

SNAKE - Projet APL Décembre 2019

Yannis SAGAL - Louis DAUBIÉ



Introduction

Notre mission était de concevoir le jeu très connu du Snake en le codant de A à Z en C89. Notre serpent ne naquit pas en un jour, mais aujourd'hui, à force de le nourrir de pommes et autres fruits virtuels, il est enfin disposé à vous laisser essayer de le dompter... Mais attention, il est féroce et goulou !

Présentation & Organisation du code

Nous avons d'abord pensé notre Snake, mais par facilité et pour gagner du temps, nous avons décidé de mettre tout notre code dans un seul fichier, tout en restant cependant organisés dans notre vision du jeu.

Dans notre fichier `main.c`, nous avons déclaré les variables globales en haut du fichier : la position, les fruits, le score, et le cas d'échec.

Pour notre terrain, nous avons décidé de mettre un peu d'originalité en créant un damier dans les tons sombres pour mieux faire ressortir le serpent et les fruits magiques.

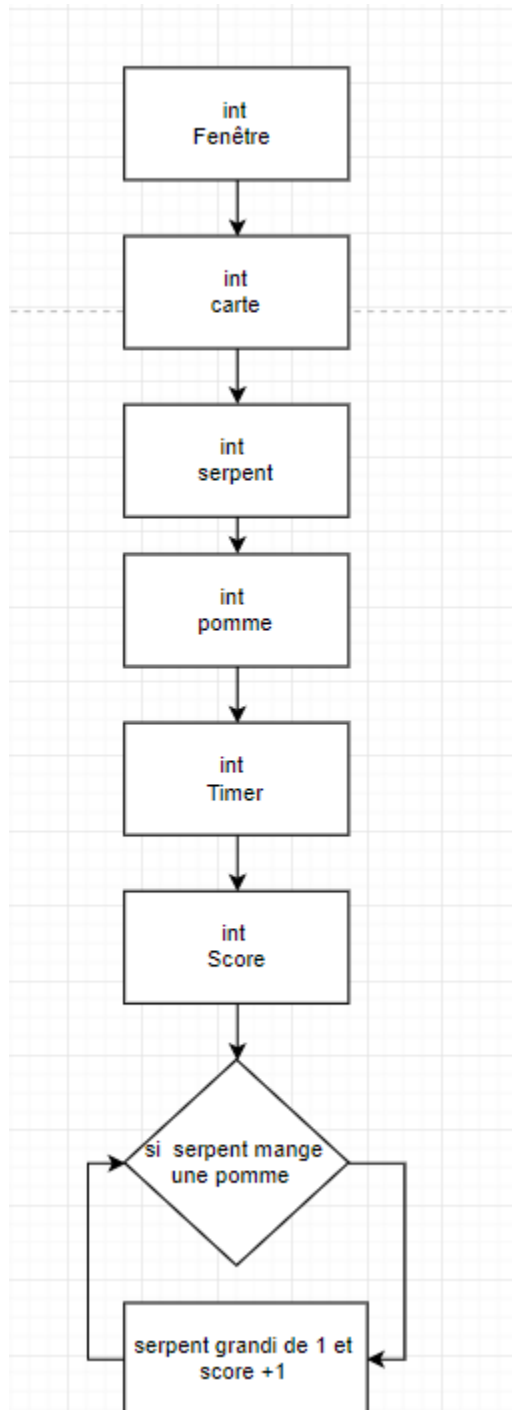
Malheureusement, il se trouve que notre serpent, au début de chaque partie, est tout timide, tout chétif ! Mais au fur et à mesure de la partie et qu'il traque ses proies, il grandit, grandit et grandit encore jusqu'à devenir un magnifique boa constricteur !



Serpent timide au début de la partie

Structure du code

Voici la structure de notre programme sous la forme d'un diagramme :



Tout d'abord nous avons défini la largeur et hauteur du terrain.

Nous avons initialisé des variables globales de manière à y avoir accès partout et pouvoir l'appeler dans n'importe quelle fonction.

Nous souhaitons créer un damier pour le terrain, et pour cela nous avons déclaré 2 variables "i" et "j" pour pouvoir alterner les couleurs.

Il a fallu ensuite positionner les fruits de façon aléatoire ; nous avons donc utilisé la fonction "srand".

La carte : il s'agit du point névralgique du programme puisque c'est ici que se déroule la partie.

