

Nell TELECHEA-LOJOU

BUT1 Informatique Fontainebleau

Benjamin BRIBANT

RAPPORT JEU SNAKE

SAE DEV 1.1

TABLE DES MATIERES

Contenu

- I. Introduction
- II. Description des fonctionnalités
- III. Explication du découpage des fichiers sources
- IV. Fonctionnement du serpent
- V. Conclusion

INTRODUCTION

I. Introduction

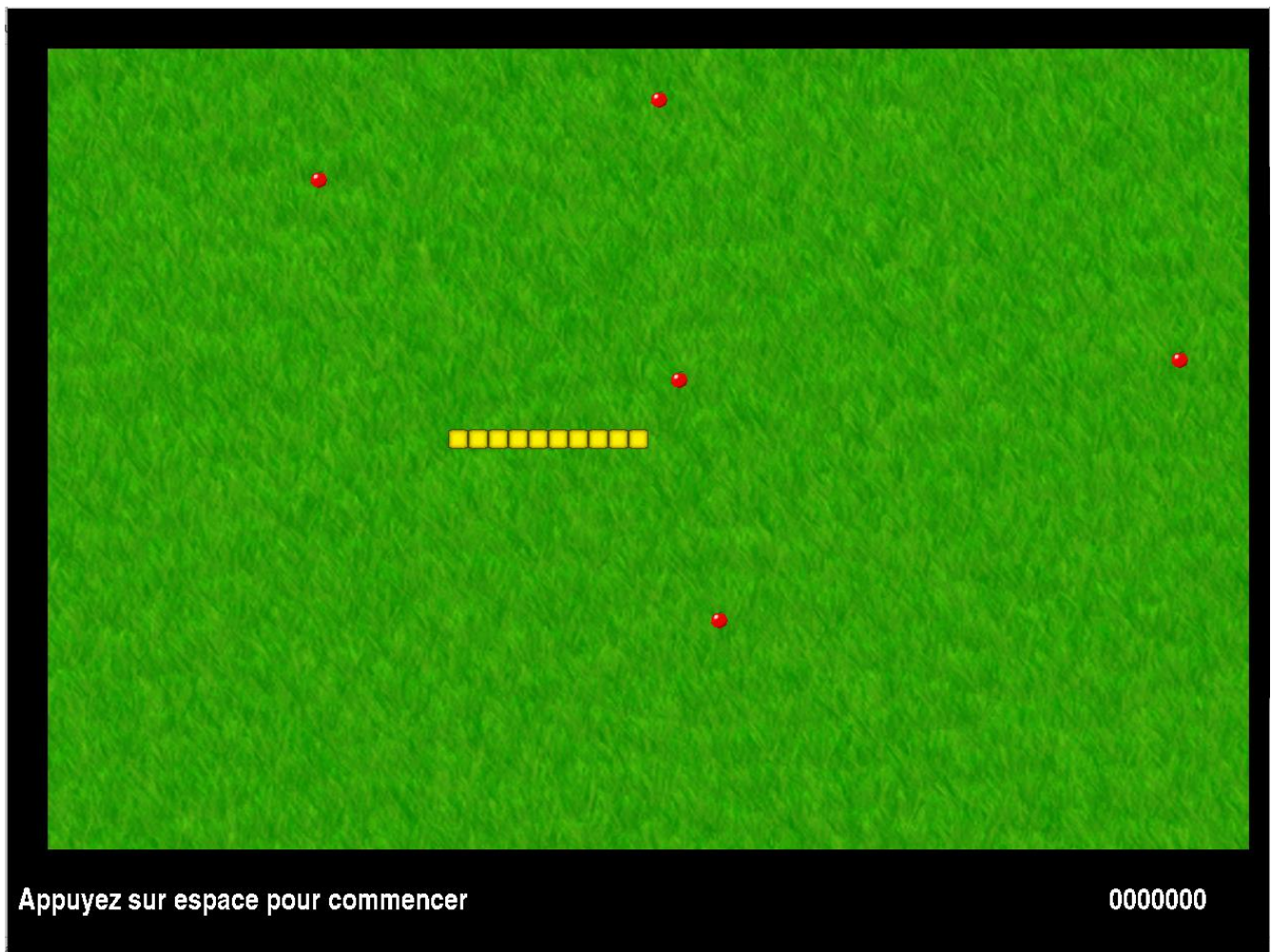
Dans ce rapport, nous allons présenter notre projet Snake et l'expliquer pas à pas.

