

# CODING CLUB

EPITECH LYON



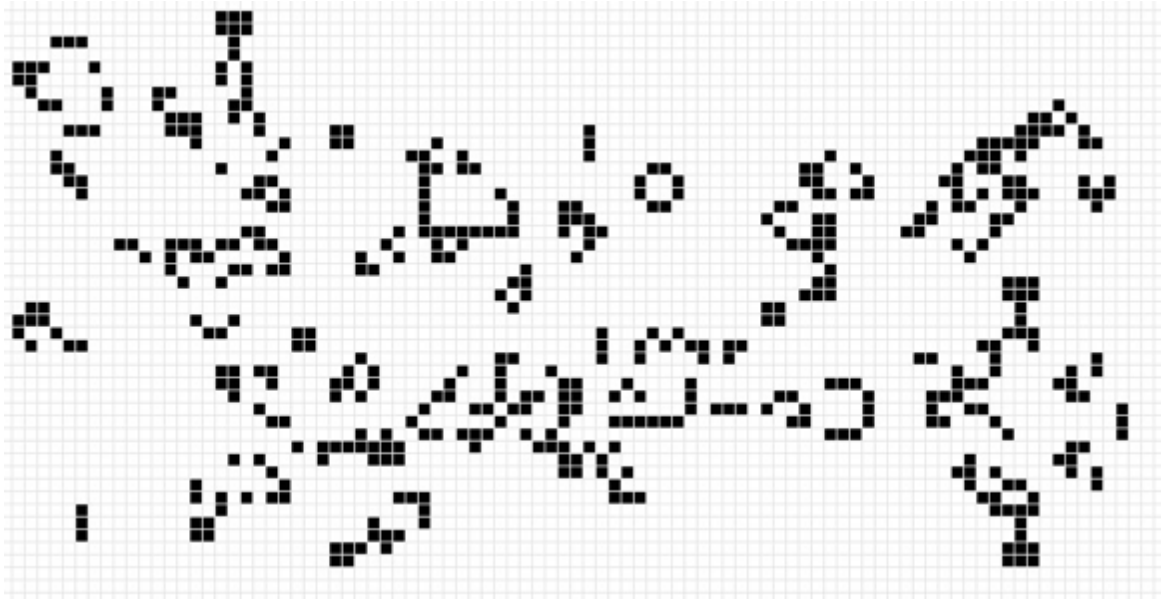
## Game of Life

---

VERSION 1.2.1

# I. Introduction

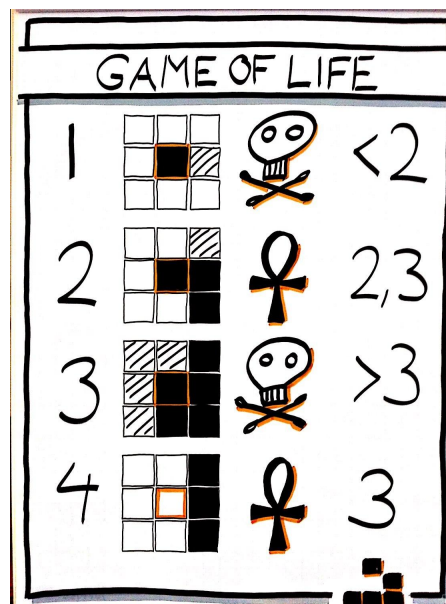
Game Of Life est un automate cellulaire créé par John Conway. Ce jeu a été créé en s'inspirant de phénomènes biologiques et nous permet d'observer l'évolution de plusieurs cellules dans un plan.



## II. Règles du jeu

Le jeu se déroule dans une grille à deux dimensions. Les cellules évoluent au tour par tour selon des règles simples :

- Toute cellule vivante avec moins de deux voisins vivants meurt (appelée sous-population).
- Toute cellule vivante avec plus de trois voisins vivants meurt (appelée surpopulation).
- Toute cellule vivante avec deux ou trois voisins vivants vit, inchangée, jusqu'à la génération suivante.
- Toute cellule morte avec exactement trois voisins vivants prend vie.



## III. Application

Votre objectif est de recréer le **Game Of Life** de Conway en Python.

Pour commencer, instanciez une nouvelle partie et chargez une map :

```
from cc_gameoflife.core import *

game = GameOfLife()
game.load_map("maps/map.txt")
```

- `game.grid` est une liste de toutes les cellules

Chaque cellule à des propriétés :

- `cellule.is_alive` indique si la cellule est en vie au tour actuel.
- `cellule.has_neighbor(x, y)` indique si elle a un voisin aux coordonnées relatives renseignées.
- `cellule.alive` est à définir (True/False), elle indique si la cellule sera en vie au prochain tour ou non.

Exemple

```
>>> cellule.has_neighbor(0, 1)
True
```

Cela signifie que la cellule a un voisin au dessus d'elle.

### Maintenant nous pouvons recréer le Game Of Life.

Dans une boucle infinie, comptez les voisins de chaque cellule de `game.grid` et, selon les règles du **Game Of Life**, renseignez le champs `cellule.alive` de la cellule pour qu'elle soit en vie (ou non) au prochain tour.

À la fin du tour, appelez `game.show_grid()` pour afficher la grille.

⚠ Pensez à rajouter un délai de 0.1s entre chaque itération pour pouvoir admirer le résultat.