



# RAPPORT

---

MATTHIEU PETIT

# INDEX

## Introduction - 3

I. Règles

II. Sujet

## Fonctionnalités - 4

I. Menu

II. Jouer

III. Meilleurs scores

IV. Paramètres

# Introduction

## I. Règles

**Snake**, est un jeu vidéo dans lequel le joueur contrôle une ligne qui forme le corps d'un serpent.

Ce corps est en mouvement continu et la partie s'arrête lorsque la tête du serpent entre en contact avec un des bords de l'écran ou son propre corps.

Le joueur peut diriger le serpent selon les directions suivantes : Haut, Bas, Gauche, Droite.

Des pastilles et des obstacles sont présents sur la grille de jeu, et la partie s'arrête si la tête du serpent entre en contact avec l'un des obstacles.

Si une pastille est mangée, le corps du serpent s'allonge de deux cases.

Pour passer au prochain niveau, le serpent doit manger toutes les pastilles présentes.

Chaque niveau présente une pastille et un obstacle de plus que le niveau précédent, et la vitesse du serpent est augmentée.

## II. Sujet

Il nous a été demandé pour ce projet de réaliser notre version du **Snake**.

Notre programme doit comporter un menu permettant de :

- Lancer une partie
- Quitter le jeu
- Changer la taille de la grille de jeu
- Changer la taille initiale du serpent
- Changer le nombre de pastilles au premier niveau

Par défaut :

- La taille de la grille jeu est de **60** lignes par **40** colonnes.
- La taille initiale du serpent est de **10** cases.
- Le premier niveau comporte **5** pastilles et **0** obstacles.

# Fonctionnalités

## I. Menu

Le jeu comporte un Menu permettant de :

- Lancer une partie
- Afficher les Meilleurs scores
- Ouvrir les Paramètres
- Quitter le jeu



## II. Jouer

Lorsque le joueur lance une partie, il arrive sur la grille de jeu et le serpent est placé au milieu de l'écran.

Le **score**, le **niveau** en cours et le **temps écoulé** sont affichés en bas de l'écran.

Lorsque la partie se termine, l'écran **Game Over** apparaît.



# 1. Game Over

Cette fenêtre affiche le **score** obtenu ainsi que le **niveau** atteint.

Le joueur peut retourner au **menu** en appuyant sur **Echap**.

Le bouton **pseudo** (*voir suite*) permet au joueur de changer son nom.

Le bouton **meilleurs scores** renvoie directement vers la fenêtre du même nom.



*Pseudo*

*Meilleurs scores*

## III. Meilleurs scores

Cette fenêtre affiche les meilleurs scores, atteints par les précédents joueurs.

Ces scores sont lus à partir du fichier *scores*.

Le bouton **reset** permet de supprimer ce fichier.

Le bouton **pseudo** permet au joueur de changer le nom qui s'affichera dans cette même fenêtre.



*Reset*

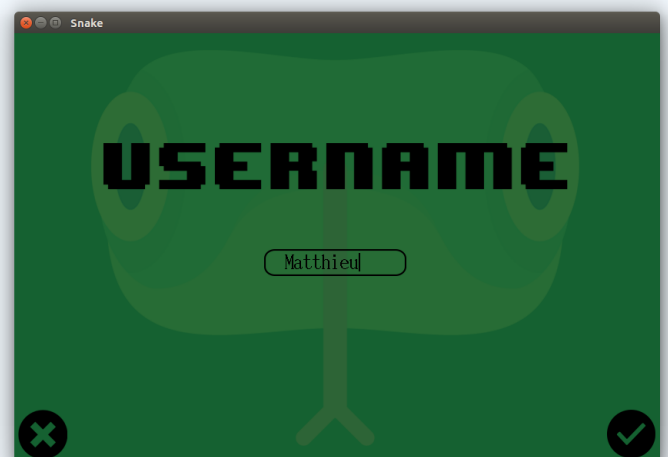
*Pseudo*

# 1. Pseudo

Le joueur a la possibilité de changer le nom qui l'identifie dans la fenêtre des **meilleurs scores**.

Le bouton **valider** et la touche **Entrée** permettent de valider le changement.

Le bouton **annuler** et la touche **Echap** permettent d'annuler tout changement.



