SAE S2.01 – Développement d'une application
Compte rendu

Tuteuré par : - BARREPERSYN Sylvain
- SERVIERES Corinne

IUT de Rodez / 2021 – 2022 / 1ère année
BUT Informatique

ALZURIA Mathys, BOIS Axel, CHEBANA Mohammed

Sommaire

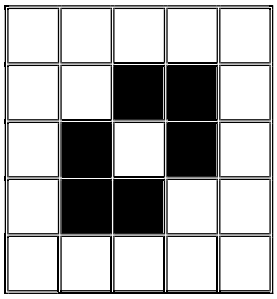
Table des matières

Automate cellulaire : Le Jeu de la Vie

Règles de base

Le Jeu de la Vie est un programme informatique imaginé dans les années 70 par le mathématicien John Conway. Le Jeu de la Vie se joue sur un tableau de case carrées et que l'on peut prendre aussi grand qu'on veut, et on imagine que chaque case peut héberger une cellule. Si la case héberge une cellule, elle devient noire, sinon elle reste blanche.

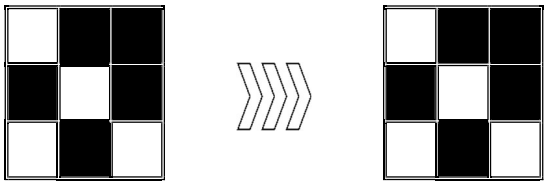
Exemple :



Maintenant le but est de faire évoluer une population de cellules au tour par tour selon des règles simplistes et défini à l'avance.

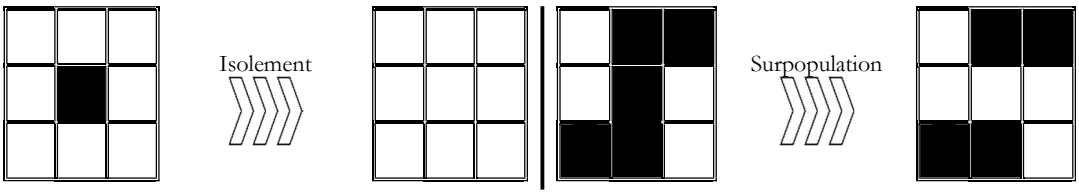
Règle de survie

Une cellule vivante le reste au tour suivant en étant entouré de 2 ou 3 cellules. Dans d'autres cas elle meurt d'isolement ou de surpopulation.



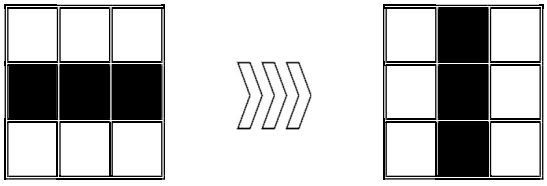
Règle de mort

Une cellule vivante meurt d'isolement si elle n'est entourée que de 1 ou 0 cellules. Elle meurt également de surpopulation si elle est entourée de plus de 3 cellules :



Règle de vie

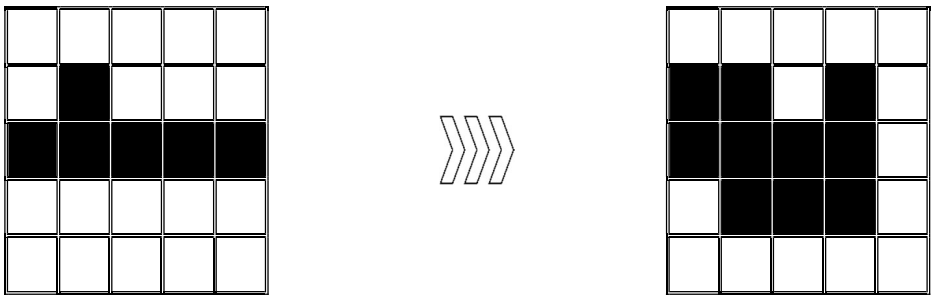
Si une case vide est entourée d'exactement 3 cellules vivantes, elle devient vivante au tour suivant :



