Jeu du Serpent - Rapport

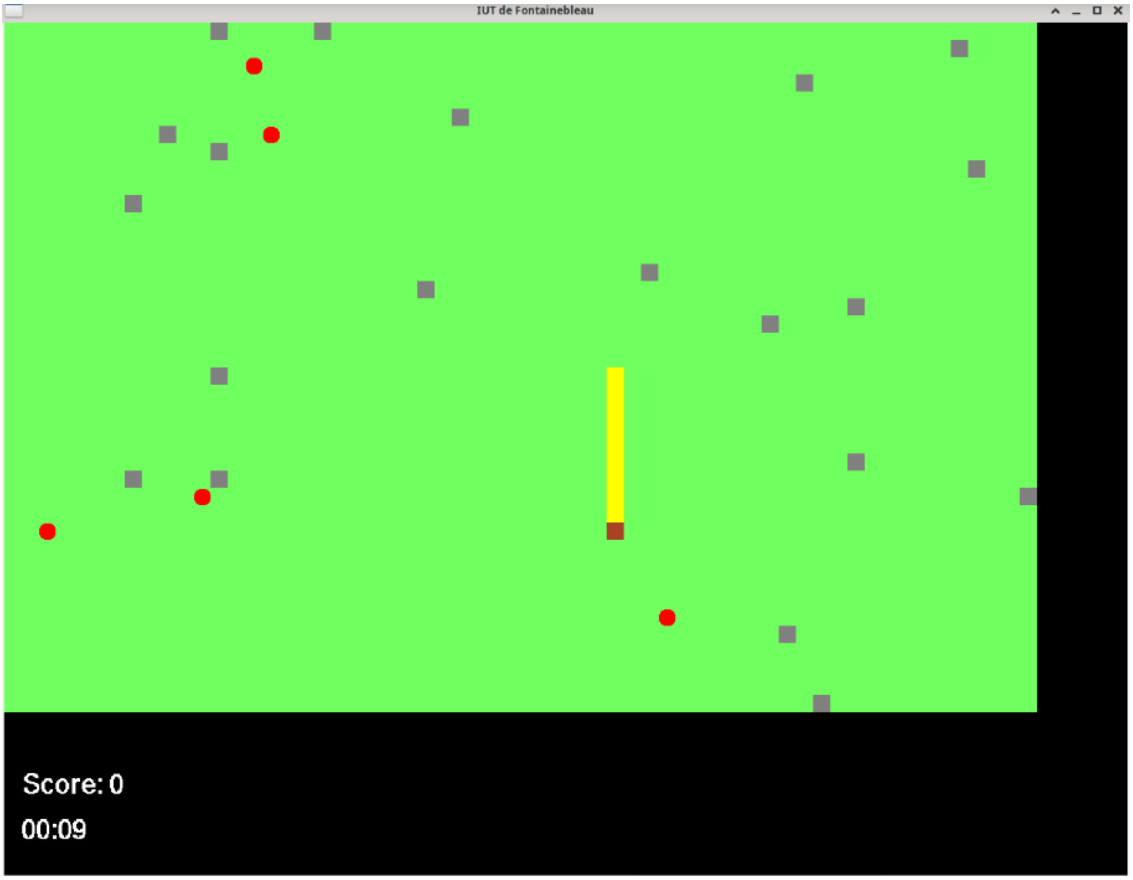
Thomas Rognant, Tom Momméja

Table des matières

Introduction

Pour notre projet, nous avons dû réaliser le jeu du serpent en C89. C'est un ancien jeu qui était sur les premiers portables. Ce jeu consiste à contrôler les déplacements d'un serpent sur une zone délimitée avec des bordures ou sans bordures. Le joueur doit manger des pommes avec le serpent afin de le faire grandir. Si le joueur se touche lui-même avec le serpent, le jeu est perdu. Ce jeu a beaucoup de versions différentes qui permettent de dynamiser le jeu.

Fonctionnalités



Notre jeu dispose des fonctionnalités suivantes :

- Interface Graphique : Le jeu commence en créant une fenêtre, affichant un message pour permettre au joueur de se préparer à commencer la partie.

