

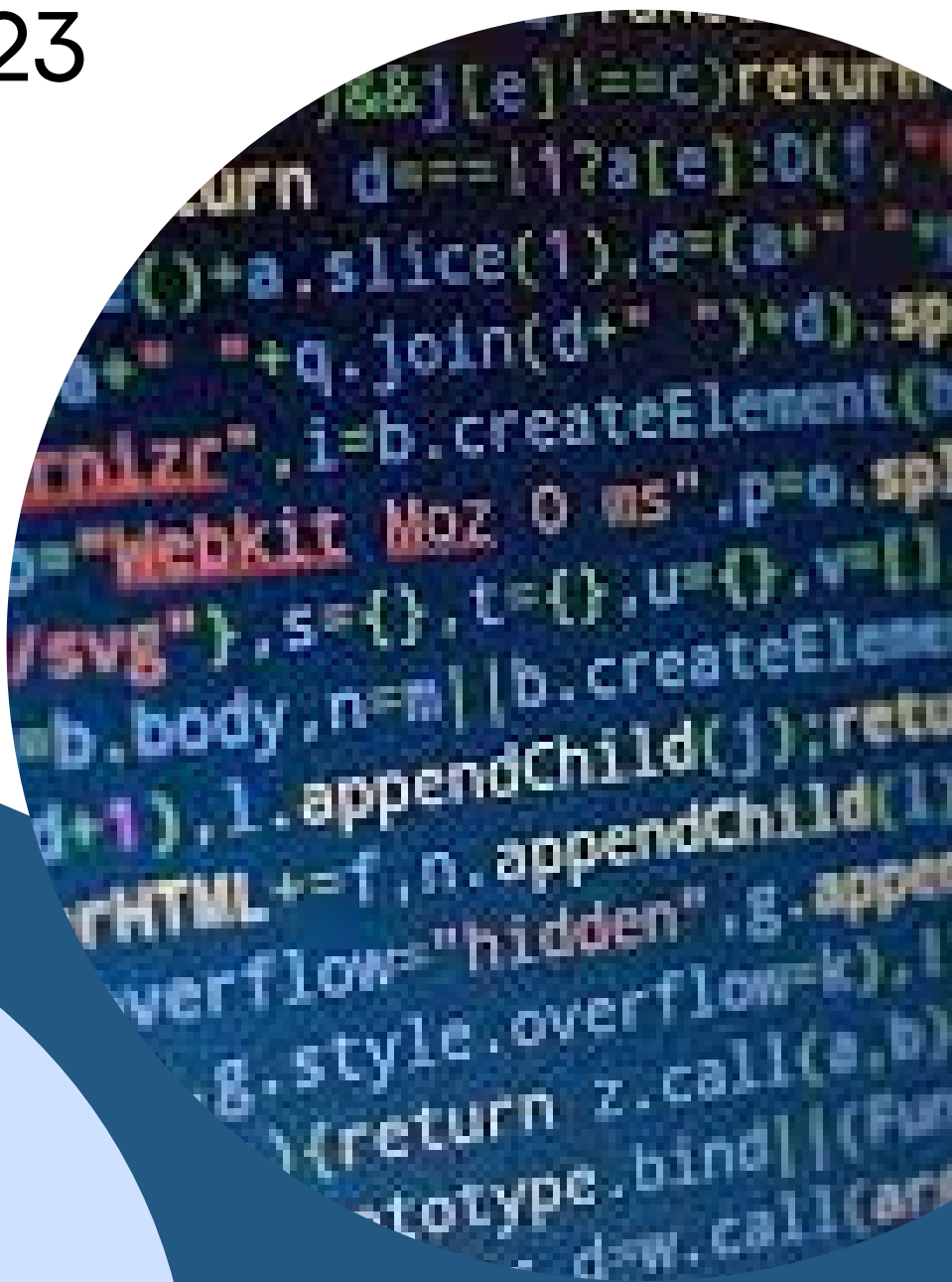
Rapport Snake

2022-2023

Réalisé par

BRIDJA Abed

SOLAR Dimitri



SOMMAIRE

Introduction	(Page 1)
Brainstorming	(Page 2)
Difficultés	(Page 3)
Fonctionnalités.....	(Page 4)
Fonctionnalités 2.....	(Page 5)
Fonctionnalités 3.....	(Page 6)
Découpage de Fichiers.....	(Page 7)
Découpage de Fichiers 2	(Page 8)
Explication des Données Représentant le Serpent	(Page 9)
Transformations au Cours de la Partie.....	(Page 10)
Conclusion	(Page 11)

INTRODUCTION

Dans le cadre de notre parcours, nous avons un projet à réaliser dans le cadre de notre formation de BUT informatique à l'IUT de Fontainebleau. Le projet en question, à pour objectif de réaliser le jeu Snake, également connu sous le nom de serpent, est un classique intemporel qui a captivé des générations de joueurs depuis sa création.

Ce jeu simple mais addictif a vu le jour à la fin des années 1970 et a évolué au fil des décennies, conservant son attrait nostalgique tout en continuant à séduire de nouveaux adeptes. L'essence du jeu réside dans sa simplicité : le joueur contrôle un serpent qui se déplace à travers un espace en constante expansion.

L'objectif principal est de guider le serpent pour qu'il mange des points de nourriture dispersés sur l'écran, ce qui entraîne la croissance progressive du serpent. Cependant, cette croissance s'accompagne d'un défi croissant, car le joueur doit éviter de heurter les murs ou de se mordre la queue. Avec des commandes basiques mais exigeantes, le jeu Snake incarne l'équilibre délicat entre la stratégie et la réactivité, créant une expérience de jeu aussi captivante qu'accessible.



BRAINSTORMING

