TD 10

Jeu de la vie

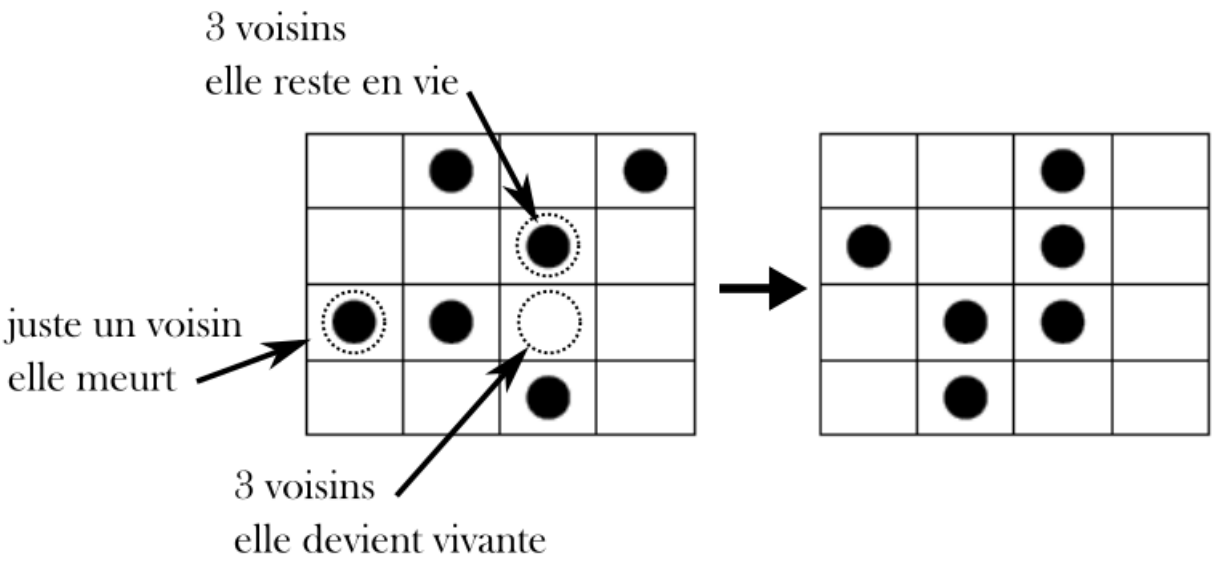
Lycée Louis Thuillier - Informatique - PCSIB -

Principe du "jeu"

Le jeu de la vie n'est pas un vrai jeu mais ce qu'on appelle un automate cellulaire : il consiste en une grille régulière de « cellules » contenant chacune un « état » (vivante ou morte) qui peut évoluer au cours du temps. Imaginé par John Conway (1937 - 2020), il se déroule sur une grille à deux dimensions dont les cases, appelées cellules, peuvent prendre deux états distincts : vivantes ou mortes.

A chaque étape, l'évolution d'une cellule est déterminée par l'état de ses huit voisines :
- ▷ Une cellule morte possédant **exactement** trois voisines vivantes devient vivante, sinon elle reste morte.
- ▷ Une cellule vivante possédant deux ou trois voisines vivantes reste vivante, sinon elle meurt.

Par exemple :



| ***Application 1 :*** *Représenter la grille de droite à l'étape suivante ?*

L'évolution d'une grille est un processus répétitif qui est parfait à confier à un ordinateur !

Implémentation sous Python

En informatique, on représentera une grille par un tableau *array* du module *numpy* composé :
- ▷ de 0 pour les cellules mortes
- ▷ de 1 pour les cellules vivantes

Ainsi la cellule de gauche ci-dessus s'écrit :

```
cellule=np.array([[0,1,0,1],[0,0,1,0],[1,1,0,0],[0,0,1,0]])
```

On pourra visualiser une cellule à l'aide de la commande *imshow*() du module *matplotlib.pyplot*, importé sous le nom *plt* :

```
import matplotlib.pyplot as plt
plt.imshow(cellule,interpolation='None',cmap='gray')
plt.show()
```

Application 2 :

1. *Écrire une fonction grille_aleatoire qui, prenant en argument un entier n, renvoie une grille de taille $n \times n$ remplie aléatoirement de cellules vivantes ou mortes.*
   Commande : *la fonction randint(a,b) du module random (à importer donc!!) renvoie un entier aléatoirement tiré dans l'intervalle $[a,b]$.*
2. *Écrire une fonction nombre_voisin qui, recevant une grille T et deux entiers i et j, renvoie le nombre de cellules vivantes voisines de la cellule de coordonnées (i,j).*
   Attention ! *Question compliquée, il faudra prendre le temps de bien y réfléchir* ***au brouillon*** *!!*
3. *Écrire une fonction grille_suivante qui, recevant en argument une grille T, renvoie la grille obtenue à l'étape suivante.*

Bonus : pour faire une jolie animation!!

```
from matplotlib import animation

def jeu(grille,dt=10):
        fig = plt.figure()
        image=plt.imshow(grille,interpolation='None',cmap='gray')
        def mis_a_jour(_):
                nonlocal grille
                image.set_data(grille)
                grille=grille_suivante(grille)
                return image,
        anim=animation.FuncAnimation(fig,mis_a_jour,interval=dt,blit=True)
        plt.show()
```