

# PROJET : Jeu de la vie

LEVREAU Ghislain  
Polytech Marseille 4A Info

## Spécifications

MISE A JOUR		
DATES	LIBELLES	Version
14/12/2022	Création du Projet	V0.1
05/01/2023	Avancements majeurs	V0.2
12/01/2023	Ajout de chaque spéc	V0.3
29/01/2023	Finissions avant rendu	V0.4
05/01/2023	Relecture et dernières mises au point	V0.5
06/01/2023	Remise en page totale	V0.6

# Table des matières

<i>Table des matières</i>	1
<b>I. Introduction</b>	2
1) Objectif du projet	2
2) Contexte	2
3) Règles du jeu	3
<b>II. La mission</b>	4
1) Technologies	4
2) Caractéristiques fonctionnelles	4
3) Qualité attendue	6
<b>III. Spécifications</b>	7
1) Avoir des explications	8
2) Afficher structures connues	9
3) Choix des cellules	10
4) Lancer le jeu	11
5) Avancer étape par étape	12
6) Choisir la vitesse	13
7) Stopper la simulation	14
8) Réinitialiser le plateau	15
9) Diagramme de classe	16

---

# I. Introduction

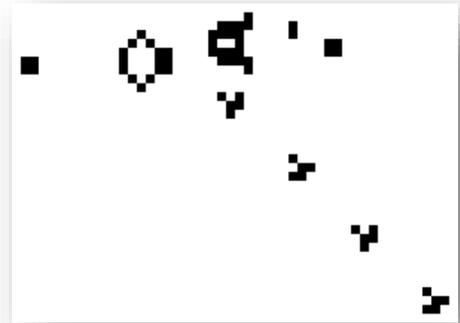
## 1) Objectif du projet

Au cours du semestre 7 de notre formation de 3 ans au sein de Polytech Marseille dans le domaine Informatique, nous devons réaliser un Projet. Nous sommes donc 2 sur le projet et devons gérer toutes les parties :

- Initialisation
- Lancement
- Qualité
- Développement
- Tests

Pour cela nous seront encadrés par un professeur responsable du projet en question, pour nous Sébastien Mavromatis sera notre encadrant.

Le but de notre projet est de programmer [Le jeu de la vie](#).



## 2) Contexte

Le jeu de la vie est un automate cellulaire imaginé par John Horton Conway en 1970. Le jeu de la vie est un outil de simulation de la vie artificielle basé sur des règles mathématiques simples, ce qui en fait un sujet de recherche important pour les mathématiques, la biologie et les sciences informatiques. C'est un « jeu à zéro joueur », puisqu'il ne nécessite aucune intervention du joueur lors de son déroulement.

Le jeu de la vie a un impact sur les mathématiques, la biologie, la philosophie et les sciences informatiques. Il a une signification profonde en raison de sa capacité à simuler la vie artificielle et à explorer les concepts liés à la vie, tels que la naissance, la mort, la reproduction et l'évolution. Cela a permis de nouvelles découvertes dans ces domaines, ce qui en fait un outil précieux pour les scientifiques et les chercheurs.

Le jeu de la vie est souvent utilisé pour simuler des systèmes complexes, tels que les réseaux de communication, les systèmes de transport et les systèmes biologiques. Il peut également être utilisé pour explorer des concepts liés à la vie artificielle, tels que la création de vie à partir de rien et l'évolution de la vie au fil du temps.

### 3) Règles du jeu

Le jeu de la vie se présente sous la forme d'un quadrillage, plus ou moins grand, dont l'état de chaque case à un instant  $n$  est déterminé par des règles et par l'état des cases qui l'entourent, à l'instant  $n - 1$ .

On parle de cellule à la place de case. Une cellule est soit vivante soit morte, suivant ses 8 voisins à l'étape d'avant.

Les 5 situations possibles d'une cellule dans le Jeu de la Vie sont :

- **Naissance** : Si une cellule a exactement trois voisines vivantes, elle naît et devient vivante.
- **Survie** : Si une cellule a deux ou trois voisines vivantes, elle survivra au tour suivant.
- **Solitude** : Si une cellule a moins de deux voisines vivantes, elle meurt de solitude.
- **Surpopulation** : Si une cellule a plus de trois voisines vivantes, elle meurt de surpopulation.
- **Mort** : Si une cellule n'a ni trois voisines vivantes pour naître, ni deux ou trois voisines vivantes pour survivre, elle restera morte.

Au cours de la simulation, ces règles sont appliquées à chaque cellule du plateau, et le jeu continue jusqu'à ce que toutes les cellules soient mortes ou qu'une situation stable se soit établie.

En ce qui concerne la taille du plateau, il n'y a pas de limites strictes quant à sa taille. Cependant, il est important de choisir une taille qui convienne à l'expérience de jeu souhaitée et qui soit gérable pour les algorithmes utilisés pour simuler le jeu. De plus, il est également important de prendre en compte les limites techniques du système sur lequel le jeu sera joué (par exemple, les ressources de traitement et de mémoire disponibles).

Il est important de noter que le Jeu de la Vie est un système de simulation déterministe, ce qui signifie que le même jeu avec les mêmes conditions de départ produira toujours les mêmes résultats.

