

IUT Fontainebleau



# Projet Tutorés 1.1

## SNAKE

APL | 20 décembre 2019

## Table des matières

INTRODUCTION .....	2
LES DIFFÉRENTES FONCTIONNALITÉS DU PROGRAMME.....	2
1) <b>DEPLACEMENT DU SERPENT</b> .....	2
2) <b>TIMER</b> .....	3
3) <b>FONCTIONNEMENT DES TOUCHES DIRECTIONNELLES</b> .....	3
4) <b>LES TOUCHES ‘ESPACE’ ET ‘ECHAP’</b> .....	4
5) <b>MENU</b> .....	5
6) <b>LES PASTILLES</b> .....	5
PRÉSENTATION DE LA STRUCTURE DU PROGRAMME .....	6
EXPLICATION DE LA FORME DES DONNÉES QUI REPRÉSENTENT LE SERPENT .....	6
LA VITESSE EN FONCTION DU SCORE .....	7
CONCLUSIONS PERSONNELLES .....	7

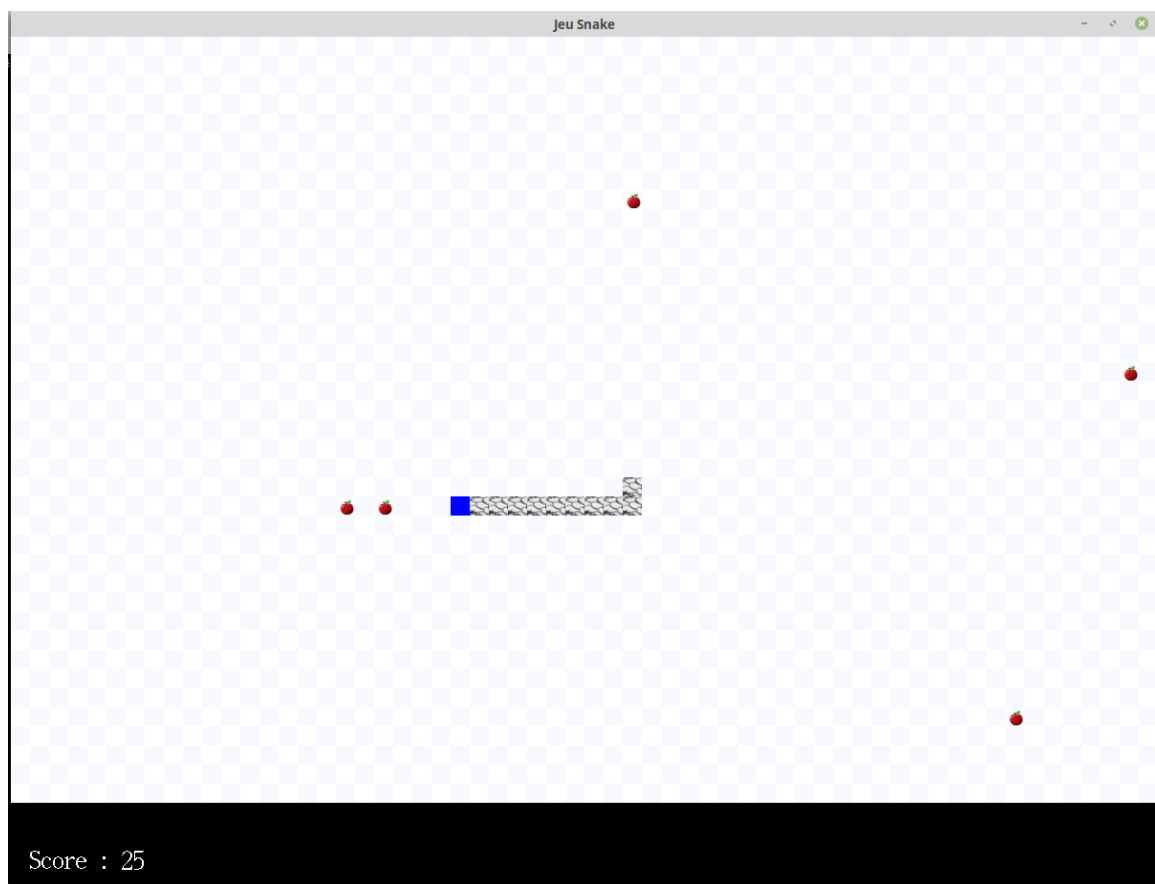
## INTRODUCTION

En cette fin d'année 2019, nous a été proposé la réalisation d'un Projet en langage C. Il s'agit d'un **Snake**, l'objectif étant de diriger un serpent vers des pommes et d'éviter une collision entre le serpent et le mur ou avec sa propre queue. Il nous a été mis à notre disposition une bibliothèque graphique « **graph. h** » afin de pouvoir écrire des applications graphiques sans avoir à maîtriser le fonctionnement et l'API de la couche X11. Ensuite, c'est à nous de faire appel aux connaissances acquises lors du premier module d'APL et de les appliquer.

