

Fiche Explicative du projet Snake

Titre du projet : Jeu Snake en Python

Description : Dans le cadre de mon BTS SIO option SLAM, j'ai développé un jeu Snake en utilisant le langage de programmation Python. Le jeu repose sur une interface graphique simple mais efficace, conçue avec la bibliothèque Pygame. L'objectif du jeu est de contrôler un serpent qui se déplace sur l'écran, en évitant les obstacles et en mangeant des pommes pour grandir. Le score augmente à chaque pomme consommée, et la partie se termine si le serpent entre en collision avec les bords de l'écran ou avec lui-même.

Fonctionnalités principales :

- Interface utilisateur interactive et intuitive.
- Déplacement fluide du serpent à l'aide des touches fléchées du clavier.
- Gestion des collisions et conditions de fin de jeu.
- Système de score affiché en temps réel.
- Augmentation progressive de la difficulté avec la croissance du serpent.

Technologies utilisées :

- Langage de programmation : Python
- Bibliothèque graphique : Tkinter

Objectifs pédagogiques : Ce projet m'a permis de renforcer mes compétences en programmation Python, en particulier dans les domaines suivants :

- Utilisation de bibliothèques externes.
- Gestion des événements et des interactions utilisateur.
- Conception de jeux vidéo simples.
- Gestion de la logique de jeu et des algorithmes de collision.

Perspectives d'amélioration :

- Ajout de niveaux avec des obstacles variés.
- Intégration de sons et de musiques pour améliorer l'expérience de jeu.
- Sauvegarde et chargement des scores.
- Portage du jeu sur d'autres plateformes.

```

GAME_WIDTH = 800
GAME_HEIGHT = 800
SPEED = 100
SPACE_SIZE = 50
BODY_PARTS = 3
SNAKE_COLOR = "#00FF00"
FOOD_COLOR = "#FF0000"
GRID_SIZE = 50
diff_verif = 1
colorTemp1 = "#F5B041"
colorTemp2 = "#F39C12"

window = Tk()
window.title("Snake")
window.resizable(False, False)

# Créer le canvas
canvas = Canvas(window, bg=colorTemp1, height=GAME_HEIGHT, width=GAME_WIDTH)

# Ajouter la grille
for i in range(0, GAME_WIDTH, GRID_SIZE):
    for j in range(0, GAME_HEIGHT, GRID_SIZE):
        if (i//GRID_SIZE + j//GRID_SIZE) % 2 == 0:
            color = colorTemp1
        else:
            color = colorTemp2
        canvas.create_rectangle(i, j, i+GRID_SIZE, j+GRID_SIZE, fill=color, outline="")

```

```
def change_direction(new_direction):
```

```
    global direction
```

```
    if new_direction == 'left':
        if direction != 'right':
            direction = new_direction
    elif new_direction == 'right':
        if direction != 'left':
            direction = new_direction
    elif new_direction == 'up':
        if direction != 'down':
            direction = new_direction
    elif new_direction == 'down':
        if direction != 'up':
            direction = new_direction
```

```
def check_collisions(snake):
```

```
    x, y = snake.coordinates[0]
```

```
    if x < 0 or x >= GAME_WIDTH:
        print("GAME OVER")
        return True
```

```
    if y < 0 or y >= GAME_HEIGHT:
        print("GAME OVER")
        return True
```

```
    for body_part in snake.coordinates[1:]:
        if x == body_part[0] and y == body_part[1]:
            print("GAME OVER")
            return True
```

```
    return False
```