

Yann KERAUDREN

groupe 2

Titouan LERICHE

Rapport d'avancement projet SNAKE

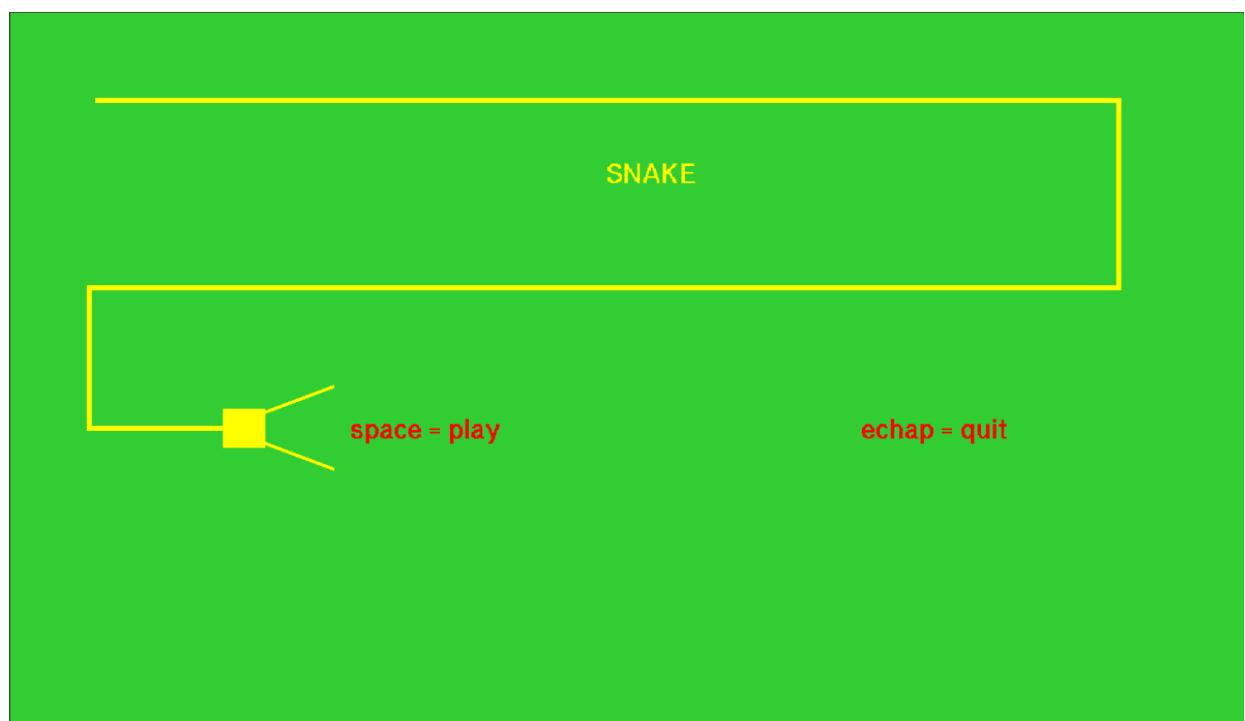


Table des matières

Introduction	1
Organisation du projet	2
Fonctionnalités du programme	3
Les différents fichiers	4
description des données du serpent.....	5
Conclusion personnel	6

1) Introduction

Description du projet :

Le jeu snake est un jeu en deux dimensions, qui consiste à déplacer un serpent sur un plateau délimité. L'objectif est de manger des pommes afin que le score augmente mais cela augmente aussi la taille du serpent ainsi que sa vitesse de déplacement, le nombre d'obstacles ou encore le nombre de pommes sur le plateau. L'augmentation de la taille et de la vitesse de déplacement rend le jeu plus complexe pour éviter de sortir des limites du plateau ou de se rentrer dedans.

Pour créer notre SNAKE, nous avons utilisé le langage de programmation C 89, nous avons essayé de faire un design simple et épuré. Nous avons voulu créer une belle ergonomie du jeu avec une prise en main simple et rapide. Nous avons choisi d'ajouter la variante que lorsque le serpent mange une pomme, celui-ci se déplace plus vite.

