

# RAPPORT PROJET

RÉALISÉ PAR : ZANA DIAMOUTENE & LE QUANG HUY

## Introduction

Le projet a été réalisé par Zana Diamoutene et Le Quang Huy. Concernant Léo Mothe, notre coéquipier, nous l'avons relancé à plusieurs reprises pour nous fournir sa partie, mais en vain.

## I) Rapport sur l'Implémentation du Jeu de la Vie (Réalisé par ZANA DIAMOUTENE)

### A) Introduction

Ce rapport détaille la conception et l'implémentation d'une simulation du "Jeu de la Vie" développée par Zana Diamoutene. Le "Jeu de la Vie" est un automate cellulaire inventé par John Conway en 1970, qui simule l'évolution de cellules sur une grille en fonction de règles simples. L'objectif de ce projet est de fournir une simulation visuelle du jeu de la vie avec une interface utilisateur permettant à l'utilisateur de suivre et d'interagir avec l'évolution de la grille.

### B) Fonctionnalités Implémentées

La simulation du "Jeu de la Vie" comprend les fonctionnalités suivantes :

Initialisation de la Grille : La grille est initialisée avec des cellules vivantes (humains) et mortes selon une probabilité définie.

Calcul du Nombre de Voisins : Chaque cellule calcule le nombre de voisins vivants qu'elle possède, nécessaire pour appliquer les règles d'évolution.

Simulation de l'Évolution : L'évolution de la grille est simulée selon les règles du "Jeu de la Vie". Les cellules vivantes peuvent naître, survivre ou mourir en fonction du nombre de voisins vivants.

Affichage Graphique : L'interface utilisateur Tkinter est utilisée pour afficher la grille de manière graphique. Les cellules vivantes sont représentées par des rectangles noirs, tandis que les cellules mortes sont représentées par des rectangles blancs.

Affichage des Statistiques : Une étiquette affiche en temps réel le nombre d'humains morts et le nombre d'humains nés lors de la simulation.

Réinitialisation du Jeu : Une fonctionnalité permet à l'utilisateur de réinitialiser le jeu à son état initial.

Contrôle de la Vitesse : Un bouton permet à l'utilisateur de contrôler la vitesse de la simulation.

Bouton Quitter : Un bouton permet à l'utilisateur de quitter l'application.

## B.1) Fonctionnalités non Implémentées

Implémentation d'un bouton "Suivant" pour permettre à l'utilisateur de simuler une étape de l'évolution du jeu à chaque clic.

## C) Choix de Conception

Plusieurs choix de conception ont été faits pour réaliser cette simulation :

Utilisation de Classes : Les classes ont été utilisées pour représenter les cellules, la grille et l'interface graphique, offrant ainsi une structure modulaire et une meilleure organisation du code.

Utilisation de Tkinter : Tkinter a été choisi comme bibliothèque graphique pour sa simplicité d'utilisation et son intégration native avec Python.

Méthode de Simulation : La simulation de l'évolution de la grille a été implémentée en deux étapes : la création d'une nouvelle grille temporaire basée sur l'état actuel, puis la mise à jour de l'état de chaque cellule en fonction du nombre de voisins vivants.

### Difficultés Rencontrées

Pendant l'implémentation, quelques difficultés ont été rencontrées, notamment :

Compréhension des règles du "Jeu de la Vie" et leur traduction en code.

Gestion des événements utilisateur dans Tkinter pour permettre l'interaction avec la grille.

### Points Positifs

Malgré les difficultés rencontrées, plusieurs points positifs ont émergé de ce projet, notamment :

La modularité du code grâce à l'utilisation de classes, facilitant la gestion et la manipulation des différentes composantes du jeu.

L'interface graphique conviviale et intuitive, permettant à l'utilisateur de suivre facilement l'évolution du jeu.

La mise en œuvre réussie des fonctionnalités supplémentaires, améliorant l'expérience utilisateur globale.