# II. La mission

## 1) Technologies

Nous avons décidé de faire 3 versions de notre Jeu de la vie avec une certaine évolution.

- Une **version Console** qui utilisera le langage de programmation Python. Python est un langage de programmation populaire pour les projets de science des données et les simulations, ce qui en fait un choix logique pour le jeu de la vie. D'autres technologies telles que les bibliothèques de simulation pourraient également être utilisées.
- Une **version Web** avec HTML, CSS et JavaScript. HTML sera utilisé pour structurer le contenu de la page, CSS pour le styliser et JavaScript pour apporter de la fonctionnalité au jeu. Cette version apportera une interface pour l'utilisateur ainsi que des boutons, des couleurs ...
- Une **version Unity** avec le langage de programmation C#. Unity est un moteur de jeu populaire qui permet de développer des jeux en 2D ou 3D pour différentes plateformes. C# est un langage de programmation orienté objet qui sera utilisé pour écrire du code pour la logique du jeu. Des bibliothèques telles que Unity UI pourraient également être utilisées pour améliorer l'interface utilisateur.

Chaque support peut offrir une expérience utilisateur différente. Par exemple, la version Unity peut offrir une expérience plus immersive avec des graphismes en 3D, tandis que la version console peut être plus simple et plus rapide. De plus : en développant plusieurs versions d'un même jeu, il est possible de tester différentes idées et de développer différentes fonctionnalités pour voir lesquelles fonctionnent le mieux. Cela peut permettre d'améliorer constamment le jeu et de l'adapter aux préférences et aux besoins des utilisateurs.

## 2) Caractéristiques fonctionnelles

Nous allons donc devoir programmer Le jeu de la vie, pour cela tous les outils sont disponibles. Le monde sera modélisé par un tableau rectangulaire de dimensions HEIGHT sur WIDTH, dans lequel les cases auront soit la valeur 0 (cellule vide, pas de bactérie) soit la valeur 1 (cellule occupée par une bactérie). La valeur d'une case devra changer au fil des itérations en fonction de l'état de ses premières voisines et des règles de vie et de mort

On aura des fonctionnalités générales comme :

- **Simulation du Jeu de la Vie** : La capacité de simuler correctement le Jeu de la Vie en utilisant les règles décrites précédemment.
- **Personnalisation des règles** : La possibilité de personnaliser les règles du jeu pour des simulations plus intéressantes ou pour explorer différents comportements.

- **Interface utilisateur** : Une interface conviviale pour permettre aux utilisateurs de contrôler la simulation, de la personnaliser et de visualiser les résultats.
- **Contrôle du jeu de la simulation** : La possibilité de contrôler le jeu, start and stop.
- **Statistiques et visualisations** : La possibilité de visualiser les statistiques sur le nombre de cellules vivantes, le nombre de générations.
- **Génération de la soupe primitive** : Le plateau sera généré (obligatoirement ou non) de façon aléatoire suivant une densité.

Puis chacune de nos 3 versions auront leurs fonctionnalités propres.

#### **a) Format console**

- **Entrée utilisateur en ligne de commande** : Une interface utilisateur basée sur la ligne de commande pour permettre aux utilisateurs de contrôler la simulation. Il pourra rentrer la taille du plateau et la densité de cellules vivantes.
- **Contrôle du jeu** : Il pourra choisir aussi de continuer ou pas la simulation, et donc de passer à l'itération suivante.
- **Affichage en console** : Une visualisation en console des résultats de la simulation pour permettre aux utilisateurs de suivre l'évolution du Jeu de la Vie.
- **Fonctionnalités minimales** : Un jeu de fonctionnalités minimales pour permettre aux utilisateurs d'effectuer des simulations rapides et simples

#### **b) Format Web**

- **Interface utilisateur graphique** : Une interface utilisateur graphique pour permettre aux utilisateurs de contrôler la simulation en utilisant des boutons, des menus déroulants et autres éléments graphiques.
- **Affichage en HTML/CSS** : Une visualisation en HTML/CSS des résultats de la simulation pour permettre aux utilisateurs de suivre l'évolution du Jeu de la Vie.
- **Interaction utilisateur** : La possibilité pour les utilisateurs d'interagir avec le plateau en cliquant sur les cellules pour les activer ou les désactiver.

#### **c) Format Unity**

- **Interface utilisateur graphique avancée** : le plateau de jeu sera affiché en 3D avec des cellules vivantes représentées par des objets 3D tels que des cubes ou des sphères.
- **Interaction utilisateur** : l'utilisateur pourra interagir avec des boutons sur l'écran pour Jouer, mettre en pause, reset, random, accélérer le jeu, agrandir le plateau.
- **Animation du jeu** : le jeu pourra être animé pour montrer l'évolution des cellules au fil des générations.
- **Effets visuels et sonores** : des effets visuels et sonores peuvent être ajoutés pour renforcer l'expérience utilisateur, tels que des explosions pour les cellules qui meurent ou des sons pour les cellules qui naissent.

### 3) Qualité attendue

Durant le développement du jeu de la vie nous devrions respecter la qualité sur plusieurs critères.

**Performance** : Le jeu de la vie doit fonctionner de manière fluide et sans temps de chargement excessif, même sur des ordinateurs ou des appareils moins performants. La vitesse de la simulation doit être ajustable grâce à une barre de défilement, et les transitions doivent être fluides.