Un jeu Snake est un jeu où l'on contrôle un serpent, qui doit manger des fruits, la plupart du temps des pommes, pour grandir. Si le serpent touche un bord ou une partie de son corps, la partie se termine. Le but est de manger le plus de pommes possibles. Dans notre version du jeu, le serpent accélère à chaque fois qu'il a mangé trois pommes.

DESCRIPTION DES FONCTIONNALITES

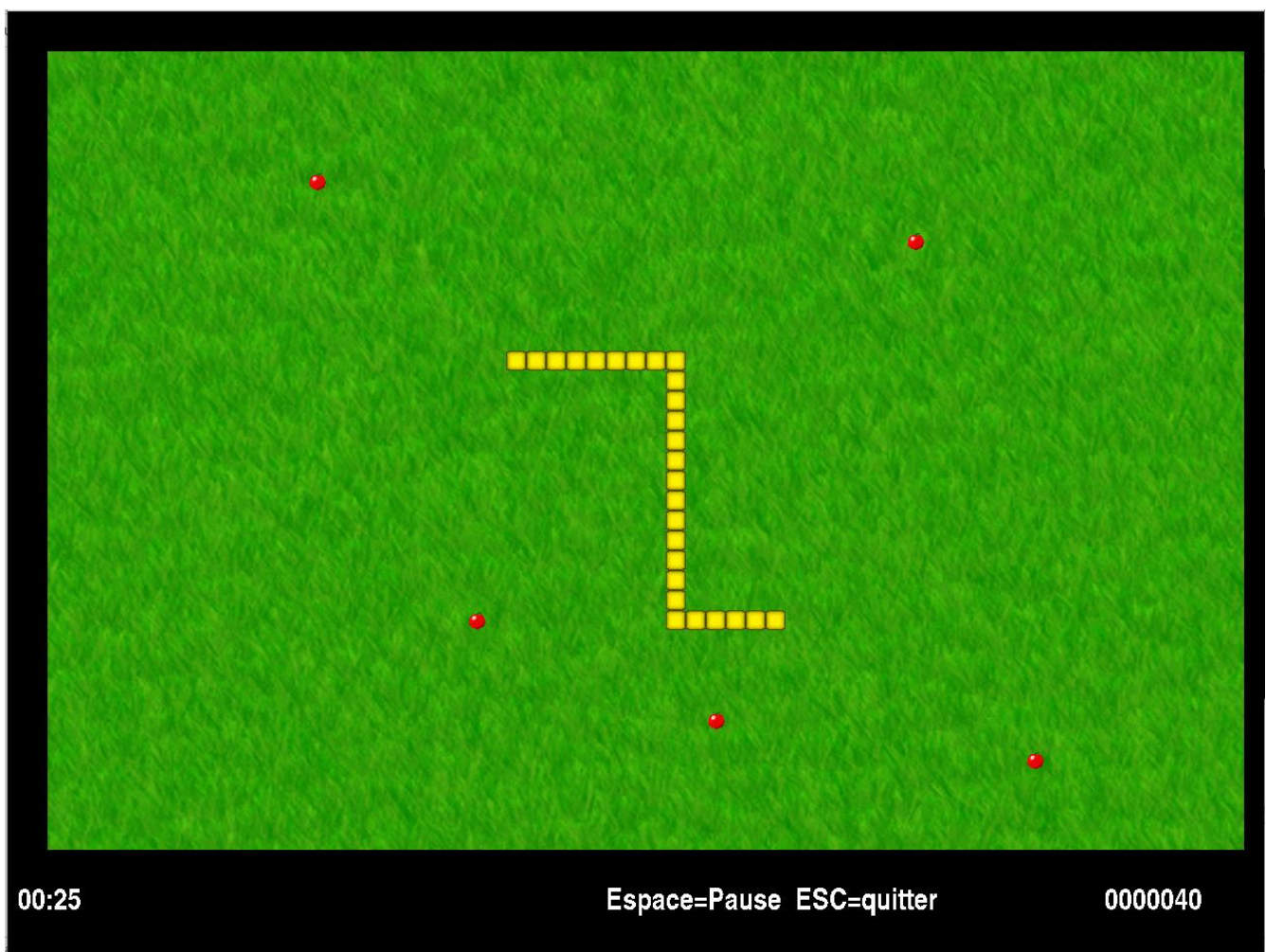
II. Description des fonctionnalités

Quand on lance le jeu, une fenêtre se crée avec l'écran du jeu et demande d'appuyer sur la touche échap pour