

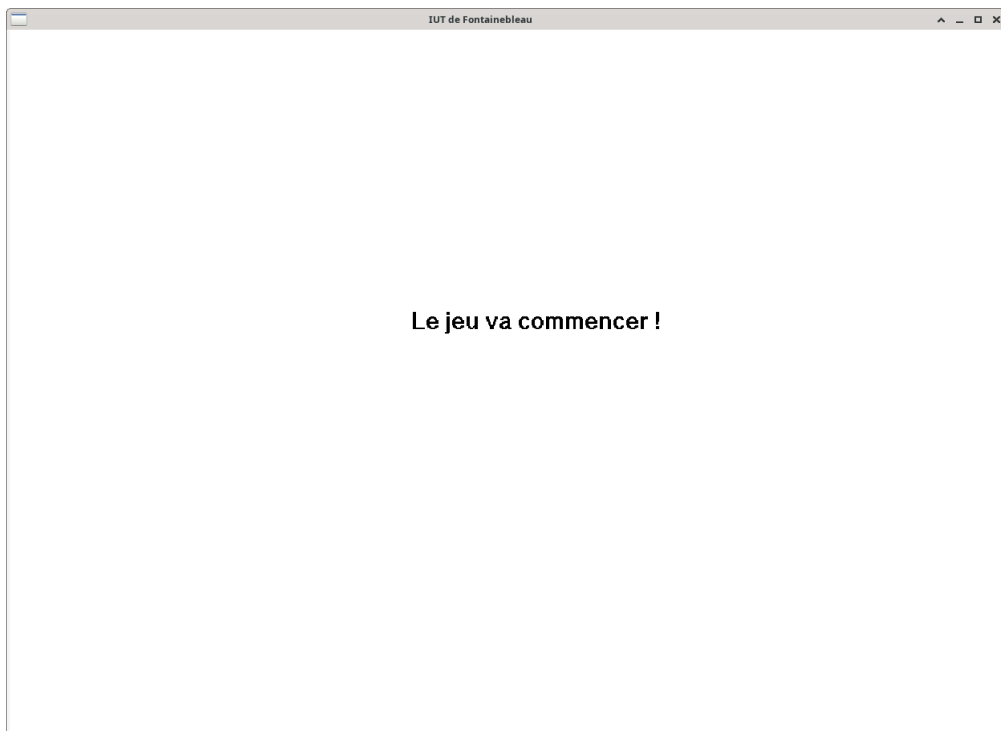
RAPPORT SAE11_2023

- **Introduction**

Il s'agit du célèbre jeu "Snake" remanier avec nos idées. Le joueur contrôle un serpent afin qu'il mange des pommes et que celui-ci grandisse petit à petit et devient lui-même un obstacle. Au fur et à mesure du jeu et que le serpent grandit, le jeu se corse. Ici, on ne grandit pas d'une case en une case mais de deux cases en deux cases. Ceci permet d'accélérer le jeu puisque le terrain fait 40 lignes de 60 colonnes. De plus, le serpent fait initialement 10 segments, contrairement à la version originale qui ne comporte qu'un seul segment. Le jeu de base ne contient pas d'obstacle, hormis le serpent lui-même et les bordures du terrain. Ici, nous avons ajouté des obstacles aléatoirement sur le terrain pour corser le jeu.

