Étude et modélisation d'un système proie-prédateur par un automate cellulaire

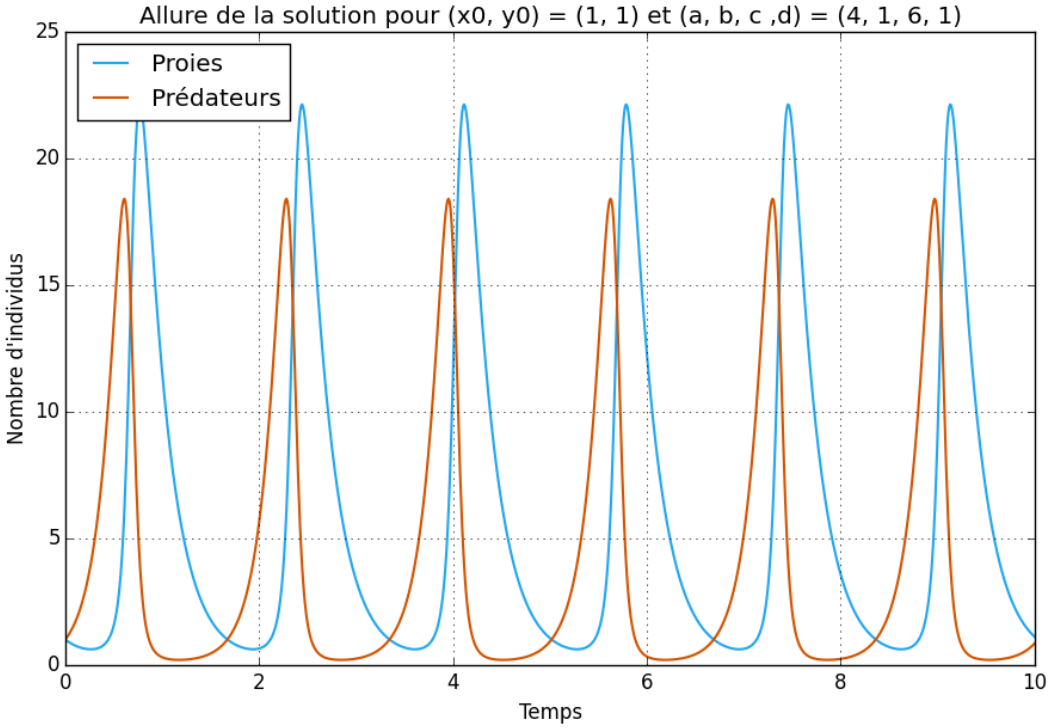
I. Le modèle proie-prédateur de Lotka-Volterra (1925)
II. Modélisation par un automate cellulaire
III. Résultats et comparaison des modèles

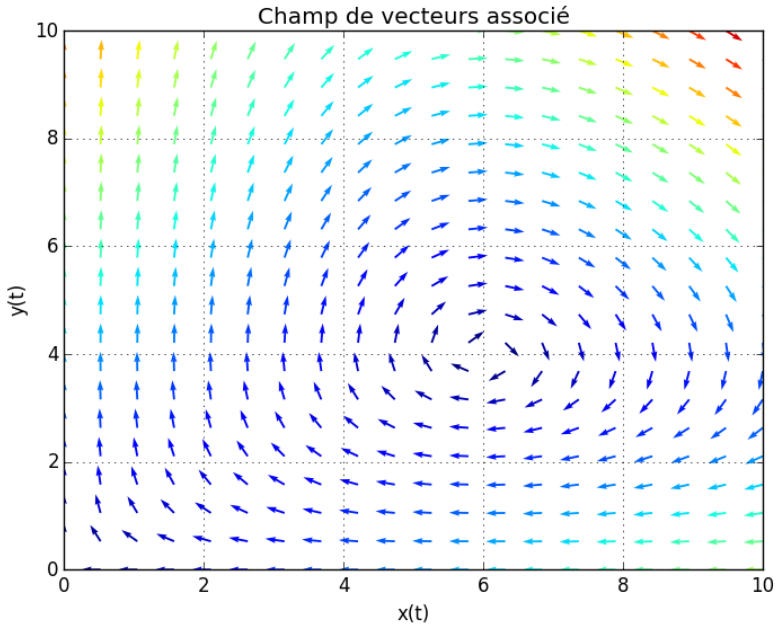
Équations de Lotka-Volterra :

