Lenglemetz Zoé
Duchez Rémi
Casandjian Ewan

Dossier projet : ISN

Jeu de la vie

Année 2017-2018

Au début, l'enjeu principal de notre projet était de réussir à allier l'art et l'informatique mais des aspects techniques nous ont contraints à revoir cette idée. En effet, notre idée de base était de crée un cube de LED au format 8*8*8. Mais, le coté électronique, et plus particulièrement l'alimentation du cube et de ses différents étages, à représenter un problème majeur. Ainsi voyant que nous n'arrivions pas à résoudre ce problème, nous avons décidé de changer de projet.

Nous avons donc choisis de plutôt allier un aspect mathématique, ici la logique, avec l'informatique. Dans un désir de rester dans une approche ludique, nous avons entamé des recherches pour trouver des jeux correspondant à nos critères.

Dans un premier temps, nous avons d'abord pensé à des jeux tels que les échecs, les dames et le jeu de la vie.

Mais en prenant un peu de recul, nous avons compris que l'intelligence artificielle nécessaire pour créer un jeu de dame ou d'échecs était très complexe.

Notre choix s'est donc porter sur le jeu de la vie. En effet, ce jeu répondait à notre désir d'allier les mathématiques à l'informatique, tout en ayant la possibilité de coder en python.

Le jeu de la vie est un automate cellulaire pensé vers 1970 par un certain John Horton Conway. Depuis ce jour-là et encore aujourd'hui, le jeu de la vie reste l'un des automates cellulaire les plus connus.

Mais qu'est-ce qu'un automate cellulaire ?

Un automate cellulaire peut être considéré comme un objet mathématique et informatique qui évolue étapes par étapes selon des règles définie. Cette évolution est, dans une certaine mesure, une imitation des capacités autoreproductrices que possèdent les cellules constituant les êtres vivants.

Les règles définissant un automate cellulaire peuvent être diverses et variés et selon ces dernières différents dessins apparaissent de par la coloration des cellules en fonction de leur état.

Ainsi, on peut donc dire que le jeu de la vie, n'est en réalité, pas considéré comme un véritable jeu car il ne nécessite pas la présence d'un joueur pour fonctionner. Le principe est simple, l'automate est constitué d'une grille dont les cases sont appelées « cellules », ces dernières peuvent prendre deux états différents, «mort» ou «vivant».

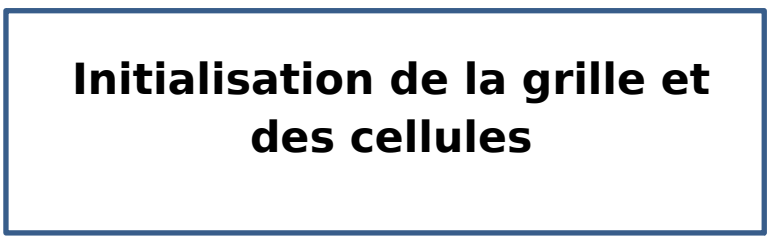
La règle que nous avons décidé d'appliquer était que chaque cellule vivante ne possédant pas au moins deux voisins devait mourir tandis qu'une cellule morte présentant au moins deux voisins vivant devenait vivante à son tour. Mais le changement d'état des cellules étant automatique, l'interaction avec la personne était très faible voir quasi-inexistante.

Cependant, nous voulions faire en sorte qu'il y ait un minimum d'interaction pour rendre la chose plus attrayante. Pour cela nous avons décidé de laisser le choix des cellules de départ à l'utilisateur et non d'utiliser des cellules prédéfinies. De plus, ce dernier peut à tout moment réinitialiser la grille et les cellules.

On peut ainsi considérer que le principe des automates cellulaires constitue le squelette de notre projet. Cependant il fallait maintenant le travailler et le mettre en place pour en arriver petit à petit au jeu de la vie.

Pour coder ce projet, nous avons donc utilisé le langage python. Ce dernier étant relativement simple, possédant une très grande bibliothèque et pouvant être utilisé par de nombreuses applications, il convenait parfaitement à notre projet.

Schéma du jeu de la vie :



Lancement de l’automate cellulaire

L’état des cellules dépend des règles prédéfini. Ici, une cellule morte possédant exactement trois voisines vivantes devient vivante tandis qu’une cellule vivante possédant au moins deux voisines vivantes

La boucle sur laquelle est basé l’algorithme se répète tant que cela est possible

Répartition des taches :

Fonction objet	Tkinter	Algorithme	Incrémentation	Rédaction du dossier

Zoé	Zoé	Zoé		
	Rémi	Rémi		Rémi
			Ewan	Ewan

Démarche collaborative :

