



BUT Informatique Rodez 2021 - 2022
SAÉ 2.01 - Développement d'une application
Référents : Corinne Servières
Sylvain Barrepersyn



UNIVERSITÉ TOULOUSE I

RODEZ

Dossier

SAÉ Développement d'une application

ENJALBERT Anthony - FROMENT Jean-François
2021 - 2022

Table des matières

I - Introduction	4
II - Diagramme de cas d'utilisation	5
III - Diagramme de classe	6
IV - Description des tests effectués	7
V - Manuel utilisateur	9
VI - Bilan	10
VII - Conclusion	11

I - Introduction

Dans le cadre de la SAE 2.01, nous avons pour but de proposer une interface permettant de simuler le jeu de la vie. L'interface doit permettre de modifier les règles du jeu ainsi que la vitesse de progression du jeu. L'utilisateur par le biais de l'interface doit aussi choisir les cellules qui sont vivantes ou mortes. Enfin, il doit être possible de lancer, arrêter et reprendre la simulation.

Le jeu de la vie est un jeu "sans joueur", en effet ce jeu ne nécessite pas l'intervention du joueur lors de son déroulement. Le joueur se contente de modifier les règles, choisir la taille de la grille et de rendre des cellules mortes vivantes ou inversement par un clic.

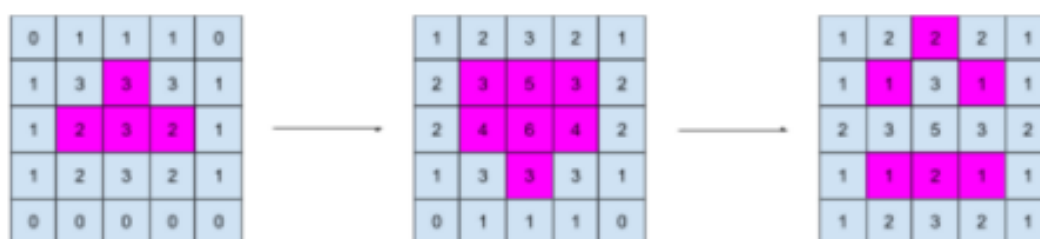
Le jeu se déroule donc sur une grille dont les cases appelées cellules peuvent prendre des états différents : "vivante" ou "morte". Une cellule est soit morte soit vivante.

Une cellule possède huit voisins, se sont les cellules adjacentes horizontalement, verticalement et diagonalement. Ceci est toujours le cas si la grille était infinie, mais dans le cas de la SAE ce n'est pas le cas. Certaines cellules (par exemple dans les coins) peuvent avoir moins de voisins.

L'état d'une cellule est déterminé par l'état de ses cellules voisines. Pour illustrer cela, utilisons les règles de base, celles de John Conway.

Une cellule morte possédant exactement trois cellules voisines vivantes devient vivante (naissance d'une nouvelle cellule). Une cellule vivante qui possède deux ou trois cellules voisines vivantes reste vivante, si ce n'est pas le cas la cellule meurt.