2) Organisation du projet

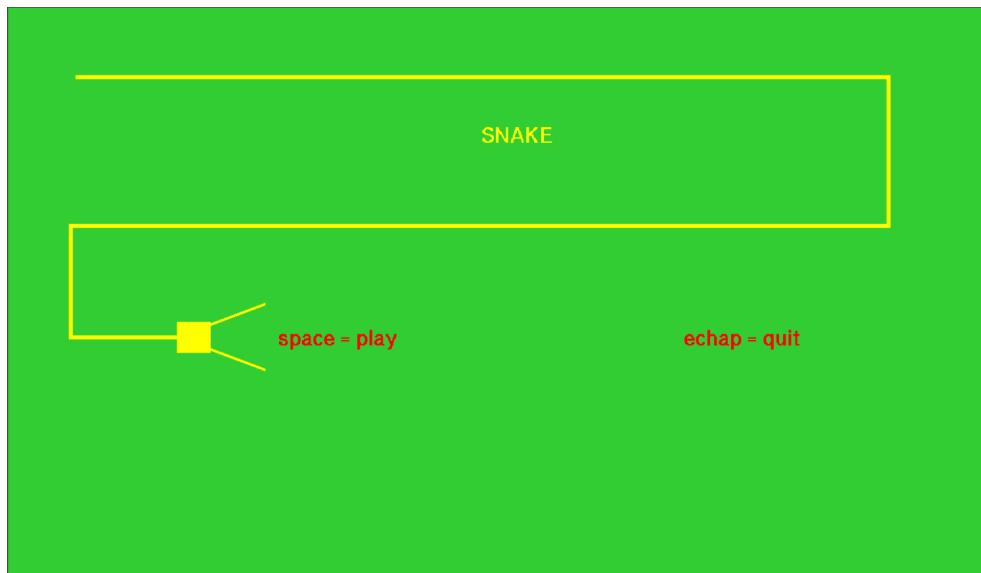
Nous avons divisé le projet en deux grandes parties principales pour éviter les conflits en travaillant sur les même morceau de code :

- Dans une partie nous avons regroupé toute la programmation des mouvements du serpents, de l'apparition des pommes, des obstacles.
- Dans la seconde partie, on s'est occupé de tout le graphique visuel du jeu, avec les affichages des scores, du temps, les écrans de commencement et de fin et le menu de pause.

Nous nous sommes occupés de ces deux parties en parallèle, Yann KERAUDREN s'est penché sur la première partie et Titouan LERICHE sur la deuxième mais nous nous sommes entraînés afin d'avancer plus rapidement et plus efficacement.

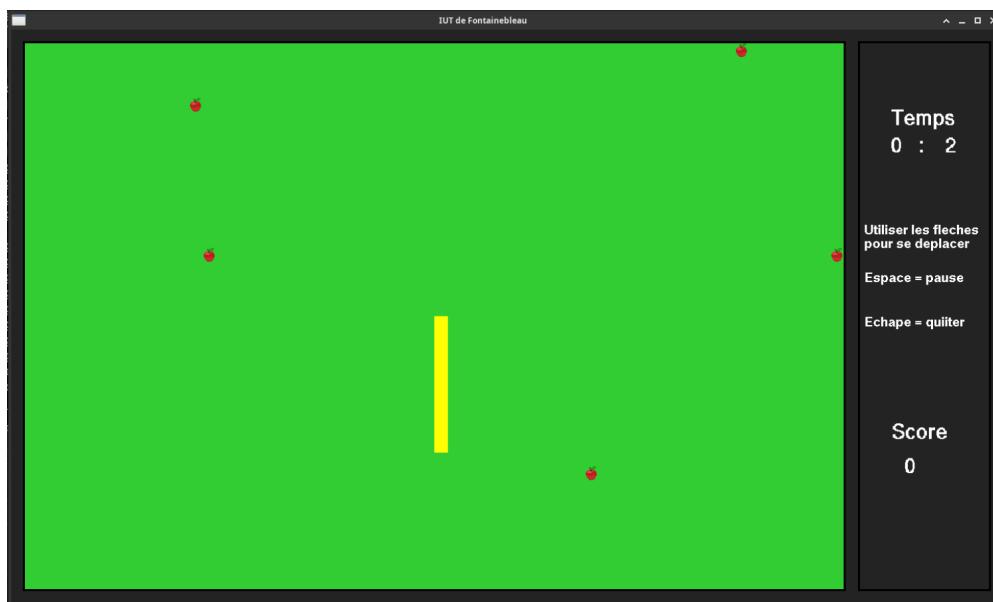
Nous ne voulions pas que l'un de nous deux reste bloqué trop longtemps sur le même problème, donc dès que l'un de nous deux était bloqué et ne comprenait pas comment résoudre le problème, l'autre ne prenait pas d'avance de son côté, mais venait aider à trouver une solution.