## LES DIFFÉRENTES FONCTIONNALITÉS DU PROGRAMME

### 1) DEPLACEMENT DU SERPENT

Pour ce fait, l'ensemble de cette partie du code a été placée dans une fonction qui a été mise à son tour dans une boucle infinie qui ne s'arrête que lorsque le jeu est terminé (à savoir qu'il y a collision entre le serpent et le mur ou sa propre queue). En effet, le serpent évolue sans cesse dans la direction qui lui a été envoyée par une touche appuyée. Lorsque le serpent fait collision avec la bordure de l'affichage ou sa propre, la partie est terminée, comme dit précédemment. Pour cela, j'ai nommé un pointeur initialisé à 0 dans une boucle infinie et qui prend la valeur 1 si les coordonnées du serpent sont égales aux coordonnées de la bordure. Et si la variable vaut 1 alors on sort de la boucle.

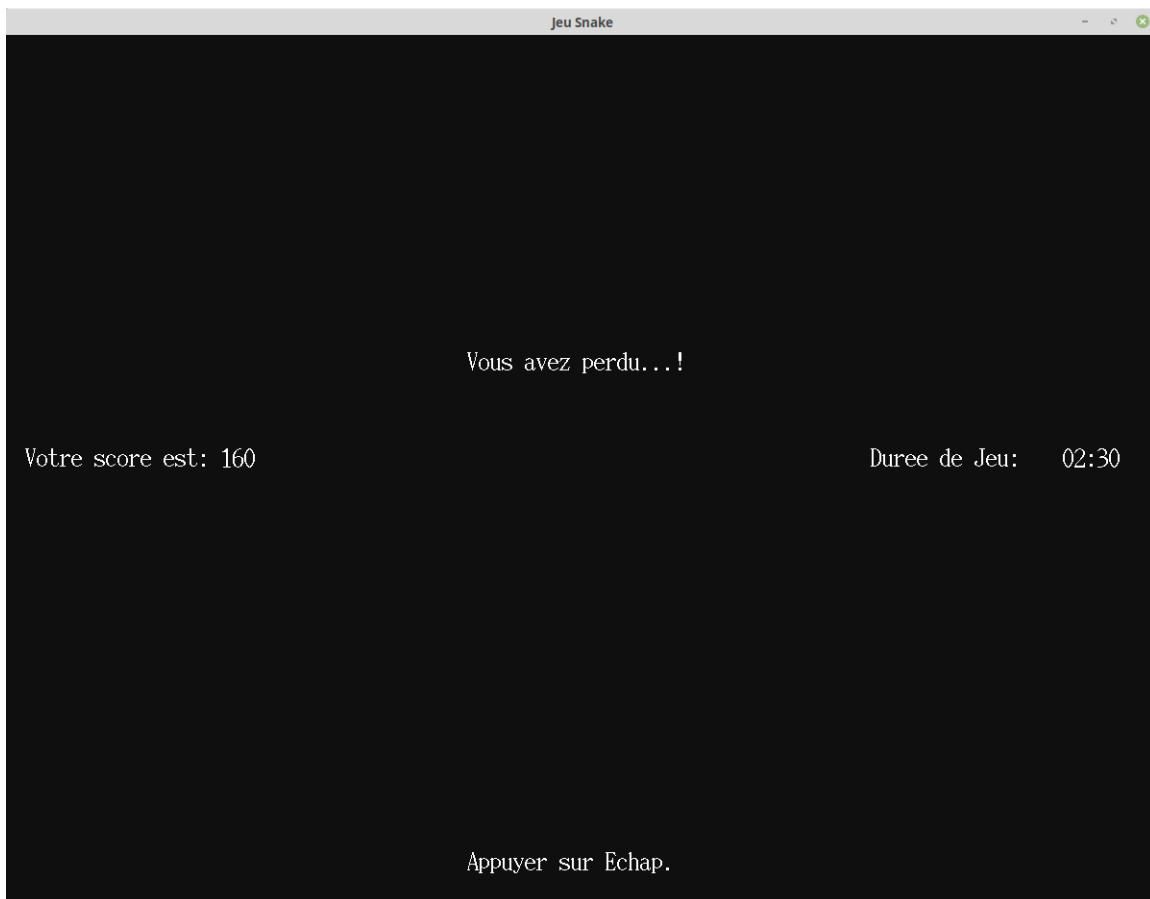


## 2) TIMER

De la même manière le « **timer** » se situe dans une boucle infinie. En effet, dès que l'instant présent dépasse l'ancien instant présent (qui est, par conséquent, vieux d'une seconde), alors l'ancien instant présent prend la valeur de l'instant présent immédiatement après s'être fait dépasser (cette phrase est on ne plus ambiguë, mais très précise).

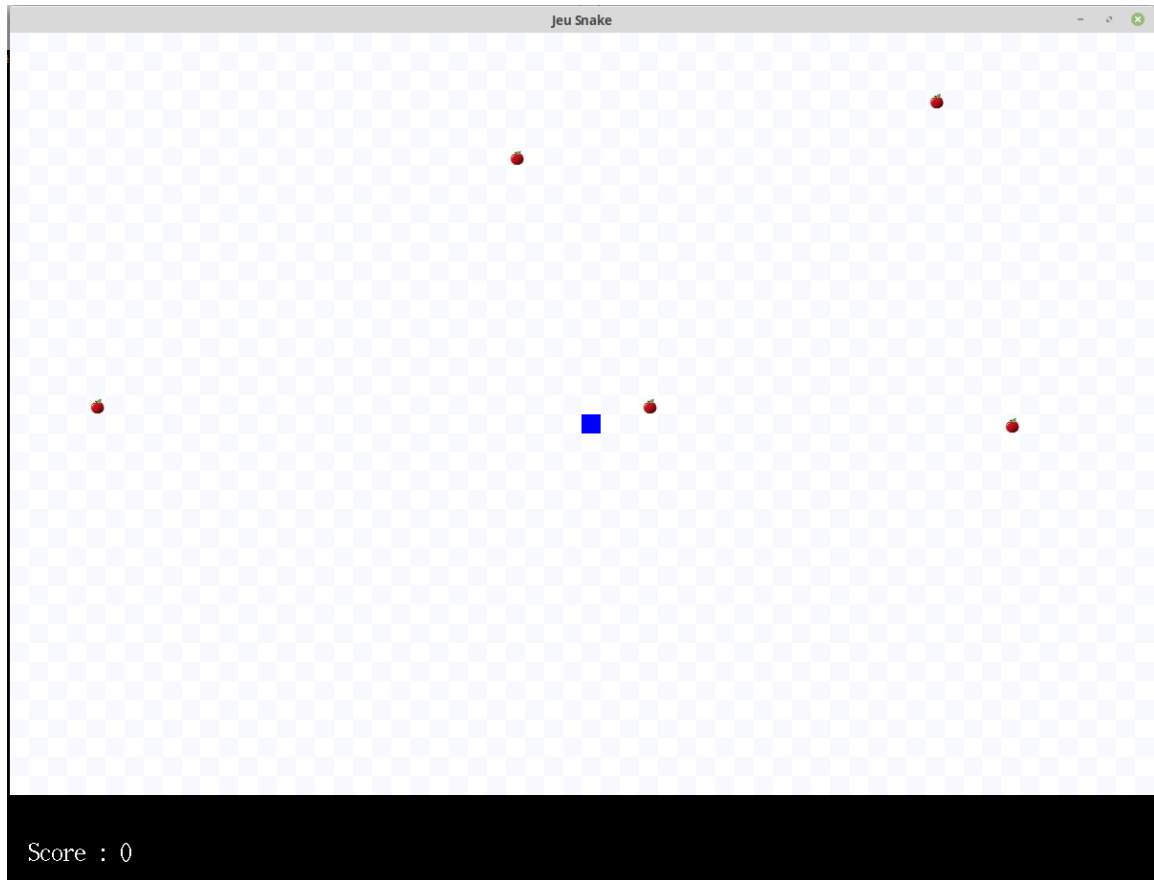
Ainsi, il suffit tout simplement d'afficher une chaîne de caractère qui est actualisée toutes les secondes par un compteur pour représenter le « **timer** » dans le jeu.

