

Rapport SAE 1.1

Lenny FOULOU, Groupe 3
Théo GROLL, Groupe 6

Sommaire

- **Description**
- **Histoire du jeu**
- **Fonctionnalités**
- **Découpage des fichiers**
- **Données du serpent**
- **Conclusion personnelle**

Description

Le jeu du serpent, connu sous le nom de Snake, met en scène un serpent mobile évoluant dans une grille. Le joueur contrôle ce serpent afin de collecter des pastilles disséminées sur le terrain, tout en évitant de percuter les limites de la grille ou sa propre queue. Chaque pastille mangée fait croître le serpent. La partie se termine si le serpent entre en collision avec les bords du terrain ou avec lui-même. Le jeu offre des commandes simples permettant au joueur de diriger le serpent à l'aide des touches directionnelles et de mettre en pause ou quitter la partie selon ses besoins. Notre projet vise à recréer cette expérience de jeu classique en utilisant le langage C89 et une bibliothèque graphique spécifique, en intégrant des fonctionnalités additionnelles pour enrichir le gameplay.

Histoire du jeu

Dans le lointain royaume de Far Far Away, un serpent méchant évoluait dans les vastes plaines, insatiable et vorace. Ce serpent était différent, non pas par sa nature, mais par ses désirs. Il était affamé, non pas de proies ordinaires, mais de pommes scintillantes et juteuses qui poussaient dans les jardins enchantés de l'endroit.

Mais avec le passage du temps, une présence étrange commença à se manifester dans ces terres magiques. Tout d'abord, l'imposant Shrek, le protecteur bien-aimé du royaume, apparut. Sa mission : protéger le royaume des forces du mal.

Puis, émergeant des ombres, le célèbre Chat Potté fit son entrée. Agile et rusé, il se joignit à la quête pour vaincre le serpent méchant, cherchant à protéger les précieuses pommes.

Cependant, le véritable tour de l'intrigue se produisit lorsque Jo, l'âne de compagnie fidèle de Shrek, se joignit à la bataille. Jo se sentit investi d'une mission sacrée : aider à sauver le royaume en empêchant le serpent de dévorer toutes les pommes et d'étendre son emprise maléfique sur ces terres enchantées.

Et ainsi, le combat était lancé. Le serpent méchant, avec sa faim insatiable pour les pommes, était confronté à ces héros mythiques, chacun ayant ses propres intentions.

L'objectif ultime était clair : le serpent méchant devait être arrêté avant qu'il ne dévore la centième pomme, déclenchant ainsi une calamité sur le royaume de Far Far Away.

Fonctionnalités

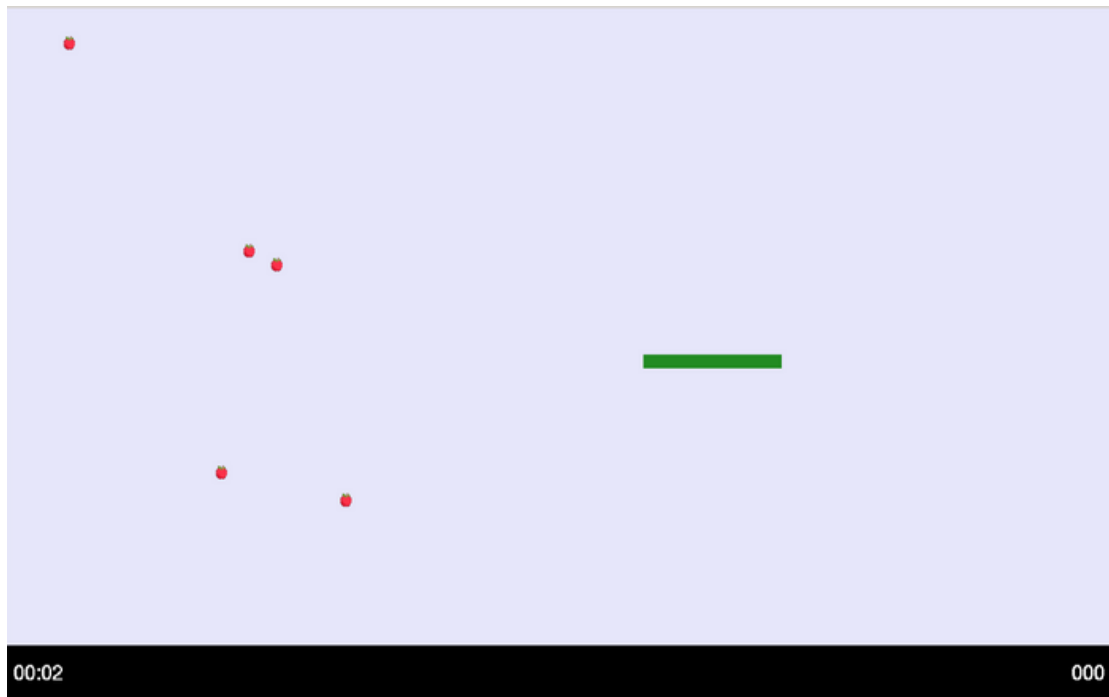
Écran de démarrage :