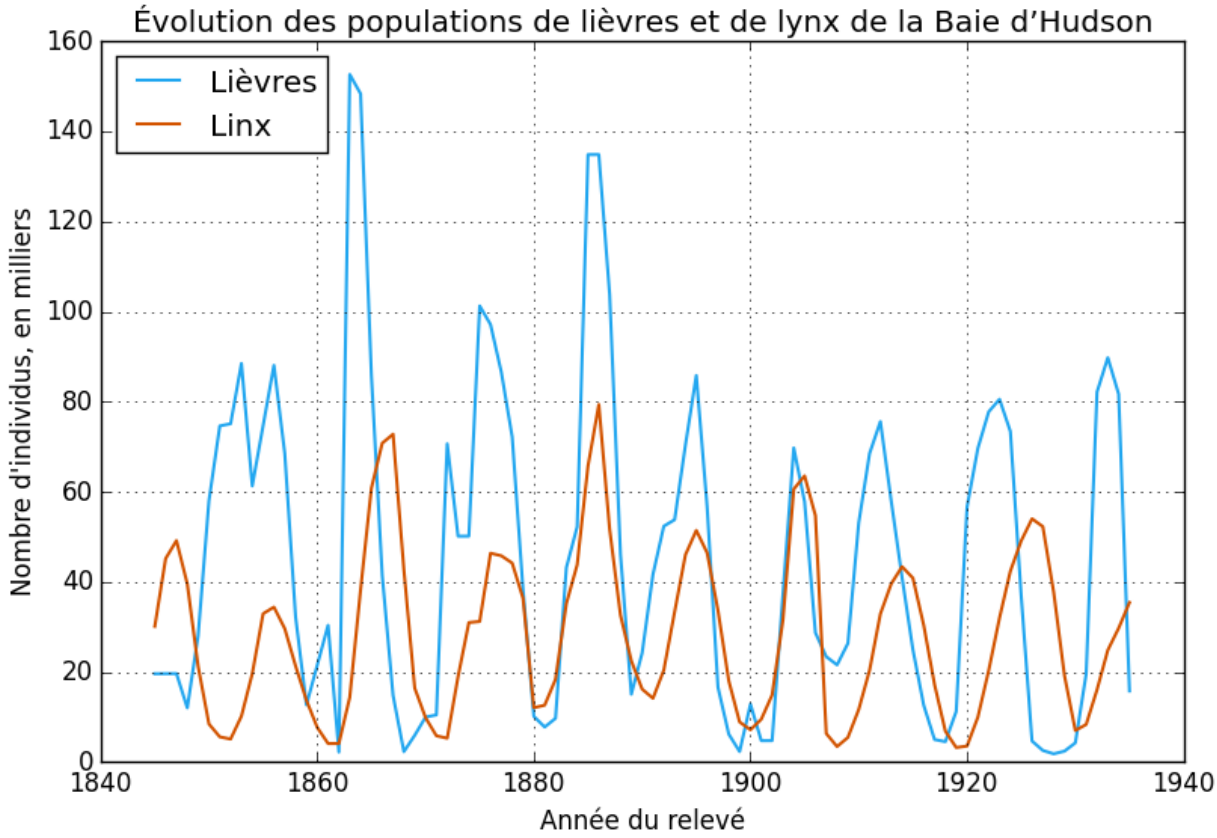
$$(LK) : \begin{cases} x'(t) = x(t)(\alpha - \beta y(t)) \\ y'(t) = y(t)(-\gamma + \delta x(t)) \\ x(0) = x_0,\ y(0) = y_0 \end{cases}$$

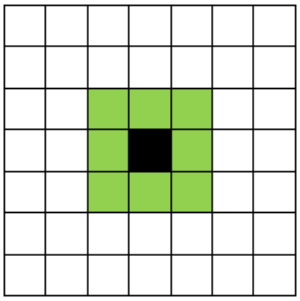
(LK) est caractérisé par :

- x_0 : population initiale de proie ;
- y_0 : population initiale de prédateur ;
- α : taux de reproduction des proies ;
- β : taux de mortalité des proies face aux prédateurs ;
- γ : taux de perte des prédateurs (mort naturelle ou émigration) ;
- δ : taux de reproduction des prédateurs.