Pour créer le damier nous avons fait une boucle for avec des conditions, dans le but de générer les couleurs alternativement.

Pour l'affichage du timer, nous avons créé une condition qui évalue si le nombre de secondes est supérieur ou égal à 0, temps à partir duquel la partie est commencée.

Concernant le score, nous avons écrit une condition pour incrémenter le score de 5 points à chaque fois que le serpent mange un fruit.

De plus, sa taille augmente d'un carré lorsqu'il mange un fruit. Nous avons comme projet de créer un fruit spécial qui donne un bonus, mais malheureusement nous n'en avons pas eu le temps !

Pour animer et contrôler le snake, nous avons créé la fonction "Snake Avance" dans laquelle nous avons déclaré la position initiale x et y dans un tableau qui a pour valeur de départ 0 et qui, grâce à un boucle for, calcule les coordonnées de la queue et de la tête du serpent lorsqu'il grandit.

La vitesse du serpent est gérée par la fonction "usleep" que nous avons définie à 50000 (50 millisecondes).

Vous contrôlerez le serpent à l'aide des flèches directionnelles du clavier (haut, bas, droite, gauche), direction que nous gérons dans un switch.

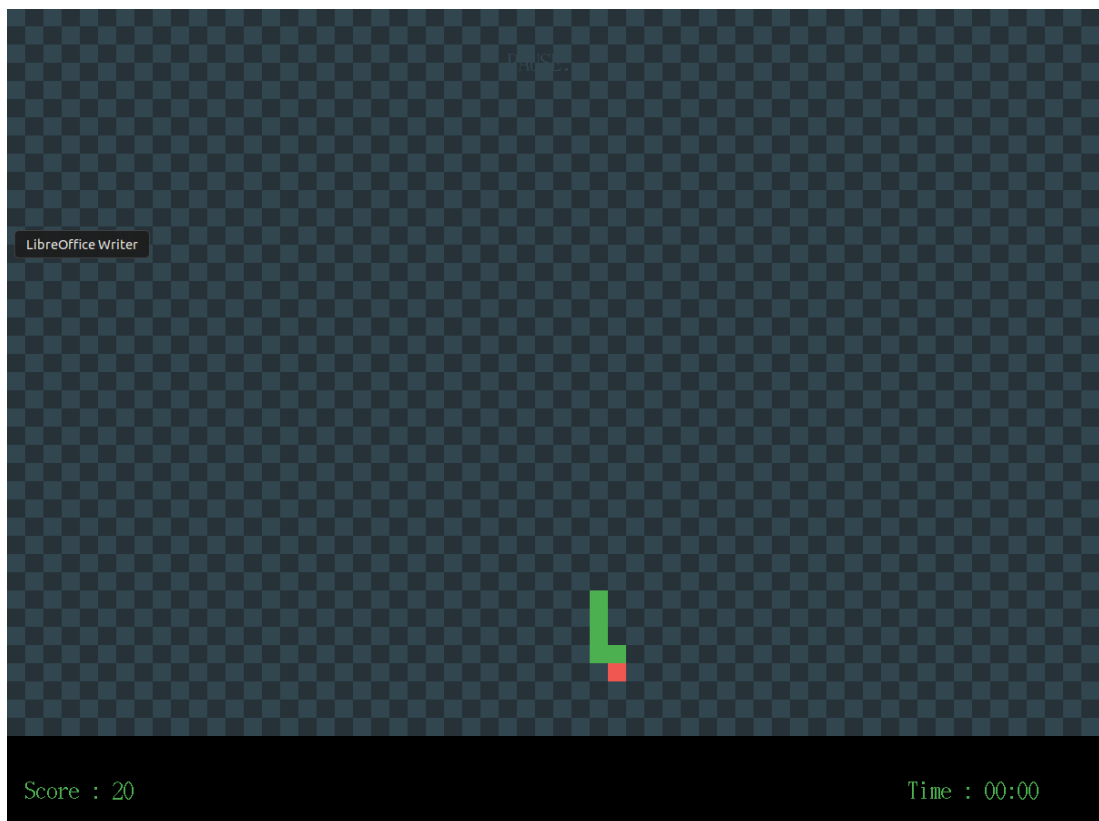
Quand le joueur perd, nous avons tenté d'afficher "game over" mais nous n'avons pas réussi.

La valeur de la variable "echec" passe à 1 quand le serpent rentre en collision avec un des murs ou s'il se mord la queue.