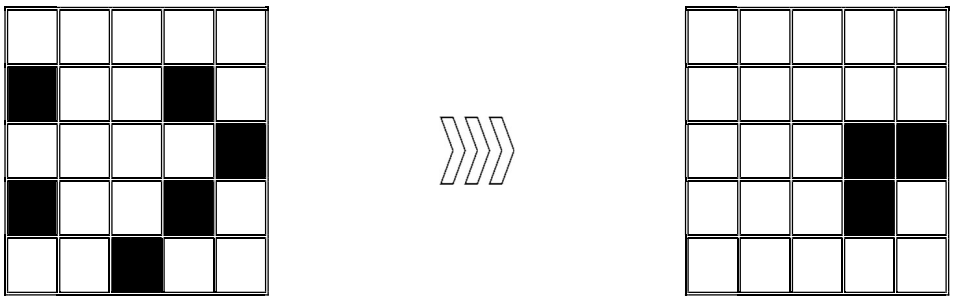
ALZURIA Mathys, BOIS Axel, CHEBANA Mohammed

Premier test

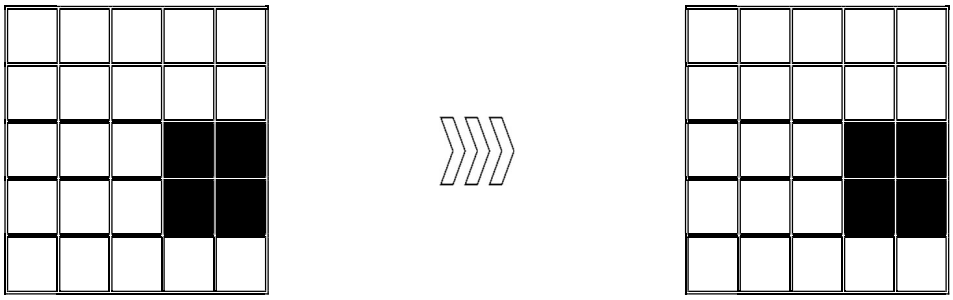
Testons maintenant ces règles sur une population test :



n = 0 n = 1



n = 2 n = 3



n = 4 n = 5

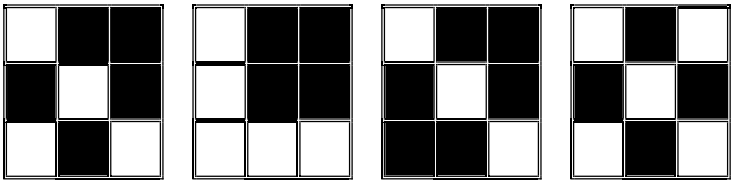
On peut voir qu'au fil des générations, des cellules naissent, meurent, mais au bout de la 5ème génération, plus rien ne se passe. Il ne se passera d'ailleurs jamais rien peu importe le nombre de générations futures. C'est ce qu'on appelle une structure stable. Le tout premier exemple était également une structure stable. Les structures stables sont des structures particulières du Jeu de la Vie, il existe aussi les structures oscillantes et les structures en mouvement.

Structures particulières

Comme dit précédemment, on peut retrouver des structures particulières avec les règles de base du Jeu de la Vie.

Structures stables

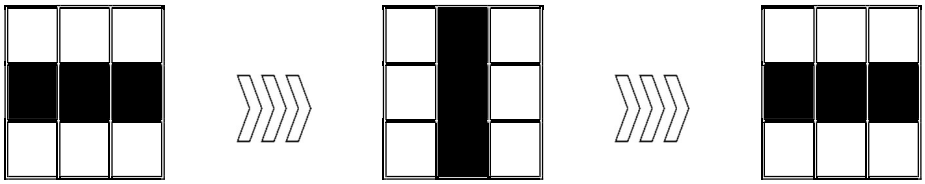
Une structure stable est une structure de cellule qui n'admet ni naissances ni mort et ce peu importe le nombre de générations qui passe :