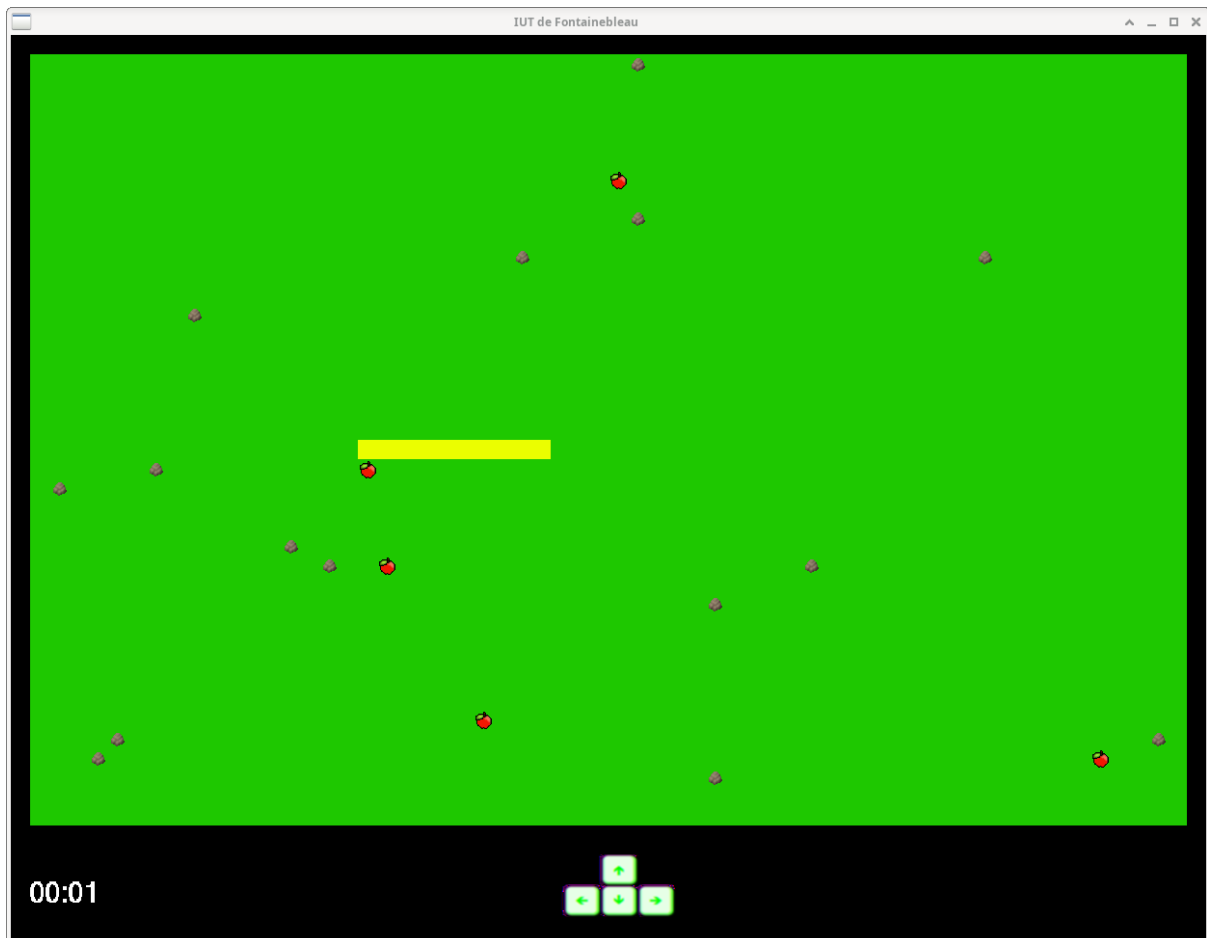
- **Fonctionnalités**

Avant la partie : Avant que le jeu commence, nous affichons une fenêtre qui dure 3 secondes. Ce temps permet au joueur de se préparer à jouer :