**Fiabilité** : Le jeu de la vie ne doit pas s'arrêter de fonctionner de manière inattendue et doit pouvoir gérer les erreurs de manière appropriée. Les boutons et les fonctionnalités du jeu doivent être fiables et ne pas se bloquer ou se déconnecter.

**Sécurité** : Les données sensibles telles que les informations utilisateur ou les informations de sauvegarde doivent être cryptées et protégées contre les accès non autorisés. Les données sensibles ne doivent pas être stockées sur le disque dur de l'utilisateur sans protection.

**Compatibilité** : Le jeu de la vie doit être compatible avec les principales plates-formes, telles que Windows, MacOS, iOS et Android. Il doit également être compatible avec les navigateurs web populaires tels que Google Chrome, Mozilla Firefox et Apple Safari.

De façon plus détaillée, lors du développement et de nos tests, nous vérifierons ces points essentiels :

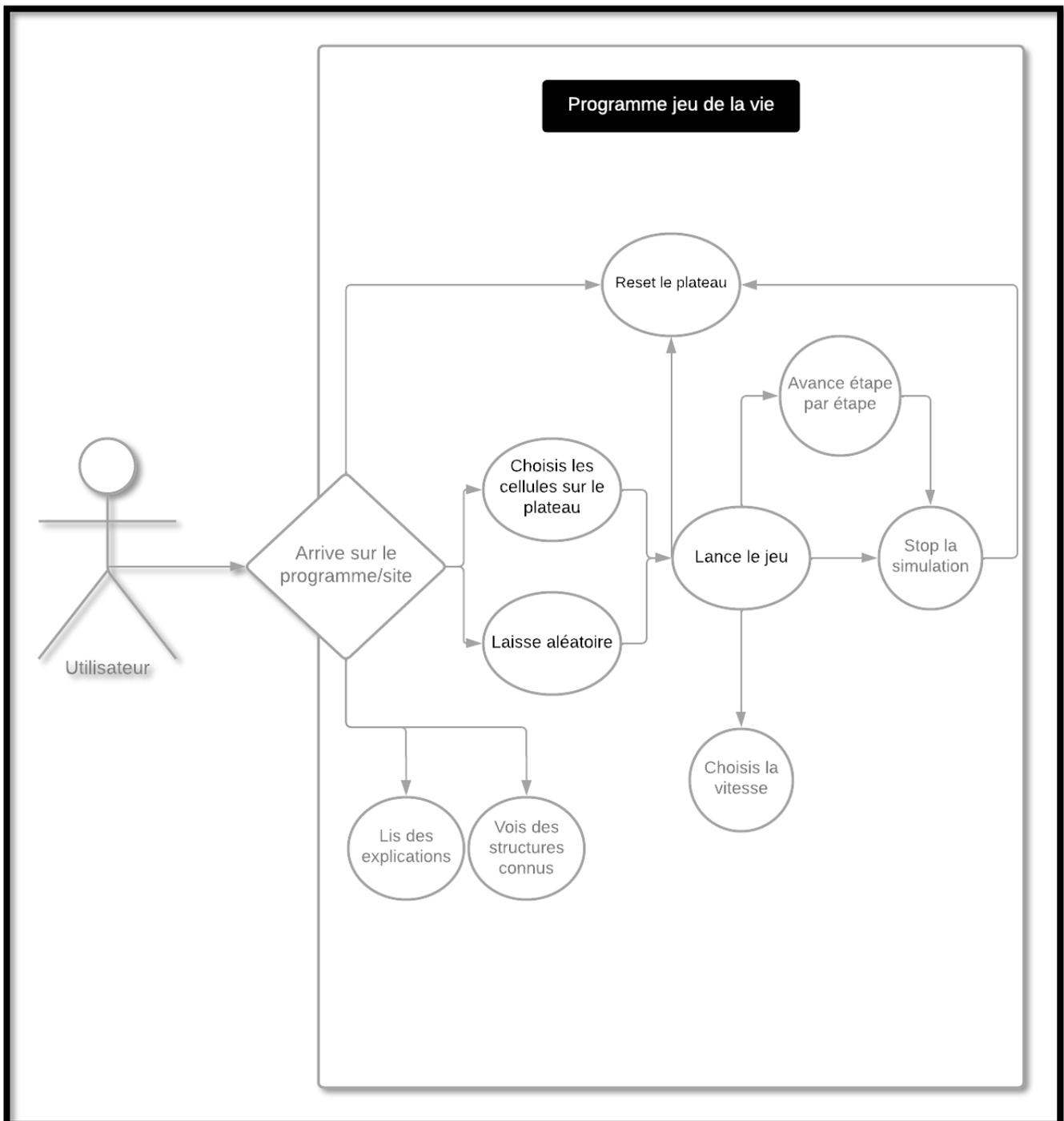
- **Conformité aux règles du jeu** : Les 5 situations possibles d'une cellule doivent être respectées et implémentées correctement pour garantir l'exactitude du jeu.
- **Plateau de jeu** : La taille du plateau de jeu doit être suffisamment grande pour permettre une observation détaillée des résultats, tout en restant pratique pour l'utilisateur.
- **Interaction utilisateur** : La possibilité de cliquer sur le plateau pour rendre une cellule vivante ou pas doit être fluide et intuitive. Les boutons Start/Stop, Reset, et Random doivent être facilement accessibles et clairement identifiés.
- **Compteur de génération et de cellules vivantes** : Les informations sur le nombre de générations et de cellules vivantes doivent être clairement affichées et mises à jour en temps réel.
- **Scroll bar** : La scroll bar qui permet de choisir la vitesse du jeu doit fonctionner de manière fluide et stable.
- **Performance et rapidité** : Le jeu doit être rapide et performant, même pour des plateaux de jeu de grande taille.
- **Esthétique** : La qualité graphique et la présentation générale doivent être soignées pour garantir une expérience utilisateur agréable.
- **Ergonomie** : L'interface utilisateur doit être intuitive et facile à utiliser pour tous les utilisateurs, sans nécessiter de formation préalable.
- **Robustesse** : Le jeu doit être robuste et stable, sans plantage ni erreur d'exécution.