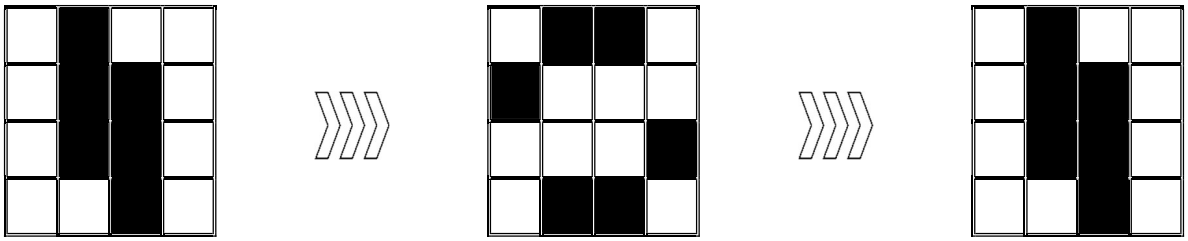
Bateau Bloc Navire Tube

Structure oscillante

Une structure oscillante est une structure qui retrouve sa configuration initiale au bout de n génération :



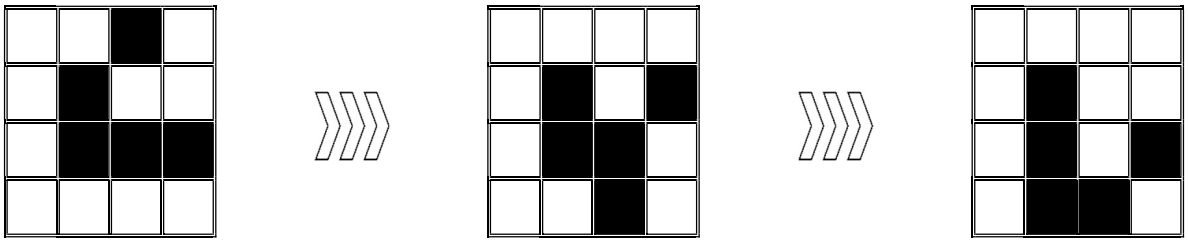
Clignotant



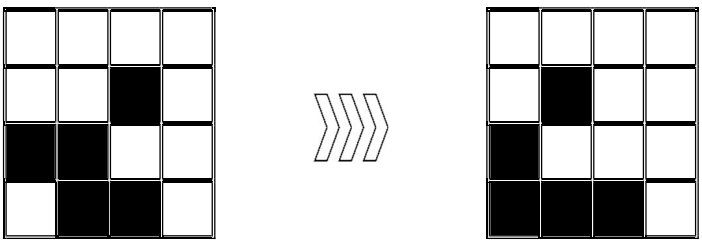
Crapaud

Structure en mouvement

Une structure en mouvement est comme une structure oscillante, qui va donc retrouver sa configuration initiale au bout de n génération mais en s'étant déplacé d'une case en diagonale et qui va donc se déplacer indéfiniment.