L'écran de démarrage présente une interface sobre et intuitive. Deux boutons distincts sont affichés, offrant au joueur des choix clairs pour interagir avec le jeu.

1. Bouton 'Jouer' : Ce bouton est destiné à lancer le jeu. En cliquant dessus, le joueur accède à la fenêtre de jeu où il pourra contrôler le serpent, collecter des pommes et interagir avec les différents éléments du jeu.
2. Bouton 'Quitter' : Ce bouton permet de fermer la fenêtre du jeu. En appuyant sur ce bouton, le joueur quitte l'application et revient à l'environnement ou à la plateforme d'où il a lancé le jeu.

Fenêtre de jeu :



La fenêtre de jeu présente une zone centrale délimitée par un fond violet clair, constituant l'espace jouable où se déroule l'action principale du jeu. Cette zone représente l'environnement avec lequel évolue le serpent et où les pommes apparaissent pour être collectées.

Le joueur contrôle le serpent dans cette zone violette, avec pour objectif de collecter les pommes tout en évitant les obstacles et les limites imposées. Plusieurs conditions sont définies pour la défaite du joueur :

- Collision avec sa propre queue : Si le serpent entre en contact avec sa propre queue, la partie est perdue.
- Collision avec les bords de la fenêtre : Si le serpent atteint les bords de la fenêtre, la partie est également perdue.
- Collision avec le rectangle noir en bas de la fenêtre : si le serpent entre en contact avec ce rectangle, la partie est perdue.

Dans le jeu, cinq pommes sont réparties aléatoirement dans la zone de jeu. Lorsque le serpent passe sur une pomme, sa taille augmente de deux cases. Cette mécanique incite le joueur à collecter les pommes pour accroître le score et la taille du serpent.

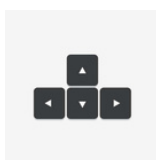
Le temps écoulé est affiché dans le coin inférieur gauche, tandis que le score s'affiche dans le coin inférieur droit. Le score est calculé en multipliant par cinq le nombre de pommes consommées par le serpent. Cette disposition permet au joueur de suivre facilement le temps passé et d'évaluer son score en fonction du nombre de pommes collectées.



Pause et Reprise du Jeu : En appuyant sur la touche "Espace", le jeu se met en pause. Pour reprendre le jeu après une pause, il suffit de réappuyer sur la touche "Espace". Cette fonctionnalité permet au joueur de suspendre temporairement l'action pour reprendre le contrôle du serpent quand il le souhaite.

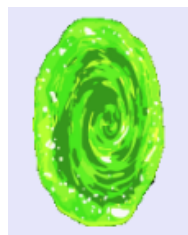


Quitter le Jeu : Pour fermer la fenêtre du jeu et quitter l'application, le joueur peut utiliser la touche "Échap". Cette action permet une sortie rapide du jeu, ramenant l'utilisateur à l'environnement ou à la plateforme d'où il a lancé le jeu.



Contrôle du Serpent : Les touches directionnelles (flèches ← ↑ → ↓) permettent de déplacer le serpent dans la direction souhaitée. À noter que le serpent est en mouvement constant, se déplaçant dans sa dernière direction connue si aucune touche n'est pressée. Cette mécanique engage le joueur à réagir rapidement pour éviter les collisions et à planifier ses déplacements pour atteindre les objectifs du jeu.

Lorsque le joueur atteint un certain score, des avertissements sont déclenchés pour annoncer l'arrivée imminente des boss. Chaque boss est précédé par un effet visuel unique à l'emplacement auquel il va apparaître, signalant ainsi sa venue. Ces indications surviennent 10 points avant l'apparition de chaque boss, soit l'équivalent de deux pommes consommées.