### Points Négatifs

Cependant, quelques points négatifs sont également à noter :

La complexité de certaines parties du code, notamment la gestion des événements et la simulation de l'évolution de la grille.

La nécessité de maintenir la cohérence entre les différentes parties de l'application, ce qui peut parfois être laborieux.

En conclusion, ce projet a été une expérience enrichissante dans la compréhension et la mise en œuvre des automates cellulaires, avec des résultats satisfaisants malgré quelques difficultés rencontrées en cours de route.

## II) Rapport Final sur le Jeu de Serpent (Réalisé par Le Quang Huy)

## A) Introduction

Le Jeu de Serpent est un classique du jeu vidéo où le joueur contrôle un serpent pour manger de la nourriture apparaissant aléatoirement à l'écran. Lorsque le serpent mange de la nourriture, il s'allonge et le score du joueur augmente. L'objectif du jeu est de maintenir le serpent en vie aussi longtemps que possible en évitant de heurter les limites de l'écran ou son propre corps.

## B) Les composants principaux du jeu

MyApp (Application) : Il s'agit de la partie la plus haute de l'application, contenant à la fois la fenêtre de l'interface utilisateur et le canevas pour dessiner le jeu.

PlanetTK (PlanèteTK) : Cette classe définit les constantes pour les directions.

Element (Élément) : Classe de base représentant les éléments du jeu, comprenant à la fois le serpent et la nourriture.

Snake (Serpent) : Classe fille d'Element, décrivant le serpent dans le jeu. Elle comprend des attributs tels que la taille, la vitesse, la direction de déplacement et le corps du serpent.

Food (Nourriture) : Classe fille d'Element, représentant la nourriture apparaissant dans le jeu.

PlanetAlpha (Planète Alpha) : Classe décrivant la planète dans le jeu,

avec des fonctions pour générer aléatoirement des positions de nourriture et gérer les positions des éléments sur la planète.

SnakeGame (Jeu de Serpent) : Cette classe est le centre de toute la logique du jeu. Elle gère à la fois le serpent, la nourriture et la planète, et contrôle le flux du jeu.

C) Bilan de ce qui a été fait / pas fait par rapport aux prévisions

Ce qui a été fait :

Implémentation réussie du jeu de serpent en utilisant la bibliothèque Tkinter de Python.

Création d'une planète virtuelle sur laquelle le serpent évolue.

Ajout de fonctionnalités telles que le déplacement du serpent, la consommation de nourriture et la détection des collisions.

Mise en place d'un mécanisme de score pour suivre les performances du joueur.

Ajout d'un bouton "Restart" permettant de redémarrer le jeu en cas de fin de partie.

Amélioration de l'interface utilisateur avec des graphismes simples mais efficaces.

Ce qui n'a pas été fait :

La mise en place d'un bouton de pause pour permettre aux joueurs de mettre le jeu en pause.

L'augmentation de la vitesse du serpent à chaque fois que le score augmente.

L'implémentation d'un système de sauvegarde des scores les plus élevés.

## D) Difficultés rencontrées et résolutions (ou pas)

Les difficultés rencontrées comprenaient la gestion des mouvements du serpent, la gestion des collisions et la gestion des événements utilisateur. Ces difficultés ont été résolues grâce à une approche méthodique de débogage et de recherche de solutions en ligne.

## E) Bilan final

Points positifs :

Réussite de la mise en œuvre du jeu de serpent avec des fonctionnalités de base.

Ajout d'un bouton "Restart" pour améliorer l'expérience utilisateur.

Graphismes simples mais esthétiquement agréables.

Points négatifs :

Manque de fonctionnalités avancées telles qu'un bouton de pause et un système de sauvegarde des scores les plus élevés.

Certaines fonctionnalités prévues n'ont pas été implémentées, comme l'augmentation de la vitesse du serpent en fonction du score.

