

Snake en Python, le plus simplement possible

Version 0

Pygame

On importe pygame avec :

```
import pygame
from pygame.locals import *
```

Le second import sert à quitter le jeu proprement.

Constantes

On crée quelques constantes :

```
HAUTEUR = 600 # hauteur de la fenetre
LARGEUR = 600 # largeur de la fenetre
BLOC = 60
```

```
NOIR = (0, 0, 0)
ROUGE = (255, 0, 0)
JAUNE = (255, 255, 0)
VERT = (0, 255, 0)
CYAN = (0, 255, 255)
```

```
FPS = 30
```

initialisation

On initialise le jeu :

```
pygame.init()
horloge = pygame.time.Clock()
fenetre = pygame.display.set_mode((LARGEUR, HAUTEUR))
pygame.display.set_caption('Snake')

pygame.display.update()
```

Boucle Infinie

Tous les jeux comportent une boucle infinie. Celle-ci ne contient pas grand chose :

- quitter le jeu,
- remplir la fenêtre de noir

- faire avancer l’horloge
- mettre à jour les affichages

Boucle infinie

```
while True:
    for event in pygame.event.get():
        if event.type == QUIT:
            pygame.quit()
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                pygame.quit()
    fenetre.fill(NOIR)
    horloge.tick(FPS)
    pygame.display.update()
```

Boucle infinie

- La boucle `for event in...` permet de récupérer les événements “cliquer sur la croix” ou “appuyer sur Escape” et quitte le jeu dans ce cas.
- Ensuite on dessine la fenêtre, remplie de noir
- On fait avancer l’horloge
- On affiche tout ça

Version 1

Ecrire du texte

Cette fonction nous permettra d’écrire facilement le score

```
def drawText(text, font, surface, x, y):
    textobj = font.render(text, 1, CYAN)
    textrect = textobj.get_rect()
    textrect.topleft = (x, y)
    surface.blit(textobj, textrect)
```

Taille de la police, valeur du score

```
font = pygame.font.SysFont(None, 48)
```

et

```
score = 0
```

Le serpent

Le serpent est une double liste

```
snake = [[3, 3], [2, 3], [1, 3]]
```

Le premier élément est sa tête, elle est en [3,3] ensuite vient son corps. Il commence donc avec une taille de 3.

Dessiner le serpent

Dans la boucle infinie, avant l'horloge :

```
for elt in snake[1:]:
    pygame.draw.rect(fenetre, VERT,
                     (elt[0] * BLOC, elt[1] * BLOC,
                      BLOC, BLOC))
pygame.draw.rect(fenetre, JAUNE,
                 (snake[0][0] * BLOC, snake[0][1] * BLOC,
                  BLOC, BLOC))
```

Le corps est vert et la tête jaune.

Afficher le score

On utilise notre fonction créée plus tôt :

```
drawText(str(score), font, fenetre,
         0.2*LARGEUR, 0.2*HAUTEUR)
```

Version 2

Capturer les touches du jeu

On ne capturerait que "Escape" et le clic sur la croix. On ajoute les flèches.

```
if key:
    if key[pygame.K_UP]:
        print("up")
    if key[pygame.K_DOWN]:
        print("down")
    if key[pygame.K_LEFT]:
        print("left")
    if key[pygame.K_RIGHT]:
        print("right")
```

Capturer les touches du jeu

Pour l'instant elles ne font rien d'autre qu'afficher du texte

Version 3

Diminuer la vitesse de rafraîchissement

FPS = 3

Déplacer le serpent

On commence par créer une direction (= la vitesse)

```
direction = (1, 0)
```

Déplacer le serpent

Chaque pression d'une flèche change la direction :

```
key = pygame.key.get_pressed()
if key:
    if key[pygame.K_UP]:
        direction = (0, -1)
    if key[pygame.K_DOWN]:
        direction = (0, 1)
    if key[pygame.K_LEFT]:
        direction = (-1, 0)
    if key[pygame.K_RIGHT]:
        direction = (1, 0)
```

Déplacer le serpent

Ensuite la tête.

C'est l'ancienne tête, qui s'est déplacée :

```
head = [snake[0][0] + direction[0],
        snake[0][1] + direction[1]]
```

Déplacer le serpent

Le corps se déplace.

1. On ajoute la tête au début :

```
snake.insert(0, head)
```

2. On perd un élément de fin :

```
snake.pop(-1)
```

Version 4

La mort du serpent

Il meurt :

- s'il quitte l'écran
- si sa tête est dans son corps

La mort du serpent

```
if head in snake[1:] \
    or head[0] < 0 \
    or head[0] > LARGEUR / BLOC - 1 \
    or head[1] < 0 \
    or head[1] > HAUTEUR / BLOC - 1:
    pygame.quit()
```

Version 5

Fluidité

Le jeu n'est pas fluide.

On va mettre à jour les éléments du jeu toutes les 1.5 secondes et afficher 30 frames par secondes.

Il nous faut deux variables supplémentaires :

1. Une valeur pour décider quand mettre à jour
2. Un compteur

Fluidité

```
FPS = 30
MAJ = 15
```

```
# ...
```

```
# juste avant la boucle infinie
compteur = 0
```

Fluidité

Dans la boucle infinie

```
if compteur == MAJ:
    compteur = 0
    head = [snake[0][0] + direction[0],
```

```

        snake[0][1] + direction[1]]
    # autres événements

    # On augmente le compteur
    # tout à la fin de la boucle infinie
    compteur += 1

```

Nourriture

On crée d'abord une nouvelle liste :

```
pomme = [8, 3]
```

Nourriture

On dessine la pomme comme la tête, mais en rouge

```

pygame.draw.rect(fenetre, ROUGE,
                  (pomme[0] * BLOC,
                   pomme[1] * BLOC,
                   BLOC, BLOC))

```

Nourriture

Puis on détecte la collision avec la pomme.

En cas de collision :

1. Le score augmente
2. Une nouvelle pomme est créée. La boucle `while` empêche la pomme d'apparaître sur le serpent

Nourriture

```

from random import randint
# ...

# Dans la boucle infinie
if snake[0] == pomme:
    score += 1
    while pomme in snake:
        pomme = [randint(0, LARGEUR / BLOC - 1),
                  randint(0, HAUTEUR / BLOC - 1)]

```

Nourriture

S'il n'y a pas de collision le serpent diminue, sinon il conserve sa taille

```
else:  
    snake.pop(-1)
```

Conclusion

Conclusion

C'est terminé...

Snake en 100 lignes (peu commentées) avec le minimum d'instructions. On peut faire beaucoup plus court mais c'est déjà très simple

Conclusion

- Python permet notamment de créer des jeux,
- Créer un jeu avec Pygame n'est pas difficile,
- Il nous faut quelques constantes, quelques éléments de jeu (serpent, tête)
- Une boucle infinie dans laquelle
 1. On lit les saisies de l'utilisateur
 2. On effectue les calculs (nouvelle tête, collisions etc.)
 3. On met à jour les éléments graphiques