

Snake Python

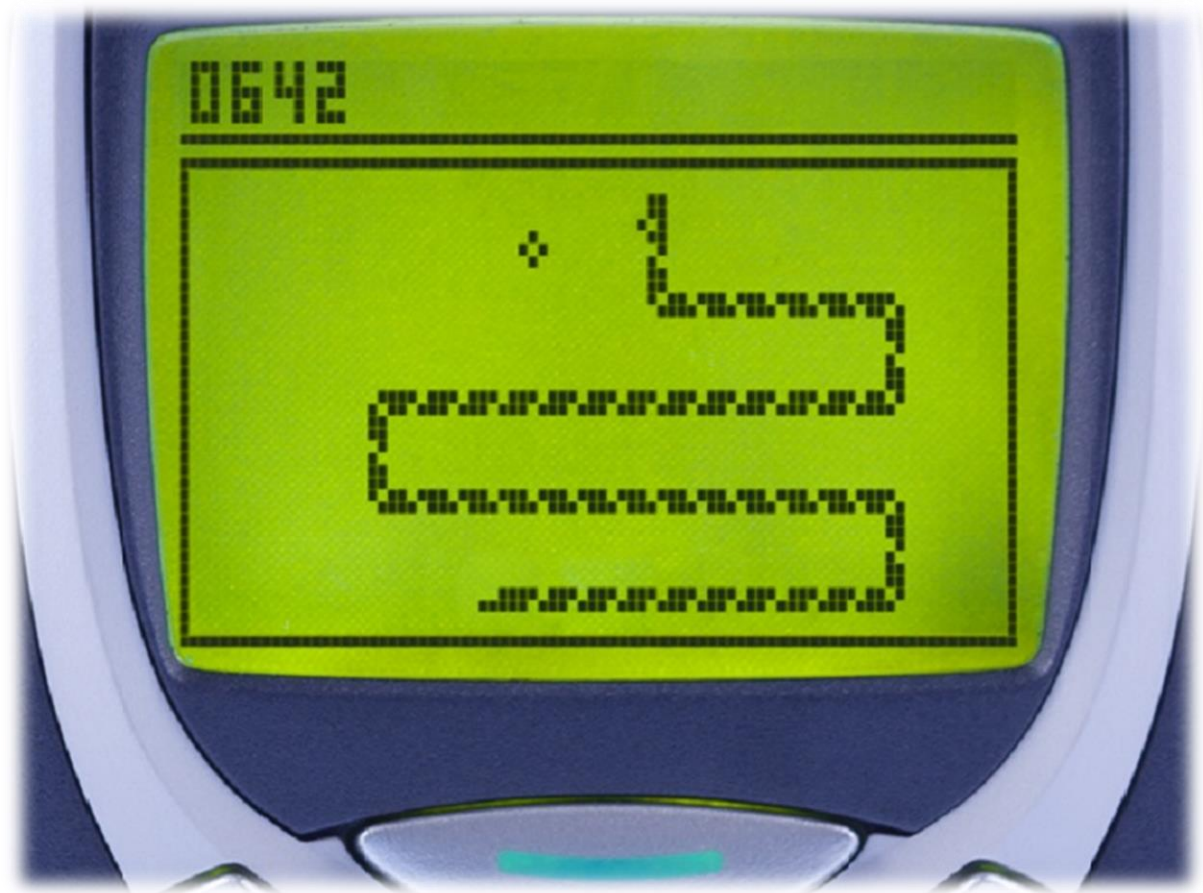
version 1.3.1



DIVERSITY
by EPITECH

I. Introduction

Snake est un jeu originellement sorti en 1997 sur le téléphone portable Nokia 6110. Programmé par Taneli Armanto, il obtint le titre du tout premier jeu mobile, et causa le succès de ce genre depuis sa sortie jusqu'à notre génération actuelle.



* « When asked why he called the new software, "git," British slang meaning "a rotten person," he said. "I'm an egotistical bastard, so I name all my projects after myself. First Linux, now git." », source.