commencer la partie. A ce moment, 5 pommes apparaissent aléatoirement sur la grille.



DESCRIPTION DES FONCTIONNALITES

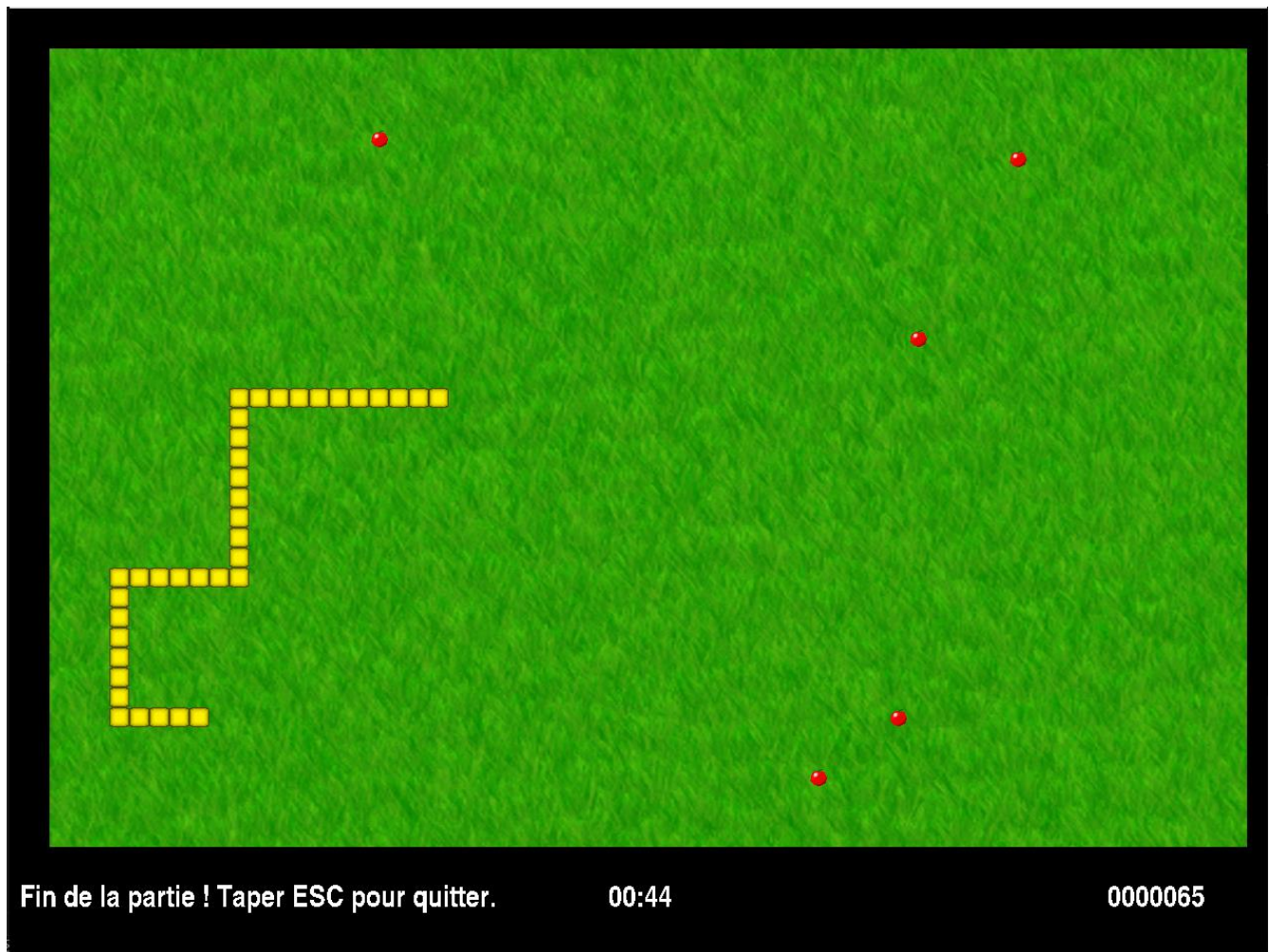
Quand on commence la partie, le serpent avance tout seul dans une direction choisie aléatoirement, et un chronomètre se lance. En appuyant sur les flèches directionnelles, il change de direction. Il peut tourner à droite et à gauche, mais il ne peut pas faire demi-tour. En appuyant sur la touche espace, le jeu se met en pause. Pour arrêter la pause, il suffit d'appuyer de nouveau sur la même touche.



