Hamza BAZZAOU

Yanis HAMOUDI

Rapport SAé

Sommaire :

———————————————————————————————

1 • Introduction

Dans ce Système d'Apprentissage Expérimental (SAé), nous devions programmer un jeu Snake en langage C89. Le but du joueur est de contrôler une longue et fine ligne, semblable à un serpent, qui doit slalomer entre les bords de l'écran et les obstacles qui parsèment le niveau. Pour gagner chaque niveau, le joueur doit faire manger à son serpent un certain nombre de pastilles, similaires à de la nourriture (dans notre cas, il s’agira de pommes), allongeant à chaque fois la taille du serpent. Alors que le serpent avance inexorablement, le joueur ne peut que lui

indiquer une direction à suivre (en haut, en bas, à gauche, à droite) afin d'éviter que la tête du serpent ne touche les murs ou son propre corps, auquel cas il risque de mourir.

2 • Description des fonctionnalités du programme

Dans cette partie, nous allons décrire toutes les fonctionnalités du jeu chronologiquement :

1) Menu

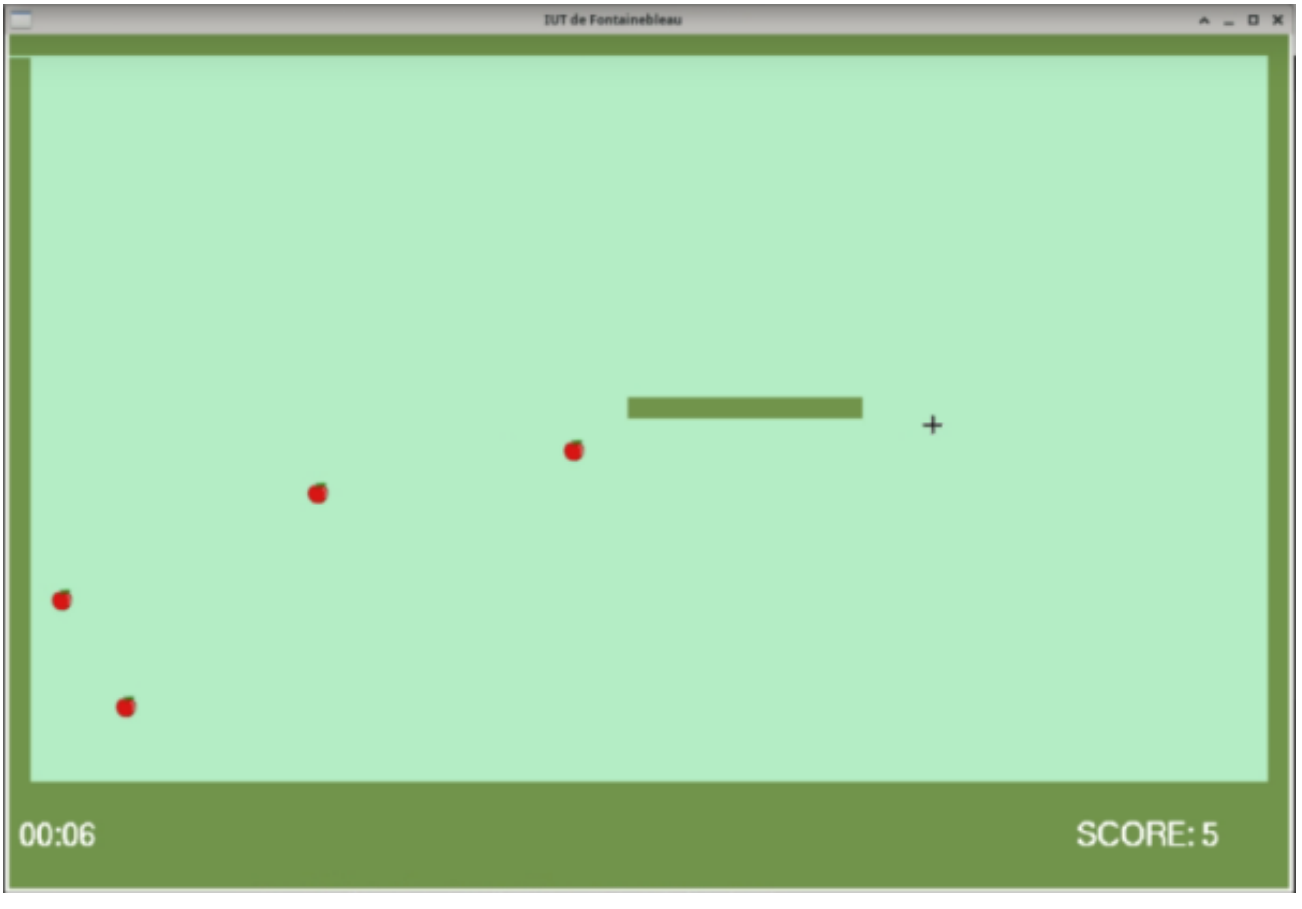
Sur cet interface on peut apercevoir 3 boutons "Commencer", "Mode Obstacles" et "Exit" qui nous indique de choisir le mode de jeu auquel on souhaite jouer ou bien qi on veut quitter le jeu. Chaque mode renvoie à une fonction appeler "terrain" qui va crée un tableau avec un nombre de colonne et de ligne identiques à chaque mode. Mais à chaque mode, la fonction jeu changera pour rendre le niveau plus difficile.