A la fin de chaque partie s'affiche le score de la partie et le temps qu'a duré la partie comme on peut le voir sur la figure suivante



## 3) FONCTIONNEMENT DES TOUCHES DIRECTIONNELLES

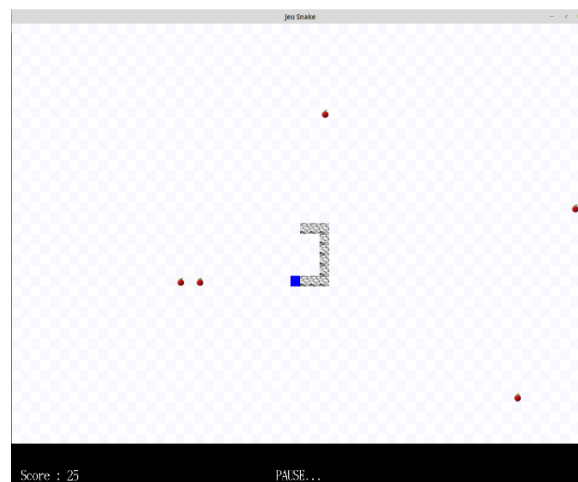
Cette partie du programme est placée dans une fonction qui est elle-même placée dans la boucle qui ne s'arrête que lorsque la partie est terminée. En effet, cette fonction est placée immédiatement après l'affichage du terrain de jeu dans la boucle. Ainsi, au début du jeu, le serpent ne bouge pas tant qu'on n'appuie pas sur une touche. Dès qu'on appuie sur une touche, le serpent prend la direction de la touche qui a été déclarée auparavant. On peut le voir sur la figure ci – joint.



#### 4) LES TOUCHES 'ESPACE' ET 'ECHAP'

Le fonctionnement de ces deux touches est spécifique au programme et elles sont bloquantes. Ainsi la touche '**ESPACE**' met le jeu en pause et la touche '**ECHAP**' quant à elle interrompt immédiatement le jeu. Le code gérant ces touches est dans la même fonction que celle des touches directionnelles mais une condition est imposée pour leur fonctionnement. Pour la touche :

- '**ESPACE**' : si la touche est appuyée, rien ne se passe et tant qu'elle n'est pas réappuyée, rien ne se passe toujours.



-‘ECHAP’ : si la touche est appuyée, on quitte immédiatement le programme.

### 5) MENU

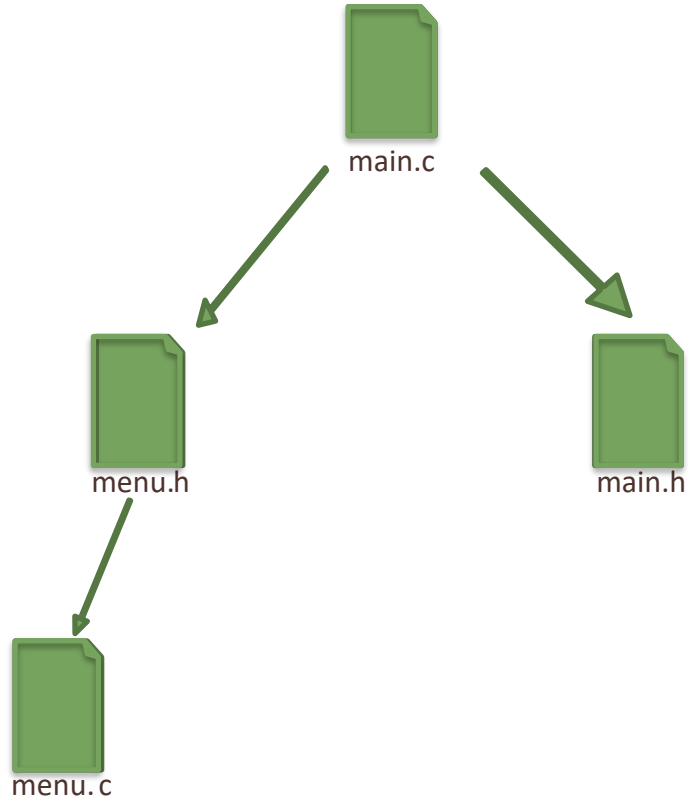
Un fichier a été dédié spécialement au menu du jeu. Dans ce fichier, se trouve une fonction qui gère les clics des souris et renvoie une valeur en fonction des zones cliquées. Vous pouvez le voir sur la figure suivante.