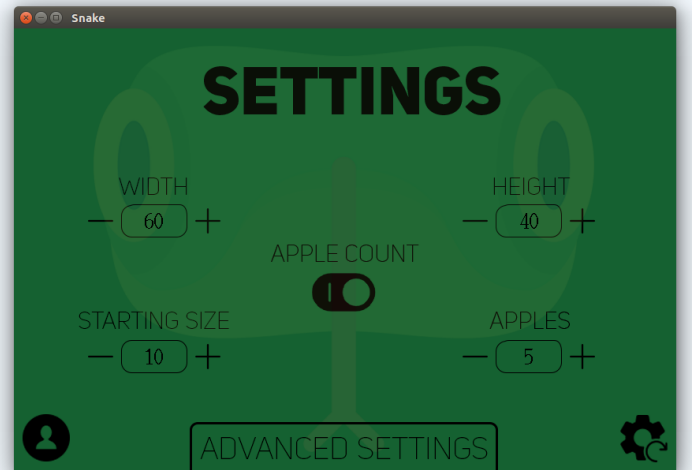
*Annuler*

*Valider*

## IV. Paramètres

Le joueur a la possibilité de modifier dans les paramètres respectivement, la **largeur**, la **hauteur**, la **taille initiale**, le **nombre de pastilles** présentes au niveau 1, et peut choisir d'afficher le **nombre de pastilles** qu'il a mangé.

Le bouton **pseudo** permet de changer le nom du joueur, et le bouton **advanced settings** permet d'accéder aux **paramètres avancés**.



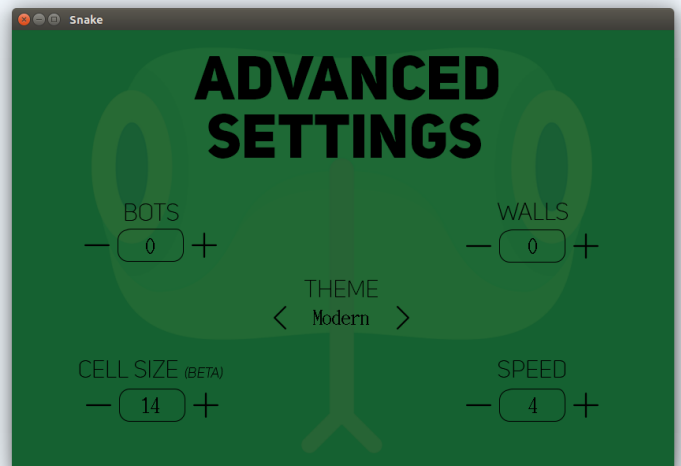
*Pseudo*

*Reset*

Ces paramètres sont stockés dans le fichier settings. Le bouton **reset** permet d'effacer ce fichier et de mettre les paramètres par défaut.

## V. Paramètres avancés

Cette fenêtre permet de modifier respectivement, le **nombre de bots**, le nombre d'**obstacles** présents au niveau 1, la **taille d'une case** (*beta*), la **vitesse** initiale du serpent, et le **thème** appliqué lors d'une partie.



# Structure

## I. Fichiers sources

Par souci de cohérence et de lisibilité, l'entièreté du code est en anglais à l'exception des commentaires qui sont en français.

Ce projet est composé de 4 fichiers **sources** (.c) ainsi que leurs **headers** (.h) correspondants. Pour éviter les erreurs d'inclusion et par clarté, un fichier **header** supplémentaire contient l'intégralité des structures utilisées.

### 1. Main

Le fichier **main.c** contient naturellement la boucle d'exécution principale.

Les fonctions qui gèrent le **dessin**, l'affichage des **données** (score, niveau, temps, nombre de pommes *si activé* ), la **pause**, le passage au **niveau suivant**, ainsi que l'écran de **fin de jeu** sont dans ce fichier.

### 2. Menu

Le fichier **menu.c** contient toutes les fonctions nécessaires à l'écran du **Menu**.

Voici la liste des tâches effectuées par ces fonctions :

- Commencer l'exécution et lire, ou créer le cas échéant les fichiers *scores* et *settings*
- Gérer l'affichage du **Menu**
- Initialiser tous les paramètres en cas de lancement d'une partie
- Afficher et gérer les **meilleurs scores**
- **Quitter** totalement le programme et **sauvegarder** les paramètres

### 2. Snake

Le fichier **snake.c**