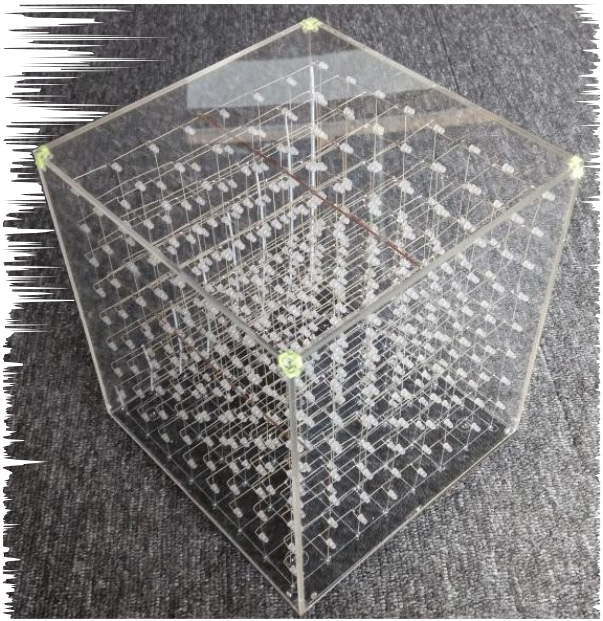
Pour créer notre projet nous avons utilisé différentes plateformes collaboratives dont Github et Google doc.

La plateforme Github nous a permis de remplir un cahier de bord et de nous partager nos recherches, les besoins que nous avions pour compléter notre code et enfin de consulter l'avancement du projet et de suivre l'évolution du code.

La plateforme Google doc quant à elle nous a servi pour la rédaction du dossier et le coté algorithmique du projet

Ainsi nous pouvions suivre l'avancement du projet et rajouter petit à petit des choses au code ou corriger des erreurs qui empêchaient le fonctionnement de ce dernier. Nous avons de plus, profité des cours d'ISN pour nous retrouver et en discuter.

Compte rendu personnel : Rémi



Il est maintenant temps de parler de mon travail personnel et de mon rôle dans le groupe. Comme nous l'avons

vu un peu plus tôt, notre idée de base était de crée un cube de LED au format 8*8*8.

De mon côté j'ai alors commencé à programmer les LED de façon à toute les définir dans un tableau pour pouvoir les réutiliser rapidement et leur fournir l'électricité nécessaire. Malheureusement quand nous avons essayé, malgré tous les efforts de mes camarades et moi-même, nous n'arrivions pas a allumé les LED des différents étages du cube.

Face à ce problème, nous avons donc décidé de changer de projet et de nous rediriger vers les automates cellulaires et les jeux.

En effet, pour réaliser ce projet, nous avons dû entamer des recherches auxquelles j'ai participé dans le but de trouver un projet répondant aux critères que nous avions définis.

Nous avons ainsi trouvé notre projet de l'automate cellulaire qu'est le jeu de la vie et le jeu de dame.

J'ai donc commencé à coder pour créer un jeu de dame mais au fur et à mesure que j'avançais, je me suis rendu compte que le mouvement de pion par une intelligence artificielle qui aurait donc représenté l'adversaire à battre était bien trop complexe. Je me suis donc orienté sur le jeu de la vie.

Mais comme nous l'avons appris en cours, avant de commencer à coder, il faut écrire un algorithme.

Avec l'aide de ma camarade Zoé, nous nous sommes donc lancés dans la création de ce dernier.

Dans cet algorithme nous avons décrit les grandes étapes que devrait suivre notre code. Dans un premier temps la création de la grille et des cellules, des boutons, puis différentes étapes comme l'incrémentation etc...

Une fois l'algorithme de base rédigé, nous avons entamé le codage de cet automate cellulaire, fondement même de notre projet.

En effet, le coté algorithmique nous a permis de mieux orientés nos recherches et donc d'utiliser fonctions comme la fonction objet etc ...

J'ai ainsi découvert le module python susnommé Tkinter. Ce dernier nous a permis à Zoé et moi de crée l'interface graphique de notre jeu de la vie tandis qu'Ewan s'occuper de l'incrémentation.

Ici grâce à Tkinter, tandis que Zoé crée les boutons, essentiels pour lancer le jeu, moi je me suis chargé de crée la grille.

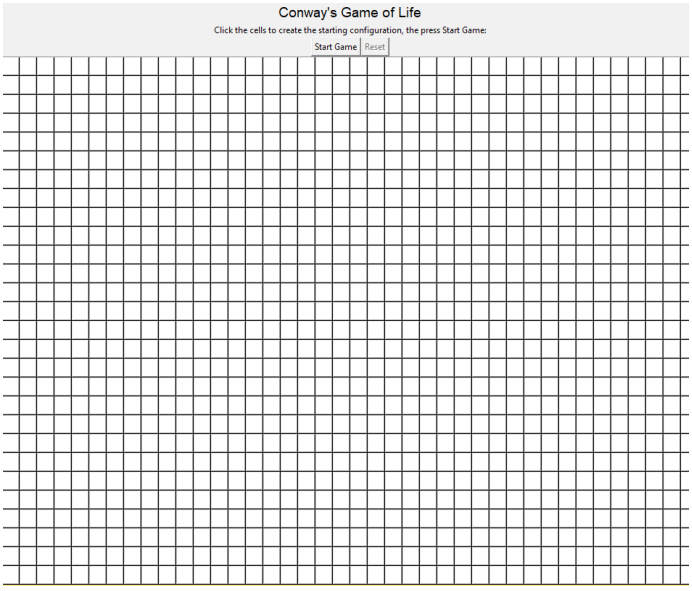
J'ai pour cela réutilisé les connaissances mise en œuvre pour coder le jeu de dame et donc le damier. Pour que la grille soit plus facile à réutiliser j'ai codé la grille avec des coordonnées.

```
from tkinter import *
from math import*
fen=Tk()
Taille_fenetre= 1000


Fenetre = Canvas(fen,bg ="white", height = Taille_fenetre, width = Taille_fenetre) #On crée une surface qui va nous permettre de tracer notre grille et nos lignes.

Nombre_Case = 10
Taille = Taille_fenetre /Nombre_Case
CoteGauche = 0
Bordure = 100

while CoteGauche <= Taille_fenetre : # On trace les lignes verticales
    Fenetre.create_line(Bordure,Bordure + Taille*CoteGauche, Taille_fenetre - Bordure, Bordure + Taille*CoteGauche,fill='black')
    CoteGauche += 1

CoteHaut = 0
while CoteHaut <= Taille_fenetre : # On trace les lignes verticales
    Fenetre.create_line(Bordure + Taille*CoteHaut,Bordure, Bordure + Taille*CoteHaut, Taille_fenetre - Bordure,fill='black')
    CoteHaut += 1

fen.mainloop() # On place la boucle permettant l'execution de la grille.

