Jeu de la vie

Groupe :

Moulouel Myriem
Boulard Lilian

Mini-projet : Le Jeu de la Vie

Le jeu de la vie de Conway représente l'évolution d'une population de cellules contenue dans un tableau bidimensionnel. Chaque case du tableau contient 0 ou 1 cellule et on simule l'évolution de la population en divisant le temps en une suite d'instants et en calculant (suivant des règles décrites plus loin) la population à chaque instant.

Règles d'évolution :

Pour savoir l'état d'une case à l'étape n + 1, on regarde son état et celui de ses 8 voisines à l'instant n.

- Si elle est vide et qu'elle a exactement 3 cases voisines occupées, elle devient occupée par une nouvelle cellule. Sinon, elle reste vide.
- Si elle est occupée et qu'elle a précisément 2 ou 3 cases voisines également occupées, la cellule qui occupe la case survit. Sinon la cellule disparaît.

Présentation du projet

Il s'agit de programmer le Jeu De La Vie en python, en utilisant la synchronisation avec des threads.

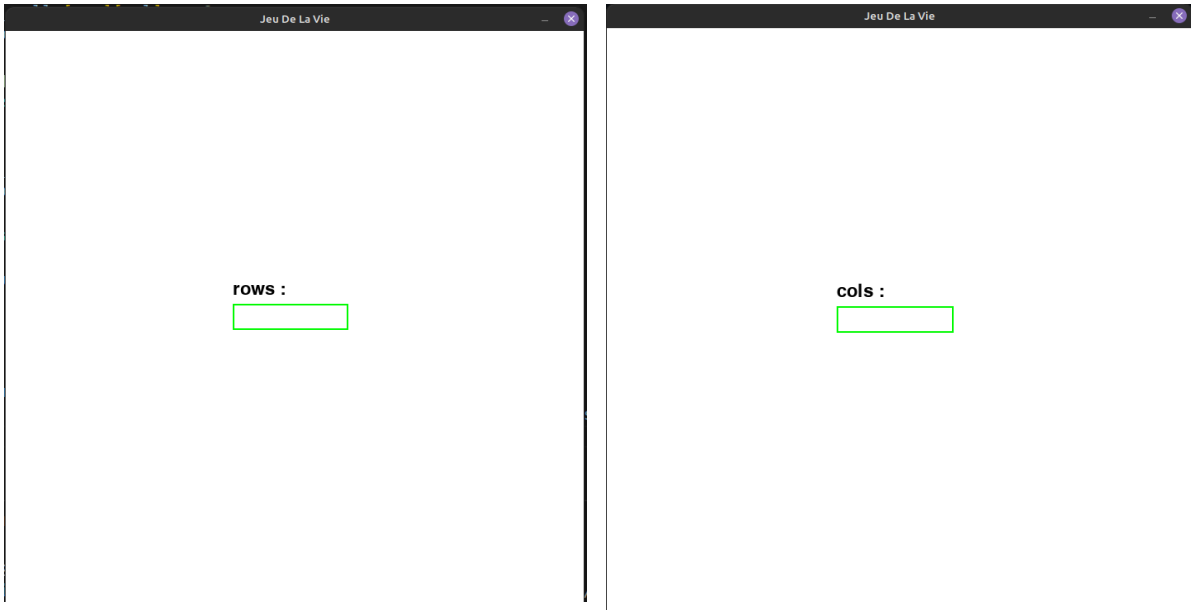
- Première approche : séquentielle, sans synchronisation, dans le fichier V1_main_sequencial.py.
- Deuxième approche : n_row * n_col threads qui calculent les (n_row * n_col) cases de la grille, dans le fichier V2_main_synchro.py.

Lancement du programme

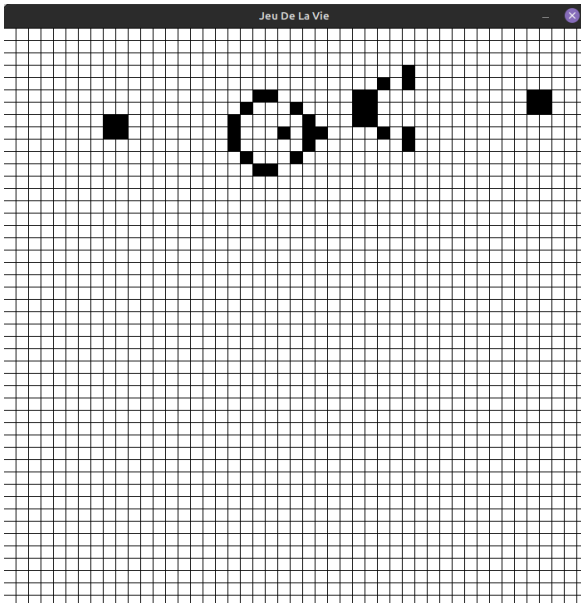
- Télécharger le projet, lancer le fichier main avec une commande python du type : python V2_main_synchro.py pour lancer la version synchronisée et python V1_main_sequencial.py pour lancer la version séquentielle.
- S'assurer d'avoir les bibliothèques python requises comme pygame, threading, time.

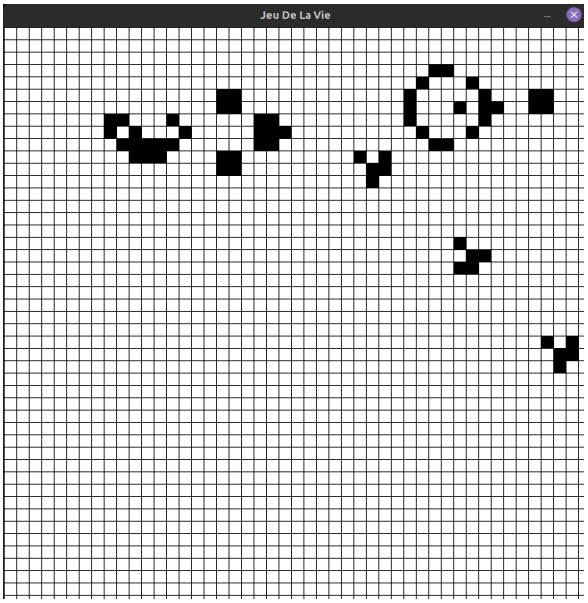
Si elle n'est pas déjà installée pygame : https://www.pygame.org/wiki/GettingStarted

- Rentrer le nombre de lignes et de colonnes que vous désirez



- Sélectionnez les cellules vivantes initiales de votre choix pour commencer le jeu avec le clic-gauche de la souris, désélectionnez avec le clic-droit puis Entrer
- Observez le jeu !





- Pour arrêter le jeu, fermez la fenêtre active.

La partie synchronisée :

On a créé (nb lignes * nb colonnes) de threads qui s'exécutent à l'infini en calculant le nombre de voisins de chaque cellule, puis la valeur de la cellule à l'étape suivante.

Pour s'assurer que le calcul du nombre de tous les voisins soit effectué avant d'entamer le changement de valeur aux cellules de la grille, on utilise deux barrières de synchronisation, l'une va s'assurer que le calcul du nombre de voisins est terminé et l'autre que le changement de cellule est terminé.

Variables globales :

Écrites dans le fichier Ressources/variables.py

- La largeur et la hauteur de la grille :

Width, Height = 700, 700

- Les couleurs utilisées pour l'affichage des cases de la grille :

White = (255, 255, 255), Black = (0, 0, 0)

- Variable utilisée dans la version synchronisée pour ralentir l'affichage : REFRESH_RATE = 0.25