Le voisinage de Moore

La modélisation *Wa-Tor*

Étape 1 : Comportement des proies :

1. Déplacement aléatoire de la proie.
2. Peut laisser derrière elle une nouvelle proie, alors variable interne réinitialisée à 0.
3. Sinon, compteur interne incrémenté de 1.

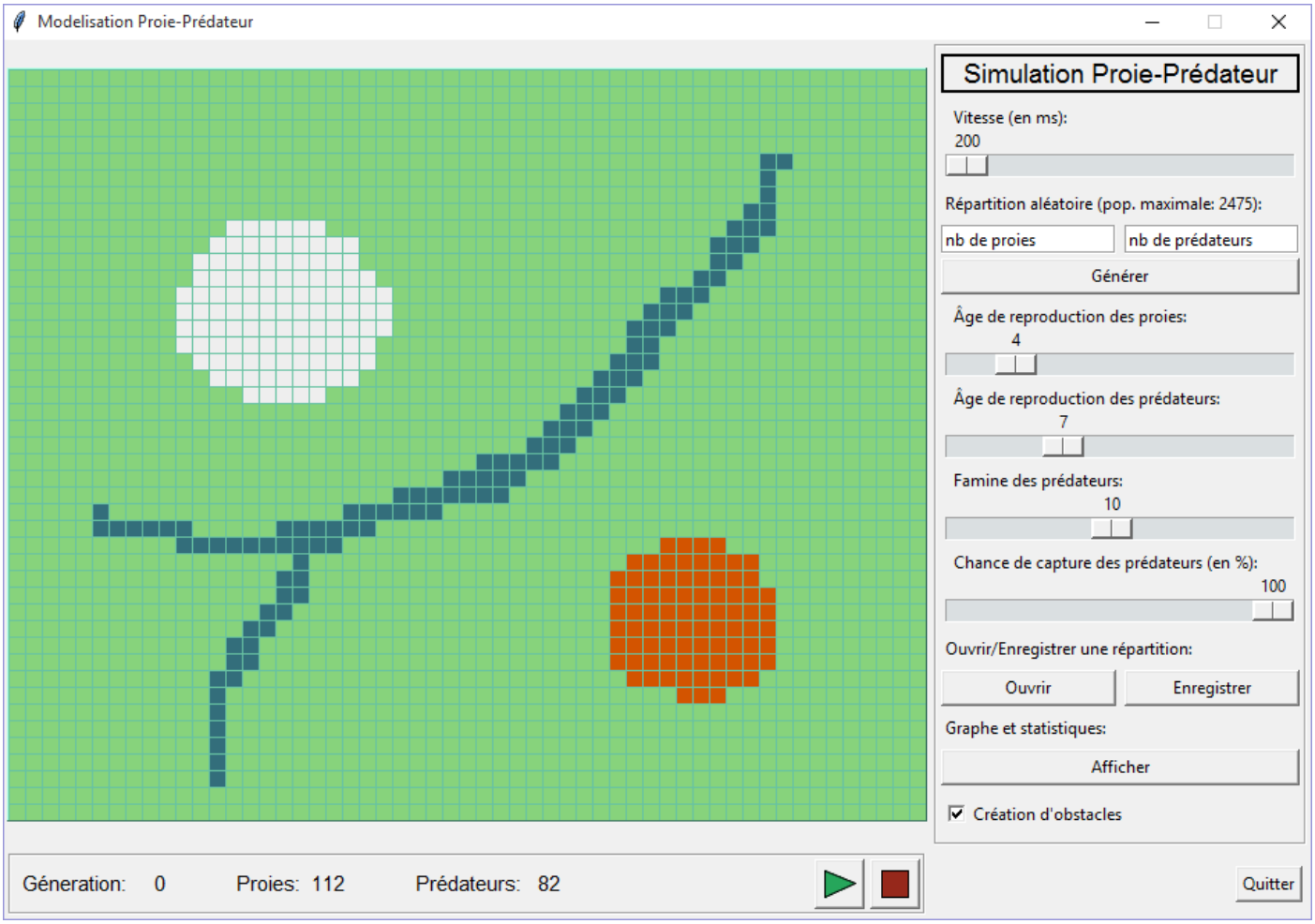
Étape 2 : Comportement des prédateurs :