n = 0 n = 1 Planeur n = 2



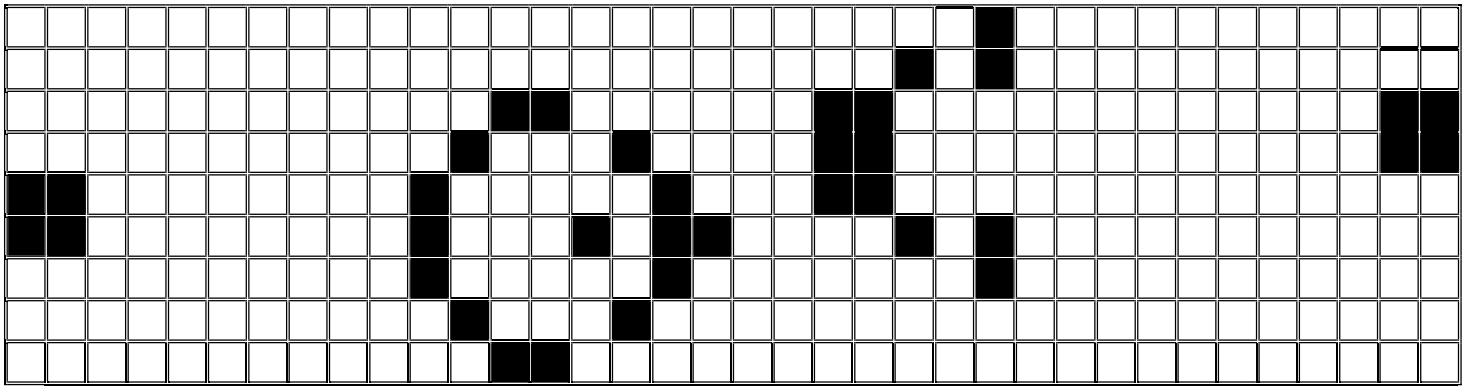
n = 3 n = 4

Qu'est ce qui en découle ?

Ce sont les planeurs qui vont intéresser John Conway et plus particulièrement une structure oscillante découvert bar Bill Gosper, le premier « canon à planeur », le canon à planeur de Gosper.

Canon à planer de Gosper

Pour faire simple le canon à planer de Gosper est une structure oscillante, qui donc retrouve sa forme initiale au bout de n génération MAIS qui crée également un planeur. Etant donnée qu'un planeur ne disparais jamais, cette structure crée à intervalle régulier des cellules. C'est la première configuration qui à un nombre de cellules tendant vers l'infini.



Canon à planeur de Gosper

Il y a eu des études sur cet automate cellulaire qu'est le Jeu de la Vie qui ont démontré que le jeu est dit Turing complet, ce qui veut dire vulgairement que l'on peut programmer avec, que c'est aussi complexe qu'un ordinateur. Il faut rappeler que ce qui fait la beauté de la chose c'est que tous ces comportements découlent de 2 lois extrêmement simple. D'ailleurs personne en voyant les 2 règles n'aurait pu prédire qu'une telle complexité aurait pu en découler. C'est ce qu'on appelle l'émergence, le Jeu de la Vie est donc un algorithme émergeant.

Un algorithme émergent

L'émergence est à la base issue d'une idée philosophique formalisée au XIX° Siècle et qui peut se résumer par l'adage : « le "tout" est plus que la somme de ses parties ». Cette idée est contrintuitive car la somme des parties d'un "tout" forme le "tout" lui-même or cette idée nous dit que le "tout" est supérieur. On peut dire aussi que l'émergence c'est l'apparition imprévisible d'un comportement complexe au sein d'une population aux caractéristiques uniques simples, d'où le fait que le "tout" soit supérieur à la somme de ses parties.

On a donc ici un automate cellulaire régis par deux règles simple (tellement qu'un enfant pourrait jouer au Jeu de la Vie), duquel émerge des comportements imprévisibles tel qu'une croissance infinie, plus ou moins rapide, ou même le fait que le jeu soit Turing complet.

Projet SAE – Développement d'une application

Base du projet

On nous a commandé une application capable de simuler le Jeu de la Vie avec une IHM.

Client

Ce projet SAE – Développement d'une application est tuteuré par :
- SERVIERES Corinne
- BARREPERSYN Sylvain

L'application simulant le Jeu de la Vie nous a été commandé par :
- BARREPERSYN Sylvain

Equipe

Pour mener à bien cette SAE, nous avions un groupe composé de 3 personnes :
- ALZURIA Mathys
- BOIS Axel
- CHEBANA Mohammed

Premier Brainstorming

Quand le projet nos a été soumis, nous avons fait un premier brainstorming pour se mettre dans le bain et trouver une voie de progression.

De cette première réunion est sorti les voies de progressions suivantes :
- Ouverture d'un repository GitHub
- Ouverture d'un drive commun
- Création d'un dépôt Git distant avec une branche par membre du groupe
- Comment programmer le Jeu de la Vie en tant que tel (programmation des règles de vie et de mort)
- Comment l'IHM sera dans les grandes lignes au niveau visuel
- Quelles sont les différentes interactions entre l'utilisateur de l'application et le système
- Répartition de qui fait quoi pour commencer le projet
- Tentative d'exploration du sujet : est-il possible de modifier les règles du Jeu de la Vie ?