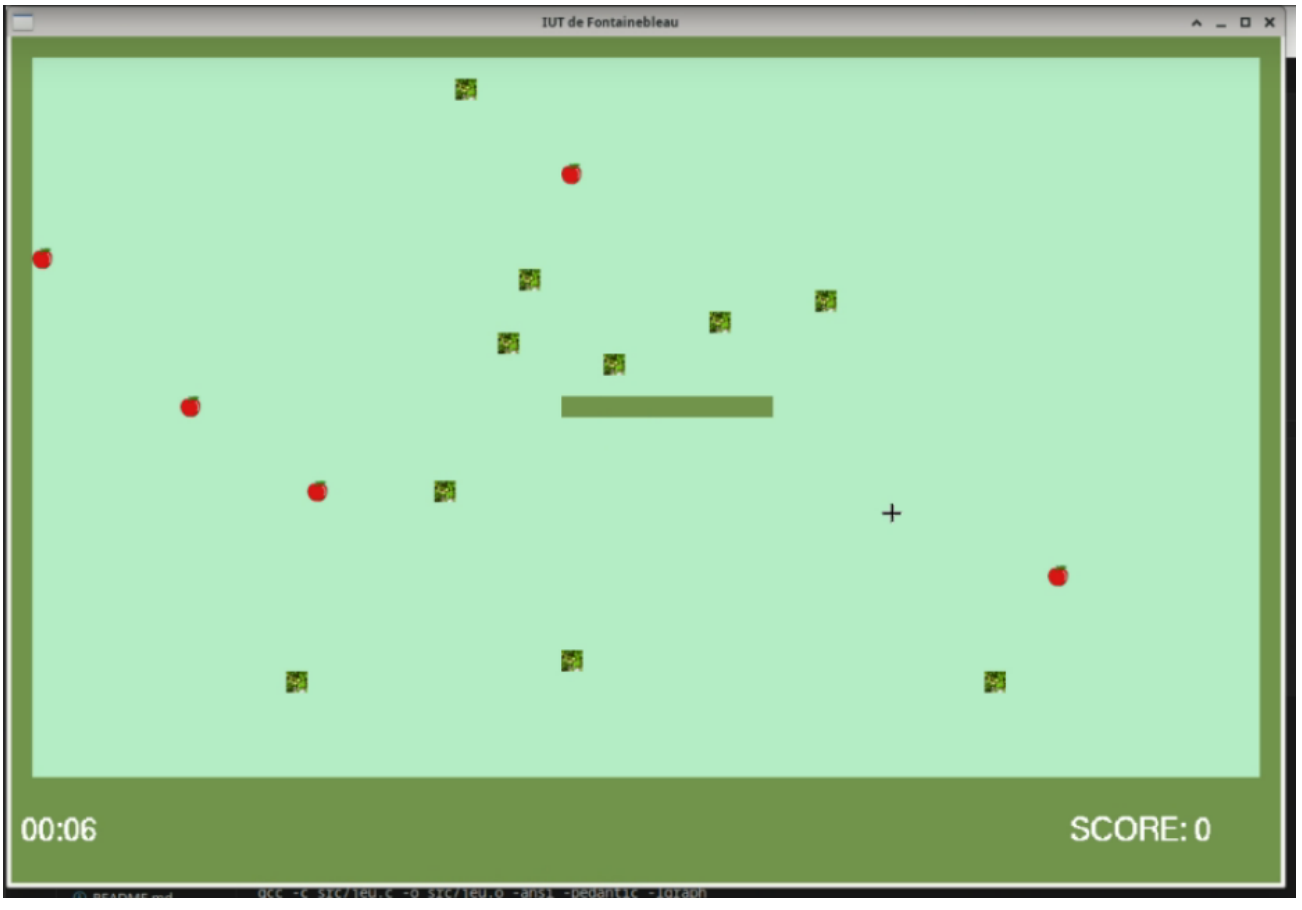
2) Les Modes

Sur cette interface, on peut apercevoir 3 boutons "Commencer", "Mode Obstacles" et "Exit" qui nous indiquent de choisir le mode de jeu auquel on souhaite jouer ou bien si on veut quitter le jeu. Chaque mode renvoie à une fonction appelée "terrain" qui va créer un tableau avec un nombre de colonnes et de lignes identiques à chaque mode. Mais à chaque mode, la fonction jeu changera pour rendre le niveau plus difficile.