L'apparition des boss est liée à des seuils spécifiques dans le jeu. À des paliers de score particuliers, soit respectivement 30 points (équivalant à 6 pommes consommées), 200 points (soit 40 pommes), et 350 points (équivalant à 70 pommes), des boss font leur apparition. Ces étapes-clés marquent l'intensification du défi pour le joueur, car l'arrivée de ces boss représente des obstacles supplémentaires à surmonter.



Si le joueur entre en collision avec un boss pendant le jeu, cela entraîne une défaite immédiate.

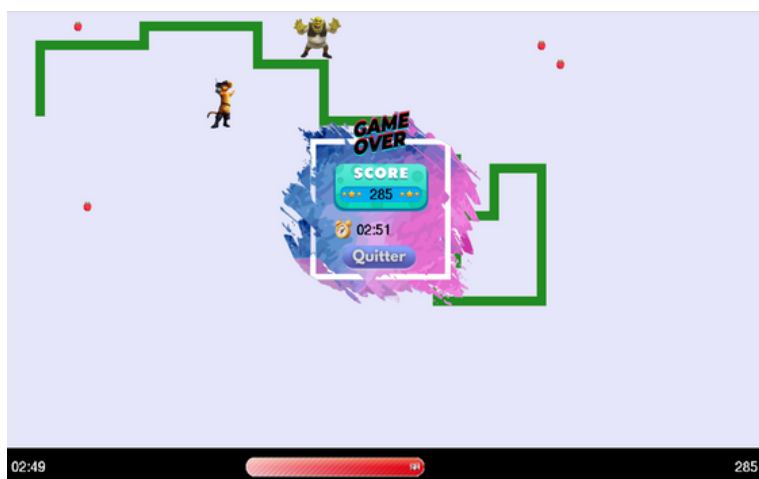
Chaque boss possède ses propres mouvements distinctifs et sa propre vitesse lorsqu'il apparaît dans le jeu. Le comportement spécifique de chaque boss ajoute une dimension stratégique au défi, demandant au joueur de s'adapter à des mouvements et des vitesses uniques pour survivre.

Dans la fenêtre de jeu, une barre de progression apparaît au centre en bas. Elle se met à jour tous les 10% et progresse en fonction des étapes franchies vers la condition de victoire. Cette barre reflète la progression du joueur vers l'objectif de score, fixé à 500 points (équivalent à la consommation de 100 pommes). Cela permet au joueur de suivre visuellement sa progression vers la victoire au cours de la partie.

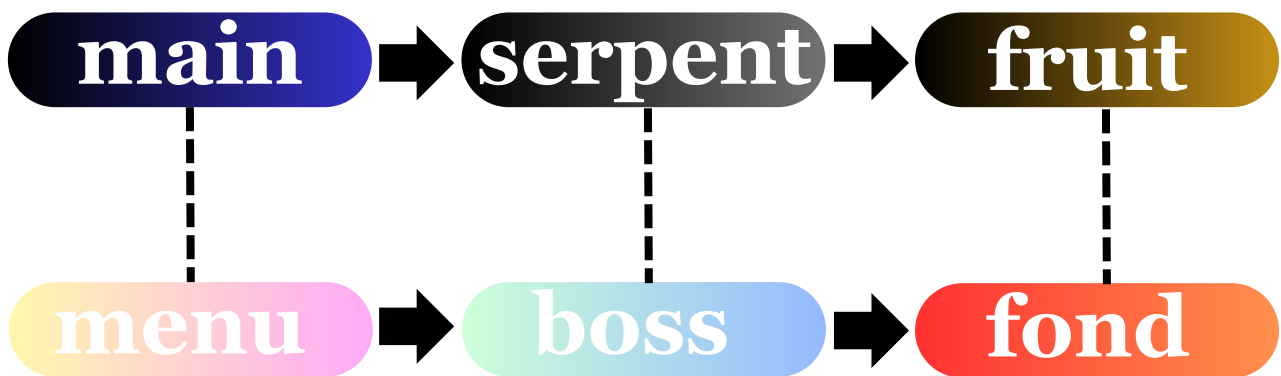


Écran de fin :

Lorsque le joueur perd, le jeu s'interrompt et affiche au centre de la fenêtre l'écran de défaite, si la condition de victoire de 500 points n'a pas été atteinte, ou l'écran de victoire dans le cas contraire. Ces écrans présentent le temps écoulé et le score obtenus durant la partie. De plus, un bouton "Quitter" est disponible ; en appuyant dessus, le joueur peut fermer le jeu.