La programmation peut commencer

Rôle de chacun :

Pour que chacun sache quoi faire et sur quelle partie avancer en priorité, des rôles ont été distribué entre les membres du groupe :

Chef développeur : BOIS Axel
Développeurs : BOIS Axel, CHEBANA Mohammed, ALZURIA Mathys

Dépôt Git Branch :
- Master : ALZURIA Mathys
- Alpha : CHEBANA Mohammed
- Beta : BOIS Axel

Chef maquette figma : CHEBANA Mohammed
Développeurs artistiques (figma) : CHEBANA Mohammed, BOIS Axel

Gestionnaire communication / Rédacteur : ALZURIA Mathys

Diagramme de l'application

Vous trouverez en suivant le diagramme de cas d'utilisation et le diagramme de classe de notre application.

Diagramme de cas d'utilisation

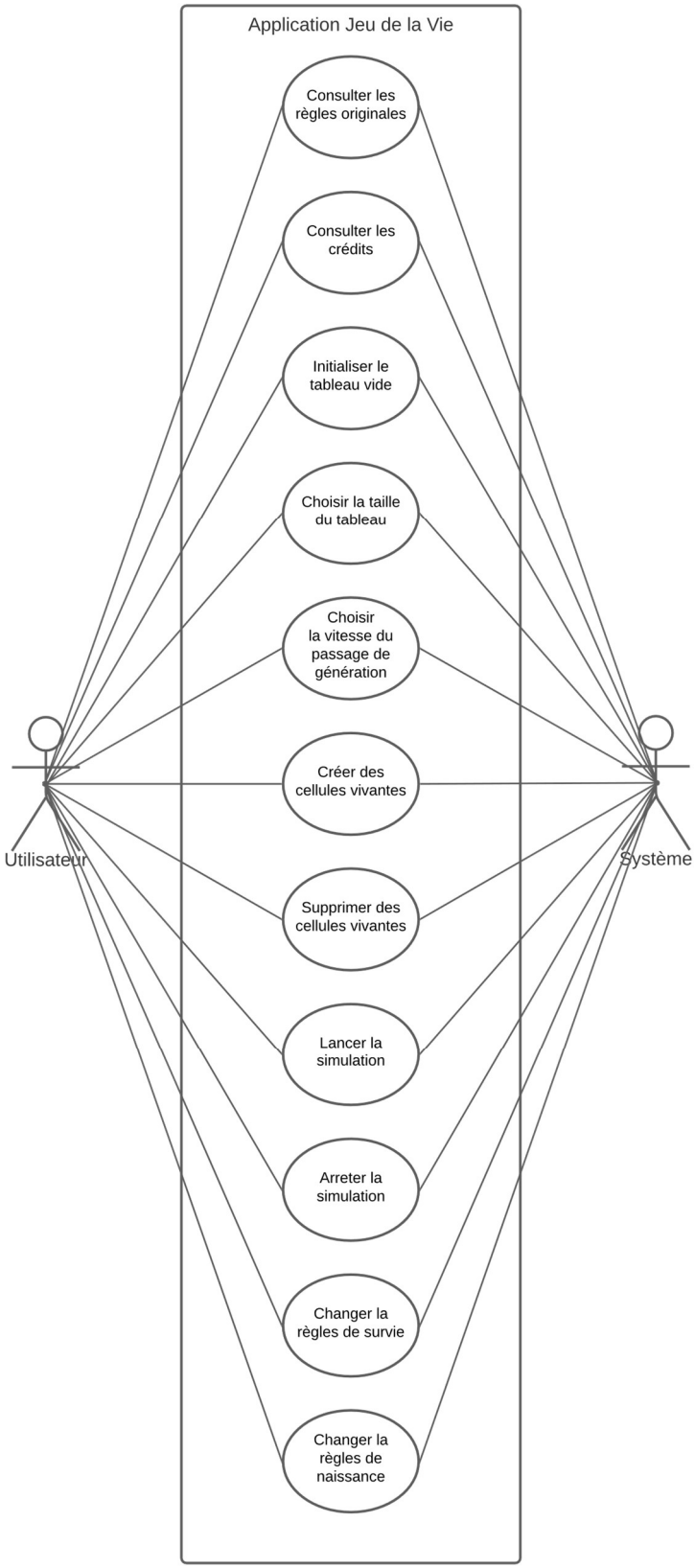


Diagramme de classe

Test

Nous avons fait des tests du Jeu de la Vie à la main pour comprendre comment pouvoir le programmer. Nos tests correspondaient donc à tester les différentes règles du Jeu de la Vie sur différentes populations. Nous avons également testé notre exploration du sujet.