II. Consignes

- * Vous allez devoir compléter des fonctions du jeu Snake sous Python avec la librairie Pygame (spoiler : ça va être dur section)
- * Si vous avez des questions, demandez à votre voisin de gauche, puis de droite. Coopérez !
- * En cas de blocage, et après avoir tenté la consigne précédente, n'hésitez pas à demander à un Cobra, ils sont très gentils et sont là pour vous aider.
- * Vous avez entièrement le droit d'utiliser internet comme vous le souhaitez pour vous renseigner, nous le conseillons fortement.
- * Si vous n'avez jamais fait de Python ou de programmation auparavant et que vous souhaitez apprendre quelques notions de base avant de vous lancer : demandez à un Cobra de vous assister et rendez [ici](#) pour apprendre les bases.
- * *N'hésitez pas à faire des bonus et ajouter/modifier des fonctionnalités à votre programme, la seule limite est votre imagination ! (Et 2h30 environ)*

- * Amusez-vous ! Nous ne sommes pas là pour apprendre mais pour découvrir, vous avez le droit de ne pas réussir à compléter le sujet. 😊

III. Pourquoi il fait tout noir ?

Nous allons tous d'abord commencer par dessiner les éléments de notre jeu sur la fenêtre d'affichage.

Grâce à la librairie Pygame, (et à nos superbes fonctions que l'on a déjà codées pour vous !) vous allez pouvoir réaliser cette tâche en un rien de temps.

Vous avez 3 éléments à afficher :

Self : la tête de notre snake

Apple : les pommes qui vous permettront de grandir et de gagner des points

Score_board : votre score actuel

Ces éléments étant des *Object*, ils possèdent des fonctions qui leur appartiennent, comme la fonction **draw()** leur permettant d'être dessiné sur votre écran.

Si vous ne savez pas comment utiliser les fonctions d'un *Object*, voici un bref exemple :

```
takumi.drift()
```

À vous de jouer !

Pour cette tâche ça se passe ici ligne 150 ↓

```
140 .....'''DRAW'''
141 .....pygame.draw.rect(window, BLACK, (0, 0, window_length, window_height))
142 .....for tail in tail_list_object:
143 .....    if tail.time == 0:
144 .....        tail.remove_self()
145 .....        elif tail.x == head.x and tail.y == head.y and die_from_tail == True:
146 .....            reset_game()
147 .....        else:
148 .....            tail.time -= 1
149 .....            tail.draw()
150 .....
151 .....pygame.display.update()
```

IV. Le mouvement du prédateur

Désormais vous pouvez visualiser l’affichage de votre snake, ainsi que les pommes qu’il désire manger.
Mais petit bémol : il reste immobile, figé sur place.

Vous allez régler cela en ajoutant un peu de mouvement, l’Object `head` possède 3 variables importantes en lui :

`head.direction` : un *str* qui vous indiquera la direction vers laquelle la tête est tournée. (« up » / « down » / « right » / « left »)

`head.x` : un *int* définissant la position sur l’axe X de la tête de votre snake.

`head.y` : similairement à `head.x`, il définit la position sur l’axe Y de la tête de votre snake.

Grâce à ces 3 variables, vous devriez pouvoir coder un bref algorithme ajoutant un mouvement continu à votre snake.

À vous de jouer !

Cette fois-ci c’est ligne 121 que ça se passe ↓

```
120 ..... ' ' 'SNAKE MOVE' ' '
121
122
```

V. Attention à ne pas vous cogner

Votre snake est enfin prêt à chasser toutes les pommes qu'il trouvera sur sa route, enfin, presque.

Lors de vos essais vous avez dû remarquer que dans sa folie meurtrière, votre snake ne s'arrêtera pas devant les bordures de votre écran et disparaîtra derrière celles-ci.

Votre tâche ici sera d'empêcher votre snake de s'enfuir de sa cage.

Pour cela vous aurez à votre disposition 4 variables :

`window.length` : taille de votre écran en largeur.