(C'est difficile de voir la pause étant donné que c'est une image, mais le serpent est en pause dessus.)

DESCRIPTION DES FONCTIONNALITES

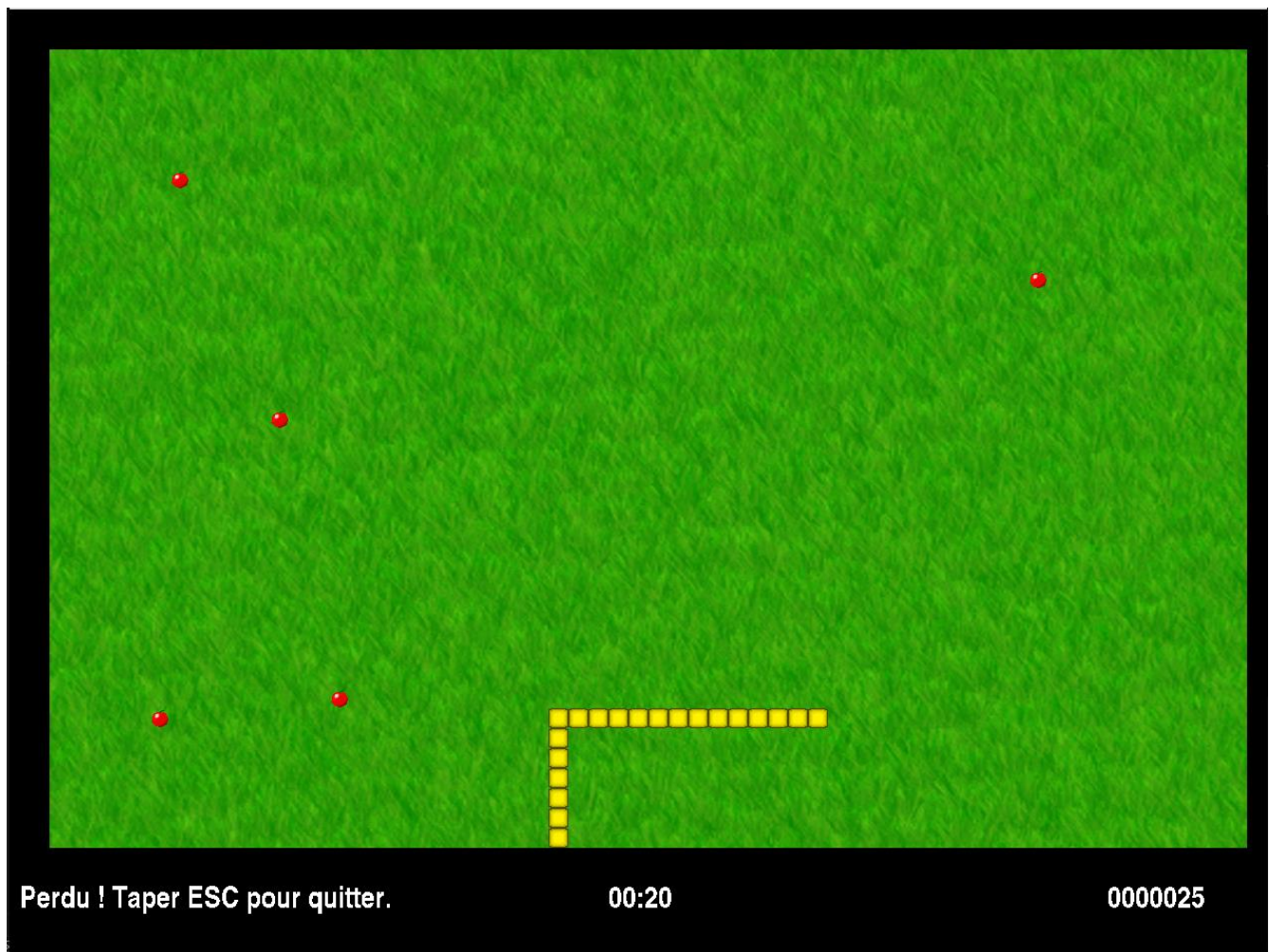
A chaque fois que le serpent mange une pomme, il grandit de 2 cases, le score augmente (5 fois le nombre de pommes mangées) et une autre pomme apparaît sur une case aléatoire sur le terrain. Pour quitter le jeu, appuyer une fois sur la touche échap, le score obtenu durant la partie sera affiché. On appuie de nouveau sur échap pour quitter le jeu.



