

Afficher des images en fond d'écran en PyGame

Un fond d'écran va rendre ton jeu plus joli. Utiliser des images donnera un meilleur résultat qu'une simple couleur de fond..

Utiliser un motif répétitif

Dans un jeu vidéo, il est parfois utile d'utiliser une image en motif (pattern) qui se répète pour couvrir tout l'écran. Voici comment procéder en PyGame.

Tu peux télécharger des images de fond dans la section téléchargement. Dans les exemples suivants, remplace le nom de fichier par celui dont tu as besoin.

1. Charger l'image du motif :

```
1 | background_tile = pygame.image.load('background.png')
```

2. Dessiner l'image en répétition :

```
1 | for x in range(0, largeur_ecran, background_tile.get_width()):  
2 |     for y in range(0, hauteur_ecran, background_tile.get_height()):  
3 |         screen.blit(background_tile, (x, y))
```

Le code fait exactement ça :

- Il commence tout en haut à gauche de l'écran.
- Il pose ton image carrée une fois.
- Ensuite, il avance vers la droite et pose à nouveau ton image, juste à côté de la première, sans laisser d'espace.
- Il fait ça jusqu'à remplir toute la ligne du haut.
- Quand il arrive au bout, il descend d'une ligne, et recommence depuis le début.
- Il fait ça ligne après ligne, jusqu'à ce que tout l'écran soit rempli.

Comme ça, tu obtiens un beau motif avec ton image, qui se répète partout sur l'écran !

Exemple complet

```
1 | import pygame  
2 | import sys  
3 |  
4 | # Initialisation  
5 | pygame.init()  
6 | largeur_ecran = 800  
7 | hauteur_ecran = 600
```

```

8
9 # Création de la fenêtre de jeu
10 screen = pygame.display.set_mode((largeur_ecran, hauteur_ecran))
11 clock = pygame.time.Clock()
12
13 # position initiale du joueur
14 player_x = 100
15 player_y = 100
16 vitesse = 5
17
18 # Chargement des images
19 player = pygame.image.load('player.png')
20 background_tile = pygame.image.load('sol_paves.png')
21
22
23 # Boucle de jeu
24 running = True
25 while running:
26     for event in pygame.event.get():
27         if event.type == pygame.QUIT:
28             running = False
29
30     # Mise à jour de l'écran
31     # Remplir l'écran avec le motif répété
32     for x in range(0, largeur_ecran, background_tile.get_width()):
33         for y in range(0, hauteur_ecran, background_tile.get_height()):
34             screen.blit(background_tile, (x, y))
35
36     screen.blit(player, (player_x, player_y))
37     pygame.display.flip()
38
39     # Limiter la vitesse d'exécution (60 FPS)
40     clock.tick(60)
41
42 pygame.quit()
43 sys.exit()

```

:: box-light 🍷 on ajouté les variables `largeur_ecran` et `hauteur_ecran`

::

Exemple avec déplacement du personnage

```

1 import pygame
2 import sys
3
4 # Initialisation
5 pygame.init()
6 largeur_ecran = 800
7 hauteur_ecran = 600
8
9 # Création de la fenêtre de jeu
10 screen = pygame.display.set_mode((largeur_ecran, hauteur_ecran))
11 clock = pygame.time.Clock()

```

```

12
13 # position initiale du joueur
14 player_x = 100
15 player_y = 100
16 vitesse = 5
17
18 # Chargement des images
19 player = pygame.image.load('player.png')
20 background_tile = pygame.image.load('sol_paves.png')
21
22
23 # Boucle de jeu
24 running = True
25 while running:
26     for event in pygame.event.get():
27         if event.type == pygame.QUIT:
28             running = False
29
30     keys = pygame.key.get_pressed()
31
32     if keys[pygame.K_LEFT] or keys[pygame.K_q]:
33         player_x -= vitesse
34     if keys[pygame.K_RIGHT] or keys[pygame.K_d]:
35         player_x += vitesse
36     if keys[pygame.K_UP] or keys[pygame.K_z]:
37         player_y -= vitesse
38     if keys[pygame.K_DOWN] or keys[pygame.K_s]:
39         player_y += vitesse
40
41     # Mise à jour de l'écran
42     # Remplir l'écran avec le motif répété
43     for x in range(0, largeur_ecran, background_tile.get_width()):
44         for y in range(0, hauteur_ecran, background_tile.get_height()):
45             screen.blit(background_tile, (x, y))
46
47     screen.blit(player, (player_x, player_y))
48     pygame.display.flip()
49
50     # Limiter la vitesse d'exécution (60 FPS)
51     clock.tick(60)
52
53 pygame.quit()
54 sys.exit()

```

Astuces

- Choisis une image qui s'assemble bien pour éviter les coupures visibles.
- Assure-toi que la taille du motif est optimisée pour éviter des répétitions trop évidentes.
- Pour un effet plus dynamique, ajoute un léger déplacement du motif à chaque frame pour simuler un défilement.

Avec cette technique, tu peux créer facilement des arrière-plans immersifs et fluides !

Bonus

Pour les plus courageux: essaye d'afficher un mur autours de la fenêtre.

Cela peut se faire avce 3 boucles:

1. Une pour le mur du haut
2. Une pour les murs verticaux
3. Une pour le mur du bas