`window.height` : taille de votre écran en hauteur.

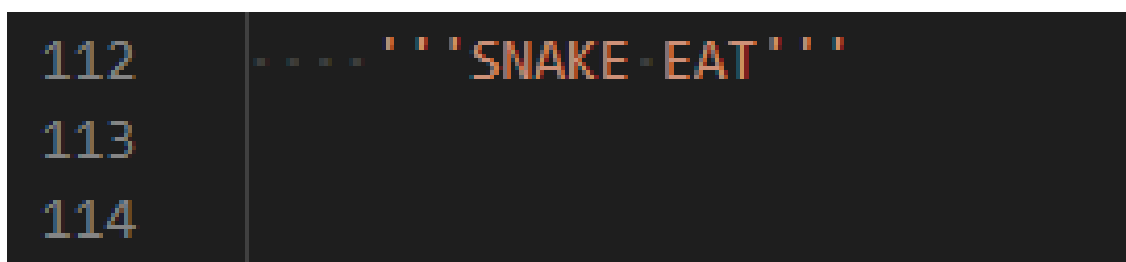
`head.length` : taille de la tête de votre snake en largeur.

`head.height` : taille de la tête de votre snake en hauteur.

À chaque fois que votre snake dépassera s'approchera de la bordure, vous devrez vous assurer qu'il n'ira pas plus loin que cette dernière.

À vous de jouer !

Rendez-vous juste en dessous de SNAKE EAT ↓



VI. Un festin plus riche que le BBQ du BDE

Votre jeu est presque complet !

Tout ce qu'il vous reste à faire, c'est de faire en sorte que votre snake ne meurt pas de fin et puisse manger autant de pomme qu'il le désire afin de grandir sans limite.

Cette fois-ci, vous allez devoir vous montrer autodidacte, car vous n'aurez pas d'explication sur les Object à utiliser, vous allez devoir les examiner par vous-même pour comprendre leur utilité.

Les Object/variables/fonctions en question :

apple.x

apple.y

apple.spawnNewLocation()

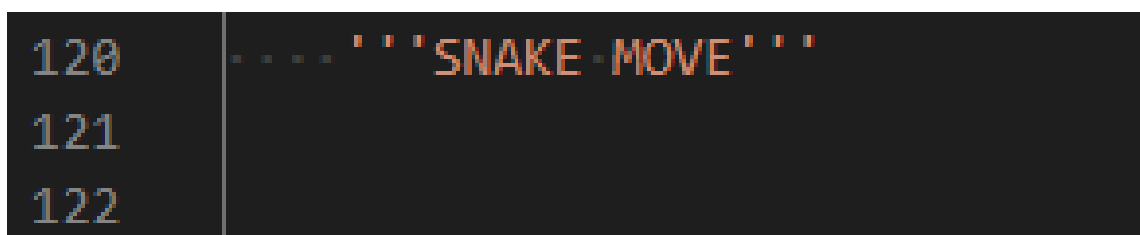
tail_length

score_board.score

Petit conseil : jetez un coup d'œil aux Class au tout début de votre fichier. Si vous n'y comprenez rien ou ne savez pas ce qu'est une Class, demandez un coup de main !

À vous de jouer !

Le grand final c'est en dessous de SNAKE MOVE ↓



VII. Bonus

Félicitations !

Vous avez réussi à venir à bout de chacune des tâches de ce sujet !

Vous êtes désormais un(e) professionnel(le) du Python sous la librairie Pygame, vous pouvez vous en vanter.

Mais faites preuve de modestie, pourquoi pas réaliser quelques bonus afin d'élargir vos compétences ?

Voici une petite liste de bonus conseillés si vous n'avez pas d'idée :

- * Rajoutez différents types de fruit
- * Faire un mode 2 Joueurs
- * Mettre un peu d'ambiance sonore
- * Rajouter des obstacles comme des murs ou des pièges
- * Coder une IA contrôlant votre snake
- * *Absolument n'importe quel autre bonus dont la seule limite est votre imagination !*