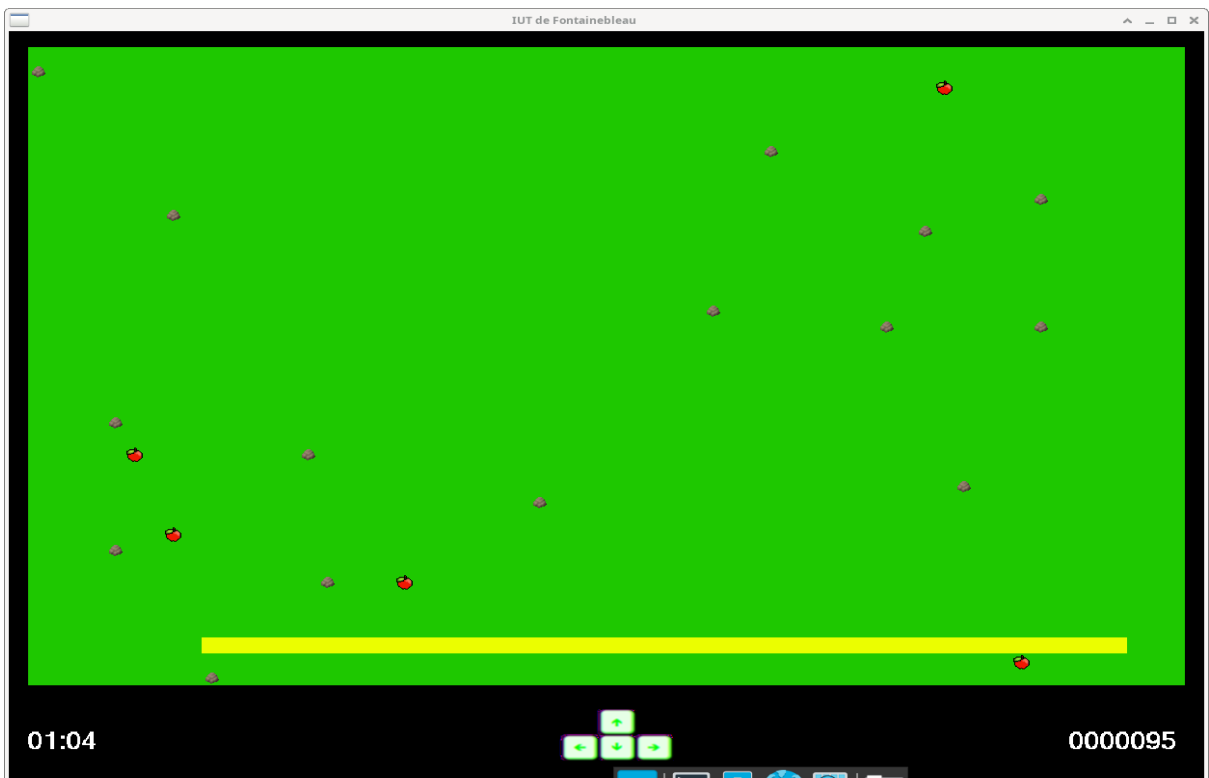
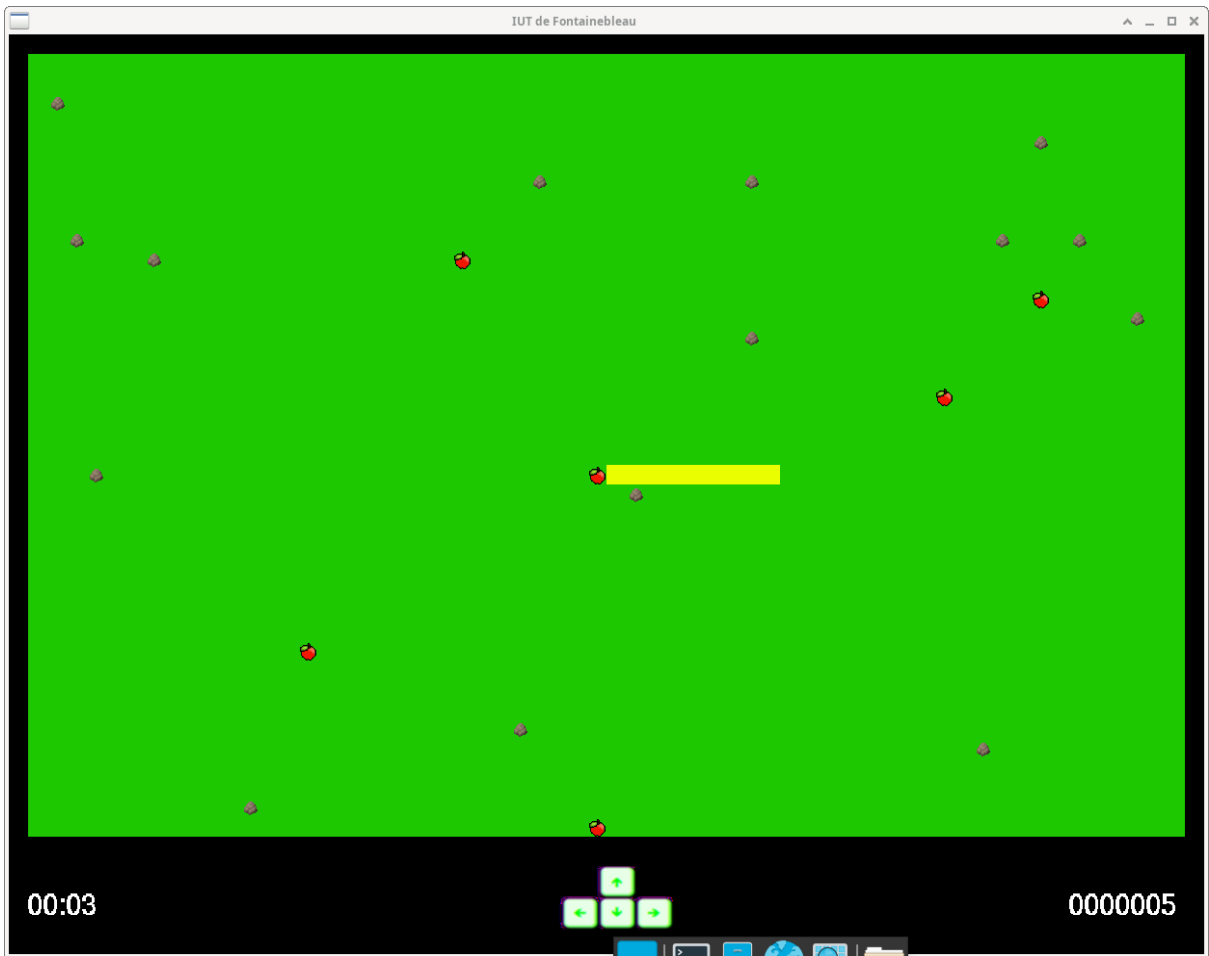


