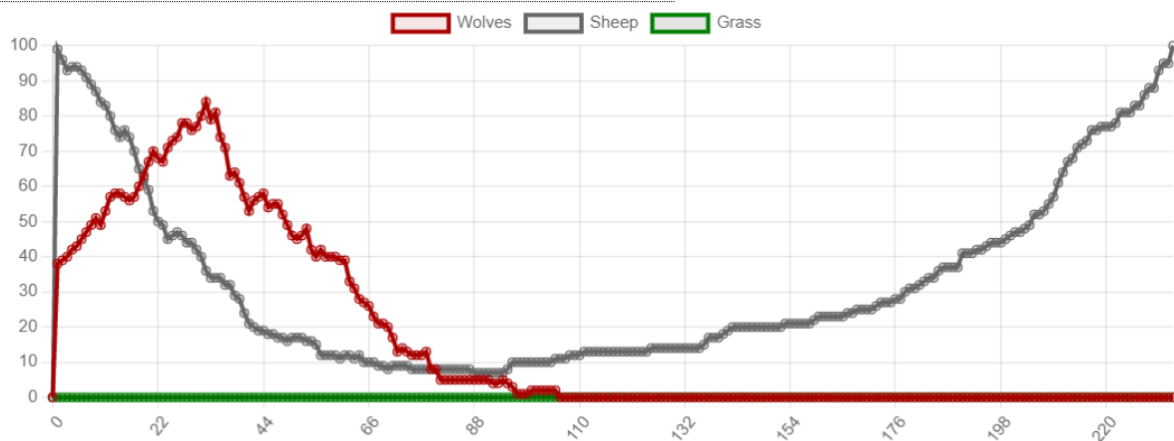
Exemple avec des paramètres initiaux menant à une éradication de toute la population :

Initialisation avec 10 M (moutons) et 10 L (loups), herbe désactivée, taux de reproduction : 0.04 (M), 0.05 (L), gain d'énergie en mangeant : 4 (M), 20 (L), énergie maximale initiale : 40 (M & L)



Cas où seuls les moutons survivent :

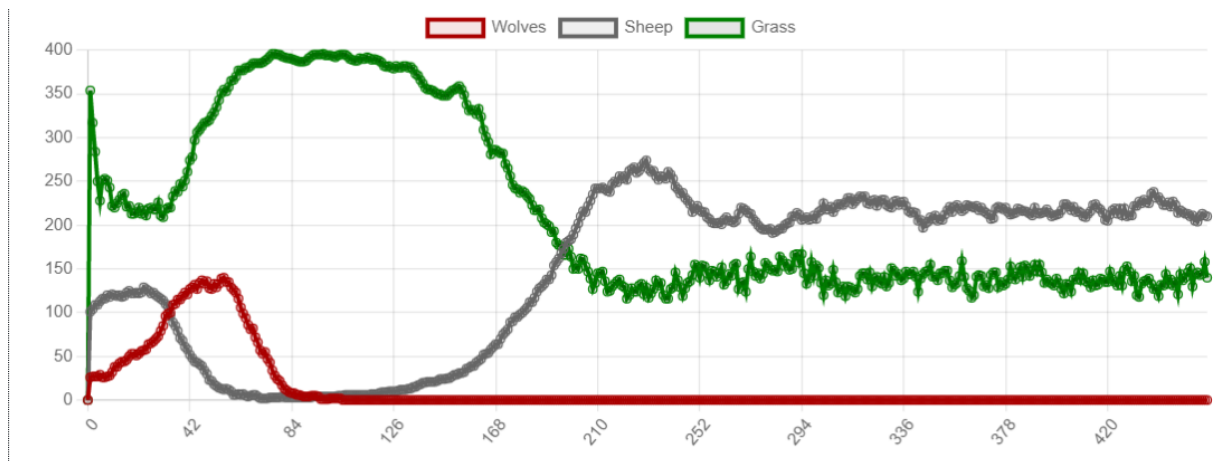
Initialisation avec 100 M (moutons) et 40 L (loups), herbe désactivée, taux de reproduction : 0.02 (M), 0.05 (L), gain d'énergie en mangeant : 4 (M), 20 (L), énergie maximale initiale : 40 (M & L)



Il n'y a aucun cas où les loups survivent car lorsqu'il n'y a plus aucun mouton, ils ne peuvent plus subvenir à leur besoin et la population décroît et converge vers 0.

Cas où seuls les moutons survivent, avec l'herbe activée. On observe que l'herbe activée stabilise la population de moutons autour d'un équilibre final.

Initialisation avec 100 M (moutons) et 24 L (loups), herbe activée, taux de reproduction : 0.04 (M), 0.05 (L), gain d'énergie en mangeant : 4 (M), 20 (L), énergie maximale initiale : 75 (M), 40 (L), temps de repousse de l'herbe : 5



Conclusion :

Avec notre modèle, il est compliqué d'obtenir de nombreuses oscillations dans la population de loups et de moutons. Rendre les loups plus "intelligents" pourrait pallier ce problème. En effet dans la réalité les loups chassent et se dirigent vers les moutons, alors que dans notre modèle, si la population de moutons diminue, les interactions entre un loup et les moutons devient extrêmement peu probable.

Une amélioration de notre modèle serait de définir un taux de précision de chasse, permettant aux loups de se placer sur une case mouton afin de se nourrir. Ce taux permet d'éviter de décimer la population de moutons, car un loup n'a pas 100% de réussite lorsqu'il chasse.