Ces critères de qualité doivent être pris en compte dans la conception, le développement et la livraison du produit final pour garantir une expérience de jeu de qualité pour les utilisateurs.

# III. Spécifications

Le but des spécifications est d'avoir une description exhaustive des fonctionnalités prises sous différents angles.

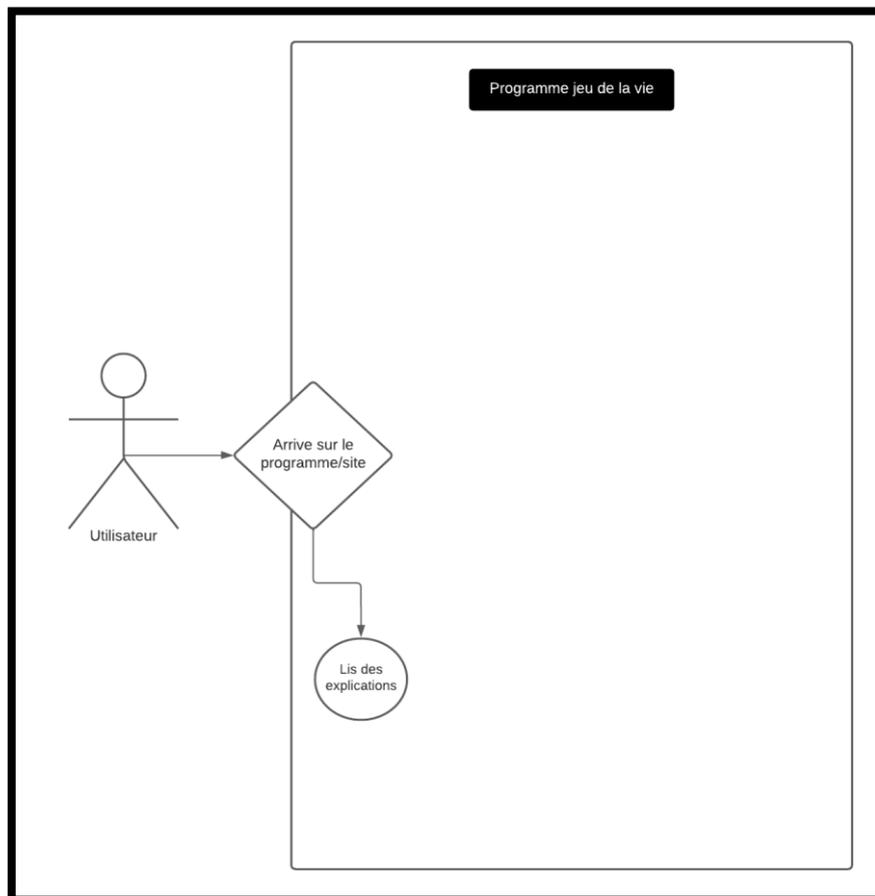
Voici le diagramme des « use cases » incluant tous les use cases important de notre système.



## 1) Avoir des explications

### a) Description

L'utilisateur pourra cliquer sur un bouton pour faire apparaître une page flottante, un widget avec quelques informations et explications sur le but du jeu, les règles, les possibilités...