## III) Rapport : Choix du Jeu (Réalisé ZANA DIAMOUTENE)

Ce rapport présente l'implémentation d'une boîte de dialogue permettant à l'utilisateur de choisir entre deux jeux : le "Jeu de la Vie" et le "Jeu du Serpent".

Le code comprend deux classes principales :

**GameChoiceDialog:** Cette classe représente la boîte de dialogue de choix du jeu. Elle affiche deux boutons pour sélectionner le jeu souhaité.

**MyApp:** Cette classe représente l'application principale. Elle crée une instance de la boîte de dialogue de choix du jeu et lance le jeu sélectionné par l'utilisateur.

## Fonctionnement

Lorsque l'application est lancée, une boîte de dialogue s'affiche avec deux options de jeu : "Jeu de la Vie" et "Jeu du Serpent".

L'utilisateur peut choisir l'un des jeux en cliquant sur le bouton correspondant.

Une fois le choix effectué, l'application démarre le jeu sélectionné dans une fenêtre principale.

## Points Positifs

**Interface Conviviale:** La boîte de dialogue offre une interface simple et conviviale pour choisir le jeu.

**Flexibilité:** L'utilisateur peut facilement choisir entre deux jeux différents, offrant



ainsi une expérience personnalisée.

## Points Négatifs

Limitation aux Deux Jeux: L'application est limitée aux deux jeux proposés. Il n'y a pas de mécanisme pour ajouter d'autres jeux facilement.

## Ajouts et Fonctionnalités Supplémentaires

Boutons de Contrôle: Ajout de boutons pour réinitialiser ou mettre en pause le jeu sélectionné.

Gestion des Erreurs: Possibilité d'ajouter une gestion des erreurs pour les cas où l'utilisateur ne sélectionne aucun jeu ou ferme la boîte de dialogue.

## Difficultés Rencontrées

Intégration avec Tkinter: Assurer une intégration correcte de la boîte de dialogue avec Tkinter et le reste de l'application.

Gestion des Choix Utilisateur: Mettre en place un mécanisme pour récupérer le choix de l'utilisateur et lancer le jeu correspondant.

L'implémentation de la boîte de dialogue de choix du jeu offre une manière conviviale et interactive pour l'utilisateur de sélectionner le jeu qu'il souhaite jouer. Malgré sa simplicité, cette fonctionnalité constitue une étape importante pour offrir une expérience utilisateur personnalisée et flexible.

## Conclusion

En conclusion, ce projet a été une expérience enrichissante dans la conception et l'implémentation de deux jeux différents, à savoir le "Jeu de la Vie" et le "Jeu du Serpent". Réalisé par Zana Diamoutene et Le Quang Huy, ce projet a permis d'explorer divers aspects de la programmation, notamment l'utilisation de la bibliothèque Tkinter pour créer des interfaces utilisateur interactives et la mise en œuvre de logiques de jeu complexes.

Le fichier contenant le jeu de la vie est nommé `game_classes.py`, tandis que le fichier contenant le jeu du serpent est nommé `gamesnake.py`. Enfin, le fichier contenant la classe `MyApp`, permettant de jouer et de choisir entre les deux jeux proposés, est nommé `main.py`.

Malgré quelques difficultés rencontrées en cours de route, telles que la compréhension des règles du "Jeu de la Vie" et la gestion des mouvements du serpent dans le "Jeu du Serpent", ce projet a abouti à la réalisation de jeux fonctionnels et amusants pour les utilisateurs.

L'ajout de fonctionnalités supplémentaires, telles que les boutons de contrôle et la gestion des erreurs, ainsi que la création d'une interface conviviale pour choisir entre les jeux, ont contribué à améliorer l'expérience utilisateur globale.

En fin de compte, ce projet a permis d'acquérir de nouvelles compétences en programmation et de consolider les connaissances existantes, tout en fournissant

des jeux divertissants et interactifs pour les utilisateurs à apprécier.