- Écran de Jeu : Un rectangle vert correspondant à la zone où le serpent peut se déplacer sur une fenêtre noire qui contient des informations de jeu.

- Initialisation du Jeu : Le serpent commence au centre de l'espace de jeu avec une longueur initiale de 10 segments. Les pommes, représentées par des cercles rouges, sont générées de manière aléatoire à des emplacements différents. Les obstacles, représentés par des rectangles gris, sont également positionnés aléatoirement dans l'espace de jeu.

- Déplacement du Serpent : Le serpent se déplace dans quatre directions : droite, gauche, haut et bas. Le déplacement du serpent est géré en temps réel, avec une mise à jour régulière grâce à la fonction DeplacerSerpent.

- Gestion du Temps : Un chronomètre affiche le temps écoulé depuis le début du jeu, mesuré en minutes et secondes. La mise à jour du temps est effectuée chaque seconde.

- Score : Le score du joueur est affiché en temps réel à l'écran. Le score est augmenté chaque fois que le serpent mange une pomme.

- Collision et Fin de Jeu : Les collisions avec les bords de l'espace de jeu, les obstacles ou le serpent lui-même entraînent la fin du jeu. Un écran de fin s'affiche, indiquant "Game Over" et le score final.

- Pause du Jeu : Le joueur peut mettre le jeu en pause en appuyant sur la barre d'espace. La touche Echap permet de quitter le jeu à tout moment.

- Dynamisme et Réactivité : Le jeu réagit aux entrées du joueur en temps réel, permettant un contrôle dynamique du serpent. La réactivité du jeu est maintenue grâce à l'utilisation de temporisations (Attendre) et de mises à jour régulières.

Découpage du programme

Le découpage en différents fichiers source dans le programme du jeu du serpent a été réalisé dans le but d'améliorer la lisibilité, la maintenabilité et la modularité du code. Chaque fichier source se concentre sur des fonctionnalités spécifiques, ce qui facilite la compréhension du code et permet une gestion plus efficace des dépendances.

3.1 Fichiers et leur utilité

- main.c: Gère la logique principale du jeu, les entrées utilisateur et l'intégration des différentes fonctionnalités. Utilise les fonctions définies dans serpent.c, pommes.c, obstacles.c et graphique.c.

- serpent.c: Gère la logique du serpent, son déplacement, sa croissance, et les collisions associées. Utilise les fonctions définies dans serpent.h, pommes.h, obstacles.h et graphique.h.

- pommes.c: Gère la génération des pommes, leur affichage, et les collisions avec le serpent. Utilise les fonctions définies dans pommes.h, serpent.h et graphique.h.

- obstacles.c: Gère la génération des obstacles, leur affichage, et les collisions avec le serpent. Utilise les fonctions définies dans obstacles.h, pommes.h, serpent.h et graphique.h.

- graphique.c: Gère l'interface graphique du jeu, l'affichage de l'écran, des messages, etc. Utilise les fonctions définies dans graphique.h.