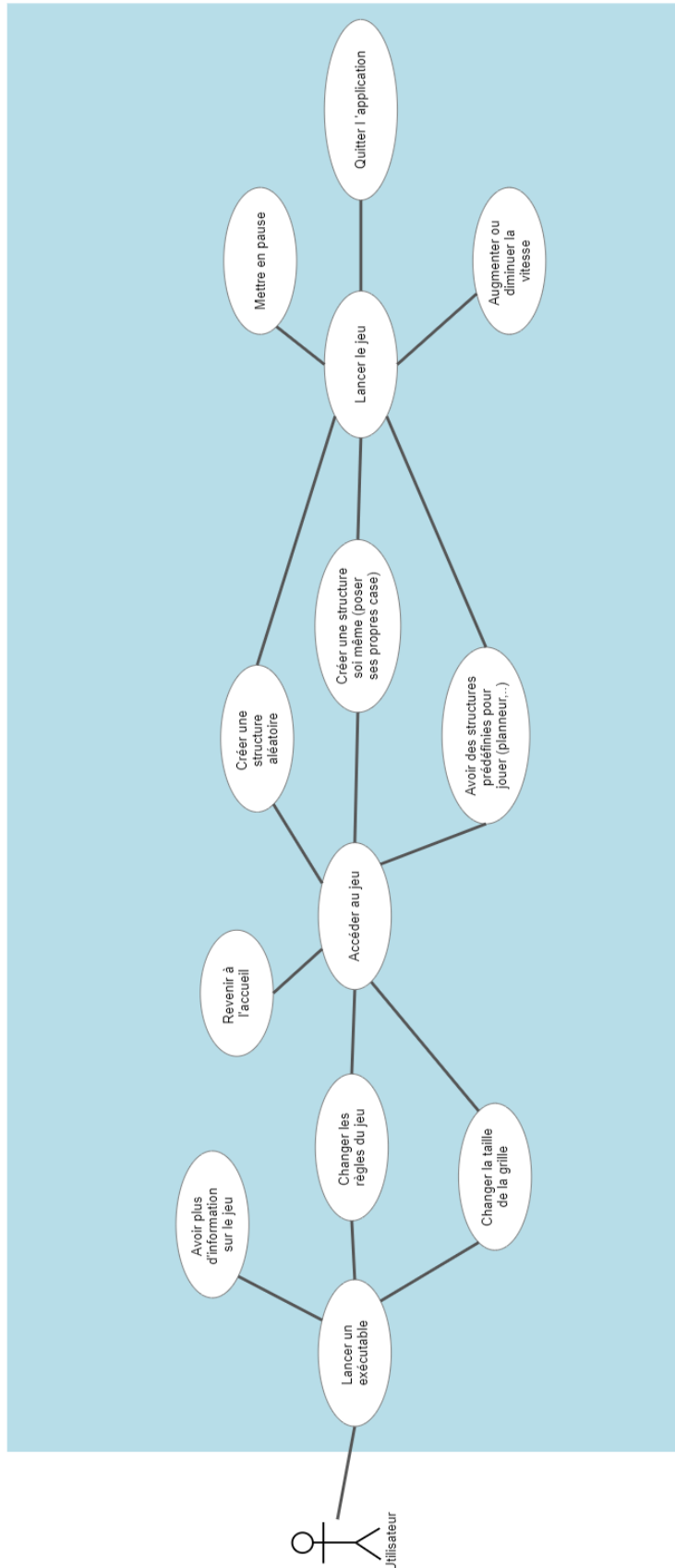
Voici un exemple de manière illustré (extrait du sujet de la SAE)



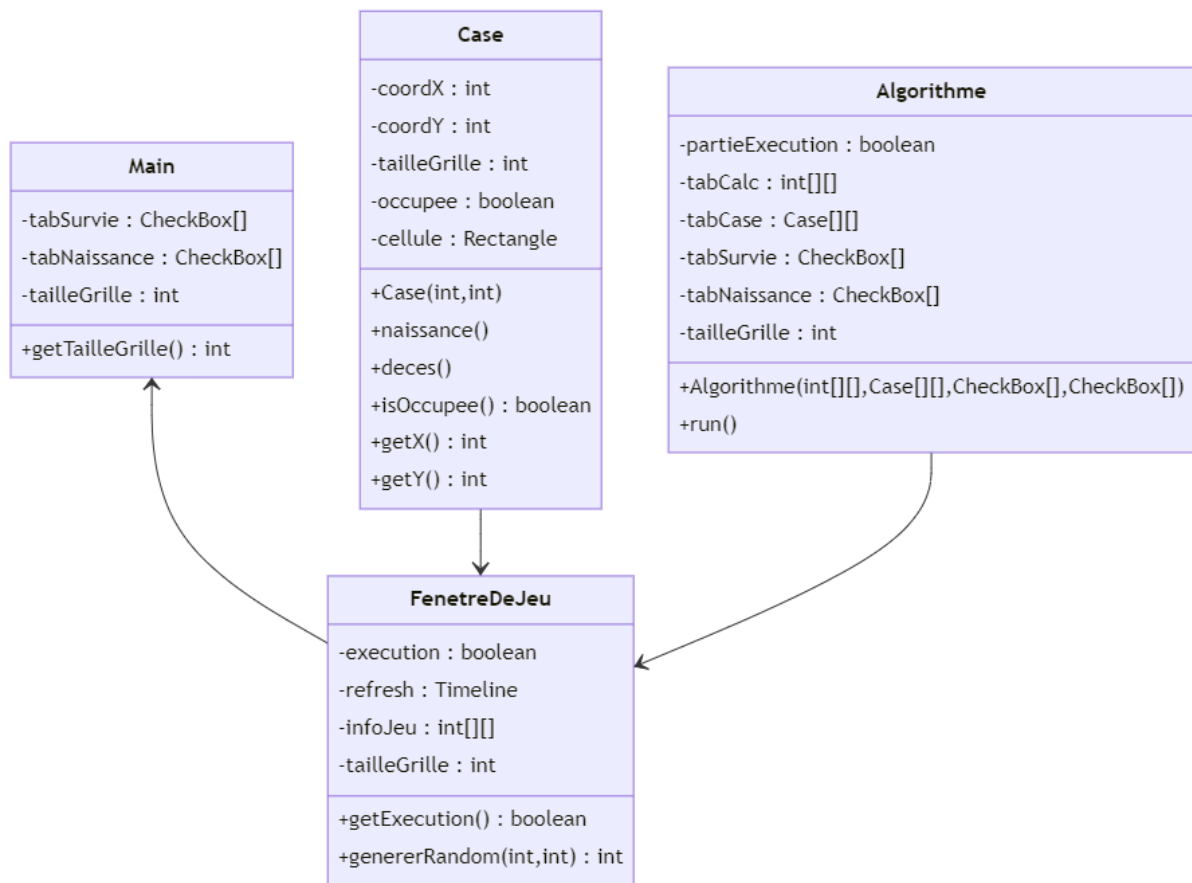
Notre travail est disponible en suivant le lien ci dessous