3) Fonctionnalités du programme



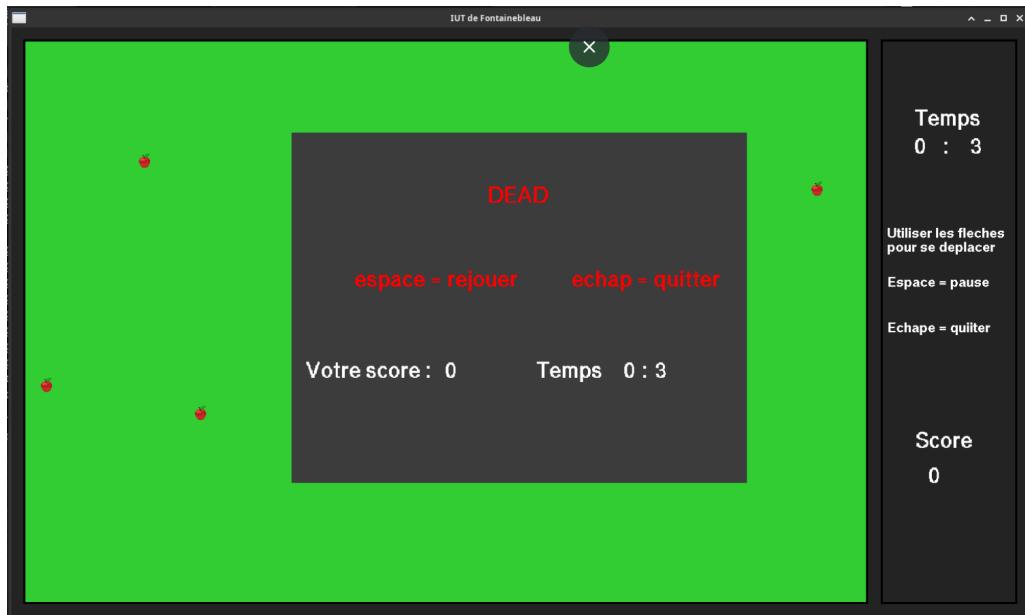
sur l'écran de lancement du jeu, on a deux choix :

- Si on appuie sur espace, la partie se lance et le serpent commence à se déplacer directement vers le bas.
- Si on appuie sur echap, on quitte le jeu et cela ferme la fenêtre automatiquement.



Une fois dans la partie, on a un visuel sur le temps qui s'écoule depuis le début de la partie et le score qui augmente de 5 à chaque fois que l'on mange une pomme.

on peut toujours appuyer sur echap pour quitter la partie et fermer la fenêtre. Lorsque l'on appuie sur espace de nouveaux, le jeu se met en pause, le serpent arrête de bouger et le temps s'arrête.



lorsque le serpent touche les contours de la fenêtre de jeux ou que l'on touche le corps du serpent, la partie est finie. Le joueur a perdu et l'écran de fin de la partie s'affiche instantanément.

On peut voir le score final de la partie et le temps de jeu.

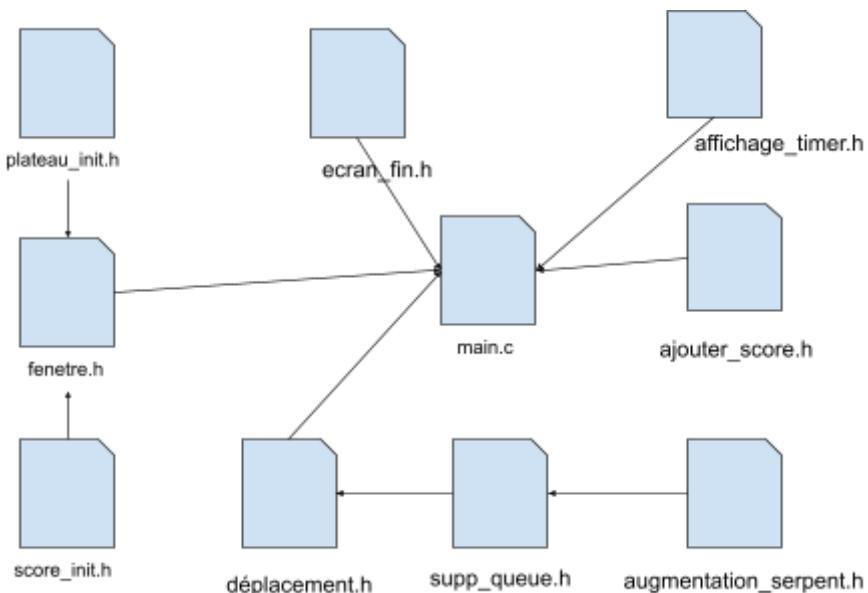
Comme sur l'écran de lancement, on peut choisir de relancer une partie en appuyant sur espace ou alors on peut quitter le jeu en appuyant sur echap.

