

MASTERE 1 DATA & IA



RAPPORT DU PROJET Computer Science Basics

HENRI OTHNIEL GOHI



Professeur

Horairy HAKIM

SOMMAIRE

I- PROCEDURE DE DEVELOPPEMENT	1
II- FONCTIONNALITES DEVELOPPEES	2
III- DIFFICULTES RENCONTREES.....	3

I- PROCEDURE DE DEVELOPPEMENT

- Création d'une classe Block() pour l'affichage des éléments du jeu sur le terrain de jeu
- Stocker les cellules occupées par le snake dans une liste.
Exp : body = [Block(x=4, y =4), Block(x=5, y =4), Block(x=6, y =4)]
x : colonne et y : ligne
- Pour dessiner le snake sur le terrain de jeu, il suffit de dessiner les blocks de même couleur dans les cellules représentées par les coordonnées du snake.
- C'est le même processus pour dessiner les bords du terrain de jeu, la nourriture du serpent et les obstacles.

Comment déplacer le snake ?

- Création de la fonction **move_snake()** qui sera exécutée de base toutes les 100 ms.
- Cette fonction permettra de mettre à jour les coordonnées du snake suivant les directions dans lesquelles on se déplace.
- Il nous faut donc une variable « direction » pour savoir dans quelle direction l'on se déplace (bas, gauche, haut, droit).
- C'est l'utilisateur qui pourra mettre à jour cette variable en appuyant sur les touches directionnelles du clavier.

Comment calculer les coordonnées du snake après un déplacement ?

- **Règle 1** : La tête du snake est toujours le dernier élément de la liste body.
- **Règle 2** : A chaque déplacement, on crée une nouvelle tête qu'on ajoute à body et on supprime la queue du snake, c'est-à-dire le premier élément de la liste body.

Exp : **Si direction = « RIGHT » :**

$x_new_head = x_old_head + 1$

$y_new_head = y_old_head$

Si direction = « TOP » :

$x_new_head = x_old_head$

$y_new_head = y_old_head - 1$

Si direction = « LEFT » :

$x_new_head = x_old_head - 1$

$y_new_head = y_old_head$

Si direction = « DOWN » :

$x_new_head = x_old_head$

$y_new_head = y_old_head + 1$

Que faire lorsque le snake mange la pomme ?

Si la tête du snake occupe la même cellule que la nourriture :

- On génère une nouvelle nourriture
- On augmente la taille du snake d'une cellule
- On ne supprime pas la queue
- On incrémente le score de 2

II- FONCTIONNALITES DEVELOPPEES

Voici quelques fonctionnalités clés que nous avons développées dans notre jeu Snake :

- ✓ **game_screen** : Affichage du terrain de jeu et du serpent
- ✓ **notice_game()** : Affichage des instructions du jeu

- ✓ **move_snake()** : Gestion des contrôles pour déplacer le serpent (par exemple, les touches fléchées)
- ✓ **game_over()** : Détection de collision avec les bords du terrain de jeu ou avec les obstacles ou avec le corps du serpent
- ✓ **auto_generate_food()** : Génération aléatoire de la nourriture sur le terrain de jeu
- ✓ **check_collision_snake_food()** : Gestion de la croissance du serpent
- ✓ **check_collision_snake_food()** : Mise à jour du score
- ✓ **game_paused()** : Possibilité de mettre le jeu en pause et de le redémarrer
- ✓ **high_score()** : Le meilleur score
- ✓ **check_collision_snake_food()** : Augmenter la vitesse de jeu chaque fois que le serpent mange cinq (5) pommes
- ✓ Son et musique

III- DIFFICULTES RENCONTREES

Au cours de la réalisation de notre projet, nous avons rencontré des difficultés de tout genre :

- Compréhension et utilisation de la bibliothèque Pygame
- Compréhension des dimensions de la fenêtre (colonne, ligne)
- Affichage du corps du serpent au cours de son déplacement
- Développement de la fonctionnalité **pause/play**
- La gestion du meilleur score
- Respect des bonnes pratiques pep 8

Cependant, nous ne nous sommes pas apitoyés sur notre sort pendant toute la durée de la réalisation du projet. Nous avons eu recours à plusieurs solutions que nous citerons par la suite relativement à chaque difficulté rencontrée.

- ✓ Pour ce qui concerne la compréhension et l'utilisation de Pygame, nous avons fait recours à un tutoriel sur YouTube dont le lien est le suivant :

<https://www.youtube.com/watch?v=WxEmlz009U&list=PLKeQQTikvsqkeJlhiE8mXwskOhXLKdl8m>

Ce tutoriel nous a également aidé à comprendre le dimensionnement des fenêtres et des objets à afficher sur la fenêtre suivant la longueur et la largeur.

Nous nous sommes également servi de la documentation de la bibliothèque Pygame pour comprendre certains concepts.

[Pygame Front Page — pygame v2.1.4 documentation](#)

- ✓ Pour ce qui concerne l'affichage correcte du corps du serpent, nous avons fait recours au professeur qui nous a donné une à laquelle nous n'avions pas pensé. Cette idée a résolu le problème.
- ✓ Le développement de la fonctionnalité **pause/play** est finalement parti d'une intuition
- ✓ La résolution du problème de meilleur score s'est faite grâce à la lecture d'un article dont le lien est le suivant :
[Réalisation et affichage d'un classement \[débutant\] - Python \(developpez.net\)](#)

L'idée nous est venue après avoir compris qu'il fallait enregistrer les scores dans un fichier texte

- ✓ Pour le respect des bonnes pratiques pep8 :
[\(32\) Write Python Code Properly! - YouTube](#)