

Utiliser un joystick dans PyGame (bonus facultatif)

Un joystick (ou joypad, manette) peut être utilisé dans un jeu PyGame pour déplacer un personnage. Ce périphérique est souvent composé de sticks analogiques et de boutons. PyGame permet d'accéder à ces éléments via son module `joystick`.

Initialisation du joystick

Avant de lire les mouvements du joystick, il faut l'activer dans PyGame :

```
1 | pygame.init()
2 | pygame.joystick.init()
3 |
4 | # Activer le premier joystick détecté
5 | joystick = pygame.joystick.Joystick(0)
6 | joystick.init()
```

⚠ Ton joystick doit être branché **avant** de lancer le programme.

Lire les mouvements du stick

La majorité des joysticks ont **deux axes analogiques** (gauche/droite et haut/bas). On peut lire ces axes avec :

```
1 | x_axis = joystick.get_axis(0) # Axe horizontal
2 | y_axis = joystick.get_axis(1) # Axe vertical
```

Chaque axe renvoie une valeur entre **-1.0** (à gauche ou en haut) et **1.0** (à droite ou en bas). Le centre correspond à **0.0**.

Appliquer le mouvement au joueur

Voici comment déplacer un joueur avec un joystick, en tenant compte d'une vitesse :

```
1 | vitesse = 5
2 | player_x += x_axis * vitesse
3 | player_y += y_axis * vitesse
```

Le personnage se déplace plus ou moins vite selon l'inclinaison du stick. Cela donne un mouvement fluide et progressif.

Exemple de boucle de jeu avec joystick

```

1 | import pygame
2 |
3 | pygame.init()
4 | pygame.joystick.init()
5 | joystick = pygame.joystick.Joystick(0)
6 | joystick.init()
7 |
8 | screen = pygame.display.set_mode((800, 600))
9 | player = pygame.image.load("joueur.png")
10 | x, y = 400, 300
11 | vitesse = 5
12 | clock = pygame.time.Clock()
13 | running = True
14 |
15 | while running:
16 |     for event in pygame.event.get():
17 |         if event.type == pygame.QUIT:
18 |             running = False
19 |
20 |         x_axis = joystick.get_axis(0)
21 |         y_axis = joystick.get_axis(1)
22 |
23 |         x += x_axis * vitesse
24 |         y += y_axis * vitesse
25 |
26 |         screen.fill((0, 0, 0))
27 |         screen.blit(player, (x, y))
28 |         pygame.display.flip()
29 |         clock.tick(60)
30 |
31 | pygame.quit()

```

Bonnes pratiques

- **Test de zone morte** : comme le joystick n'est jamais parfaitement stable, on peut ignorer les petites valeurs proches de 0 (zone morte).

```

1 | if abs(x_axis) < 0.1:
2 |     x_axis = 0

```

- **Limiter la position** pour que le personnage ne sorte pas de l'écran (voir article précédent).
- Vérifie combien de joysticks sont branchés avec `pygame.joystick.get_count()` si tu veux gérer plusieurs manettes.

Avec cette base, tu peux facilement ajouter un support joystick dans ton jeu PyGame et permettre aux joueurs de se déplacer autrement qu'au clavier.