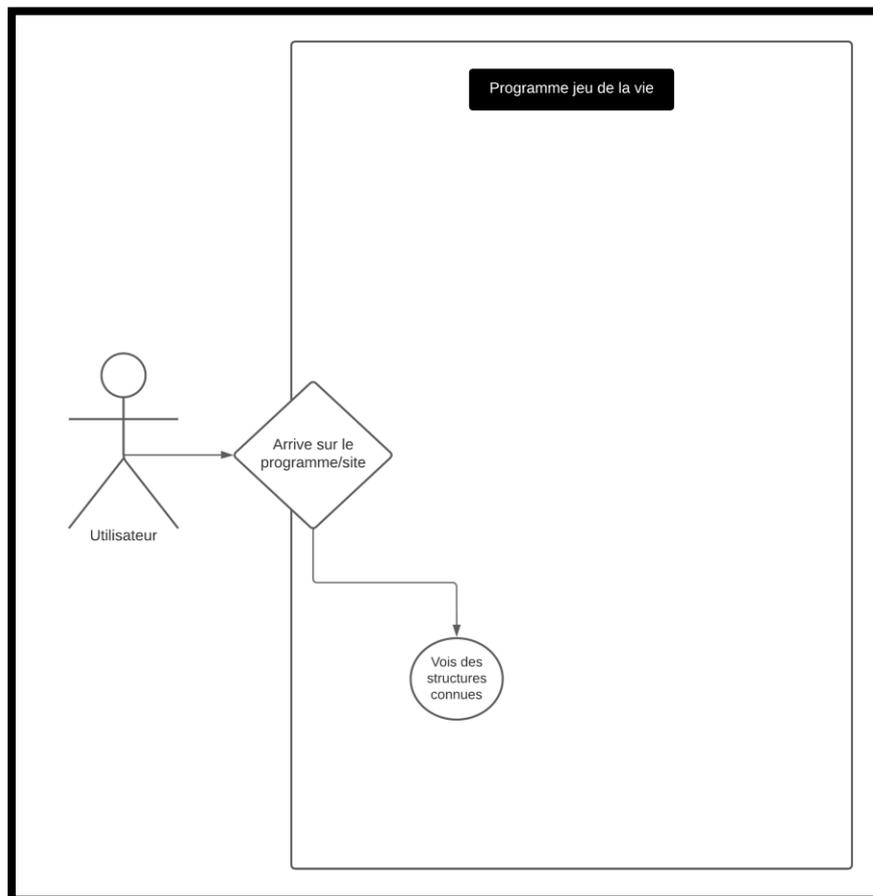
### b) Scénario

L'utilisateur arrive sur le site, il est un peu perdu, il ne connaît pas vraiment le jeu. Il voit alors un bouton « **Informations** ». Il clique dessus, une page apparait et il peut lire le but du jeu. Ensuite il ferme cette page, pour pouvoir jouer au jeu avec toutes les informations nécessaires en tête.

## 2) Afficher structures connues

### a) Description

L'utilisateur après avoir un peu essayer le jeu, verra des structures intéressantes, qui persistent dans le temps, qui créées d'autres structures ... Alors il pourra se demander s'il existe des structures connues, intéressantes à observer. Un bouton sera disponible pour afficher une page flottante qui lui montrera des structures intéressantes.



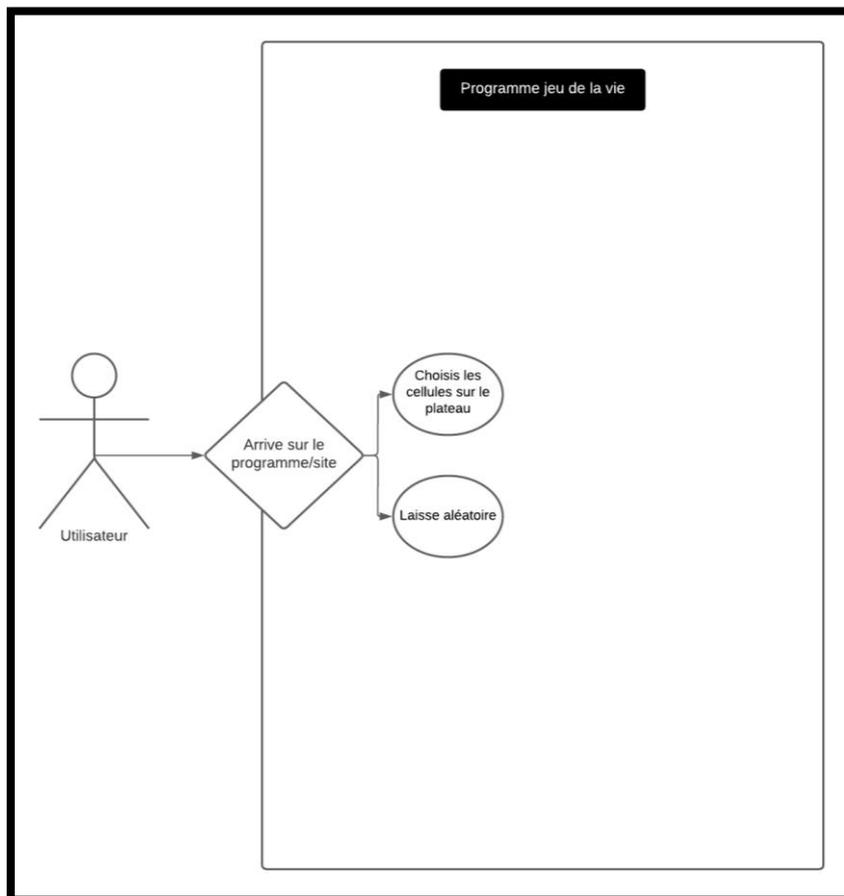
### b) Scénario

L'utilisateur à bien essayé le jeu, et il aimerait découvrir des structures qui font des situations intéressantes. Il clique alors sur la page flottante et navigue dans les différentes structures, il en voit une qui l'intéresse, il peut alors la reproduire.

### 3) Choix des cellules

#### a) Description

L'utilisateur une fois sur le site, aura la possibilité de choisir les cellules qu'il veut rendre vivante. Pour cela il pourra soit cliquer sur les cellules de son choix, ce qui changera leur couleur et donc les rendra vivantes. Ou soit il pourra cliquer sur un bouton « *aléatoire* » qui remplira le plateau de cellules vivantes de façon aléatoire.