1. Mort du prédateur s'il y a famine.
2. Sinon, si une ou plusieurs proie(s) dans son voisinage, capture et déplacement du prédateur vers la proie. Peut laisser derrière lui un nouveau prédateur.
3. Compteur de famine incrémenté de 1 si pas de proies adjacentes, compteur de reproduction aussi si pas de naissance.

```
class ProiePredateur
    def __init__
    def creer_obstacle
    def nouv_grille
    def change_param
    def coord_milieu
    def coord_proie
    def coord_predat
    def coord_mort
    def proie_vie
    def predat_vie
    def tue_cellule
    def cel_voisines_proie
    def cel_voisines_predat
    def animation
    def start
    def pause
    def change_bouton
    def import_motif
    def save_motif
    def pop_aleatoire
    def affiche_graphes
```

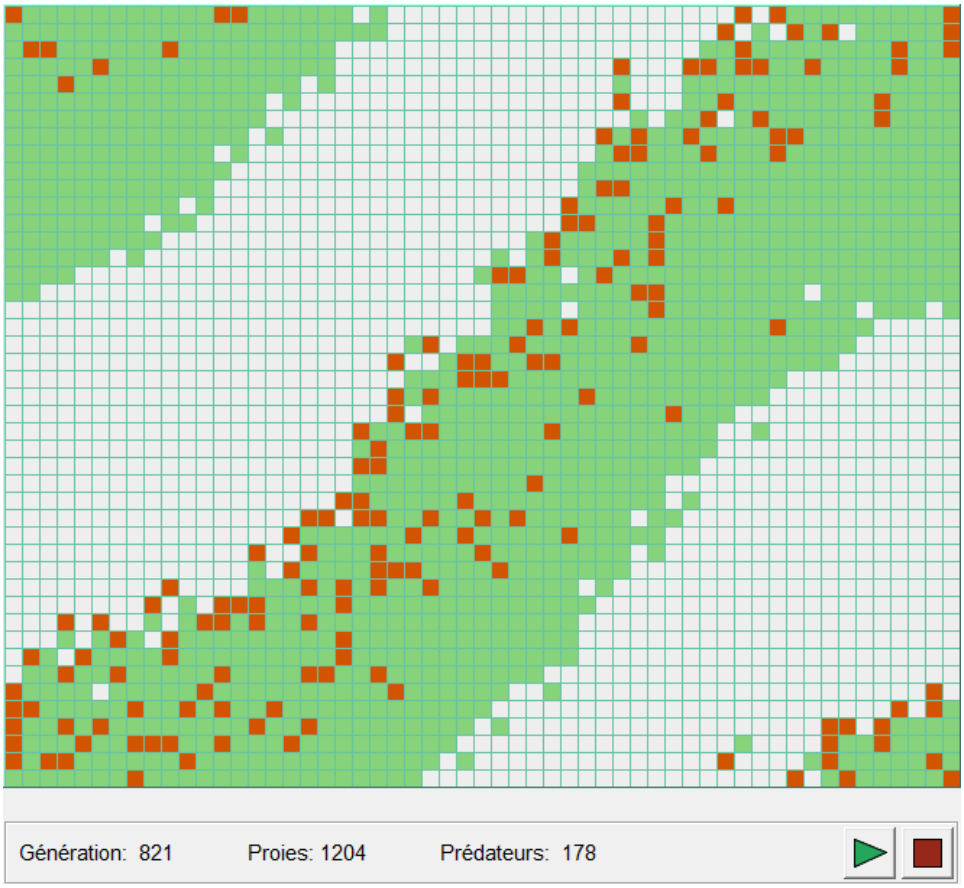
```python
from tkinter import *
from random import randint, choice
from time import clock
import matplotlib.pyplot as plt
import numpy as np
import pickle #Permet d'enregistrer des données avec conservation de leur type
#Module donnant accès aux fenêtres de recherche de fichiers:
from tkinter.filedialog import asksaveasfile, askopenfile


class ProiePredateur(Frame):
    """La classe principale de la simulation proie prédateur et ses méthodes."""
    def __init__(self, master):
        Frame.__init__(self, master)
```