Lien vers le dépôt distant : <https://github.com/anthoo21/SAEJeuDeLaVie>

II - Diagramme de cas d'utilisation

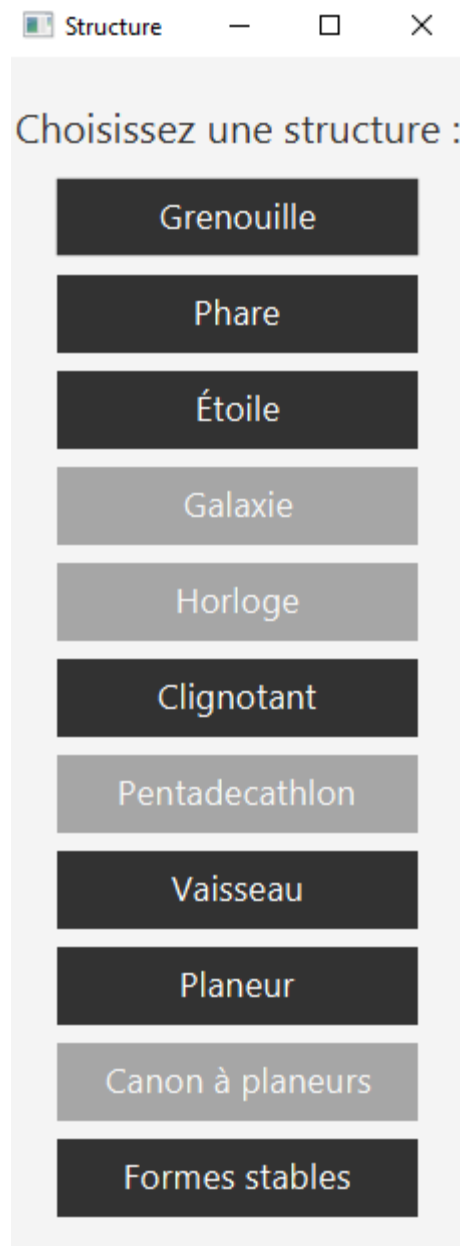


III - Diagramme de classe



IV - Description des tests effectués

Pour la description des tests effectués, vu que nous avons utilisé une interface graphique, nous n'avons pas effectué beaucoup de tests. Cependant, nous en avons utilisé pour la taille de la grille, pour qu'elle soit comprise entre deux intervalles. Nous avons aussi effectué des tests du côté du choix des figures. En effet, si une figure fait 35 pixels de longueur, mais que la grille n'en fait que 30, on ne doit pas pouvoir placer la figure.



Toujours dans le choix des figures, il ne faut pas pouvoir placer deux figures en même temps. Du coup, il nous a fallu, pour chaque figure, supprimer la figure précédente avant de placer la nouvelle.

Aucun autre test n'a été effectué, vu que la plupart des interactions avec l'interface graphique se faisait via des boutons, et non via du texte que l'utilisateur doit remplir à un moment.

V - Manuel utilisateur

Notre application permet une simulation du jeu de la vie. Elle est composée de deux pages.

Après avoir lancé l'exécutable, vous arrivez sur la page d'accueil.