Test à la main règle Jeu de la Vie

Voir « Automate cellulaire : le Jeu de la Vie – Premier test »

Test à la main Exploration du Sujet

Nous nous sommes donc posé la question : est-il possible de modifier les règles du Jeu de la Vie ?

Pour rappel :

Règle de vie : 1 case entourée de 3 cellules vivantes
Règle de survie : 1 cellule entourée de 2 cellules
Autres cas = mort

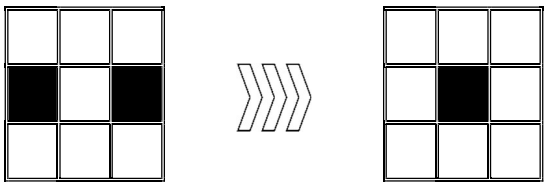
On a donc dans le code : - associé à la règle de vie, l'information 3
- associé à la règle de survie l'information 2

Ces informations sont modifiables et on peut donc les remplacer par n'importe quelle valeur entre 0 et 8 (car une case ne peut pas avoir mois de 0 cellule autour d'elle ou bien plus de 8 cellules autour d'elle).

On peut donc imaginer la situation suivante :

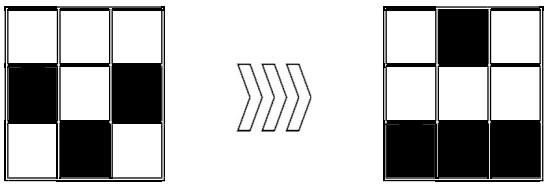
Règle de vie

Si une case vide est entourée d'exactement **2** cellules vivantes, elle devient vivante au tour suivant :



