



Pour ce premier projet d'informatique nous avons pour mission de créer un *Snake*.

## I] – Plan d'attaque

### **A) Un *Snake*, mais qu'est-ce donc ?**

Dans le *Snake* original, le joueur dirige un serpent grandissant au fur et à mesure qu'il mange des pommes. Le but est de diriger le serpent le plus long possible. Celui-ci meurt s'il rencontre un obstacle en l'occurrence lui-même ou les bords de la fenêtre de jeu. Les pommes apparaissent une par une, ou il y a strictement une pomme sur le plateau.

### **B) Notre petit serpent à nous**

Dans le cadre du projet, nous avons été invités à apporter nos propres modifications aux règles du jeu.

Nous voulions commencer par avoir un écran titre (ainsi qu'un écran de fin de partie) proposant tous deux un accès à un menu par l'intermédiaire duquel le joueur pourrait contrôler les paramètres de jeu.

Le menu de fin de partie doit proposer au joueur de rejouer sans qu'il ait à redémarrer le jeu.

Nous voulions aussi proposer (en vrac) :

- Un score (affiché au cours de la partie) correspondant aux nombres de pommes mangées
- Des décors et une musique de fond, ainsi que des bruitages pendant la partie
- Une vitesse de jeu qui augmente en fonction du score du joueur
- Un paramètre permettant au joueur de choisir la couleur de son serpent et la vitesse de jeu initiale
- Un mode deux joueurs ou chacun des deux joueurs contrôle un serpent infini et doit encercler l'autre
- Un mode torique ou le joueur doit évoluer dans une arène où la collision avec les murs est impossible

- Un mode de jeu infini où le serpent arrive dans un nouveau décor quand il sort de la fenêtre. Il aurait été intéressant que le serpent soit capable de revenir à l'endroit dont il est parti.
- Des obstacles et des missiles qui apparaissent de manière aléatoire à partir d'un certain score
- Un mode permettant la présence simultanée de plusieurs ou aucune pommes sur le plateau.
- 

## **II] - Réalisation du projet**

### **A) Modules et structures**

Nous avons utilisé pour programmer la bibliothèque upemtk ainsi que la fonction sleep du module time. Pour la génération aléatoire nous nous sommes servis du module random (notamment des fonction choice et randint) et enfin nous avons importé le module doctest.

Un algorithme présentant le fonctionnement en gros du programme est joint au rendu au format .png

### **B) Variables et fonctions au cœur du programme**

- La fonction case\_vers\_pixel nous permet de découper la fenêtre en un plateau de taille largeur plateau par longueur plateau

- parmi les variables principales de notre programme, il y a un certain nombre de listes représentant chacune un type d'objet.

Serpent représente les cases adjacentes contenant le serpent à chaque tour

Pommes représente la liste de coordonnées des cases contenant les pommes. Il en est de même pour murs (cases contenant des murs) et pour tête (cases contenant des missiles)

La variable direction est un couple de coordonnées donnant la direction dans laquelle ce dirige le serpent et le joueur peut contrôler cette variable au moyen du pavé directionnel.

La direction des missiles est donné par la liste dir\_tetes de la même longueur que tete car ils sont tous indépendants les uns des autres.

Les fonctions sont ensuite appelées en suivant le modèle de l'algorithme du programme.

.

### **C) Problèmes rencontrés**

- Nous ne sommes pas parvenus à inclure la dimension sonore dans notre jeu, car nous avons eu des difficultés pour trouver, installer et manipuler les modules permettant la lecture de fichier audio avec python (par exemple pyaudio ou pygame). Nous avons donc vite abandonné cette idée.

D'autres obstacles mineurs ont été écartés avec l'ajout de nouvelles fonctions de quelques lignes çà et là. (par exemple la fonction `cache_serpent()` pour faire en sorte que le serpent ne voit pas disparaître sa queue devant lui quand il change de plateau dans l'arène infinie).

Certaines difficultés sont expliquées plus en détails dans la fiche d'auto-évaluation (ci-jointe).

En cas de problème avec le lancement du programme, vérifiez que les images sont bien dans le même dossier et redémarrez le noyau.

## **III] - Conclusion**

Ce projet nous a donné l'opportunité de faire face à des défis stimulants, qui ont nécessité un travail de recherche et d'inventivité de notre part. La liberté dont nous jouissions en ce qui concerne l'ajout des fonctionnalités et des modes nous a permis de n'être limité que par notre imagination et nos compétences et les différents échecs d'un peu mieux connaître nos limites.

Le travail fut réparti au sein du binôme et nous pensons avoir progressé durant la réalisation du projet.

### **Sources :**

[https://fr.wikipedia.org/wiki/Snake\\_\(genre\\_de\\_jeu\\_vidéo\)#Sur\\_téléphone Nokia](https://fr.wikipedia.org/wiki/Snake_(genre_de_jeu_vid%C3%A9o)#Sur_t%C3%A9l%C3%A9phone_Nokia)

*(Les images autres que celle de l'écran-titre et l'animation ont été réalisées par nous-mêmes grâce au logiciel Piskel)*