Les modules importées et les méthodes de la classe principale « ProiePredateur »

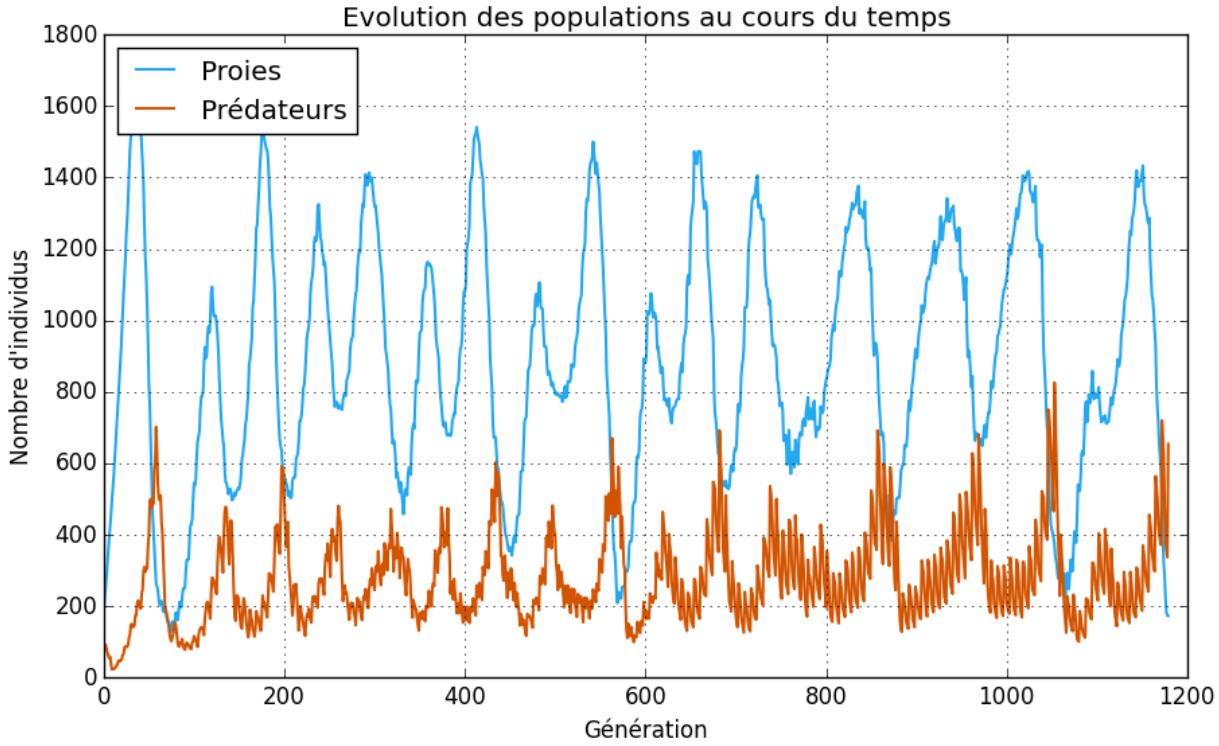


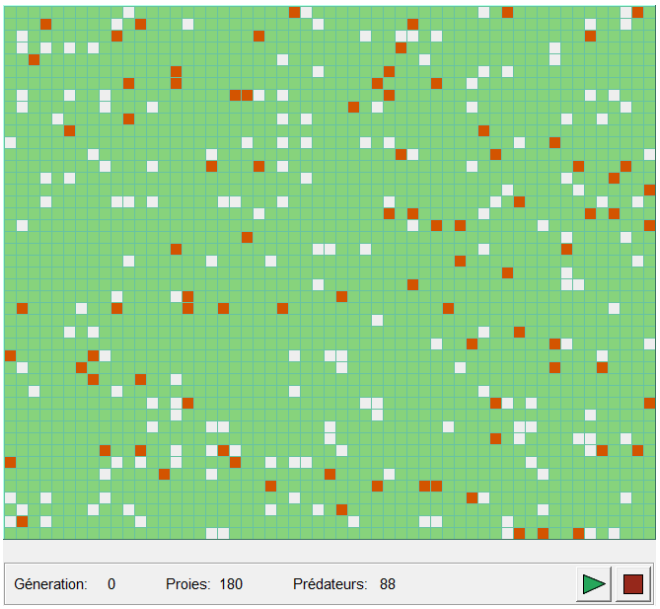
Le programme final, dans sa configuration par défaut et avec une répartition quelconque

```
def nouv_grille(self):
    "Affiche l'espace de simulation et initialise les variables."
    #Permet d'éviter une itération en trop lorsqu'une simulation est en cours
    if self.flag:
        self.flag = 0
        self.after(self.vitesse, self.nouv_grille)
        return

    self.change_bouton(1)       #Affichage du bouton Play
    self.can.delete(ALL)        #Efface le canevas

    #Initialisation des variables:

    self.generation = 0         #Compteur de génération
    #Tableau contenant respectivement le nombre de proies et
    #le nombre de prédateurs actuel:
    self.pop = [0, 0]

    #Tableau contenant les populations à chaque génération