4) Les différents fichiers

Le projet est constitué principalement d'un fichier main où il y a l'intérieur le programme principal du jeu. Pour simplifier et ne pas avoir de nombreuses lignes de code dans la fonction main nous avons décidé de séparer les fonctions qui sont indépendantes.

Puis dans un autre souci de correction de bug, il était préférable de découper les principales fonctionnalités du code pour pouvoir repérer les problèmes plus facilement.

Comme l'affichage de la fenêtre, les déplacements, la suppression de la queue, l'augmentation de la taille du serpent et l'écran de fin de jeu.



5) description des données du serpent

Le corps du serpent est représenté par un tableau de tableau qui évolue à chaque appel de la fonction déplacement selon un cycle choisi. Deux variables retiennent les coordonnées de la tête et la queue, et une variable retient l'indice de la queue dans le tableau. Lors de chaque appel, les coordonnées de la tête sont écrites à la place de l'indice de la queue et l'indice de la queue est incrémenté de 1. Ainsi, le tableau est utilisé comme un ruban qui boucle sur lui-même évitant d'utiliser trop de place mémoire.

Lorsque le serpent mange une pomme une nouvelle fonction est appelé, celle-ci recréer un nouveau tableau de tableau avec une taille supérieur de 1, de l'indice 0 jusqu'à l'indice de la queue, on insère les coordonnées de l'ancien tableau, puis on ajoute la nouvelle tête et enfin on ajoute les coordonnées restantes.

Et enfin comme le serpent s'agrandit de 2 cases, une variable nommé retard est placé à 2 lorsque le serpent mange la pomme, l'augmentation de la taille du tableau coprs serpent est effectuée et on retire 1 à la variable retard, ainsi on augmente la taille du tableau tant que la variable retard n'est pas à 0.

6) Conclusion personnel

conclusion personnel de Titouan LERICHE :

Le projet final ressemble à ce que j'avais imaginé, je suis content du résultat du travail que nous avons fourni. Il y a encore quelques problèmes d'affichage du score et du temps à la fin de la partie mais le principal du jeu tourne très correctement. Durant cette SAE j'ai pu apprendre beaucoup de choses et m'améliorer en pratiquant ce que nous avions déjà appris durant le début de l'année.

pour améliorer ce projet, a l'avenir on pourrait :

- ajouter des obstacles
- ajouter des petit boost de vitesse que le joueur peut gagner
- proposer différentes couleurs du serpents

Pour ma part, les difficultés que j'ai rencontrées étaient surtout quand il fallait faire des fichier en créant des fonctions dedans, puis ensuite appeler les fonctions dans d'autres fichiers (le fichier principal) . mais grâce à l'aide de mon collègue, j'ai fini par bien comprendre le fonctionnement de tout cette partie.

Pour la partie travail en équipe je suis content de la façon dont nous avons réparti les tâches et comment nous avons travaillé chacun de notre côté comme ensemble quand il le fallait.

Conclusion personnel de Yann KERAUDREN

Ce projet a été très formateur d'une part pour les allocations dynamiques mais aussi l'utilisation du Makfile, qui une fois bien pris en main s'avère très pratique.

Cependant, j'ai perdu beaucoup de temps en cherchant à vouloir optimiser le programme, en essayant de faire le moins d'appels possible ou encore en affichant uniquement la tête et en effaçant la queue.

Notre répartition du travail à été je pense une bonne idée, nous avons pu travailler chacun de notre côté sans se gêner, ou sans mettre nos fichiers en conflits dans le git.