Au cours de notre période de brainstorming, nous avons mis en place un programme détaillé répertoriant toutes les tâches nécessaires à l'achèvement de notre projet. Pour assurer une gestion optimale, nous avons élaboré un emploi du temps hebdomadaire qui définit clairement les missions à accomplir pendant la semaine, réservant le week-end à la finalisation des tâches en cours et à la planification des nouvelles activités pour la semaine suivante.

Nos réunions se sont déroulées à divers endroits, que ce soit dans les salles informatiques physiques ou sur la plateforme Discord, offrant ainsi une flexibilité essentielle à notre équipe. Ces sessions étaient cruciales pour discuter de l'évolution constante de notre projet, échanger des idées novatrices que nous avons conceptualisées, et revisiter les fonctionnalités que nous avons temporairement écartées, les intégrant dans un fichier commun où nous partagions des idées en mode brouillon.

La collaboration ne se limitait pas à la semaine de travail régulière, car nous avons également utilisé les weekends pour des échanges approfondis sur notre progression. Ces discussions couvraient divers aspects, y compris les points de vue individuels et même les désaccords, mais notre engagement envers un objectif commun était toujours maintenu, assurant ainsi la satisfaction générale de l'équipe. Cette approche collaborative a non seulement renforcé notre cohésion, mais elle a également enrichi la qualité et la diversité des idées intégrées dans notre projet, créant ainsi une dynamique de travail productive et harmonieuse.



DIFFICULTÉS

Après avoir exposé nos objectifs lors du brainstorming initial du projet, nous avons rapidement été confrontés à des difficultés. Bien que notre première session de brainstorming ait permis de définir des objectifs clairs, il était encore difficile de passer de manière concrète à la réalisation d'un véritable projet. Afin de surmonter ces obstacles, nous avons organisé une réunion pour examiner de manière approfondie nos forces et faiblesses spécifiques par rapport au projet.