    #(cet historique est utile lors du tracé de graphes)
    self.evol_pop =([], [])

    #Tableau des proies contenant pour chaque coordonnées
    #l'âge de la proie (> 0 ou 0 si pas de proie):
    self.grille_proie = np.zeros( (self.xmax, self.ymax), dtype = np.int8)
    #Pareil pour les prédateurs:
    self.grille_predat = np.zeros( (self.xmax, self.ymax), dtype = np.int8)
    #Tableau contenant la 'faim' actuelle du prédateur
    self.grille_predatfaim = np.zeros( (self.xmax, self.ymax), dtype = np.int8)
    #Tableau repérant les cases vivantes, d'obstacles et vides (-1):
    self.grille_vie = np.zeros( (self.xmax, self.ymax), dtype = np.int8)
    #Tableau contenant les cases en elles-mêmes:
    self.grille_items = np.empty( (self.xmax, self.ymax), dtype = np.object)
```



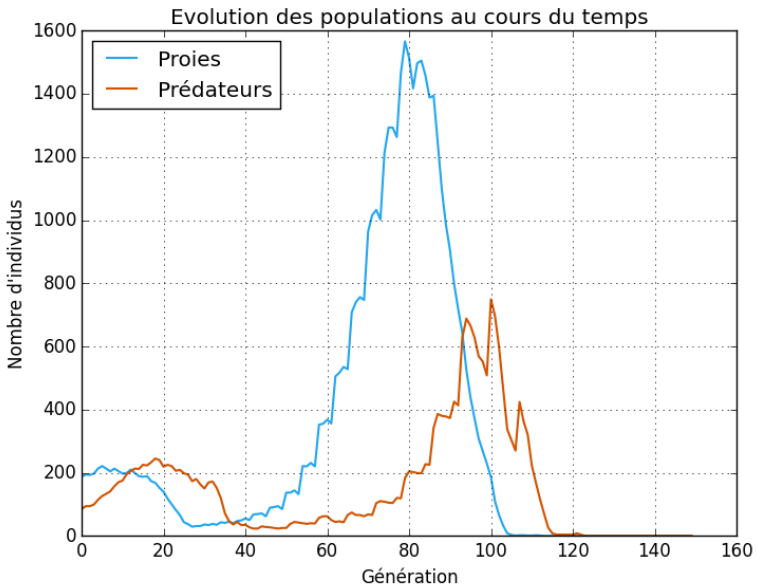
Le phénomène migratoire d'une configuration stable



