

Pygame-4



Gestion des collisions

Il peut être parfois utile de savoir s'il y a « collision » de deux images, par exemple pour éviter que le joueur rentre dans un mur, ou encore pour faire perdre le joueur s'il touche un objet.

Pour cela on peut utiliser le test suivant : if position_perso.colliderect(rect_objet):

```
# Détection de collision
if vaisseau.colliderect(astrect) or vaisseau.colliderect(astrect2):
    fenetre.blit(fond, (0, 0))
    fenetre.blit(logo, vaisseau)
    fenetre.blit(asteroid, astrect)
    fenetre.blit(asteroid, astrect2)

    texte = font.render("GAME OVER", True, (255, 0, 0))
    fenetre.blit(texte, (350, 280))
    pygame.display.flip()

    pygame.time.delay(2000)
    pygame.quit()
    sys.exit("Le vaisseau a été détruit par un astéroïde !")
```

Ce code vérifie si le vaisseau touche l'un des deux astéroïdes, puis :
Affiche une dernière image avec un message "GAME OVER",
Attend 2 secondes,
puis ferme proprement le jeu.

```
# Détection de collision
if vaisseau.colliderect(astrect) or vaisseau.colliderect(astrect2):
```

vaisseau, astrect et astrect2 sont des rectangles Pygame (Rect), liés à des images (logo pour le vaisseau, asteroid pour les astéroïdes).

colliderect() vérifie si deux rectangles se touchent ou se superposent.

Le or signifie que si l'un ou l'autre des astéroïdes entre en collision avec le vaisseau, on déclenche la suite.

```
fenetre.blit(fond, (0, 0))
fenetre.blit(logo, vaisseau)
fenetre.blit(asteroid, astrect)
fenetre.blit(asteroid, astrect2)
```

blit(surface, position) : affiche une image (surface) à une position donnée sur la fenêtre.
fond est l'arrière-plan.

logo est le vaisseau.

asteroid est affiché deux fois, avec ses deux rectangles respectifs (astrect et astrect2).
Ce rendu est fait manuellement ici pour figer la scène au moment exact de la collision.

```
texte = font.render("GAME OVER", True, (255, 0, 0))
```

font.render(text, antialias, color) crée une surface de texte à partir d'une chaîne.
"GAME OVER" est le texte affiché.

True active l'anti-aliasing (bords plus lissés).

(255, 0, 0) est la couleur rouge (format RGB).

```
fenetre.blit(texte, (350, 280))
```

Le texte est affiché à la position (350, 280) .

`pygame.display.flip()`

`flip()` actualise tout l'écran avec les derniers éléments dessinés.
C'est à ce moment que le joueur voit le message.

`pygame.time.delay(2000)`

Fige le programme pendant 2000 millisecondes (soit 2 secondes).
Cela laisse le temps au joueur de lire le message "GAME OVER" avant de quitter le jeu.

`pygame.quit()`

Libère les ressources utilisées par Pygame (ferme la fenêtre, etc.)

```
sys.exit("Le vaisseau a été détruit par un astéroïde !")
```

`sys.exit()` termine le programme Python.
Le message fourni s'affiche dans la console (terminal), utile pour le débogage ou les logs.