Lors de cette réunion, nous avons eu l'occasion d'analyser en détail les défis rencontrés, qui étaient en grande partie cohérents avec les difficultés identifiées lors du brainstorming initial. M. Solar Dimitri a pris en charge l'aspect design et la charte graphique du jeu, ainsi que l'organisation du dépôt Git, la gestion des fichiers, la recherche d'images et le développement de certaines fonctionnalités comme la gestion du temps ou encore le démarrage du jeu .

Quant à moi, j'ai assumé la responsabilité de l'aspect technique du serpent mais aussi des obstacles, des pommes, du système de scoring et de la gestion du game over... Notre collaboration a été extrêmement bénéfique, nous permettant de nous conseiller mutuellement et d'apporter des solutions lorsque l'un d'entre nous faisait face à des difficultés.

Cette approche collaborative a renforcé notre compréhension mutuelle des domaines de compétence de chacun et a facilité le partage d'idées et de compétences. Les échanges réguliers ont également contribué à améliorer notre efficacité, en nous permettant de résoudre rapidement les problèmes et d'avancer de manière plus fluide dans le développement du projet. Grâce à cette collaboration étroite, nous sommes parvenus à surmonter les défis initiaux et à progresser de manière significative dans la réalisation de notre projet commun.

FONCTIONNALITÉS

Barre de Chargement et Lancement du Jeu :

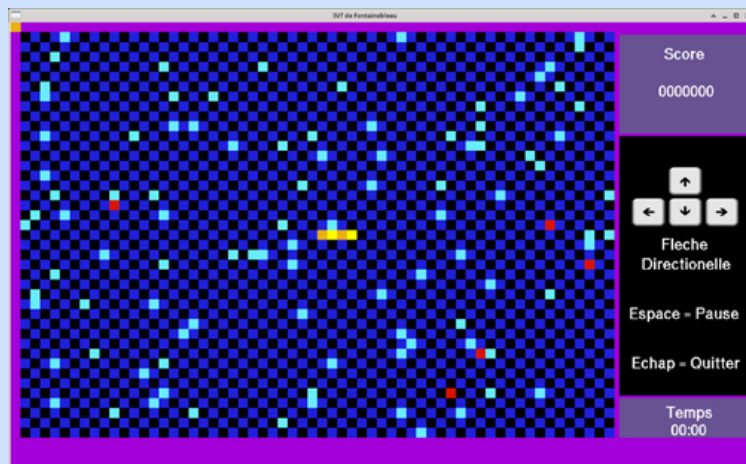
Capture d'Écran :



- Lorsque le programme démarre, une barre de chargement s'affiche dans le terminal. L'utilisateur peut lancer le jeu en appuyant sur la touche "Entrée". Une image représentative apparaît alors, signalant le début du jeu.

Interface Divisée en Quatre Sections :

Capture d'Écran :



- **Section Principale (1ère Case) :**
 - Le terrain de jeu principal, où le serpent, les pommes, et les obstacles évoluent. Le damier a été intégré pour une meilleure disposition des éléments.
- **Section Score (2ème Case) :**
 - Affiche le score du joueur, augmentant par paliers de 5 points.
- **Section Touches (3ème Case) :**
 - Liste des touches que le joueur peut utiliser pour contrôler le serpent.
- **Section Temps (4ème Case) :**
 - Affiche le temps écoulé depuis le début du jeu.

FONCTIONNALITÉS 2

Image de Fin de Partie :

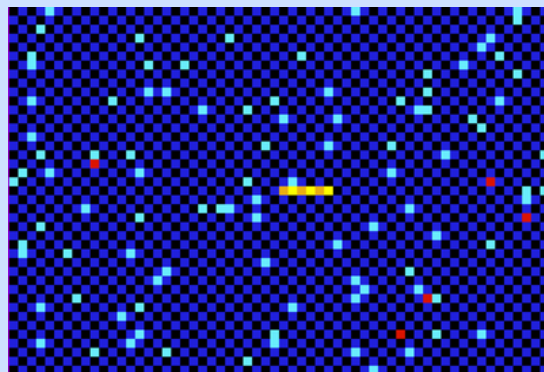
Capture d'Écran :



Lorsque le joueur perd la partie, une image s'affiche, présentant le score atteint et le temps écoulé. Cela permet au joueur de réfléchir à sa performance.