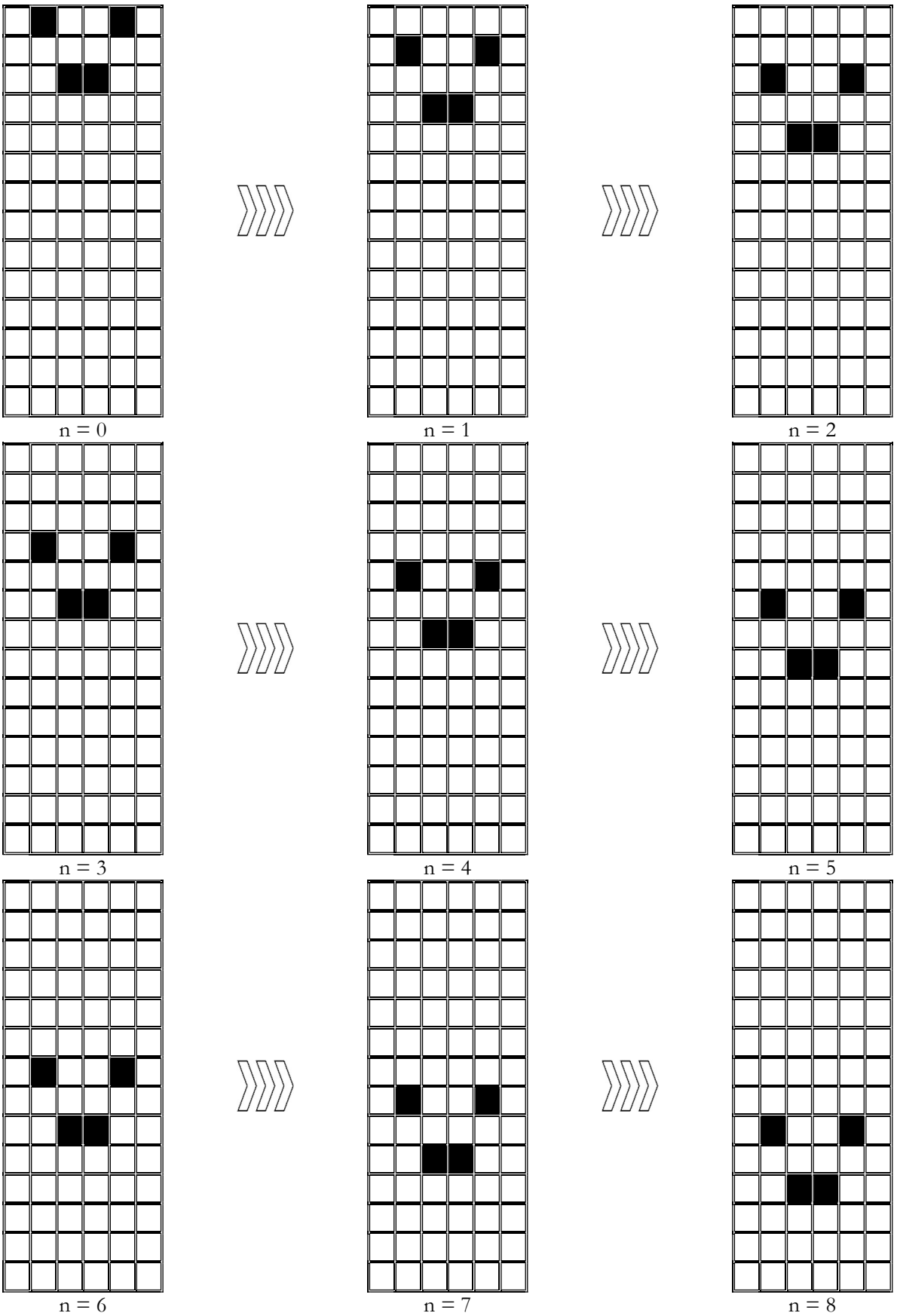
Règle de survie

Une cellule vivante le reste au tour suivant en étant entouré de **2** cellules. Dans d'autres cas elle meurt d'isolement ou de surpopulation.



Test des nouvelles règles

On a donc testé cette nouvelle configuration et on a vu très vite qu'on pouvait faire des planeurs assez facilement.



On a donc ici un planeur qui se déplace de 1 case à chaque génération à l'infini.

Conclusion

Bilan personnel

Ci-après les bilans individuels des membres de l'équipe.

ALZURIA Mathys

Cette SAE a été la plus stimulante de toutes. Ça va peut-être aussi avec le fait que je connaissais déjà le Jeu de la Vie et la catégorie d'algorithme auquel il appartient, les algorithmes émergents qui constituent une grande passion pour moi. J'avais déjà essayé à l'époque de programmer cet automate en python (le seul langage que je connaissais) sans succès car pas assez d'expérience. C'est donc aujourd'hui que grâce a mon groupe, nous réussissons là où j'avais échoué. Je pense que ce sujet était le plus intéressant parmi tous du point de vue où on peut le dépasser : se renseigner sur les algorithmes émergents ou bien imaginer et programmer un Jeu de la Vie avec 81 possibilités de règles différentes. Merci pour cette SAE.

BOIS Axel

Pour ma part j'ai trouvé que la SAE était plutôt intéressante du point de vue de la programmation/IHM car le Java Fx est un langage que l'on n'avait jamais utilisé. De ce fait on était plongé dans l'inconnu pour l'IHM mais on a réussi à se débrouiller de ce côté-là. De plus, le choix des sujets était très intéressant, en tout cas plus que dans la plupart des projets qu'on a pu réaliser qui avait des sujets qui ne sont pas « amusant » à réaliser.

CHEBANA Mohammed

De mon point de vue, cette SAÉ a été l'une des plus intéressantes de par son sujet que je ne connaissais pas du tout et qui m'a fait découvrir un aspect amusant de la programmation. J'ai pu pratiquer du figma et découvrir de nouvelles fonctionnalités.