

Pygame-1



Pygame est une bibliothèque Python permettant la réalisation simple de jeux interactifs.

Cette bibliothèque est principalement basée sur la **SDL** (Simple Directmedia Library) qui est une bibliothèque libre multiplateforme permettant la gestion du multimédia dans la programmation.

La SDL est écrite en C, mais est utilisable avec un grand nombre de langages, comme le C/C++, Pascal, Perl ou encore Python !

En résumé, cette bibliothèque nous servira principalement pour :

- L'affichage vidéo 2D
- La gestion de l'audio
- La gestion de périphériques de commandes (clavier, souris...)

Structure d'un jeu pygame

Les jeux Pygame fonctionnent quasiment tous sur la même structure :

Initialisation

Importation des bibliothèques nécessaires

Chargement des images et des sons utiles

Définition des variables du jeu

Boucle

Boucle principale du jeu (while)

Boucle d'événements (for)

gestion des contrôles (souris, clavier...)

Boucle d'animation

mouvement des images

Condition de fin de jeu (if)

condition d'arrêt de la boucle de jeu.

Importation de la bibliothèque

Après avoir installé Pygame, et avant de se lancer dans la rédaction de centaines de ligne de code, il va falloir **importer** cette librairie. Pour l'importer c'est assez simple, il suffit de faire comme pour n'importe quelle autre librairie, c'est-à-dire, ajouter en début de notre programme :

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

Ouverture d'une fenêtre

Avant toute chose, cette bibliothèque nécessite d'être **initialisée**. Pour cela, on utilise l'instruction suivante :

```
# initialisation des modules de pygame
pygame.init()
```

Une fois Pygame initialisé, nous allons demander la **création d'une fenêtre**. Pour faire ceci, il suffit d'entrer :

```
# création de la surface graphique
fenetre=pygame.display.set_mode((600,420))
```

600 et 420 définissent la taille de la fenêtre en pixels.

La fonction set_mode() est ensuite appelée avec un tuple en argument pour initialiser la fenêtre de jeu.

Affichage d'une image

Maintenant, essayons d'afficher des images, mettre une image de fond et un petit personnage.

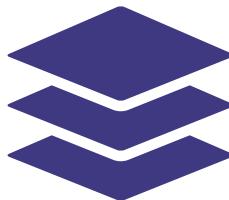
Pour commencer, il faut **pré-charger** les images au début du code, pour cela, la ligne de code à utiliser est :

```
# chargement des images
fond = pygame.image.load("fond.jpg").convert()
```

Ici, nous avons chargé l'image dans la variable « fond »

Le principe d'affichage de la SDL est à connaître pour bien afficher ses images : fenêtre est une surface vide, sur laquelle on va "coller", ou "empiler" les autres images.

Le fond doit donc être empilé sur la surface vide de la fenêtre. Cela s'apparente à un système de **calque**, donc attention à l'ordre d'appel, une image pourrait être cachée derrière une autre !



Pour mettre une image, ou « coller » une image sur la surface, on utilise la fonction **blit()** de la bibliothèque Pygame. Comme ceci :

```
# positionnement des images (calques)
fenetre.blit(fond, (0,0))
```

La fonction blit() prend comme première argument l'image pré-chargée que l'on souhaite coller, et en second argument, il va prendre les positions de l'endroit où l'on souhaite afficher notre image sous forme de tuple. Les coordonnées sont celles de l'angle en haut à gauche de l'image.

Jusqu'ici, si vous avez exécuté votre code, votre image n'est pas apparue, et c'est normal, car il faut rafraîchir, actualisé la surface pour faire apparaître notre « collage ». Pour ce faire il suffit de taper :

```
#affichage des images
pygame.display.flip()
```

Si vous voulez dessiner des formes géométriques avec pygame, jetez un coup d'œil à la documentation suivante :

<https://www.pygame.org/docs/ref/draw.html>

Voilà, maintenant vous savez afficher une image dans une fenêtre ! Mais une simple image ne constitue pas un jeu, il manque des animations, des contrôles et des interactions.