Initialisation du Damier :

Capture d'Écran :



Un damier a été mis en place, facilitant l'organisation visuelle du jeu. Chaque case du damier correspond à une unité de mesure, simplifiant le positionnement des pommes, des obstacles et du serpent.

Des obstacles ont été introduits dans le damier. Lorsqu'ils sont touchés par le serpent, cela déclenche la fin de la partie, affichant l'image de fin de partie et le score final.

FONCTIONNALITÉS 3

Barre de chargement :

Capture d'Écran :



Lorsque le programme démarre, une barre de chargement s'affiche dans le terminal. L'utilisateur peut lancer le jeu en appuyant sur la touche "Entrée". Une image représentative apparaît alors, signalant le début du jeu.

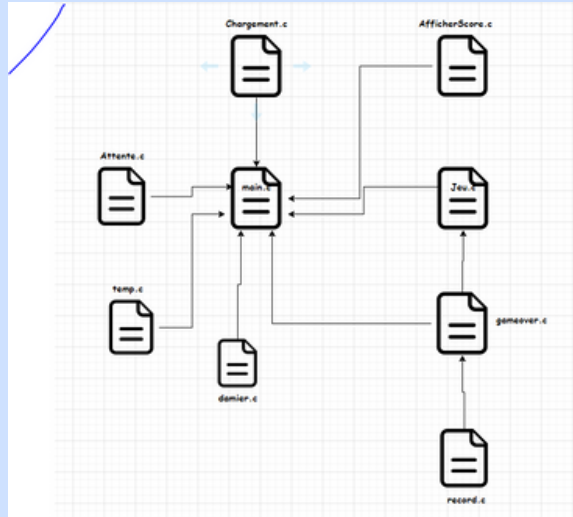
Fichier score.txt :

Capture d'Écran :

2	20
3	10
4	0
5	0
6	15
7	20
8	10
9	20
10	35
11	15
12	

Un fichier de liste de scores a été ajouté, permettant au joueur de visualiser son score actuel ainsi que le meilleur score enregistré. Cette fonctionnalité ajoute une dimension compétitive au jeu, incitant les joueurs à améliorer constamment leurs performances.

DÉCOUPAGE DE FICHIERS



Le découpage en différents fichiers source dans notre projet "Snake" a été une décision délibérée visant à améliorer la lisibilité, la maintenance et la modularité du code. Chaque fichier source est dédié à une tâche spécifique, contribuant ainsi à une organisation claire du projet. Voici les justifications pour chaque fichier :

temp.c:

Justification : Gère l'affichage du temps tout au long de la partie. Séparer cette fonctionnalité dans un fichier distinct permet de maintenir une logique claire concernant la gestion du temps, facilitant des ajustements ultérieurs si nécessaire.

attente.c :

Justification : Responsable de la pause du jeu, permettant au joueur de prendre des moments de réflexion sans compromettre le déroulement global. Cette séparation garantit une gestion propre de l'attente sans encombrer le fichier principal du jeu.

record.c :

Justification : Connecté à la logique de "gameover", il enregistre le score du joueur dans un fichier externe ("score.txt") et affiche le meilleur score atteint. Isoler cette fonctionnalité contribue à la maintenabilité du code, facilitant l'extension des fonctionnalités liées aux scores.

damier.c :

Justification : Gère l'affichage du damier de dimensions spécifiques et initialise les composants du jeu (obstacles, serpent, pommes). Ce découpage facilite la gestion visuelle et structurelle du damier sans surcharger le fichier principal.

