



BO - HUB

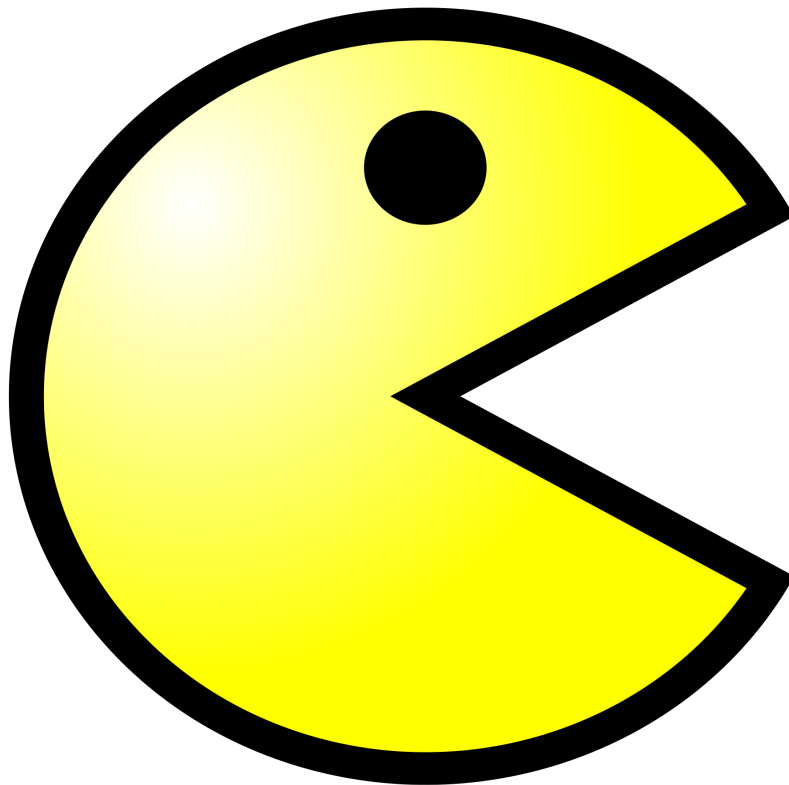
B-INN-000

Pyman

Python Game

PAC-MAN

Pac-Man est un des plus célèbres personnages du jeu vidéo. Créé en **1979** par l'entreprise japonaise Namco, le jeu mettait en scène ce fameux personnage en forme de camembert à l'intérieur d'un labyrinthe. Il devait alors récupérer toutes les pac-gommes qui s'y trouvaient sans être touché par les fantômes qui le hantaient !





PY-MAN

Aujourd'hui, vous allez créer un jeu type "Pac-man", c'est-à-dire, un jeu où vous devez récupérer des objets sans se faire toucher par un "monstre".

Pour cela, vous allez utiliser la librairie **pygame** de Python.

Pygame est une librairie python qui facilite le développement de jeux vidéos temps réel en intégrant une interface graphique.

PRE-REQUIS

Avant de commencer à programmer, vous devez installer Python si ce n'est pas déjà fait !

[Lien de téléchargement de Python](#)

Ensuite téléchargez la librairie **pygame** :

```
Terminal
~/B-INN-000> py -m pip install pygame
```

*Si la commande précédente **ne fonctionne pas**, veuillez contacter un Cobra.*

INTRODUCTION A PYTHON & PYGAME

Pour réaliser ce **Coding Club**, vous allez avoir besoin de la [Documentation de la librairie Pygame](#).



Faire vos propres **recherches** sur internet et **solliciter** un camarade est indispensable pour réaliser ce **Coding Club**.

Bien, nous pouvons commencer !

Tout d'abord vous allez créer un fichier `pyman.py`. Vous allez ensuite commencer à programmer votre jeu.

Pour cela, vous allez avoir besoin de la librairie **pygame**. Pour utiliser cette librairie, vous devez l'**importer** à votre programme:

```
import pygame
```



Maintenant que vous avez accès à cette librairie, nous allons plusieurs variables utiles:

```
title = "VotreNomDuJeu"
image_icone = "CheminVersImage"
window_size = (880, 880)

pygame.init()#On initialise pygame
window = pygame.display.set_mode(window_size)
pygame.display.set_caption(title)
icone = pygame.image.load(image_icone)
running = True
```

PREMIERE FENETRE.

Vous avez précédemment créer une variable appelée `window`. Cette variable va être notre fenetre de jeu. Cela dit, pour que votre jeu fonctionne vous devez utiliser une **boucle tant que**. Pour réussir à afficher votre fenetre de jeu, vous allez devoir remplir le code ci-dessous.

```
while ...:
    window.fill((0, 0, 0)) #on met la fenetre en noir.
    pygame.display.flip()
```

PREMIER EVENEMENT.