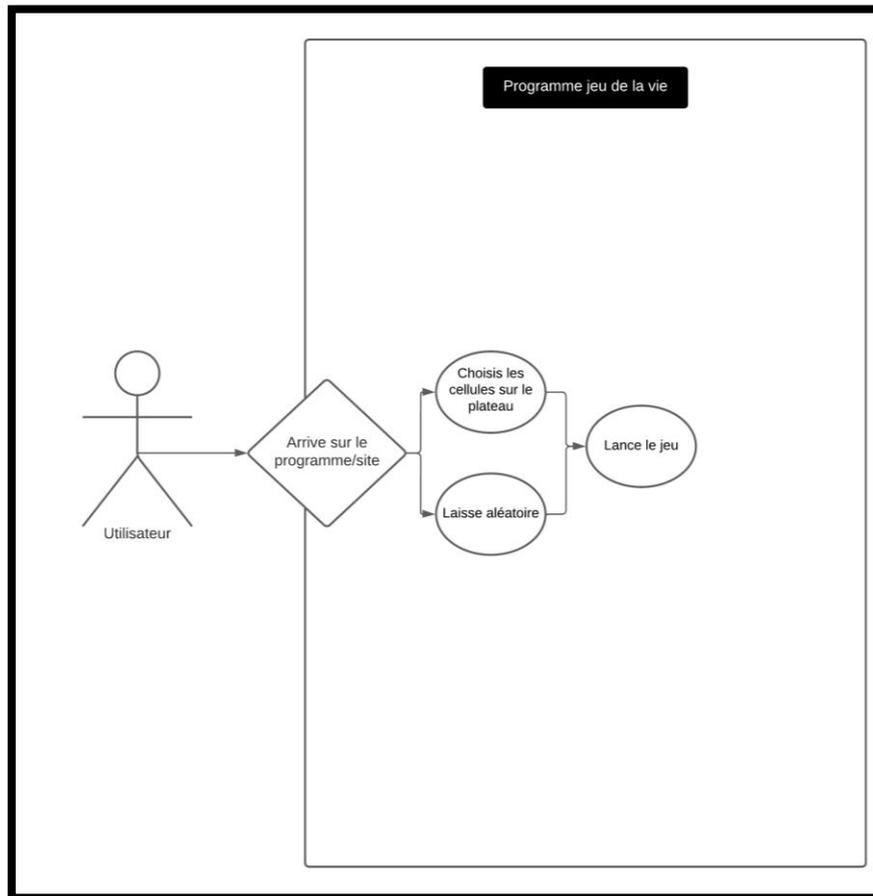
#### b) Scénario

L'utilisateur vient sur le site de notre jeu, il veut tester le fonctionnement alors il choisit aléatoirement le placement pour essayer une première fois. Ensuite, il veut lui-même placer ses cellules, il clique sur le plateau cellule par cellule pour mettre en place une structure intéressante.

## 4) Lancer le jeu

### a) Description

L'utilisateur clique sur le bouton « Play » qui lance le jeu. Le plateau est mis à jour à chaque update suivant la vitesse.