DESCRIPTION DES FONCTIONNALITES

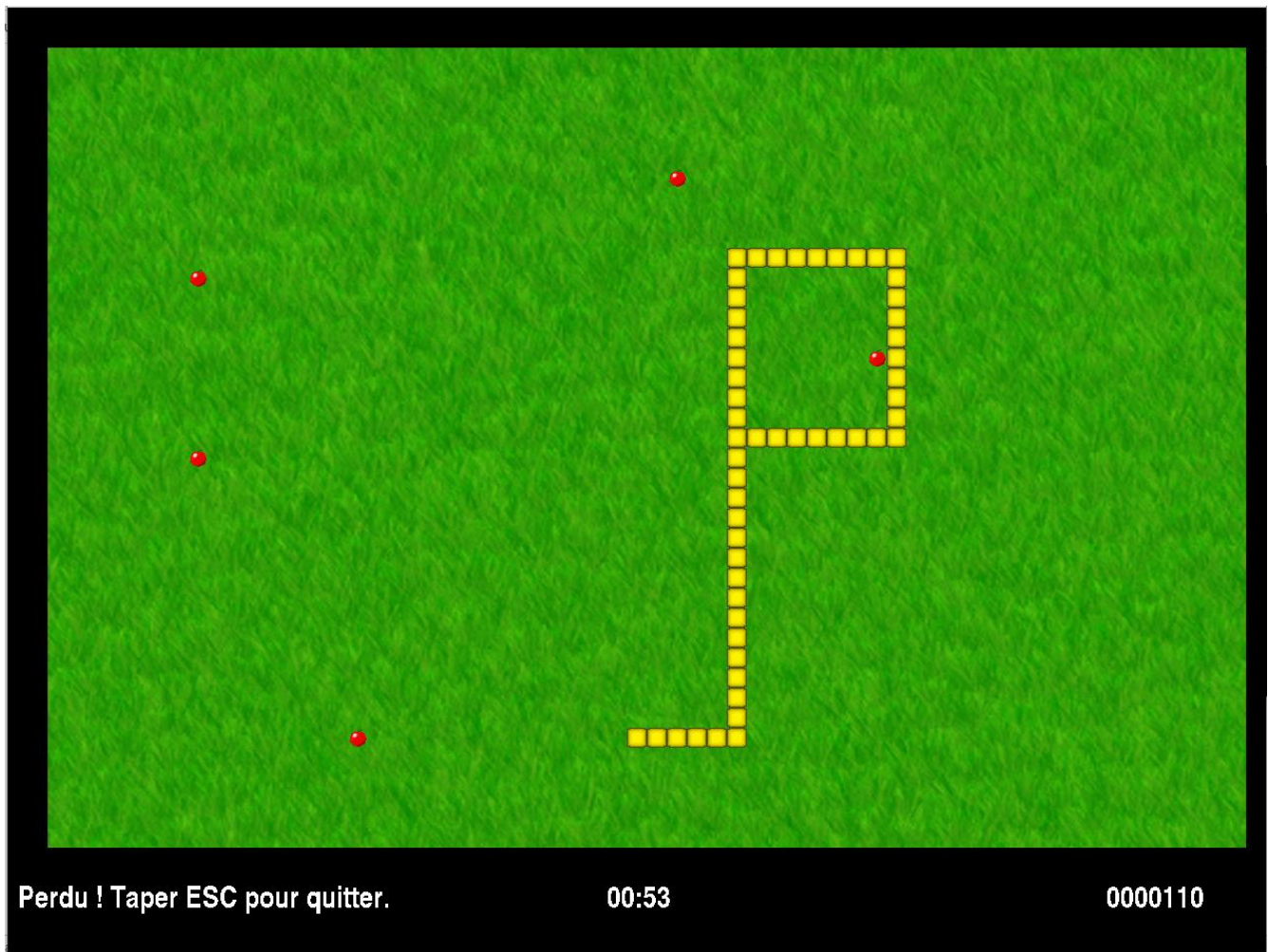
Si le serpent entre en collision avec un mur, on perd et doit faire échap pour quitter le jeu. Il se passe la même chose si le serpent entre en collision avec une partie de son corps.

