

Projet : Le Jeu de la Vie

Votre Nom

7 août 2024

Introduction

- Le jeu de la vie (jdlv) se représente sur une grille 2D (de taille théoriquement infinie) composée de cases carrées appelées cellules.
 - Chaque cellule a un état binaire : 1 pour vivante, 0 pour morte.
- Le jeu de la vie est un automate cellulaire basé sur l'évolution de la grille dans le temps.
 - Chaque étape de l'évolution est appelée génération.
 - Les cellules évoluent en fonction de leur voisinage (chaque cellule a 8 cellules voisines).
- Règles de survie et de mort pour une cellule vivante :
 - Si une cellule vivante a 0 ou 1 voisin vivant, elle meurt à l'évolution suivante (mort par sous-population).
 - Si une cellule vivante a 2 ou 3 voisins vivants, elle reste en vie.
 - Si une cellule vivante a 4 voisins ou plus, elle meurt à l'évolution suivante (mort par sur-population).
- Règle de naissance pour une cellule morte :
 - Si une cellule morte est entourée de 3 cellules vivantes, elle devient vivante (naît) à la prochaine évolution (naissance par reproduction).

Étape 1 : Génération aléatoire des cellules

Réalisez une fonction qui génère 33 cellules placées aléatoirement, sur une matrice de 10 sur 10.

Étape 2 : Application des règles du jeu de la vie

Réalisez un programme qui applique les règles du jeu de la vie à chaque génération. Limitez le nombre de générations à 100 pour éviter une boucle infinie.

Décomposition du projet

Voici les étapes du projet pour le Jeu de la Vie :

1. **Génération de la grille initiale** : Créez une matrice 10x10 et placez aléatoirement 33 cellules vivantes.
2. **Fonction de comptage des voisins** : Implémentez une fonction pour compter les voisins vivants de chaque cellule.
3. **Évolution de la grille** : Implémentez une fonction pour appliquer les règles du jeu et produire la grille de la génération suivante.
4. **Simulation du jeu** : Créez une boucle pour faire évoluer la grille à chaque génération et affichez la grille.
5. **Limitation des générations** : Assurez-vous que la simulation ne dépasse pas 100 générations pour éviter une boucle infinie.

Composition du projet

Pour structurer le projet, nous utiliserons des classes et des fonctions :

— **Classes**

— `GameOfLife`

— `__init__(self, rows, cols, num_cells)` : Initialise la grille.

— **Méthodes**

— `generate_random_cells(rows, cols, num_cells)` : Génère une grille avec des cellules vivantes placées aléatoirement.

— `count_neighbors(grid, x, y)` : Compte les voisins vivants pour une cellule donnée.

— `evolve(grid)` : Applique les règles du jeu de la vie pour faire évoluer la grille.

— `game_of_life(initial_grid, max_generations)` : Gère la simulation du jeu de la vie.