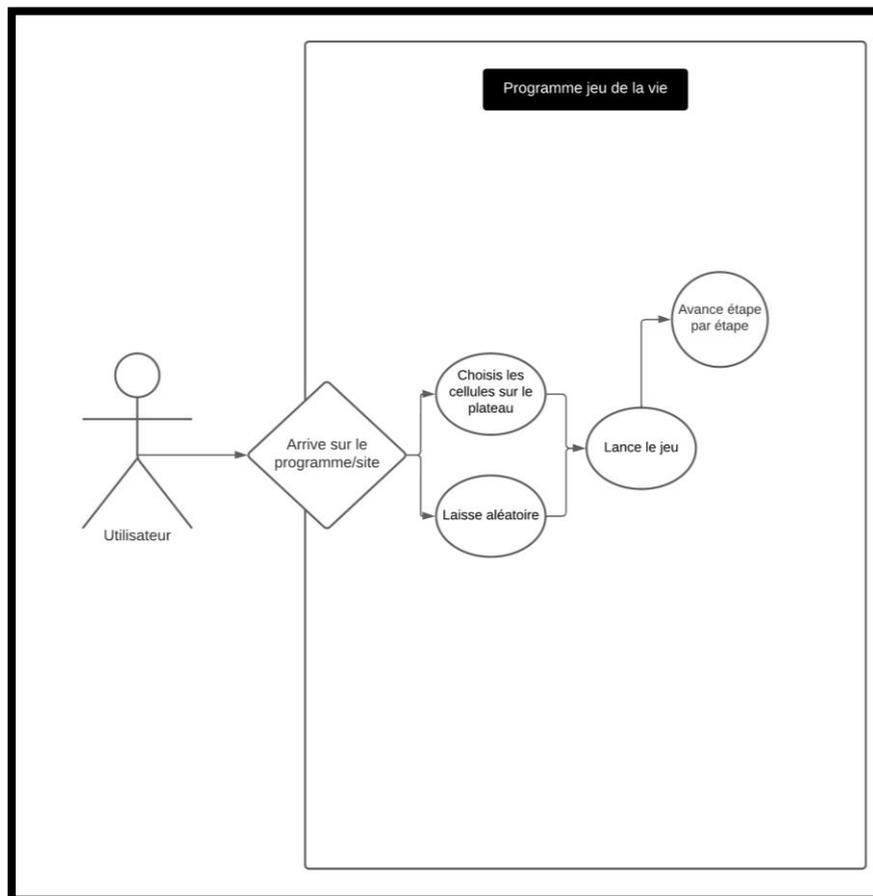
### b) Scénario

L'utilisateur a placé les cellules qu'il voulait sur le plateau, il lance alors le jeu pour voir les générations futures. Un compteur de générations et de cellules vivantes sont disponibles pour lui communiquer des informations supplémentaires.

## 5) Avancer étape par étape

### a) Description

L'utilisateur clique sur un bouton qui permet au jeu de se placer en mode « step by step », le plateau est mis à jour seulement quand l'utilisateur le veut.



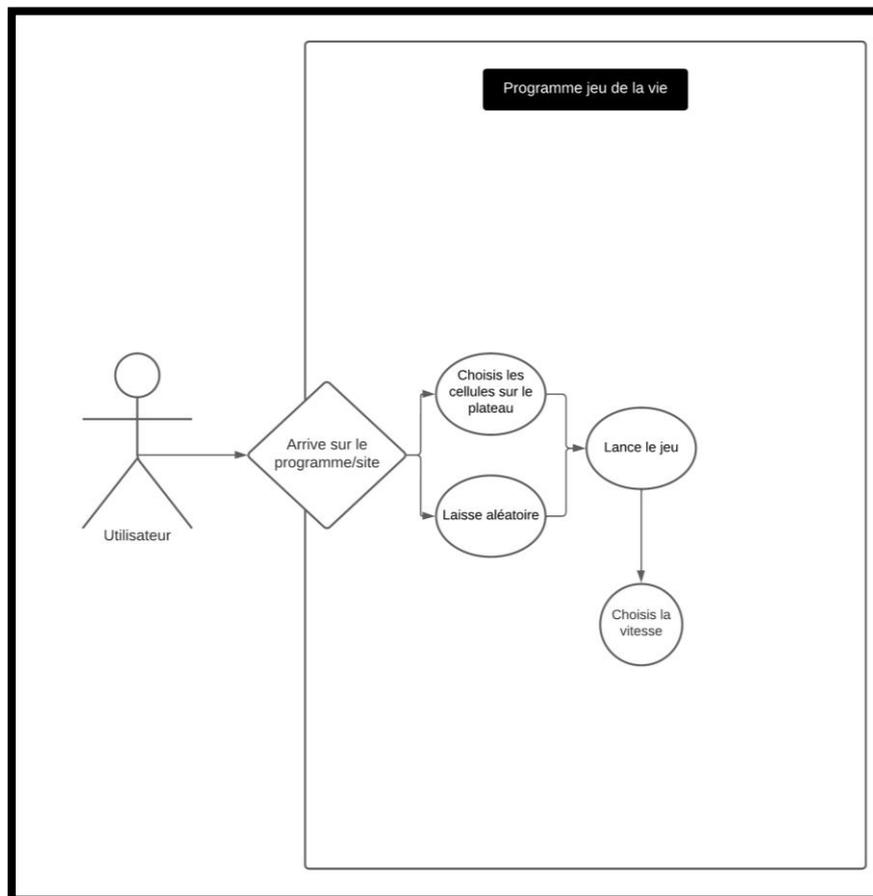
### b) Scénario

L'utilisateur veut regarder chaque étape du jeu pour bien observer ce qui se passe, il se place alors en mode étape par étape. Le jeu s'actualise seulement aux cliques de l'utilisateur.

## 6) Choisir la vitesse

### a) Description

L'utilisateur choisit la vitesse du jeu à l'aide d'une scroll bar.