Début de la partie : Le serpent se trouve toujours au milieu du terrain et fait initialement 10 segments. Les pommes, qui sont au nombre de 5, sont placées aléatoirement sur le terrain. Les obstacles, qui sont eux au nombre de 15, sont eux aussi placés aléatoirement sur le terrain. Le serpent bouge par défaut à droite. Le temps s'affiche en bas à gauche dès que le jeu commence.

Déplacement : Pour déplacer le serpent dans la direction souhaitée, le joueur devra utiliser les flèches directionnelles du clavier, comme indiqué sur la fenêtre du jeu au milieu en bas.



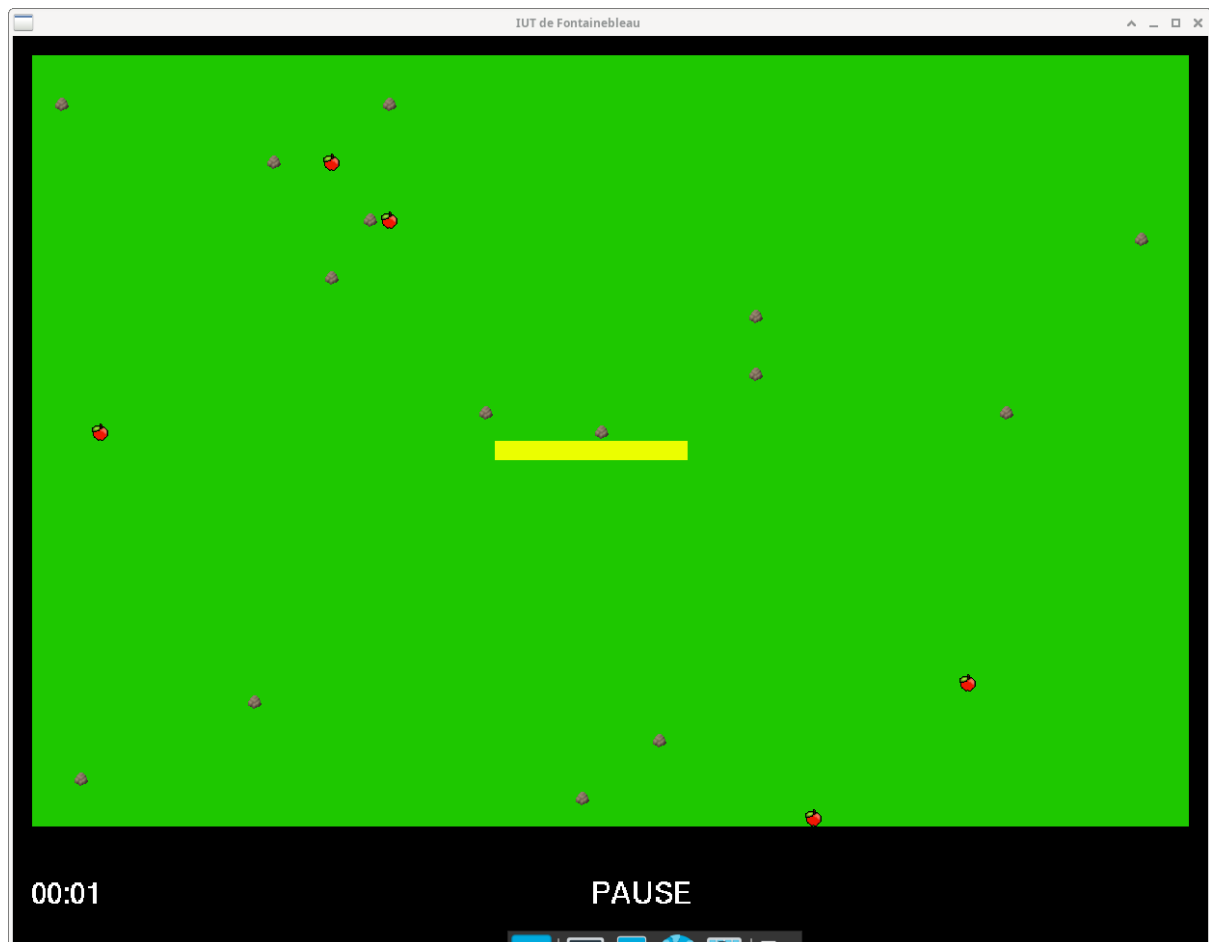
Pommes : Pour grandir, le joueur doit manger les pommes, représenté par des pommes rouges. Une pomme mangée équivaut à 5 points. Le score s'affiche ensuite en bas à droite. Si le joueur perd avant de manger une pomme, le score ne s'affiche pas. Une fois la pomme mangée, le serpent grandit de 2 segments.



