

Le jeu de la vie

Le jeu de la vie est un **automate cellulaire**.

La vie se passe dans un tableau à doubles entrées (éventuellement infini) et son évolution est une succession d'étapes. Chaque case a donc huit cases voisines. Chaque case est soit vivante soit morte. Les règles d'évolution de la vie sont :

1. si une case morte a exactement trois voisins vivant alors elle naît ;
2. si une case vivante a deux ou trois voisins vivants alors elle reste vivante ;
3. si une case vivante ne possède pas deux ou trois voisins vivants alors elle meurt (par isolement ou surpeuplement).

En appliquant ces trois règles simultanément à tout le tableau, nous déterminons son étape suivante.

1 Construire un tableau

Les tableaux qui existent par défaut en python ne sont pas à double entrée (mais on peut faire des listes de listes). Nous utiliserons donc l'extension : **numpy**.

Créez un tableau carré 20 par 20 rempli de 0. Affichez-le. Remplissez ses cases aléatoirement par des 0 et des 1

```
import numpy
monde=numpy.zeros((10,10),dtype=int)
```

```
import random
for i in range(100):
    print(random.randint(3,8), end=' ')
```

Écrire une fonction **afficher(monde)** qui affiche l'état du monde.
L'affichage doit être assez joli.

2 La vie évolue

Écrire une fonction qui étant donné un tableau et les indices **i** et **j** compte le nombre de 1 dans les huit cases voisines.

Écrire une fonction qui pour un tableau et les indices **i** et **j** détermine l'état de la cellule (**i, j**) à la prochaine étape.

Écrire une fonction qui prend un tableau en entré et retourne son étape suivante.

3 Écologie

Pour observer l'évolution du monde pendant un temps assez long, il faut améliorer l'affichage. Nous pouvons nous lancer dans de jolis graphiques.

Il faut utiliser la librairie ToolKit sous la forme `Tkinter` par exemple.

Pour dessiner, il faut créer une fenêtre (`Frame`) et dans cette fenêtre créer une zone de dessin (`Canvas`). On peut ensuite placer des objets graphiques dans cette fenêtre.

```
import Tkinter
fenetre=Tkinter.Tk()
fenetre.title("Jeu de la vie")
canvas = Tkinter.Canvas(fenetre, width=100, height=100, background='yellow')
canvas.create_rectangle((10,20,30,40),fill="#ff0000")
canvas.pack()
fenetre.mainloop()
```

Pavez la fenêtre par des carrés dont la couleur représente l'état des cases.

Écrire une fonction `affiche(monde, canvas)` qui efface le canvas (`delete()`) puis le remplit de carrés dont la couleur représente l'état des cases.

Une fenêtre peu réagir à des événements (clique de la souris, touche pressée, recadrage, etc.). Pour cela il faut lier (*bind* en anglais) un événement et une fonction.

```
def vivre_sa_vie(event):
    monde=etape_suivante(monde)
    affiche(monde,canvas)

canvas.bind("<Button-1>",vivre_sa_vie)
```

Faites réagir la fenêtre pour qu'à chaque clique le monde évolue.

Explorez les possibilités de ToolKit pour obtenir de belles versions du jeu de la vie.