Découpage des fichiers



Justification du découpage en différents fichiers source :

1. Modularité et lisibilité : Diviser le code en fichiers distincts permet de regrouper des fonctionnalités ou des composants similaires dans des modules spécifiques, facilitant ainsi la compréhension et la maintenance du code.
2. Facilité de gestion : Des fichiers plus petits et spécifiques réduisent la complexité globale du projet, ce qui facilite la navigation et la localisation des parties spécifiques du code.
3. Réutilisabilité du code : En séparant le code en modules distincts, il devient possible de réutiliser des fonctionnalités ou des classes dans d'autres projets sans avoir à transporter l'intégralité du code.

Dans ce diagramme, chaque fichier source représente un module distinct du projet. Par exemple, `main.c` est le point d'entrée principal, `serpent.c` gère la logique liée au serpent, `fruit.c` s'occupe de la gestion des fruits, et ainsi de suite.

Ce découpage permet une séparation claire des responsabilités, rendant le code plus modulaire, facile à comprendre, à maintenir et à développer. Chaque fichier source encapsule une fonctionnalité spécifique, ce qui améliore la lisibilité et la maintenabilité globale du projet.

Données du serpent

Dans le fichier serpent.c, le serpent est représenté par une structure Serpent qui contient les données le décrivant et les fonctions pour le manipuler. Voici une explication des données et des transformations que celles-ci peuvent subir au cours du jeu :

Explication des données du serpent :

- longueur: Représente la longueur actuelle du serpent en nombre de parties du corps.
- directionX, directionY: Indiquent la direction du déplacement du serpent. Ils sont utilisés pour déplacer le serpent dans le plan XY.
- corps: Un tableau représentant les différentes parties du corps du serpent. Chaque partie a une position x et y pour sa localisation sur l'écran.

Transformations possibles du serpent :

1. InitialiserSerpent() :
 - Définit les valeurs initiales du serpent, telles que sa longueur, sa position de départ et sa direction.
2. DeplacerSerpent() :
 - Déplace le serpent en mettant à jour les positions de chaque partie de son corps. Chaque partie est déplacée vers l'emplacement de la partie précédente dans le tableau, et la tête du serpent est déplacée dans la direction spécifiée par directionX et directionY.
3. VerifierCollision() :
 - Vérifie s'il y a une collision du serpent avec les bords de l'écran ou avec son propre corps. Si une collision est détectée, la fonction renvoie un indicateur d'erreur (1 pour collision, 0 sinon).
4. DessinerSerpent() :
 - Dessine le serpent à l'écran en utilisant la bibliothèque graphique. Chaque partie du corps est représentée par un rectangle coloré.

Au cours du jeu, le serpent peut changer de direction, sa longueur peut augmenter, et il peut se déplacer tout en vérifiant s'il entre en collision avec lui-même ou les limites de l'écran. Ces transformations sont gérées par les fonctions définies dans serpent.c, permettant ainsi de contrôler le mouvement et l'affichage du serpent dans le jeu.

Conclusion personnelle

“Cette expérience a été enrichissante pour moi. La conception de ce projet m’a tout d’abord appris à mener un travail de recherche. Le fait de faire des essais, d’échouer et par la suite de réussir m’a montré, je pense, une facette du métier de développeur. Travailler pendant un longtemps et réussir procure une satisfaction inexplicable et me donne envie de continuer dans cette voie. J’ai également apprécié travailler avec mon binôme. Nous avons entretenu une certaine forme de synergie pour ce projet et j’espère pouvoir à nouveau collaborer avec lui.”

Lenny FOULOU

“La réalisation de ce projet a été une aventure passionnante et enrichissante. Ce défi m’a permis de découvrir les bases du développement de jeux et de consolider mes compétences en programmation. Travailler en équipe a été une expérience gratifiante, où la collaboration et la résolution de problèmes ont été les clés. Cette expérience m’a inspiré à poursuivre dans cette voie et à approfondir mes connaissances.”

Théo GROLL