Collision avec un bord

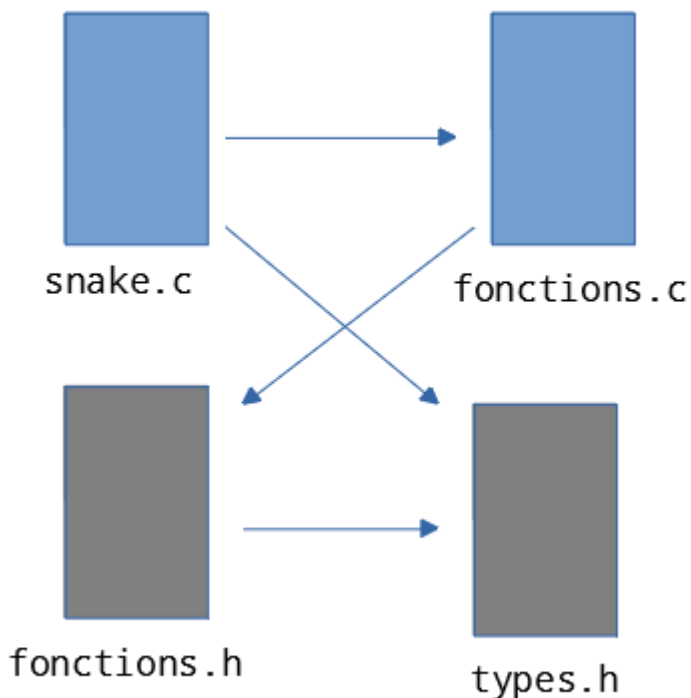


DESCRIPTION DES FONCTIONNALITES

Collision avec un segment de son corps.



III. Explication du découpage des fichiers sources



Le fichier principal du programme est **snake.c**.

L'autre fichier source est **fonctions.c**, il contient toutes les fonctions utilisées dans **snake.c**, on lui a créé un fichier d'en-tête **fonctions.h** qui regroupe tous les prototypes des fonctions.

Nous avons également créé un autre fichier d'en-tête **types.h** qui contient la déclaration des types, ici notre structure grille. On l'inclut dans **fonctions.h** qui utilise le type.