DÉCOUPAGE DE FICHIERS 2

`gameover.c` :

Identifie et gère les différentes conditions de "gameover" (collisions avec obstacles, bordures, serpent). La séparation de ces logiques complexes améliore la clarté du code principal et facilite les futures modifications.

`chargement.c` :

Contient la logique de démarrage du jeu, notamment l'initialisation d'une fenêtre. Séparer cette phase d'initialisation contribue à une gestion plus propre du lancement du jeu.

`afficherScore.c` :

Dédie un fichier à l'affichage du score pendant la partie. Cela rend cette fonctionnalité distincte et facilement modifiable sans affecter la logique centrale du jeu.

`jeu.c` :

Contient la logique principale du jeu, y compris l'initialisation des composants, les mécanismes de jeu (manger des pommes, éviter des obstacles) et la gestion des collisions. La séparation de ces aspects clés assure une lisibilité accrue et une maintenance simplifiée.

`main.c` :

Agit comme le point d'entrée du programme, reliant toutes les fonctionnalités du jeu. Intègre la logique du démarrage du jeu, la gestion des entrées utilisateur et l'appel des différentes fonctions de jeu. C'est ici que la traduction du code en image du jeu est orchestrée.

Ce découpage modulaire offre une structure cohérente au projet, facilite la collaboration avec le coéquipier, et permet des modifications ciblées sur des aspects spécifiques du jeu sans perturber l'ensemble.

EXPLICATION DES DONNÉES REPRESENTANT LE SERPENT

Au cœur du programme du jeu de serpent en langage C, la structure de données essentielle est appelée "Serpent". Cette structure est définie comme suit :

```
typedef struct {  
    Position *corps;  
    int longueur;  
} Serpent;
```

Position : Il s'agit d'une structure représentant les coordonnées x et y d'une position dans l'espace. Les coordonnées x et y indiquent la position d'un élément dans un repère, permettant ainsi de localiser chaque partie du corps du serpent sur l'écran de jeu.

corps : Cette partie de la structure est un pointeur vers un tableau dynamique de positions. En d'autres termes, le "corps" du serpent est une séquence dynamique de positions dans l'espace. Chaque élément de ce tableau représente une partie distincte du serpent, et le pointeur permet de gérer de manière flexible la taille du serpent en ajoutant ou supprimant des parties.

longueur : Il s'agit d'un entier qui indique la longueur actuelle du serpent. En d'autres termes, il reflète le nombre de parties qui composent le corps du serpent. Cette information est cruciale pour suivre l'évolution du serpent au fil du jeu, notamment lorsqu'il se déplace, mange une pomme, ou lorsqu'il y a des collisions.

TRANSFORMATIONS AU COURS DE LA PARTIE

- Dessiner Damier :

Avant d'entamer l'initialisation du serpent, le damier, représentant l'espace de jeu, est mis en place. Ce damier est une structure fondamentale pour offrir un contexte visuel au mouvement du serpent. Chaque carré du damier représente une cellule dans laquelle le serpent peut se déplacer. La fonction Dessiner Damier attribue des propriétés visuelles aux cellules, créant ainsi un fond sur lequel le serpent évoluera.

- Initialisation du Serpent :

La fonction initialiserS assume la responsabilité cruciale d'initialiser le serpent avec une longueur de 10, tout en positionnant ses parties initiales sur l'écran. Cette phase, essentielle au commencement de chaque partie, instaure un serpent prêt à être contrôlé. En définissant son état initial, cette fonction crée les conditions nécessaires pour démarrer une partie du jeu, établissant ainsi les bases du déroulement ultérieur.

- Déplacement du Serpent :

La fonction deplacerS orchestre avec précision le déplacement du serpent. En ajustant les positions de son corps en fonction de la direction choisie (droite, bas, gauche, haut), cette fonction crée l'illusion d'un serpent en mouvement. Chaque déplacement successif du serpent semble être accompagné d'une "mangeaison" de positions, conférant au mouvement une fluidité visuelle qui contribue à l'expérience immersive du jeu.

