

# Projet pygame et jeu du serpent

Cette séquence s'appuie sur :

- [https://pixees.fr/informatiquelycee/n\\_site/insi\\_term\\_projet\\_3.html](https://pixees.fr/informatiquelycee/n_site/insi_term_projet_3.html)

## 1 Introduction

Dans ce projet nous allons tenter de créer un jeu du serpent à l'aide de la bibliothèque Pygame et l'utilisation d'un des types abstraits vus en cours (pile, file, liste, ...).

Pour jouer au jeu du serpent en ligne : <https://www.snake.fr/>

## 2 Rappels sur pygame

### 2.1 Gestion de l'affichage

#### A faire vous même 1.

Saisissez, analysez et testez ce code

```
import pygame

surf = pygame.display.set_mode((800,600))
run = True
while run :
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    pygame.draw.rect(surf, (255,255,255), (10,20, 150,200))
    pygame.display.flip()
pygame.quit()
```

La ligne `pygame.draw.rect(...)` permet d'afficher un rectangle.

Toutes les options sont décrites là : <https://www.pygame.org/docs/ref/rect.html>

ATTENTION : le point de coordonnées (0,0) est en haut et à gauche de la fenêtre

La ligne `pygame.display.flip()` indique à Pygame qu'il faut afficher tout ce qui doit être affiché (cette ligne est nécessaire à partir du moment où vous cherchez à dessiner quelque chose)

Il est possible de créer des animations dans Pygame :

Nous allons déplacer notre cercle en modifiant ces coordonnées à chaque tour de boucle :

#### A faire vous même 2.

Saisissez, analysez et testez ce code

```
import pygame

LARG = 800
HAUT = 600

surf = pygame.display.set_mode((LARG,HAUT))
run = True
l, h = 20, 10
while run :
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    pygame.draw.rect(surf, (255,255,255), (l,h, 5,5))
    pygame.display.flip()
    l += 5
    h += 5
pygame.quit()
```

Comme vous pouvez le constater, nous avons un problème : les rectangles précédents restent affichés. Il faut donc effacer l'image précédente avant de pouvoir en afficher une nouvelle.

#### A faire vous même 3.

Saisissez, analysez et testez ce code

```
import pygame

LARG = 800
HAUT = 600

surf = pygame.display.set_mode((LARG,HAUT))
run = True
l, h = 20, 10
while run :
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
    surf.fill((0,0,0))
    pygame.draw.rect(surf, (255,255,255), (l,h, 5,5))
    pygame.display.flip()
    l += 5
    h += 5
pygame.quit()
```

La ligne `surf.fill((0,0,0))` permet d'effacer l'écran avant de réafficher le rectangle à une position différente.

Il est important de bien comprendre que même si vous n'avez pas d'animation à gérer, il est important d'effacer la surface avant d'afficher une nouvelle image.

Le système essaie d'exécuter le plus grand nombre de fois possible la "boucle de jeu". Sachant que plus votre microprocesseur est rapide et plus le nombre d'exécutions de la "boucle de jeu" par seconde sera grand, vous risquez d'avoir un jeu qui ne tournera pas du tout de la même façon sur 2 machines différentes. Pour éviter cet inconvénient, il est possible de limiter le nombre d'exécutions de la "boucle de jeu" par seconde :

#### A faire vous même 4.

Saisissez, analysez et testez ce code

```
import pygame

LARG = 800
HAUT = 600

surf = pygame.display.set_mode((LARG, HAUT))
run = True
l, h = 20, 10
clock=pygame.time.Clock()
while run :
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
        clock.tick(60)
    surf.fill((0,0,0))
    pygame.draw.rect(surf, (255,255,255), (l,h, 5,5))
    pygame.display.flip()
    l += 1
    h += 1
pygame.quit()
```

Nous avons ajouté 2 lignes afin de contrôler le nombre d'exécutions de la "boucle de jeu" par seconde :

- `clock=pygame.time.Clock()` permet de définir un système d'horloge
- `clock.tick(60)` permet de limiter le nombre d'exécutions de la "boucle de jeu" à 60 par seconde (nous aurons donc une fréquence d'affichage de 60 images par seconde)

Ce système de limitation est surtout important si vous avez à gérer des animations.