3) Début du jeu

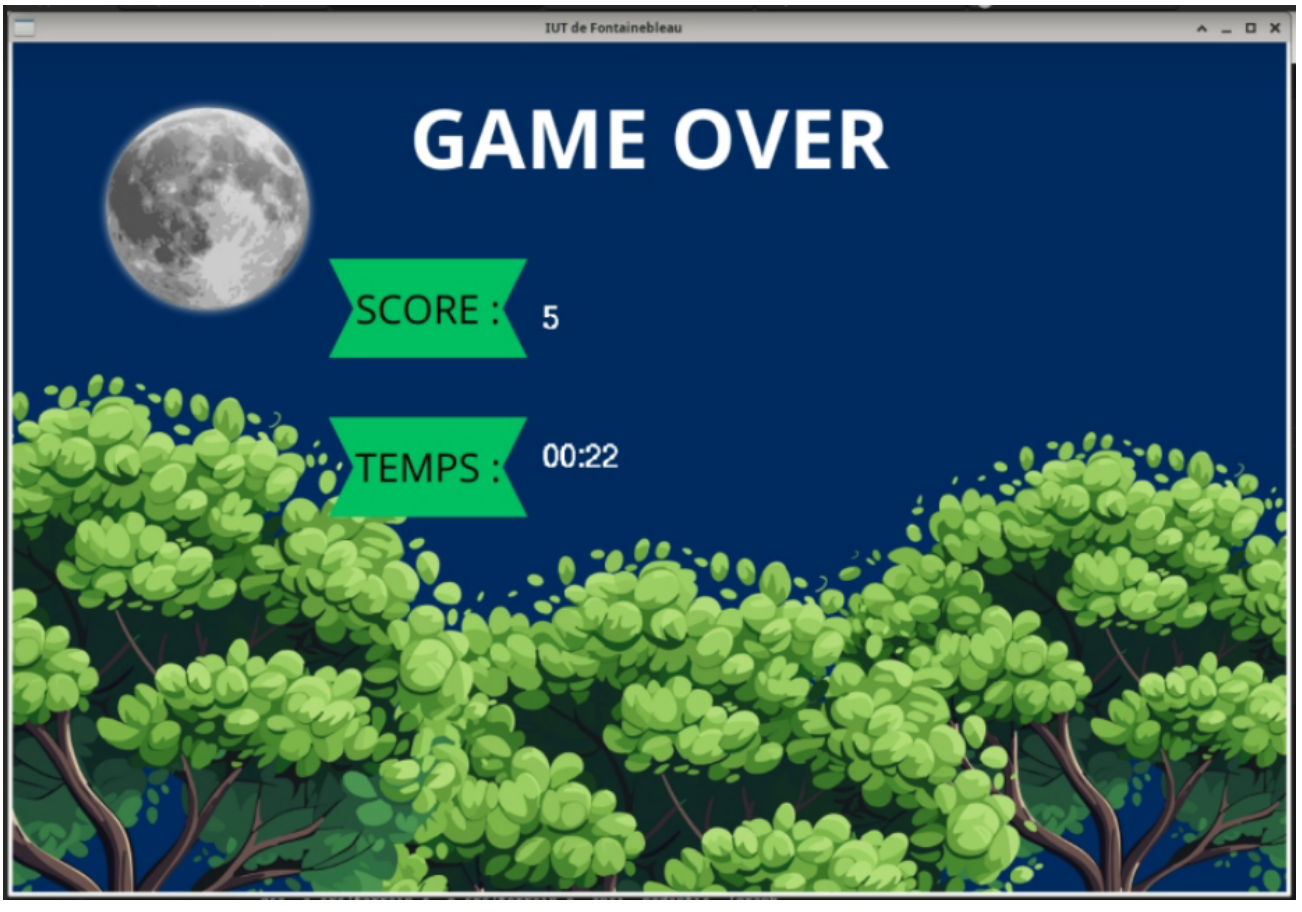
Après avoir passé le menu, nous arrivons sur l'écran de jeu. Sur cet écran, un chronomètre permet au joueur de voir le temps écoulé pendant qu'il joue. De plus, nous avons implémenté un compteur de score qui augmente à chaque pastille que le joueur/serpent mange.