Grâce à notre **Makefile** nous pouvons créer les fichiers objets.

Les fichiers **pomme.png**, **herbe.png** et **snake.png** contiennent les images utilisées dans **snake.c** et **fonctions.c** pour la création des sprites.

FICHIERS SOURCES

La structure présente dans le fichier **types.h** est l'élément principal de notre jeu, elle contient : - la grille de jeu, un tableau de 62 par 42 contenant la valeur de ce qui se trouve sur le terrain : bordure (3), serpent (2), pomme (1) et terrain vide (0)

- les coordonnées de la tête du serpent
- les coordonnées de la queue du serpent
- la direction du serpent, 0=haut 1=bas 2=gauche 3=droite
- un indice de collision, s'il passe à 1 le jeu est fini
- le score
- le score en caractères
- le nombre de pommes mangées
- le cycle du serpent, sa vitesse

Exemple du tableau réduit d'une grille de jeu :

3	3	3	3	3	3	3	3	3	3
3	0	0	1	0	0	0	0	0	3
3	0	0	0	0	0	0	1	0	3
3	1	0	0	0	0	0	0	0	3
3	0	0	2	2	2	2	0	0	3
3	0	0	1	0	0	0	0	0	3
3	0	0	0	0	0	1	0	0	3
3	3	3	3	3	3	3	3	3	3

FONCTIONNEMENT DU SERPENT

IV. Fonctionnement du Serpent

Initialisé dans la grille (valeur 2), on place la tête au centre, décide de la direction puis 9 cases derrière la tête prennent la valeur du serpent. On entre dans la boucle du jeu, les données du serpent dans la grille sont mises à jour toutes les 0.1 secondes et à chaque fois que le joueur appuie sur une flèche directionnelle dans une fonction de mise à jour de la grille.

Toutes les trois pommes mangées, la vitesse du serpent accélère, c'est-à-dire que les données du serpent sont mises à jour plus rapidement, le cycle du serpent diminue de 5000 microsecondes soit 0.005 secondes.

Les données sont prises en compte dans la fonction, on s'occupe d'abord de la queue du serpent dans une fonction puis de la tête dans une autre, toutes deux dans celle de la mise à jour.