### 6) LES PASTILLES

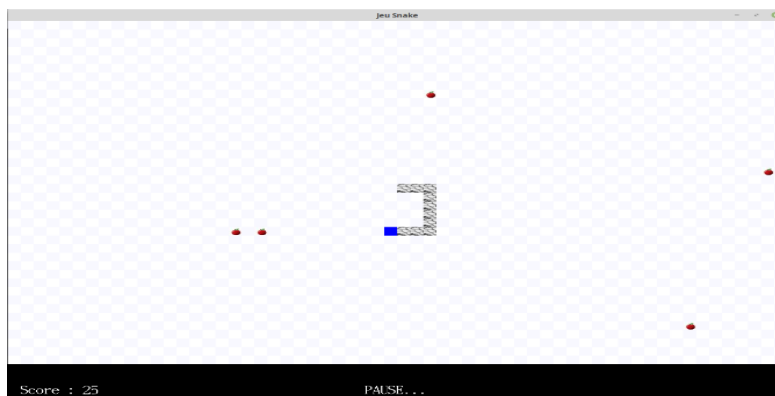
Nous avons représenté les pastilles par des pommes dont le serpent grandit quand il passe dessus. Le nombre de pomme sur le plateau du jeu est de 5, elles se placent n'importe où sur une case du tableau qui n'est pas occupée par le serpent, c'est à dire qu'une fois une pastille mangée, une autre apparaît aléatoirement sur une case non-occupée. Nous utilisons les coordonnées (x, y) pour donner les emplacements dans le tableau.

## PRÉSENTATION DE LA STRUCTURE DU PROGRAMME



## EXPLICATION DE LA FORME DES DONNÉES QUI REPRÉSENTENT LE SERPENT

C'est à l'aide d'une boucle que je parcours le serpent représenté sous forme de tableau et dont sa taille augmente au fur et à mesure qu'il mange des pastilles de pomme. Comme vous pouvez le voir sur la figure suivante



## LA VITESSE EN FONCTION DU SCORE

Pour avoir une variante comme l'a demandé le projet, nous avons mis une variante de « vitesse en fonction du score ». Pour cela, dans la boucle infinie qui s'arrête que lorsque la partie sont mis des conditions d'augmentation en fonction de la vitesse.

## CONCLUSIONS PERSONNELLES

\*\*\*Mikdaam BADAROU\*\*\*

Cette expérience représente l'aboutissement du premier projet de programmation auquel j'ai été sujet. Au début, insouciant, je commence les choses comme je le peux, je découvre la bibliothèque graphique, j'ai mes premières idées jusqu'à ce que je me rende compte qu'être en quelque sorte lâché dans la nature sans aucune expérience préalable s'avère être plus compliqué que prévu.

Cette sensation de ne pas savoir nager dans un océan bien vaste m'a poussé à aller de l'avant en m'intéressant de fond en comble sur cette bibliothèque graphique. Effectivement, confronté aux problèmes concrets rencontrés lors de la compilation, la réflexion ou encore la réalisation, on finit bien par se résoudre à devoir revoir certains points abordés en TP/TD.

Je suis très fier d'avoir pu revoir différentes notions et de les avoir redécouvertes n'ayant pas en moi ce recul qui permet de se rendre compte instantanément des erreurs à ne pas faire comme par exemple (pour ma part) : négliger le cours sur l'allocation dynamique et les pointeurs en me disant que seuls, les tableaux me suffisaient pour mes futurs programmes. Ce projet a réellement provoqué pour moi une prise de conscience de ce qu'est la programmation, de pourquoi apprend-t-on autant une notion bien qu'elle ne nous parle peu.

\*\*\*\*Antonio RANDRIAMANANJARA\*\*\*\*

Ce projet fut ma première véritable expérience que j'ai pu aboutir avec mon camarade. Malgré le manque de connaissance que je n'avais pas encore acquis, cela m'a permis d'apprendre pas mal de choses sur le codage en langage C, et de comprendre une phase sur le fonctionnement d'un programme.

Au début, c'est très motivant de savoir que l'on peut réaliser une telle chose mais une fois rentrer dedans, on se rend compte de la difficulté et du temps que cela peut prendre. Il m'est arrivé plusieurs fois d'être bloqué ou parfois perdu par moment, mais les conseils des professeurs et des autres élèves m'ont été très utiles. Même si nous pouvions encore améliorer ce Snake, je suis satisfait du résultat que nous avons obtenu. Je pense que si nous avions eu un peu plus de temps, cela nous aurait permis peut-être de le développer encore plus.