
Simulation du jeu de la vie dans un notebook Jupyter

Pierre Poulain - 2018

Ce projet consiste à simuler le jeu de la vie dans un notebook Jupyter.

1 Présentation

Le jeu de la vie¹ a été imaginé par John Conway en 1970. C'est un automate cellulaire, c'est-à-dire que chaque état conduit au suivant à partir d'un certain nombre de règles pré-définies.

David Louapre du blog Science étonnante² a publié une vidéo très bien documentée sur le sujet³.

1.1 Règles

Le jeu se déroule sur une grille à 2 dimensions. Chaque cellule de la grille peut prendre deux états : *vivant* ou *mort*. Chaque cellule de la grille a 8 cellules voisines.

À chaque étape, l'évolution de l'état d'une cellule est déterminé par l'état de ses voisines :

- Une cellule morte possédant exactement 3 voisines vivantes devient vivante.
- Une cellule vivante possédant 2 ou 3 voisines vivantes reste vivante, sinon elle meurt.

1.2 Représentation

Le jeu de la vie est représentée sur une grille à 2 dimensions dont la taille est arbitrairement fixée mais néanmoins fonction des motifs initiaux et de leur mobilité (vous verrez en effet que certains motifs se déplacent).

Les cellules mortes sont généralement représentées en blanc et les cellules vivantes en noir.

1.3 Structures remarquables

En fonction de la configuration initiale, des structures peuvent émerger : structures stables, oscillateurs, vaisseaux, fumeurs, canons... À vous de les découvrir !

2 Contraintes techniques

- Votre programme principal sera dans un notebook Jupyter. Vous pouvez créer un module pour y mettre les principales fonctions que vous appellerez depuis le notebook.
- L'animation sera réalisée par le sous-module `animation` du module `matplotlib`. La fonction `imshow` permet d'afficher une grille et d'en colorer certaines cellules.

1. https://fr.wikipedia.org/wiki/Jeu_de_la_vie

2. <https://sciencetonnante.wordpress.com/>

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

-
- David Louapre a publié le code source qu’il a utilisé pour faire sa vidéo. Vous pouvez bien sur y jeter un oeil, **mais** vous ne pouvez pas copier/coller le code tel quel. Vous devez réimplémenter vous-même le code du jeu de la vie.
 - Le code produit devra respecter les règles d’écritures PEP8 et PEP257.

3 Objectifs

Vous allez implémenter le jeu de la vie dans un notebook Jupyter en Python. À partir d’une configuration initiale choisie, l’évolution des différentes cellules sera présentée sous forme d’une animation.

Vous préparerez une douzaine de structures initiales (structures stables, oscillateurs, structures aléatoires, fumeurs...) qui seront testées lors de votre présentation.

Une évolution intéressante du projet sera d’utiliser le module `ipywidget` pour que l’utilisateur puisse choisir dans une liste, le motif de départ et un certains nombre de paramètres de la simulation (taille de la grille, nombre de pas de simulation, couleur des cellules vivantes...). Dans ce cas, il faudra éviter d’utiliser Jupyter lab qui pose parfois quelques difficultés avec ce module.

Un autre axe d’amélioration concerne l’export de la configuration sous forme d’image à chaque pas de simulation puis la génération d’un gif animé à partir des images ainsi exportées...

4 Rendu du projet

Le projet sera à rendre sous la forme d’un fichier .tgz (ou .zip) à déposer sur la plateforme avant la date et l’heure indiquées et comprenant :

- Le code Python développé (script Python et notebook Jupyter).
- Les éventuels fichiers de données utiles pour le fonctionnement du programme.
- Un fichier `README.md`, au format Markdown, qui explique **clairement** comment se lance le programme.
- Les diapos (5 à 10 max, au format PDF) avec les informations suivantes :
 - une intro présentant le projet ;
 - la ou les méthodes employée(s) ;
 - la structure du programme ;
 - les difficultés rencontrées et comment vous les avez résolues ;
 - les résultats obtenues ;
 - les axes d’améliorations ;
 - le mode d’organisation utilisé (qui a fait quoi).

5 Ressources

- Embedding Matplotlib Animations in Jupyter Notebooks :
<http://louistiao.me/posts/notebooks/embedding-matplotlib-animations-in-jupyter-notebooks/>
- Simple animation examples :
https://matplotlib.org/2.1.2/gallery/animation/basic_example.html
- Animation example code : `dynamic_image.py`
https://matplotlib.org/examples/animation/dynamic_image.html
- Python - Plotting colored grid based on values :
<https://stackoverflow.com/questions/43971138/python-plotting-colored-grid-based-on-values>
- Reproducible dashboards and other great things to do with Jupyter, Domino, 29/11/2017
<https://blog.dominodatalab.com/reproducible-dashboards-great-things-jupyter/>