Description des fonctionnalités

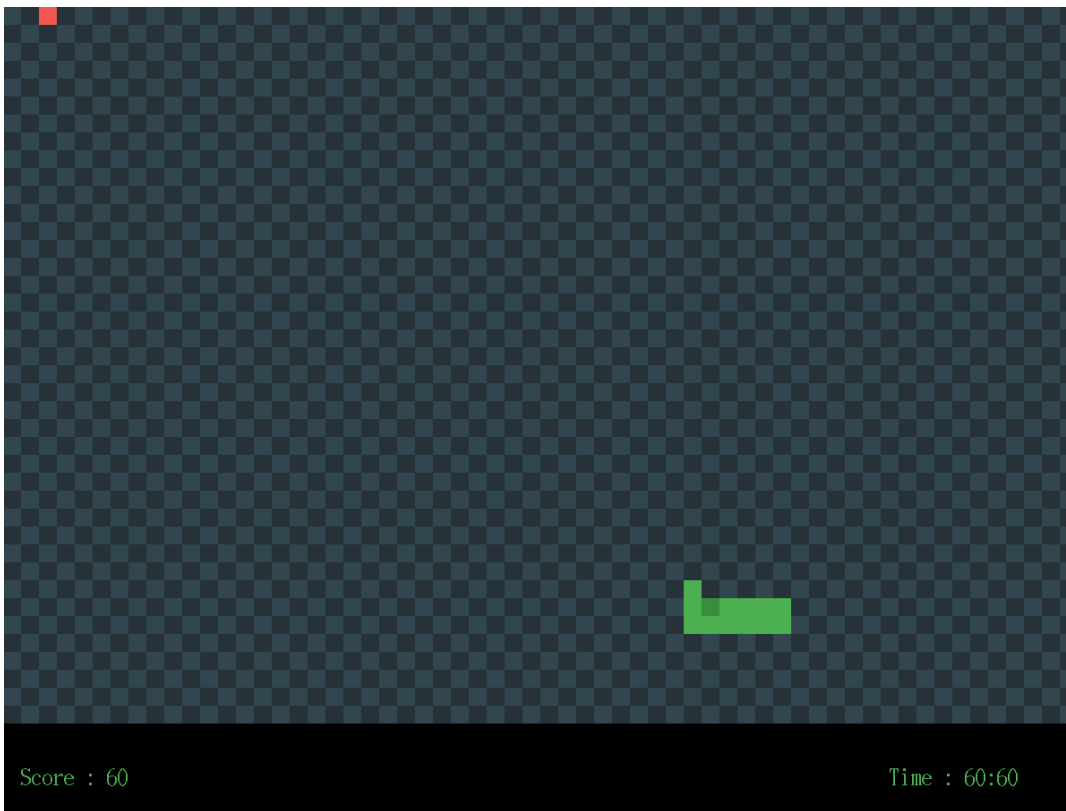
Nous avons créé une fonction dans laquelle les flèches directionnelles sont dédiées à l'animation du serpent. La barre d'espace sert à faire "Pause", mais malheureusement il n'y a pas de touche pour quitter le jeu, vous êtes obligé de jouer à l'infini !

La taille du corps augmente à chaque fruit avalé ; une boucle parente de celle de la queue, fait apparaître un nouveau fruit lorsque le précédent a été gobé.