Lorsque vous jouez à un jeu vidéo, vous allez involontairement activer des **événements**. On appelle un **événement** une interaction entre l'utilisateur (Joueur) et le programme.

Notre premier événement sera la fermeture du jeu grâce à la croix de la fenêtre ainsi que la touche **ECHAP** (complétez le programme):

```
for event in pygame.event.get():
    if event.type == pygame.QUIT ... event.type == pygame.KEYDOWN ... event.key ==
        pygame.K_ESCAPE:
        running = ...
```

Nous avons finis avec les bases du programme. Nous reviendrons sur les événements un peu plus tard, vous allez maintenant créer une **map**.

LE SAVIEZ-VOUS ?

A l'origine, lorsqu'il est sorti sur borne d'arcade au Japon en 1979, le jeu (et le héros) s'appelait en fait Puck-Man, en référence à l'expression japonaise « paku paku » qui signifie s'empiffrer comme un glouton. Le succès est bien sûr énorme (sinon vous ne seriez pas en train de lire cette fiche) et le jeu est commercialisé en 1980 aux Etats-Unis, son aspect non-violent étant bien sûr un des principaux facteurs d'expansion du soft (car cela implique que le jeu était destiné à tous les publics). Toutefois, Namco décide de changer le nom du héros et du jeu en Pac-Man car ils avaient peur que des petits rigolos s'amuse à gratter le « P » de Puck-Man pour en faire un « F », ce qui n'aurait pas été de bon augure dans les salles d'arcade majoritairement fréquentées par les adolescents.





CREATION DE LA MAP

LES CLASSES EN PYTHON

Les classes sont un moyen de réunir des données et des fonctionnalités. Créer une nouvelle classe crée un nouveau type d'objet.

Ainsi nous allons créer une classe **Map**, afin de réunir les données de la map.

Pour cela nous allons créer un nouveau fichier appelé `map.py`.

Dans ce nouveau fichier, vous allez devoir utiliser le code à compléter.

```
class Map:
    def __init__(self, file):
        self.file = file
        self.structure = 0
        self.walls = []
```

Dans le programme principal, vous allez pouvoir **importer** la classe **Map**: `from map import Map`.

Vous allez également pouvoir créer une variable `map = Map("map.txt")`

CREATION

Revenons sur notre classe **Map**. Vous pouvez désormais ajouter une méthode à votre classe:

```
def create(self):
```

Cette méthode doit importer le contenu du fichier `map.txt` et le stocker dans un tableau à **2 dimensions**, c'est-à-dire une liste contenant une liste de caractères.

Exemple (ne pas copier):

```
map.structure = [
    ['W', 'W', 'W', 'W', '\n'],
    ['W', ' ', ' ', 'W', '\n'],
    ['W', ' ', ' ', 'W', '\n'],
    ['W', ' ', ' ', 'W', '\n'],
    ['W', 'W', 'W', 'W', '\n'],
]
```

AFFICHAGE

Toujours sur notre classe **Map**, vous allez ajouter une méthode "show" telle que:

```
def show(self, window):
    self.create()
    nb_line = ...
    for line in ...:
        nb_pos = ...
        for sprite in line:
            x = nb_pos * 30
            y = nb_line * 30
            wall = pygame.rect.Rect(x, y, 25, 25)
            if sprite == ...:
                self.walls.append(...)
                #fonction pour afficher un Rect
```



```
        nb_pos = nb_pos + 1
    nb_line = nb_line + 1
```

Complétez le programme ci-dessus.

Vous pouvez désormais appeler la méthode show dans la boucle de votre programme principal.

CREATION DU PERSONNAGE

CREATION ET AFFICHAGE

Vous allez devoir créer une classe Perso qui prend en argument la map, et une méthode **show**:

```
def __init__(self, map):
    self.rect = ... #Créer un rectangle
    self.map = map

def show(self, window):
    ... #Afficher le rectangle
```

Vous pouvez également appelé la méthode show dans votre programme principal.

DEPLACEMENTS ET COLLISIONS

Pour ce qui est des déplacements, vous allez devoir utiliser les **événements** pour détecter une touche (flèches). Ensuite, vous allez créer des fonctions afin de déplacer le joueur.

Pour les collisions c'est un peu plus compliqué. Vous allez devoir trouver un moyen pour détecter une collision entre deux rectangles.

A VOUS DE JOUER !

Vous avez bien avancé. C'est maintenant à votre tour d'améliorer votre jeu.

Vous pouvez:

- Créer les fantômes
- Créer les cookies (les boules qu'il mange)
- Créer un système de score
- Redimensionner la taille de la fenetre en fonction de la taille de la map.
- ... A vous d'imaginer !