Projet Python ING3 APP

Game of Life

Sommaire

I. Description générale

Le Jeu de la Vie (ou Game of Life), créé par le mathématicien John Conway en 1970, est un célèbre automate cellulaire qui explore la dynamique des systèmes complexes à partir de règles simples. Il se déroule sur une grille bidimensionnelle infinie, où chaque cellule peut être dans l'un des deux états possibles : vivante ou morte. Les règles du jeu sont simples :

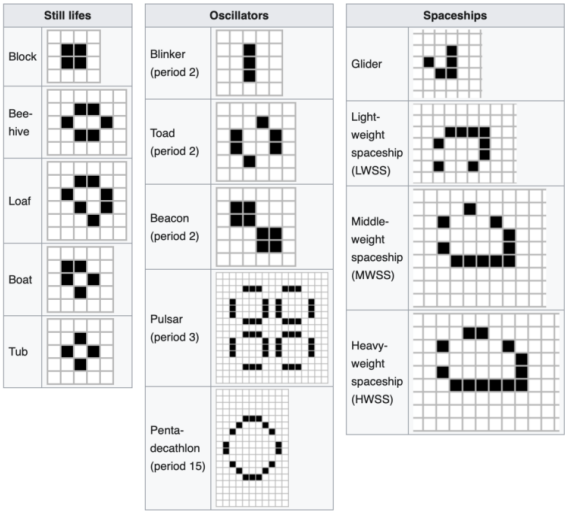
- une cellule vivante survit si elle a 2 ou 3 voisins vivants,
- une cellule vivante meurt de solitude ou de surpopulation.
- une cellule morte peut devenir vivante si elle a exactement 3 voisins vivants.

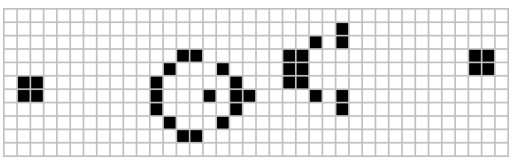
Le jeu évolue de manière itérative en fonction de ces règles, créant de structures plus ou moins complexes comme des oscillateurs, des planeurs et même des structures auto-reproductrices. Bien que le Jeu de la Vie soit déterministe, il est imprévisible à long terme.

Le Jeu de la Vie

Une caractéristique intéressante du Jeu de la Vie est sa capacité à développer des structures de cellules complexes. Celles-ci sont classifiées en quatre types :

- Les **constants** (stille lifes), qui restent en vie indéfiniment sans changer de forme tant qu'elles ne sont pas perturbées par d'autres cellules.
- Les **oscillateurs**, qui évoluent de manière cyclique, reprenant leur forme initiale au bout d'un certain temps, appelé "période".
- Les **vaisseaux**, qui se déplacent dans la grille.
- Les **générateurs**, qui créent d'autres structures.





Exemple d'un générateur: le *Glider gun*

Le but de ce projet est de réaliser un programme permettant d'expérimenter dans le Jeu de la Vie et d'étudier son fonctionnement et ses structures. Pour cela, il faudra réaliser une interface permettant à l'utilisateur de choisir l'état (vivant ou mort) de certaines cellules et de visualiser les résultats. On mettra ensuite en place des outils permettant d'étudier l'évolution des cellules dans le Jeu de la Vie.

II. Cahier des charges

Ce projet a pour but d'utiliser le Python pour réaliser une interface du Jeu de la Vie, étudier son évolution et présenter les résultats. Il se découpe en plusieurs parties centrées sur différentes parties du code. Chaque partie est constituée de fonctionnalités nécessaires qu'il faudra impérativement réaliser, ainsi que de fonctionnalités additionnelles qui permettront d'améliorer la note. Les étudiants sont libres de proposer des fonctionnalités supplémentaires. Le partage égal des tâches entre les membres du groupe est fortement encouragé. Les différentes parties peuvent être réalisées séparément par différents membres du groupe, **exceptée la première** (Moteur de Jeu), qui doit bien être comprise par tous les membres du groupe pour pouvoir réaliser les autres parties.

A. Partie Moteur de Jeu

Pour commencer ce projet sur le Jeu de la Vie, la première étape est de concevoir un ensemble de fonctions et données qui permettront de faire tourner le jeu. Ces éléments de code serviront comme base à tout le reste, ils doivent donc être conçus en amont et de manière à être réutilisés facilement par d'autres parties du programme.

Concrètement, cette partie consiste à définir comment stocker l'état de la grille à un instant *t*, et programmer les règles qui déterminent le changement d'état de *t* à *t+1*.