#### A faire vous même 5.

- Complétez le code ci-dessus pour que le jeu s'arrête si le rectangle touche un bord.
- Modifiez le programme pour que le rectangle se déplace à l'horizontal
- Modifiez le programme pour que le rectangle se déplace à la vertical

## 2.2 Gestion du clavier

Comme déjà dit plus haut, il est possible de gérer les événements "clavier" et "souris" (utilisation du clavier et de la souris par l'utilisateur). Nous allons ici uniquement nous intéresser aux événements "clavier". Si vous avez besoin d'utiliser les événements "souris" n'hésitez pas à consulter la documentation de pygame.

#### A faire vous même 6.

Saisissez, analysez et testez ce code, observez la console lorsque vous appuyez sur les touches "Entrée", "Espace" et "A" (vous pouvez aussi constater qu'il ne se passe rien quand)

```
import pygame

surf = pygame.display.set_mode((800,600))
run = True
while run :
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False
        if event.type == pygame.KEYDOWN :
            if event.key == pygame.K_SPACE :
                print ("vous avez appuyé sur la touche espace")
            elif event.key == pygame.K_a :
                print ("vous avez appuyé sur la touche A")
            elif event.key == pygame.K_RETURN :
                print ("vous avez appuyé sur la touche Entrée")
            else :
                print ("vous avez appuyé sur une touche")
    surf.fill((0,0,0))
    pygame.display.flip()
pygame.quit()
```

Vous trouverez la liste complète des touches dans la [documentation officielle de pygame](#).

#### A faire vous même 7.

- Reprenez votre script où un rectangle se déplace et adaptez ce que vous avez vu sur la gestion du clavier pour faire monter/descendre/aller à gauche/aller à droite votre rectangle. Si le rectangle touche un bord, Game Over.

# 3 Jeu du snake

## 3.1 Jeu de base

Vous devez obligatoirement vous appuyer sur un des types abstraits du chapitre (liste chaînée, pile, file, ...) afin d'implémenter le jeu du serpent.

1. Comme sur le jeu sur snake.fr, créez une fonction `dessine_damier` qui dessine un damier avec des carrés vert clair et vert foncé.  
C'est facultatif, mais cela pourrait être intéressant de définir 2 constantes (`LARG` et `HAUT`) qui serait la largeur et la hauteur d'un carré de base. Il est ensuite possible de définir une fenêtre qui soit un multiple de ces valeurs ( $20 * LARG$  et  $20 * HAUT$  par exemple). Et tout le reste peut découler de ces valeurs.
2. Créez une fonction `dessine_serpent` qui dessine un rectangle correspondant à un anneau
3. Exploitez ce que vous avez fait avant sur la gestion du clavier pour faire bouger ce rectangle
4. Versionnez ce script pour l'évaluation du projet
5. Créez une classe `Anneau` qui aura
  - \* 4 attributs (le lien avec l'anneau suivant, sa position en largeur, sa position en hauteur, sa future position en largeur, sa future position en hauteur)
  - \* 1 méthode qui basculera les futures positions en position
  - \* 1 méthode qui affichera un rectangle à la position de l'anneau
6. Sur le modèle liste chaînée créez une classe `Serpent` qui aura :
  - \* 1 attribut : le lien vers un anneau de tête
  - \* 1 méthode qui fera les mises à jour des positions de tous les anneaux
  - \* 1 méthode qui affichera tous les anneaux
7. Instanciez la classe `Serpent` et créez 5 anneaux liés les uns aux autres
8. Mettez à jour la fonction `dessine_serpent` afin qu'elle dessine un rectangle pour chaque anneau
9. Exploitez ce que vous avez fait avant sur la gestion du clavier pour faire bouger ce serpent
10. Versionnez ce script pour l'évaluation du projet
11. POUR LES PLUS RAPIDES : Dessinez une pomme et quand la tête du serpent rencontre la pomme, ajoutez-lui 5 anneaux supplémentaires
12. POUR LES HYPER RAPIDES : Ajoutez l'affichage d'un score
13. POUR LES HYPER HYPER RAPIDES : Gérez un 2<sup>e</sup> serpent pour faire des battles (celui qui fonce sur l'autre à perdu)
14. POUR LES HYPER HYPER HYPER RAPIDES : Gérez un protocole d'échange par le réseau pour pouvoir jouer en réseau

Term NSI GRILLE D'EVALUATION :	Rien d'écrit	Ne fonctionne pas	Fonctionne imparfaitement	Fonctionne avec beaucoup d'aide	Fonctionne avec aide	Fonctionne sans aide	Fonctionne de manière optimale
Noms :							
Note : /15							
Objectif 1							
Objectif 2-3-4							
Objectif 5-6-7-8							
Objectif 9-10							
Objectif 11-12							
Objectif 13-14 (bonus)							