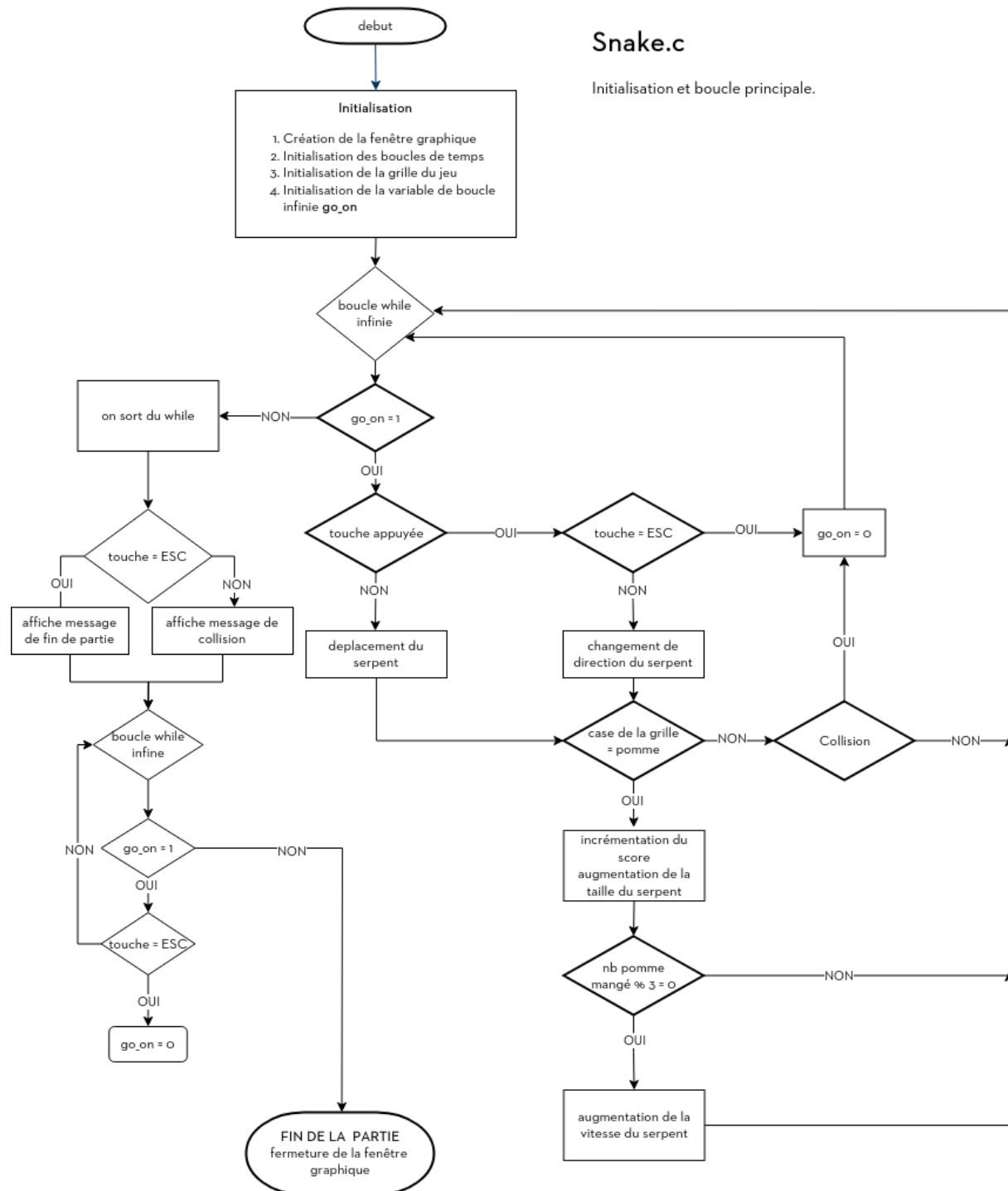
Les coordonnées de la dernière position de la queue sont recueillies et on regarde où se trouve la case qui porte la valeur du serpent autour (en haut, à gauche, à droite ou en bas) et on déplace la queue à cet endroit-là. Pour la tête, nous avons deux situations : le joueur a appuyé sur une touche ou le joueur n'a pas appuyé.

Si le joueur a appuyé sur une touche, par exemple celle du haut, on va vérifier que le serpent ne se dirige déjà pas vers le haut ou vers le bas, ensuite on vérifie que la prochaine case n'est pas un bord ou un segment du serpent car dans le cas contraire, le jeu est perdu pour cause de collision. Puis si la prochaine case a la valeur d'une pomme on met à jour le score et grâce à une autre fonction on déplace la tête du serpent et l'allonge de deux cases supplémentaires, si le serpent n'a pas mangé de pomme alors le serpent ne s'allonge pas.

Si le joueur n'a pas appuyé sur une touche, on fait les mêmes vérifications hormis celle de la touche évidemment.

Si la pause est enclenchée, il n'y a pas de mise à jour.

FONCTIONNEMENT DU SERPENT



V. Conclusion

Benjamin Bribant :

En conclusion, ce projet Snake a été aussi difficile qu'enrichissant. Il m'a permis d'apprendre à réfléchir pour surmonter des difficultés et a montré que le développement n'est pas aussi simple que certaines personnes peuvent le penser. Voir le jeu prendre forme au fur et à mesure que l'on travaillait dessus m'a donné envie de m'améliorer en développement et m'a aussi montré que cela peut être amusant.

Nell Telechea :

Pour conclure, j'ai également trouvé que le projet était difficile mais il m'a permis de mieux maîtriser les outils de programmation du C. L'usage des pointeurs et des passages par adresse était primordial, je ne les maîtrisais pas très bien au début mais à présent je les maîtrise mieux. C'était une occasion amusante d'appliquer concrètement tout ce que nous avons vu depuis le début de l'année.