Obstacles et Game Over : Les obstacles sont représentés par des petits rochers. Si le joueur touche un des rochers, il perd. Nous faisons alors apparaître un “GAME OVER” où le temps sera stoppé, le score reste affiché pour que le joueur puisse juger ses performances. De même pour les bordures ou lorsque le joueur touche une partie de son serpent. Une fois perdu, le joueur devra utiliser la touche ECHAP du clavier pour quitter la fenêtre.

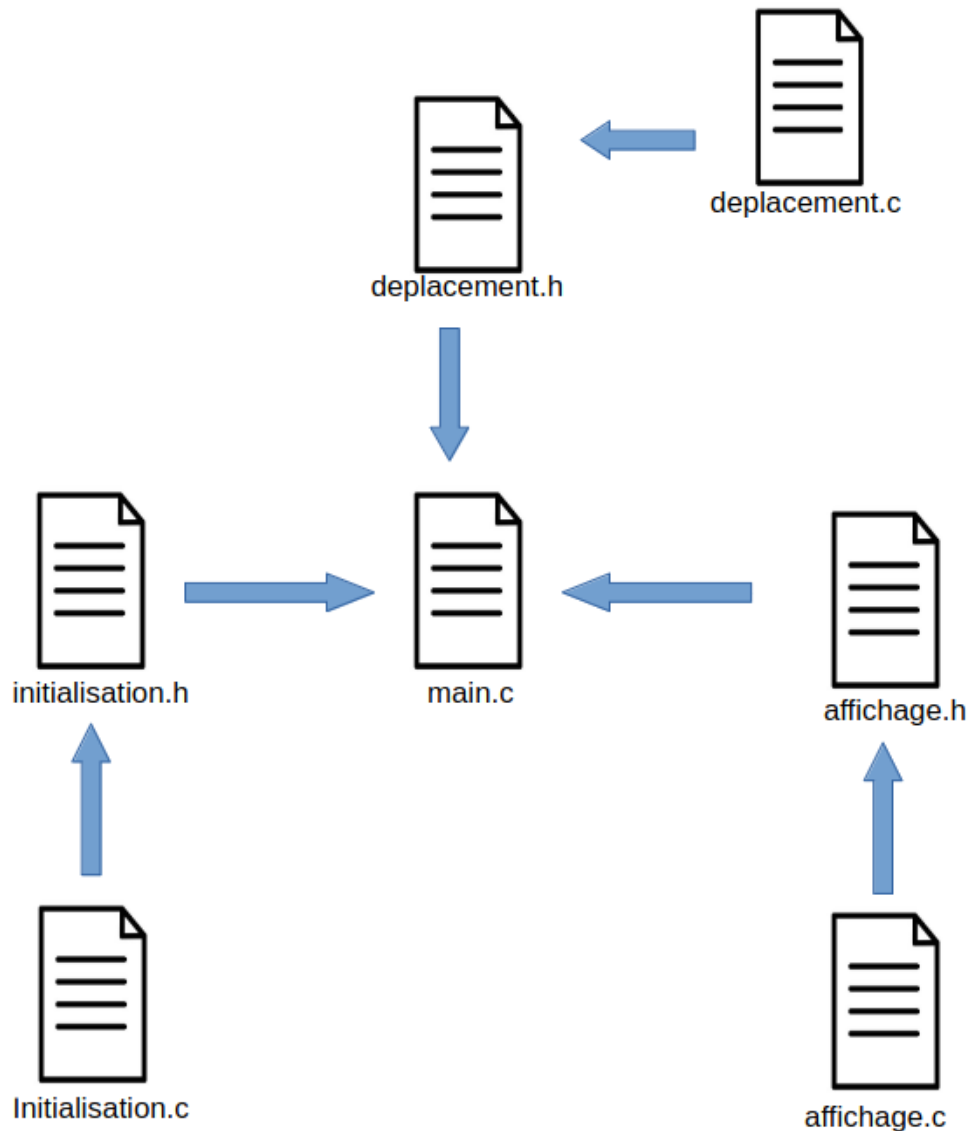


Pause : Le joueur peut mettre à tout moment pause à sa partie. Pour cela, il devra utiliser la touche ESPACE du clavier. Il sera alors indiqué à l'écran, à la place des touches, “PAUSE”. Pour revenir à la partie, le joueur devra utiliser la touche ESPACE du clavier et les touches réapparaîtront et le jeu reprendra.



Arrêter la partie : Le joueur aura à tout moment l'opportunité d'arrêter la partie avec la touche ECHAP du clavier. Ceci entraînera automatiquement la fermeture de la fenêtre.

- **Découpage des fichiers**



Nous avons découpé notre projet en 4 fichiers.

Le **main.c** permet de rassembler l'ensemble des fonctionnalités de notre snake et de joueur. Il prend donc en compte tous les fichiers d'en-tête conçu : "initialisation.h", "affichage.h" et "déplacement.h".

L'initialisation.c répertorie toutes les fonctions d'avant jeu et ne permet pas d'afficher quoi que ce soit. Ainsi, on peut y retrouver l'initialisation du serpent, des pommes et des obstacles avant même que ceux-ci soient affichés.

L'affichage.c permet cette fois-ci d'afficher tout ce que le joueur verra. Par exemple, le serpent, les pommes, le terrain en lui même, le temps etc...

Et **déplacement.c** permet toutes les logiques du déplacement du serpent, d'où son nom. Puisque les collisions avec des obstacles, le serpent et les bordures sont liées au déplacement, ce fichier permet aussi de gérer la fin de jeu et donc le Game Over.

- **Représentation des données**

Nous avons représenté nos données dans un tableau multidimensionnel de 40 lignes et 60 colonnes. Chaque valeur dans ce tableau représente un élément visible sur le terrain :