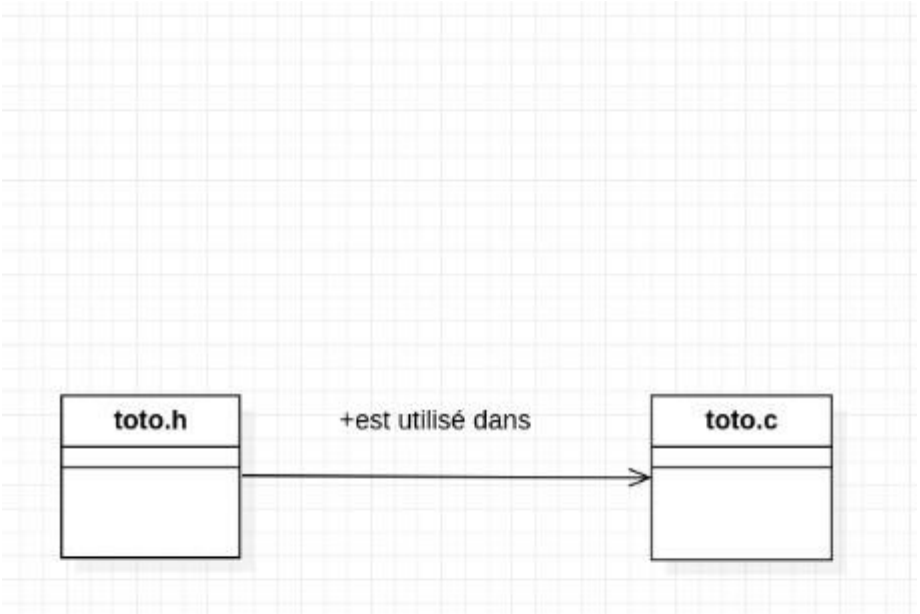
3.2 Avantages du découpage

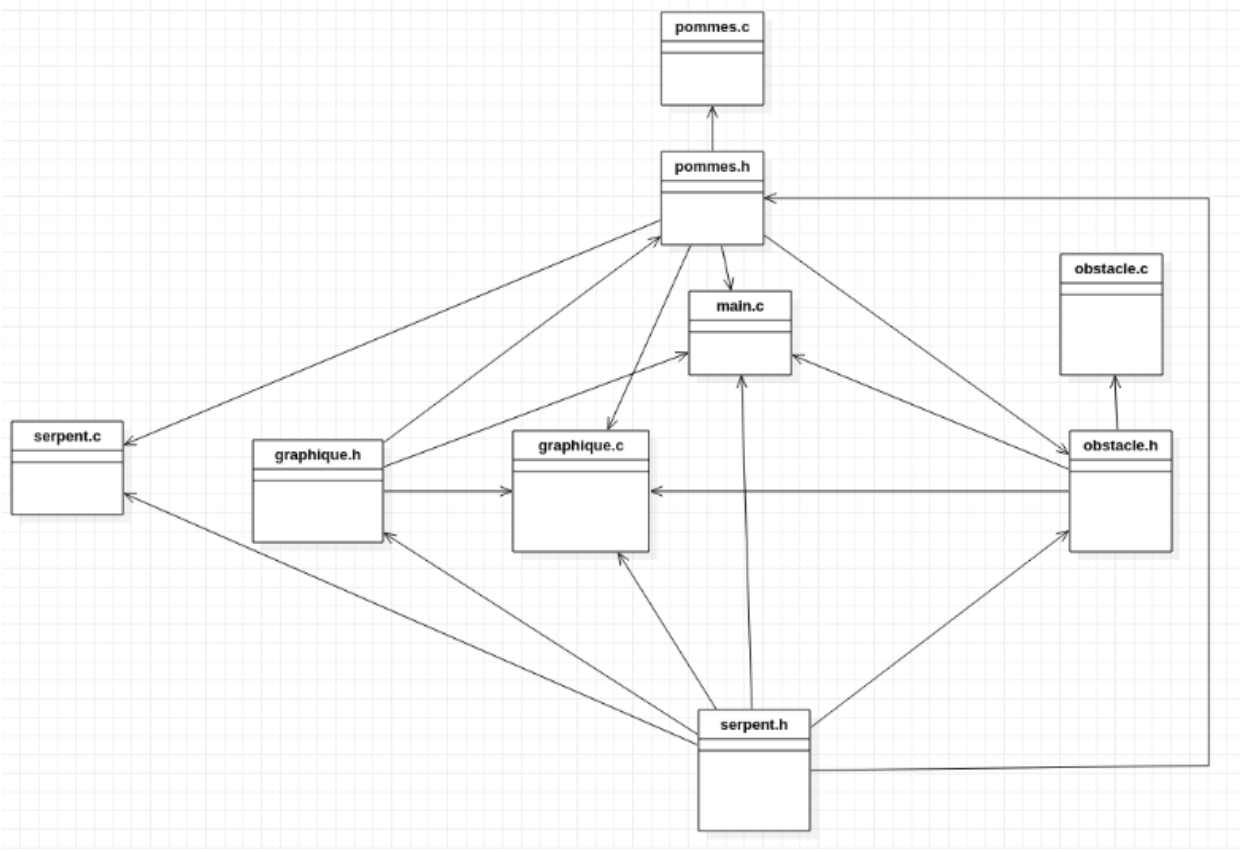
- Lisibilité accrue : Chaque fichier source se concentre sur une fonctionnalité spécifique, facilitant ainsi la compréhension du code.