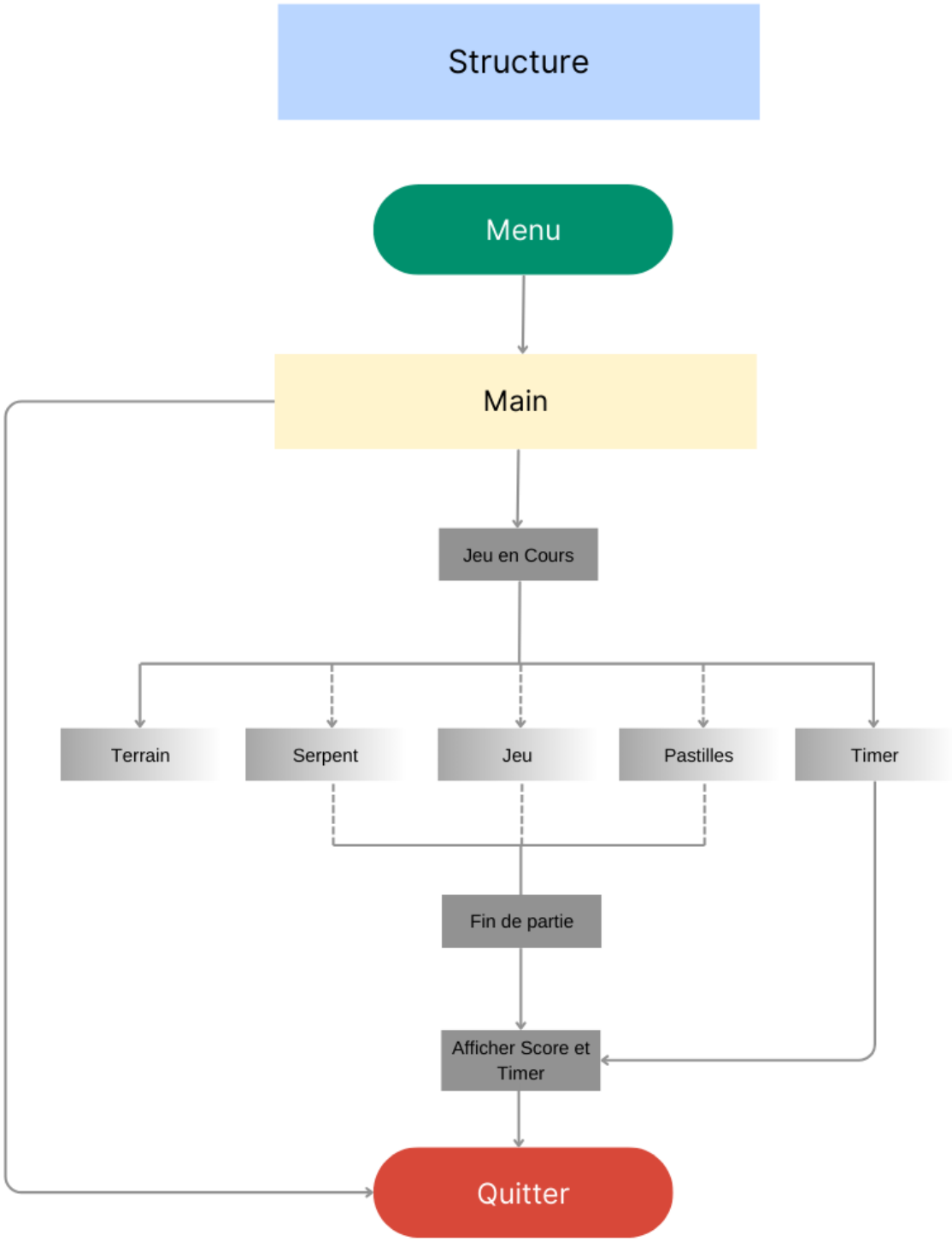


4) Fin du jeu

Dès que le serpent va heurter l'un des 4 rebords ou qu'il va toucher une partie de son corps, une fonction "jeu" sera appelée pour afficher l'écran de fin. Sur cet écran, nous retrouvons, d'une part, le temps final du joueur donné grâce au chronomètre, ainsi que le score réalisé pendant la partie. Pour quitter, il vous suffira d'appuyer sur le bouton "Échap".



2 • Structures



1. **main.c :** Gère le démarrage du jeu, l'affichage du menu, et appelle les fonctions appropriées en fonction des choix du joueur.
2. **jeu.c :** Contient la logique principale du jeu, y compris la gestion du serpent, des obstacles, des pastilles, et de la fin de partie.
3. **serpent.c :** Gère les aspects spécifiques au serpent, tels que son affichage, sa mise à jour, et les conditions de défaite liées à son mouvement.
4. **terrain.c :** S'occupe de l'affichage du terrain, y compris le fond, les obstacles, et d'autres éléments visuels.
5. **timer.c :** Gère le minuteur du jeu, mesurant le temps écoulé pendant la partie.

6. **menu.c :** Gère l'affichage du menu d'accueil, les choix du joueur, et la navigation entre les différentes options.

3 • Données

Structures du Programme :

Nous avons choisi d'organiser notre programme en utilisant des structures pour représenter les principales entités du jeu, notamment **jeu** (JEU), **terrain** (TERRAIN), **serpent** (SERPENT), **temps** (TIMER), et **pastille** (PASTILLE).