Page d'accueil :

Deux boutons sont présents, le premier (le bouton jouer) vous amènera sur la page où il est possible de simuler le jeu de la vie. Le second (le bouton plus d'information) vous conduira à la page wikipédia concernant le jeu de la vie.

De plus, sur cette page d'accueil il est possible de modifier la taille de la grille en bougeant le curseur de défilement, attention le changement de grille ne peut s'effectuer qu'une seule fois au démarrage de l'application. Si vous revenez à l'accueil depuis la page de simulation de jeu le changement de grille ne sera pas pris en compte. La taille minimum est 10X10 tandis que la taille maximale est 100X100. La taille exacte de la grille s'affiche à droite du slider.

Enfin la page d'accueil permet de changer les règles par le biais de checkbox (plusieurs choix possibles), les règles prédéfinies sont celles proposées par John Conway.

Page de simulation du jeu :

On voit la grille où vous pouvez "noircir" les cases par un clic. Les cases noircies représentent donc une cellule vivante par conséquent une case non colorier (blanche) est une cellule morte.

Sous la grille, il y a des boutons. Le bouton "accueil" ramène à la page d'accueil (là où on change les règles et la taille de la grille). Le bouton "effacer" permet de remettre la grille à son état initial c'est-à-dire avec aucune cellule vivante présente. Le bouton "start" lance le jeu tandis que le bouton "stop" arrête le jeu. Le bouton "aléatoire" noircis des cases aléatoires, des cellules vivantes sont donc répartis aléatoirement sur la grille.

Enfin, le bouton "structure" ouvrira un menu. Ce menu propose différentes structures connues pour le jeu de la vie comme le planeur. Par exemple, si vous cliquez sur planeur vous obtiendrez un planeur mais attention certaines structures sont possibles uniquement si la taille de la grille le permet (exemple : le canon à planeur).

VI - Bilan

Pour le bilan, nous aborderons les difficultés que nous avons rencontrées, et comment nous les avons résolues. La première difficulté était dans la modification des règles. En effet, il fallait pouvoir modifier les règles (choisir à partir de combien de cellules voisines une cellule meurt ou reste en vie) mais ce n'était pas aisé à mettre en place avec l'algorithme. Pour résoudre ce problème, nous avons tout simplement implémenté à l'algorithme directement le choix de l'utilisateur sur la page d'accueil. La deuxième difficulté était juste après avoir écrit l'algorithme. Nous devions faire en sorte de pouvoir modifier la vitesse de jeu. Pour cela, nous nous sommes aidés d'internet pour trouver comment faire. Nous avons donc déterminé qu'utiliser une "Timeline" était la meilleure solution, et nous l'avons donc implémenté au code. Une des plus grandes difficultés était dans le choix de l'utilisateur sur la taille de la grille, car la taille des cellules varie en fonction de la taille de la grille. Il fallait donc trouver le calcul parfait par rapport à ce que l'utilisateur choisissait comme taille de grille.

On peut donc dire que ce projet nous a posé quelques problèmes, mais que nous avons su les résoudre en faisant quelques recherches et en réfléchissant avec logique.

VII - Conclusion

Conclusion Enjalbert Anthony :

J'ai beaucoup apprécié cette SAE. J'avais déjà entendu parler du jeu de la vie auparavant, mais je ne m'étais jamais vraiment penché sur les règles ou toutes les possibilités qu'offre ce jeu.

La partie JavaFX était moins compliquée que ce que je pensais. L'algorithme était assez simple à écrire, mais certains composants de JavaFX n'ont pas été faciles à utiliser.

Conclusion Froment Jean-François :

Cette SAE était intéressante. J'ai découvert le jeu de la vie, ce jeu est très amusant, on peut s'amuser à faire plein de formes différentes plus ou moins complexes. Le sujet était donc excellent.

En revanche, les outils à notre disposition notamment JavaFX on posé quelques soucis comme le fait qu'à l'IUT JavaFX est disponible sur la version Eclipse de 2019 ou qu'on est raté des TP (de mémoire à cause d'une panne) défavorisant notre apprentissage. De plus, JavaFX et SceneBuilder ne sont pas faciles à prendre en main.

Enfin travailler à deux ne m'a pas dérangé, en effet notre ancien camarade peu sérieux est parti au lancement de cette SAE.

J'ai donc apprécié cette SAE malgré les différences que j'ai pu avoir avec les différents outils.