- Croissance du Serpent :

Lorsque le serpent "mange" une pomme, la fonction mangerPomme prend en charge l'expansion du serpent en intensifiant sa longueur. Cette augmentation de taille est réalisée en incorporant de nouvelles parties à son corps. Un aspect crucial de cette opération est la réallocation dynamique d'espace pour le tableau de positions du corps du serpent. Cette gestion dynamique de la mémoire permet au serpent de s'adapter à sa nouvelle taille, garantissant une croissance fluide et efficiente.

- Gestion des Collisions :

La fonction collisionAvecS opère une surveillance constante des collisions entre la tête du serpent et ses autres parties. La détection d'une telle collision indique la fin de la partie, marquant un point crucial dans le déroulement du jeu.

La fonction collisionAvecBordures assume la tâche d'identifier si le serpent entre en collision avec les bordures de l'écran. Cette vérification est essentielle pour éviter que le serpent ne sorte de l'espace de jeu défini, maintenant ainsi l'intégrité des règles du jeu.

La fonction collisionAvecObstacles se consacre à l'inspection des éventuelles collisions entre le serpent et des obstacles générés aléatoirement sur l'écran. Cela introduit un élément de complexité dans le jeu, ajoutant un défi supplémentaire au joueur en obligeant le serpent à naviguer habilement à travers des obstacles mouvants.

CONCLUSION

Abed Bridja

La réalisation du projet "Snake" a été une expérience riche en enseignements et en défis, marquant une étape significative dans mon parcours de programmation. Mon ressenti à l'égard de ce projet est incarné entre l'enthousiasme face à sa concrétisation et la reconnaissance des défis qui ont été difficiles à surmonter.

Au cours de ce projet, j'ai été confronté à diverses difficultés, qu'elles soient liées à la gestion des structures de données, aux mouvements du serpent mais ces défis ont été autant de tremplins pour renforcer la compréhension des concepts de programmation, me confrontant à des problématiques concrètes et stimulantes.

Néanmoins, chaque difficulté surmontée a été une victoire personnelle. La résolution de problèmes m'a poussé à explorer de nouvelles approches, à affiner mes compétences, et à approfondir ma compréhension du langage C notamment un moyen d'application du cours. Ces moments d'incertitude ont été essentiels pour la croissance en tant que débutant en programmation, m'invitant à repousser mes limites et à rechercher des solutions créatives. En fin de compte, le projet "Snake" a été plus qu'une simple réalisation technique. Il a été une aventure, une exploration personnelle de mes compétences et de ma créativité. Cette expérience a élargi mon champ de vision quant à ce que je peux accomplir en programmation, et m'a donné une nouvelle appréciation pour la fusion de la logique algorithmique et de l'expression artistique dans le monde du développement logiciel.

Solar Dimitri

Avec mon coéquipier, nous avons réparti les rôles de manière à maximiser notre progression sur ce projet. Bien que le projet ait semblé simple au départ, la réalité s'est avérée plus complexe, présentant des défis auxquels nous n'avions pas initialement pensé. La gestion du temps s'est avérée être une contrainte majeure, dépassant nos prévisions initiales et nous obligeant à résoudre des problèmes inattendus, comme l'affichage du temps en faisant apparaître un rectangle.

Sur le plan visuel, déterminer les dimensions optimales pour l'écran principal, l'écran de jeu, le damier, etc., s'est révélé être un défi de taille. La recherche des bonnes proportions a demandé une réflexion approfondie pour garantir une expérience visuelle agréable et cohérente.

Ce projet a constitué une opportunité précieuse qui m'a permis de renforcer ma compréhension du langage C. Les défis rencontrés ont été autant d'occasions d'apprendre et de perfectionner mes compétences en résolution de problèmes. En outre, travailler en équipe m'a offert une perspective nouvelle, améliorant ma collaboration et ma communication avec mon coéquipier.