- Maintenabilité améliorée : Les modifications ou ajouts de fonctionnalités peuvent être effectués de manière isolée, réduisant les risques d'erreurs et simplifiant la maintenance du code.

- Réutilisation du code : Les fonctions spécifiques à chaque module peuvent être réutilisées dans d'autres projets sans modification, favorisant ainsi une approche de développement modulaire.

3.3 Diagramme des dépendances





Représentation et Transformations du Serpent dans le Programme

Dans le programme du jeu, la représentation du serpent est réalisée à l'aide de données structurées permettant de suivre son corps, sa longueur, et ses mouvements tout au long de la partie. La structure du serpent est définie dans le fichier serpent.h et se compose principalement des éléments suivants :

4.1 Position et Structure du Serpent

- Position : La structure Position représente les coordonnées (x, y) d'une partie du serpent sur la grille du jeu. Chaque élément du corps du serpent est défini par une position.

- Serpent : Le serpent est représenté sous la forme d'une liste chaînée de positions. La variable corps est un pointeur vers la tête de la liste

chaînée. Chaque élément de cette liste est de type Position, représentant une coordonnée sur le terrain de jeu. La variable longueur indique le nombre d'éléments dans la liste, reflétant la longueur actuelle du serpent.

4.2 Effacement des positions précédentes

Avant de déplacer le serpent, les positions précédentes doivent être effacées du terrain de jeu. Cela est réalisé en parcourant la liste chaînée et en utilisant les coordonnées de chaque position pour effacer la représentation graphique du serpent.

4.3 Mise à jour des positions du corps

Après l'effacement, les positions du corps du serpent doivent être mises à jour pour refléter son déplacement. Cela est réalisé en parcourant la liste chaînée et en déplaçant chaque élément du corps à la position de l'élément précédent.

Transformations du Serpent au Cours de la Partie

5.1 Déplacement

On donne tout d’abord une direction initiale au serpent dans main.c. Lorsque le serpent se déplace, la position de sa tête est mise à jour en fonction de la direction du déplacement. Les parties du corps suivent la tête, créant ainsi un mouvement fluide. Cette transformation est effectuée par la fonction DeplacerSerpent dans le fichier serpent.c.

5.2 Croissance

Lorsque le serpent mange une pomme, sa longueur augmente, et un nouvel élément est ajouté à son corps. La transformation est gérée dans la fonction CollisionAvecPomme dans serpent.c.

5.3 Collision avec lui-même ou les obstacles

Si le serpent entre en collision avec lui-même ou avec des obstacles, la partie se termine, et des actions spécifiques peuvent être déclenchées. Cette transformation est gérée dans la fonction GestionCollision du fichier serpent.c.

L'utilisation d'une liste chaînée offre une structure dynamique qui facilite la gestion des mouvements et des transformations du serpent au cours du jeu. Chaque élément de la liste représente une partie du corps du serpent, et les opérations sur cette liste reflètent directement les actions du serpent pendant le jeu.

Conclusion

Ce projet a été une aventure riche en défis et en réussites. J’ai appris énormément sur l’utilisation des listes chaînées et de la création d’une interface graphique. Les moments difficiles m’ont permis de développer de nouvelles compétences.

Tom Momméja

Pour ma part, j’ai trouvé ce projet très intéressant. Cela ma permit de mettre en pratique ce que j’ai appris sur le Language C. Durant ce projet, je pense avoir commis des erreurs tel que l’utilisation de liste chainée pour les pommes ou encore pour les obstacles, ça rajoute de la complexité peut être inutile.

Thomas Rognant