

# CODING CLUB

EPITECH




## Subject

# Le Jeu de la Vie des Citrouilles

## Objectifs de l'atelier

Bienvenue dans cet atelier où vous allez découvrir la programmation en créant votre propre simulation de "Jeu de la Vie"!

## Qu'est-ce que le Jeu de la Vie ?

Le **Jeu de la Vie** est un "automate cellulaire" inventé par le mathématicien John Conway en 1970. Ce n'est pas vraiment un jeu au sens classique : c'est une simulation où des cellules (ici, des citrouilles ) vivent, meurent et naissent selon des règles très simples.

Malgré leur simplicité, ces règles produisent des comportements fascinants et complexes !

---

# Les Règles du Jeu

Chaque citrouille sur la grille peut être **vivante** () ou **morte** (case vide).

À chaque génération, on applique ces 4 règles :

## 1. Mort par solitude

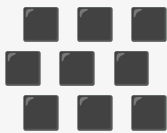
Une citrouille vivante avec **moins de 2 voisines** meurt.

Génération 1:



→

Génération 2:



(0 voisins = mort)

## 2. Survie

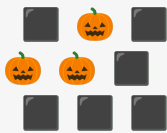
Une citrouille vivante avec **2 ou 3 voisines** survit.

Génération 1:



→

Génération 2:



(2 voisins = survie)

## 3. Mort par surpopulation

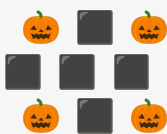
Une citrouille vivante avec **plus de 3 voisines** meurt.

Génération 1:



→

Génération 2:



(8 voisins = mort)

## 4. 🌱 Naissance

Une case vide avec **exactement 3 voisins** donne naissance à une nouvelle citrouille.

Génération 1:



→

Génération 2:



(3 voisins = naissance)



# Votre Mission

Vous devez compléter la fonction `calculer_prochaine_generation()` dans le fichier `main.py`.

## Fonctions utiles disponibles

Le fichier `wrapper.py` contient une fonctions pour vous aider :

`compter_voisins(grille, ligne, colonne)`

Compte automatiquement le nombre de citrouilles voisines autour d'une case.

### Exemple :

```
grille = [[0, 1, 0],
          [1, 0, 1],
          [0, 1, 0]]

voisins = compter_voisins(grille, 1, 1) # Retourne 4
voisins = compter_voisins(grille, 0, 0) # Retourne 2
```

## Comprendre la grille

La grille est une **liste de listes** : - `0` = case vide - `1` = citrouille vivante

```
grille = [
    [0, 0, 0],
    [1, 1, 1], # 3 citrouilles en ligne
    [0, 0, 0]
]
```



# Comment Tester Votre Code

## 1. Lancer le programme

```
python main.py
```

Ou simplement utiliser la flèche de lancement

## 2. Créer des motifs

Cliquez sur la grille pour placer des citrouilles et créer des motifs.

## 3. Démarrer la simulation

Cliquez sur le bouton "► Démarrer" pour voir votre code en action !

---

# Motifs Intéressants à Tester

## Le Clignotant (Blinker)

Trois citrouilles en ligne qui alternent entre horizontal et vertical.

Génération 1:



Génération 2:



## Le Bloc (Block)

Un carré de 4 citrouilles qui reste stable indéfiniment.



(Ne change jamais)

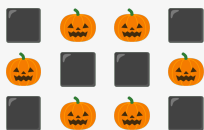
## Le Planeur (Glider)

Un motif qui se déplace diagonalement à travers la grille !



## Le Ruche (Beehive)

Une forme hexagonale stable.





# Défis Bonus

Une fois votre code fonctionnel, essayez ces défis :

## Défi 1 : Compteur de citrouilles

Ajoutez un compteur qui affiche le nombre total de citrouilles vivantes.

## Défi 2 : Modification des règles

Créez une variante avec des règles différentes. Par exemple : - Survie avec 3 à 5 voisins au lieu de 2 à 3 - Naissance avec 2 ou 3 voisins au lieu de 3 uniquement

## Défi 3 : Grille torique

Faites en sorte que les bords de la grille soient connectés (comme dans Pac-Man).

## Défi 4 : refaire la fonction qui compte les proches voisins

## Défi 5 : Créer un canon à planeurs

Recherchez le "Gosper Glider Gun" et essayez de le reproduire !



# Bon Codage et Joyeux Halloween !

N'hésitez pas à expérimenter, à casser votre code, et à le réparer. C'est comme ça qu'on apprend ! "*Les seules vraies erreurs sont celles dont nous ne tirons rien.*" - Henry Ford