-2 : Correspond à une pomme.

-1 : Correspond à un obstacle.

0 : Case dites "vide". Il s'agit en réalité de toutes les zones vertes qui matérialisent le terrain.

1-* : Toutes les valeurs au-dessus de 0 correspondent au serpent. Les données ont été introduites dans ce tableau afin que la valeur 1 corresponde toujours à la queue du serpent, la valeur la plus grande du tableau (taille du serpent) correspond à sa tête et que toutes les valeurs entre ces 2 valeurs soit le corps.

Les positions dans le tableau des pommes et obstacles n'ont aucune raison de changer contrairement au serpent. La position de sa queue et de sa tête ne fait que bouger. Ainsi, dans notre tableau :

Toutes les valeurs du serpent diminuent de 1. Ainsi, la valeur auparavant de 1 devient un 0 donc une zone verte et la valeur 2 devient un 1 donc la queue. On ajoute simplement devant la valeur la plus haute, la valeur qui correspond à la taille du serpent pour contrebalancer le fait d'avoir perdu un bout du corps du serpent à l'arrière.

Lorsque le serpent mange une pomme. Nous détectons la position de la tête et de la queue, nous augmentons ensuite toutes les valeurs de 1. Ainsi, à ce stade le serpent n'a plus de queue puisqu'il n'y a aucune valeur à 1. Pour contrebalancer le fait qu'il ait perdu la queue, on ajoute donc un 1 à la position enregistrer. De plus, puisqu'il est demandé d'augmenter de 2 segments en 2 segments, nous ajoutons la valeur de la taille du serpent devant la tête actuelle et devient donc la tête du

[illegible]

- **Conclusion personnelle (Christopher Dubreuil)**

La réalisation du jeu a été, selon moi, quelque chose de difficile sans que cela soit insurmontable non plus. Une fois les bonnes idées en tête, chaque tâche à accomplir s'est avérée simple. Mais pour en arriver là, il fallait avoir l'idée. Le plus compliqué a donc été ici. Au-delà de la complexité de la conception du projet, j'ai bien aimé le réaliser. Même si j'aurais aimé ajouter un peu plus de fonctionnalités pour rendre le jeu meilleur, le temps qui nous a été donné pour réaliser uniquement ce qui a été demandé a été suffisant.