```

Les lignes de codes ci-dessus sont celles que j'avais rédigées pour coder le damier. Je me suis donc servi de ca pour créer une grille plus grande correspondant aux critères nécessaire pour notre jeu de la vie.

Une fois la grille fini, Zoé a trouvé de nouvelles fonctions qui nous ont permis de modifier le code de la grille pour ainsi mieux l'intégrer dans le code en la simplifiant.

Ainsi nous en sommes arrivés au résultat final suivant :



Enfin j'ai activement participé à la plateforme github sur laquelle je remplissais toutes les semaines le carnet de bord et modifié les différentes colonnes en fonction de l'évolution de notre projet.

Les difficultés rencontrées :

Les difficultés rencontrées furent nombreuses. D'abord dans notre premier projet, les problèmes électroniques nous ont contraints à nous orienter sur un autre projet totalement différent. En effet, je n'arrivais pas à alimenter les LED des étages supérieurs, de plus nous avions des problèmes du côté de la construction du cube. Malgré cela nous n'avions pas tout perdu, en effet nous avons découvert de nouvelles choses aussi bien en termes de codage sous python mais également sur d'autres langages et l'utilisation de micro-ordinateurs tels que les rasberry pi.

Ensuite les difficultés rencontrées furent lors de la création de la grille. En effet, la principale difficulté que j'ai alors rencontrée était le tracé des lignes verticales. J'ai mis un certain temps à comprendre pourquoi. En effet, la solution était en fait relativement simple. Il me suffisait de recopier le code correspondant aux lignes horizontales en inversant les abscisses et les ordonnées. J'en ai ainsi déduit que la solution n'est parfois pas si compliquée que ça.

Enfin la dernière difficulté rencontrées a été de crée le paramètre de sélection des cellules vivantes par l'utilisateur. Il a alors fallu faire correspondre le clic gauche de la souris sur une case au changement d'état de la cellule.

Intégration dans le projet :

Mon projet s'insère relativement bien dans le projet global. En effet la grille que j'ai proposée au travers de l'utilisation du module python Tkinter, est la base de notre automate cellulaire de Conway ou jeu de la vie. De plus avant même de coder, la réalisation de l'algorithme avec l'aide de mes camarades nous a permis à chacun d'orienter nos recherches sur telle ou telle fonction selon nos besoins.

Enfin, mes mises a jours semestriel de la plateforme github nous a permis de suivre l'avancement du projet, le planning ainsi que les recherches a faires pour compléter, corriger et améliorer notre code.

Bilan et perspectives :

Pour conclure, les objectifs que nous nous étions fixés moi et mon groupe ont été atteints et notre automate cellulaire terminé à temps.

Cependant, de nombreuses améliorations peuvent encore y être apportées. Tout d'abord au niveau de l'interface qui pourrait être plus travaillée, en effet au jour d'aujourd'hui, l'interface de notre programme reste très simple par rapport à d'autres jeux ou automates

cellulaires que l'on pourrait trouver sur Internet. De plus une attirance plus sympathique, serait bien plus attrayante pour l'utilisateur.

Ensuite, la seconde amélioration possible serait au niveau de l'interaction. En effet, le jeu de la vie reste très limité dans son interaction avec l'utilisateur, même si nous avons cherché à améliorer cela en lui laissant le choix des cellules vivantes de départ.

Pour améliorer l'interaction on pourrait peut être proposé un support en hardware avec un panneau de LED, ou tout simplement en lui permettant d'appliquer des règles qu'il aurait choisi.

Enfin, au travers de ce travail de groupe j'ai beaucoup appris sur le langage python au travers de mes recherches aussi bien que par l'interaction avec mes camarades et les autres groupes et tout simplement de mes erreurs et échecs.

Lien Github

https://github.com/lasource2018/plus_qu-un_simple_cube.../tree/code