Fonctionnalités nécessaires

- Une matrice de cellules qui peuvent être soit vivantes, soit mortes. Le moteur de jeu doit pouvoir gérer une matrice de *N*x*N* cellules, *N* étant défini pendant l'utilisation du programme.
- Le code pour gérer les trois règles du jeu qui déterminent l'évolution des cellules.
- Une fonctionnalité permettant d'initialiser une grille aléatoirement.

Fonctionnalités additionnelles

- La possibilité de changer les règles du jeu.
- La possibilité d'avoir une grille de cellules virtuellement infinie avec une matrice dont la taille évolue avec le temps en fonction des cellules qui se trouvent à l'intérieur.
- …

B. Partie Interface

Une fois que le moteur de jeu est programmé, il faut concevoir une interface qui permette à l'utilisateur de choisir l'état de certaines cellules et de visualiser l'évolution des cellules dans le temps. Pour cela, vous pouvez utiliser les outils que vous voulez, tant qu'ils permettent d'avoir les fonctionnalités nécessaires. La note de l'interface dépendra de la simplicité d'utilisation et de l'esthétisme de l'interface.

Fonctionnalités nécessaires

- Un affichage d'une grille de 50x50 cellules minimum, dans la console ou avec une interface graphique (pygame, tkinter, …). L'interface graphique est évidemment recommandée, mais elle sera un peu plus compliquée à mettre en place.
- Un moyen de modifier l'état des cellules dans la grille présentée à l'utilisateur, soit par une entrée clavier, soit avec la souris directement (nécessite une interface graphique).
- La possibilité de voir l'évolution de la grille de cellules dans le temps en avançant à l'étape suivante ou en laissant le jeu se dérouler sans interruption.

Fonctionnalités additionnelles

- La possibilité de zoomer ou dézoomer dans la grille pour voir plus ou moins de cellules.
- La possibilité de se déplacer dans la grille pour n'observer qu'une partie.
- La possibilité de positionner des structures complexes (voir Partie I) à un endroit donné dans la grille.
- …

C. Partie Sauvegarde et Chargement

Pour les besoins de l'étude de l'évolution des cellules, on aura besoin de pouvoir sauvegarder l'état de la grille, ainsi que de charger un état donné de la grille.

Pour cela, il faut définir une technique pour enregistrer une grille et la recharger dans le programme.

Fonctionnalités nécessaires

- Sauvegarder la grille dans un fichier.
- Charger une grille depuis un fichier dans le programme.

Fonctionnalités additionnelles

- …

D. Partie Optimisation

Plus la grille du Jeu de la Vie est grande, plus il y a de cellules à considérer pour passer à l'étape suivante, plus l'exécution du programme sera lente. Il faut donc réfléchir à comment optimiser l'algorithme de calcul de l'état des cellules de la grille. Les algorithmes les mieux optimisés (qui peuvent calculer des grilles plus grandes, plus rapidement) seront valorisés.

Fonctionnalités nécessaires

- Une fonctionnalité permettant de calculer le temps de calcul pour une étape du jeu, **sans compter l'affichage**, sur une grille potentiellement très grande.
- La conception d'un algorithme le mieux optimisé possible pour gérer des grilles de très grandes tailles.

Fonctionnalités additionnelles

- …

E. Partie Présentation et Analyse de données

On souhaite pouvoir étudier l'évolution des cellules dans notre grille. Pour cela, il faudra analyser l'évolution du nombre de cellules dans le temps et tracer des graphes.

Fonctionnalités nécessaires

- Un compte du nombre de cellules vivantes à un instant *t*.
- La génération d'un graphe de l'évolution du nombre des cellules dans le temps.

- La génération d'un graphe montrant le temps de calcul en fonction de la taille de la grille.

Fonctionnalités additionnelles

- La reconnaissance et le compte de structures plus ou moins simples dans la grille.
- La génération de graphes sur les populations de différentes structures présentes dans la grille.
- …

III. Rendus

Les rendus attendus pour ce projet sont :

- Un code propre, bien rangé et commenté. **Le code doit être divisé en modules** (dossiers). Chaque partie présentée précédemment doit avoir son module contenant tout le code correspondant. Le code doit avoir un script qui lance l'interface et permet de jouer au jeu.
- Une **présentation de 15 minutes, sous la forme d'un Jupyter Notebook**. Pendant cette présentation, les élèves présenteront toutes les parties séparément en montrant des exemples de code qui seront expliqués par les étudiants qui ont travaillé dessus. Le Notebook doit contenir des images pour illustrer la présentation, et des lignes de codes permettant de lancer des exemples du jeu de la vie directement depuis le notebook, et des graphes pour analyser l'évolution des cellules.