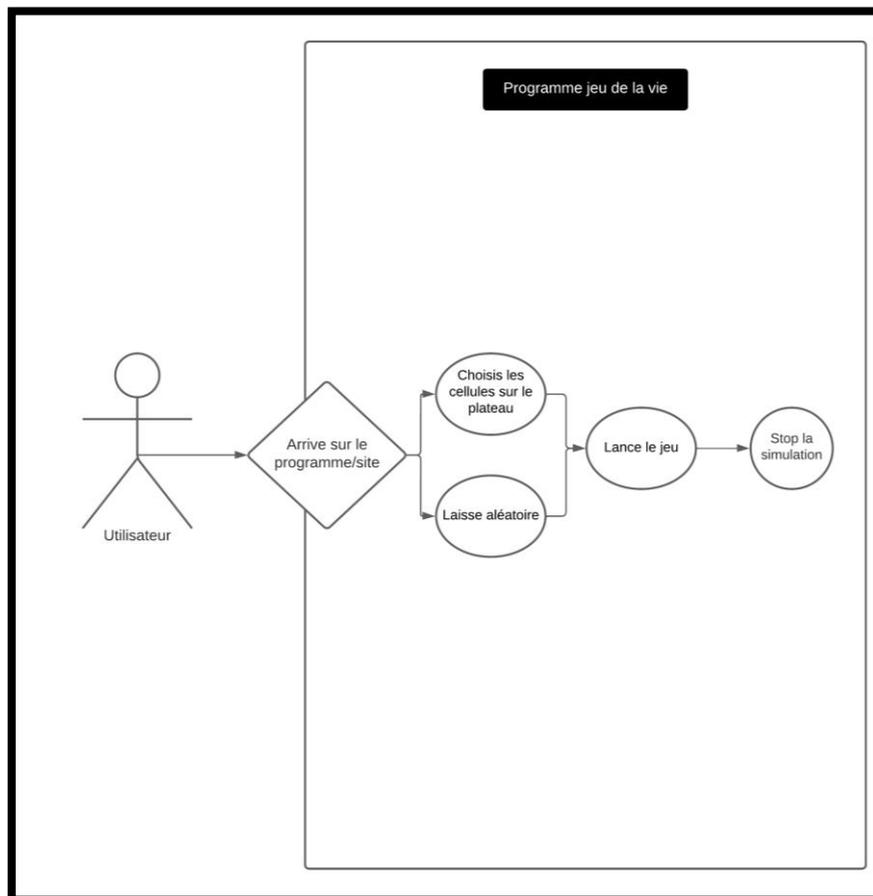
### b) Scénario

L'utilisateur veut observer le jeu et l'analyser, il diminue alors la scroll bar pour diminuer la vitesse du jeu. Il veut ensuite voir seulement si une structure survit dans le temps, il augmente au maximum la vitesse du jeu et voit si en plusieurs centaines de générations, il y a encore des cellules.

## 7) Stopper la simulation

### a) Description

L'utilisateur clique sur un bouton « Pause » pour arrêter la simulation. Il n'y a plus d'update, le plateau reste le même. Le jeu peut être relancé à tout moment avec le bouton « Play ».



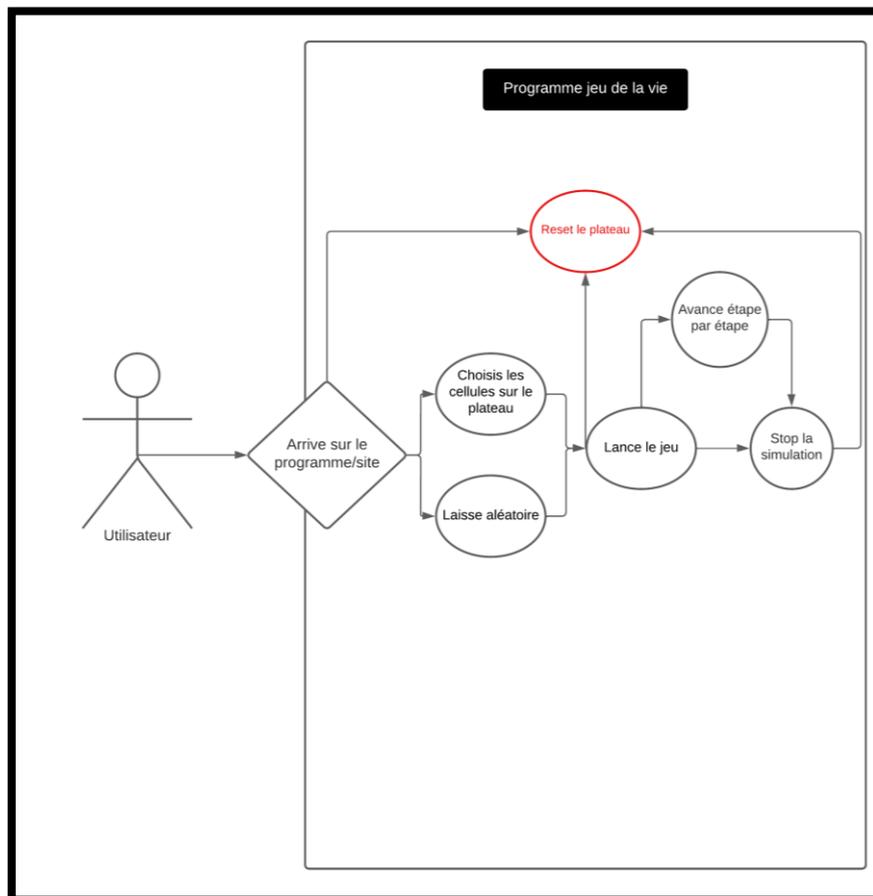
### b) Scénario

L'utilisateur veut voir une génération en particulier, il peut alors mettre le jeu en pause. Il peut le relancer quand il veut.

## 8) Réinitialiser le plateau

### c) Description

L'utilisateur peut cliquer sur un bouton « reset » qui réinitialisera le plateau, c'est-à-dire que le plateau sera remis à zéro. Toutes les cellules redeviendront « mortes ».



### d) Scénario

L'utilisateur vient de faire une simulation et veut relancer une nouvelle, il clique alors sur « reset » ce qui rendra le plateau vierge. Il peut alors remettre les cellules qu'il veut.

## 9) Diagramme de classe



Ce diagramme de classe nous montre les principales classes du projet, avec pour chacune les fonctionnalités les plus importantes.