

DNS n°03 d'informatique

A rendre le mardi 09 janvier de la rentrée.

Le jeu de la vie

Avant de démarrer je vous invite à regarder la vidéo dont le lien est donné ci – dessous (ou tapez dans votre moteur de recherche préféré : « jeu de la vie »).

<https://www.youtube.com/watch?v=S-W0NX97DB0>

La règle du jeu est la suivante. On dispose d'un damier (qu'on appellera aussi tableau dans la suite) carré de N cases de côté. Il y a donc N^2 cellules. On applique les règles ci – dessous :

- *Il n'y a que deux états possibles pour une cellule : vivant ou mort.*
- *Une cellule avec moins (strictement) de deux voisins vivants meurt d'isolement.*
- *Une cellule avec plus (strictement) de trois voisins vivants meurt de surpopulation.*
- *Une cellule avec exactement 3 cellules vivantes autour d'elle devient vivante (une naissance).*

Pour faciliter la programmation on va stocker le tableau avec le type « array » de numpy (ne pas oublier d'importer numpy). On va laisser à zéro tout le bord du tableau. Cela signifie qu'avec notre code on ne pourra pas avoir des « objets » qui partent à l'infini (regarder la vidéo si vous ne comprenez pas cette phrase). Le bord du tableau devient un « no man's land ».

A chaque itération on doit donc créer un nouveau tableau à partir des règles ci – dessus. Afin d'économiser l'espace mémoire on va travailler avec deux tableaux que l'on va appeler *Tab* et *Tabp1* (le *p1* à la fin du nom signifie « plus 1 »). A chaque itération on va placer dans *Tabp1* l'étape suivante du jeu de la vie et par une simple affectation double on va remplacer *Tab* par *Tabp1* et *Tabp1* par *Tab*. On évite ainsi de surcharger la mémoire.

1. Ecrire la fonction $F2(Tab, Tabp1)$ qui prend comme paramètre deux tableaux et qui construit dans le tableau *Tabp1* l'itération suivante du jeu de la vie.

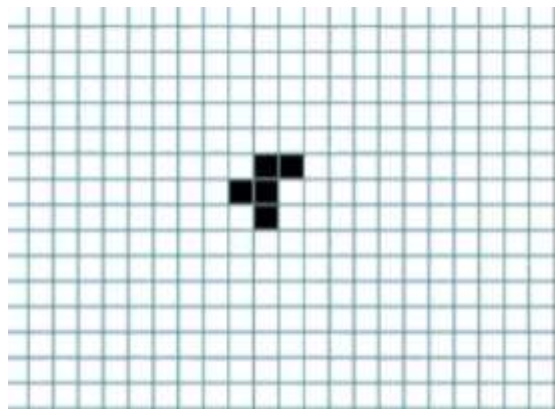


Fig : Essayez cet état initial avec environ 1100 itérations.

2. Vous pouvez essayer l'état initial constitué de 5 cellules de la figure ci – dessus. Le résultat est étonnant (cf. figure ci – dessous).

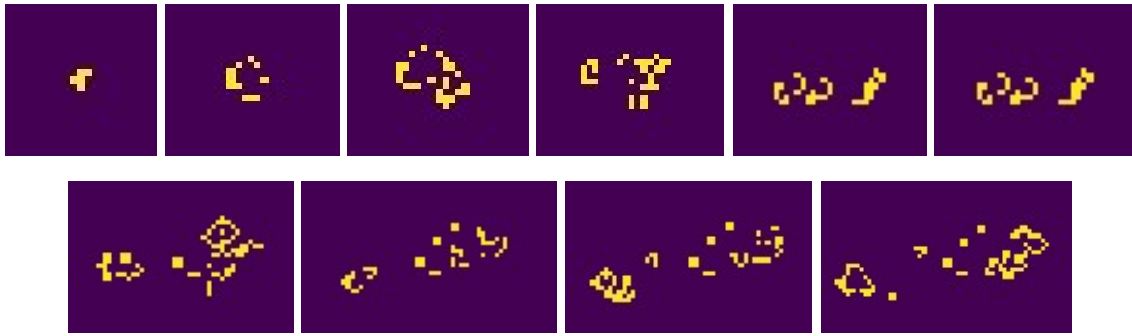


Fig : Un exemple aux itérations 0, 9, 19, 29, 39, 49, 59, 69, 79, 89

CODE CADEAU

```

8 import numpy as np
9 import matplotlib.pyplot as plt
10
11 def F1(Tab) : """ Cette fonction crée une image avec des cases de P*P pixels """
12     global P
13     H,L = len(Tab), len(Tab[0])
14     Ima = np.zeros((P*H,P*L),dtype='int')
15     for j in range(H) :
16         for i in range(L) :
17             val = Tab[j,i]
18             for a in range(P) :
19                 for b in range(P) :
20                     Ima[P*j+b,P*i+a] = val
21     return Ima
22
23 def F2(Tab,Tabp1) :
24     H, L = len(Tab), len(Tab[0])
25     for j in range(1,H-1) : """ On ne traite pas es bords de l'image de 1 à H-1 exclu"""
26         for i in range(1,L-1) : """ On ne traite pas es bords de l'image de 1 à H-1 exclu"""
27             """ VOTRE CODE
28             VOTRE CODE
29             VOTRE CODE
30             VOTRE CODE """
31     return Tabp1,Tab """ Avec l'appel de la fonction on permute Tabp1 et Tab """
32     """ Tab, Tabp1 = F2(Tab,Tabp1) """
33 N = 100 """ Taille d'un coté du tableau """
34 P = 10 """ Taille d'un coté d'une case dans Ima (cf. fonction F1) """
35 NbPas = 1000 """ no comment """
36 Tab = np.zeros((N,N)) """ Initialisation des tableaux """
37 Tabp1 = np.zeros((N,N))
38 """ Initialisation du motif de départ """
39 """ Exemple : un clignotant """
40 Tab[50,50] = 1
41 Tab[40,50] = 1
42 Tab[51,50] = 1
43
44 """ LA boucle for pour itérer le jeu de la vie """
45 for k in range(NbPas) :
46     fig = plt.figure()
47     Ima = F1(Tab)
48     plt.imshow(Ima) """ Affiche une image """
49     _m = k//1000 """ chiffre des milliers """
50     _c = (k//100)%10 """ chiffre des centaine """
51     _d = (k//10)%10 """ chiffre des dizaine """
52     _u = k%10 """ chiffre des unité """
53     nom = "Jeu2LaVie"+str(_m)+str(_c)+str(_d)+str(_u)+".png"
54     plt.savefig(nom) """ On sauvegarde dans un fichier numéroté """
55     plt.close()
56     print("pas : "+str(_m)+str(_c)+str(_d)+str(_u)) """ Pour savoir ou en est le prg """
57     Tab, Tabp1 = F2(Tab,Tabp1) """ Si vous regarder la dernière ligne de la fonction
58                               F2 vous vous rendrez compte que on a bien permuter
59                               Tab et Tabp1 """
60

```