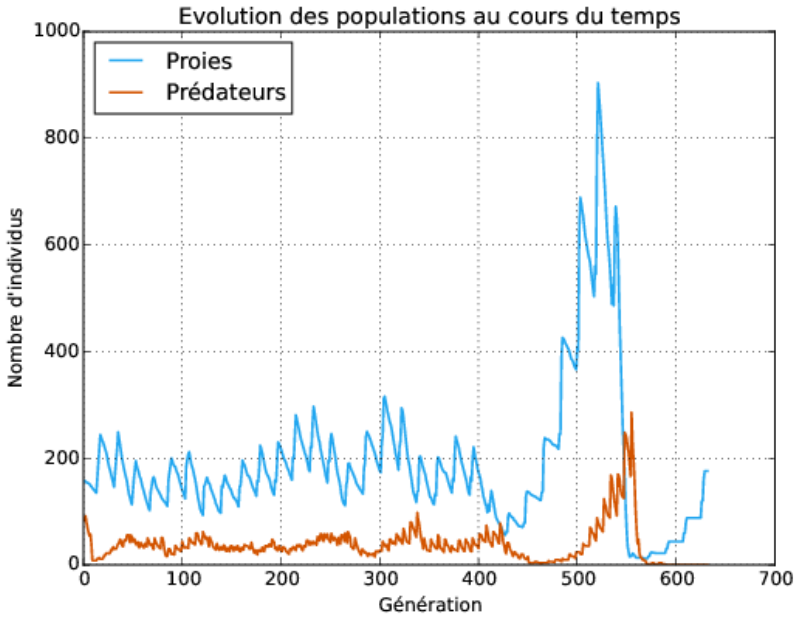


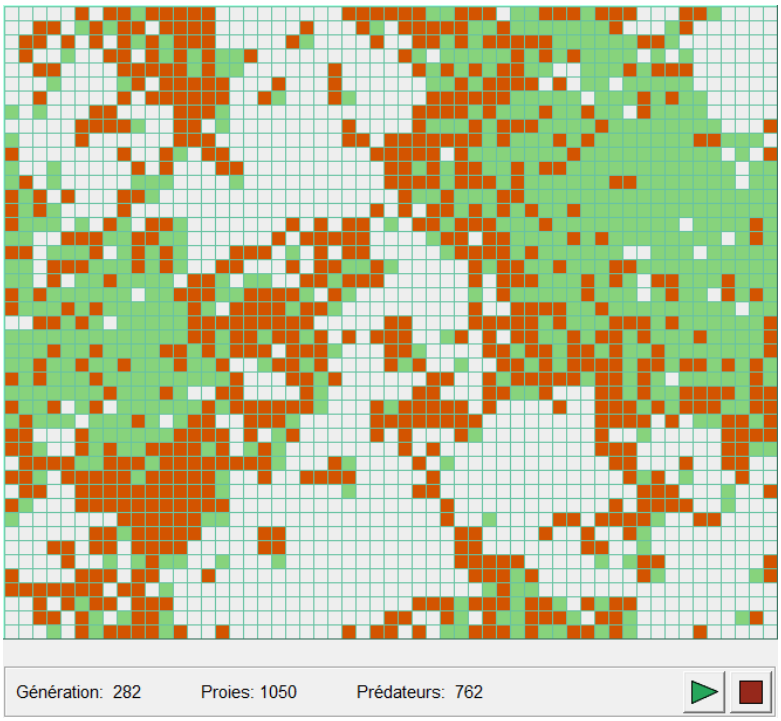
Une répartition aléatoire

Le graphe obtenu



Évolution pour α= 18: disparition des prédateurs à la génération 588.





Un exemple d'évolution pour β= 0.05

Évolution pour δ = 18, aboutissant à un système stable