1. **Structure JEU (Jeu_s) :**
   - La structure Jeu_s encapsule les informations relatives au jeu, telles que la direction du serpent (*direction*), la dernière direction (*last_direction*), un indicateur du statut du jeu (*jeu_en_cours*), le score, une variable associée à la touche (*touche*), et un indicateur de pause (*paused*).
   - L'alias de type *JEU* (typedef struct Jeu_s Jeu) crée une référence concise pour la structure Jeu_s.
2. **Structure TIMER (Timer_s) :**
   - La structure Timer_s représente le chronomètre du jeu avec des champs pour les secondes (*seconde*), les minutes (*minute*), la valeur actuelle des secondes (*seconde_actuel*), l'ancienne valeur des secondes (*old_seconde*), une chaîne de caractères représentant le temps (*timer*), et un champ supplémentaire de type unsigned long int.
   - L'alias de type *TIMER* (typedef struct Timer_s TIMER) crée une référence simplifiée pour la structure Timer_s.
3. **Structure TERRAIN (Terrain_s) :**
   - Cette structure définit les caractéristiques du terrain de jeu, avec des champs pour les coordonnées (x et y) et la couleur de fond (*fond*).
   - L'alias de type *TERRAIN* (typedef struct Terrain_s TERRAIN) offre une manière abrégée de déclarer des variables de type struct Terrain_s.
4. **Structure SERPENT (Serpent_s) :**
   - La structure Serpent_s englobe les propriétés du serpent, y compris un identifiant (*serpent*), le nombre de segments (*segment*), les positions actuelles (*pos_x et pos_y*), et les positions antérieures (*old_x et old_y*).
   - L'alias de type *SERPENT* (typedef struct Serpent_s SERPENT) crée une référence facilitant l'utilisation de la structure Serpent_s.
5. **Structure PASTILLE (Pastille_s) :**
   - Cette structure contient des informations relatives aux pastilles du jeu, comprenant un identifiant (*p*), le nombre de pastilles (*pastille*), ainsi que les positions des pastilles (*pastillex et pastilley*).
   - L'alias de type *PASTILLE* (typedef struct Pastille_s PASTILLE) simplifie la déclaration de variables de type struct Pastille_s.

L'utilisation de structures permet une organisation claire et une manipulation efficace des données, contribuant ainsi à une conception modulaire et compréhensible du programme. Les alias de type ajoutent une couche de lisibilité supplémentaire à notre code.

4 • Conclusion

Hamza BAZZAOU :

Ce SAE a constitué une opportunité clé pour développer une expertise approfondie en langage C89 et maîtriser l'outil de gestion de version Git. Bien que le projet ait été achevé dans les délais impartis, il offre encore de nombreuses perspectives d'amélioration, comme l'introduction de nouveaux modes de jeu pour renforcer sa difficulté comme par exemple les obstacles qui change de position à chaque fois que le serpent mange une pomme. Cette expérience a été enrichissante, car elle montre des similitudes avec le monde du travail en entreprise. Bosser en équipe a vraiment montré que si on ne se parle pas assez, ça peut tout foutre en l'air. Du coup, cette expérience m'a fait comprendre que se coordonner et travailler ensemble, c'est le secret pour que le projet fonctionne.

Yanis HAMOUDI :

La réalisation de ce SAE a constitué une opportunité cruciale pour approfondir mes compétences en langage C89 et acquérir une maîtrise avancée de l'outil de gestion de version Git. Bien que le projet ait été mené à terme dans les délais impartis, il offre encore de nombreuses perspectives d'amélioration, notamment par l'introduction de nouveaux modes de jeu pour renforcer sa complexité. Cette expérience s'est avérée particulièrement enrichissante, reflétant des similitudes avec le monde professionnel en entreprise. Travailler en équipe a souligné l'importance cruciale de la communication : l'absence de dialogue peut compromettre l'ensemble du projet. Ainsi, cette expérience a renforcé ma conviction que la coordination et la collaboration sont les clés du succès d'un projet.