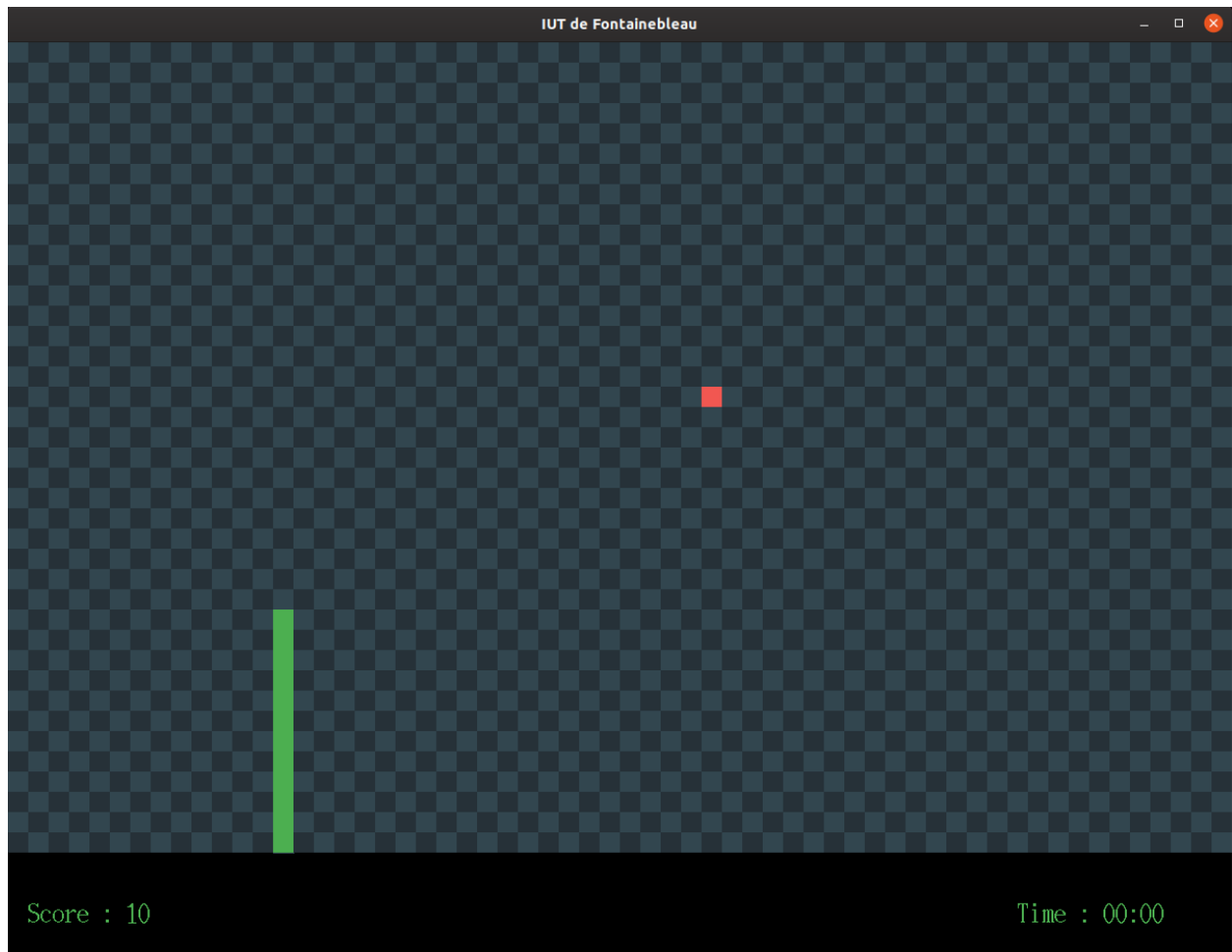


Serpent timide qui gobe un fruit

Bien sûr, si le serpent se mange lui-même ou s'il rentre en collision avec le mur, il meurt, le jeu s'arrête et le score est affiché en bas à gauche. Le joueur doit ensuite appuyer sur une touche pour fermer le jeu.



Serpent timide qui se mange lui-même



Serpent timide qui tente de faire le mur

Représentation graphique du jeu

Après avoir défini les deux constantes `LARGEUR` et `HAUTEUR`, nous avons créé une boucle pour générer le terrain en damier et ses limites, la tête du serpent symbolisée par le signe "supérieur" (`>`), et, dans une boucle dédiée, sa queue (symbolisée par la lettre "o").

Les murs sont représentés par des `"#"`.

Le score, initialisé à 0, est incrémenté de 5 pour chaque fruit mangé. Nous avons créé une boucle qui s'occupe très bien de cela.

Nous avons réussi à afficher le timer mais malheureusement il reste bloqué à 60:60.

Pour la pause, lorsque le joueur appuie sur la barre d'espace, le message "pause" s'affiche en haut de l'écran.

Conclusion

Yannis : “Pour ma part, je me suis focalisé principalement sur la taille du serpent, le timer (pour lequel j’ai malheureusement rencontré des difficultés pour le faire tourner) et le score. Globalement, je suis fier de la fluidité de notre jeu, et heureux d’avoir eu l’occasion de travailler avec Louis. ”

Louis : “Etant féru de design, je me suis occupé de la partie graphique, restant un peu sommaire malgré tout par manque de temps. Je me suis aussi chargé de placer les éléments du jeu et du choix des couleurs. Je suis fier du rendu graphique, mais cependant je suis conscient que j’aurais pu l’améliorer bien plus.”