

Détecter une collision simple (un joueur et un carré)

Commencer par la détection de collision avec **un seul carré** est parfait débuter.

Maintenant que tu sais déplacer un personnage dans PyGame, voyons comment détecter **si le joueur touche un objet** à l'écran. On commence avec **un seul objet** : un carré doré qui représente une pièce à ramasser.

Objectif

- Comprendre comment vérifier si **le joueur touche un objet**
- Utiliser un **rectangle invisible** pour détecter la collision entre le joueur (image) et le carré (rectangle)

Étape 1 : Créer un seul carré

On dessine un carré jaune à l'écran avec :

```
1 | carre_x = 300
2 | carre_y = 200
3 | taille_carre = 20
4 | couleur_carre = (255, 215, 0)
5 |
6 | # Pour détecter une collision, on crée un rectangle :
7 | carre_rect = pygame.Rect(carre_x, carre_y, taille_carre, taille_carre)
```

Ce rectangle représente la **zone** où la pièce est dessinée. Dans l'exemple du code précédent, on s'est contenté de dessiner un carré. Ici pour pouvoir détecter la collision avec le personnage, il faut enregistrer ce carré dans une variable pour effectuer nos tests plus tard.

Étape 2 : Créer le rectangle du joueur

Même si le joueur est une **image**, on doit créer un rectangle invisible autour de lui avec :

```
1 | player_rect = pygame.Rect(player_x, player_y, joueur_image.get_width(), joueur_image.get_height())
```

Ceci doit se faire dans la boucle. A chaque fois que le joueur a changé de position.

Étape 3 : Tester la collision

On utilise la méthode `.collidect()` pour savoir si les deux rectangles se touchent :

```
1 | if player_rect.colliderect(carre_rect):
2 |     print("Le joueur touche la pièce !")
```



Exemple complet avec un seul carré

Voici un petit jeu complet : un joueur (image) peut se déplacer et toucher un carré jaune. Quand c'est le cas, un message s'affiche dans la console.

```
1 | import pygame
2 | import sys
3 |
4 | # Initialisation
5 | pygame.init()
6 |
7 | # Fenêtre
8 | largeur_ecran = 800
9 | hauteur_ecran = 600
10 | screen = pygame.display.set_mode((largeur_ecran, hauteur_ecran))
11 | pygame.display.set_caption("Collision simple")
12 |
13 | # Joueur (image)
14 | joueur_image = pygame.image.load("joueur.png")
15 | player_x = 100
16 | player_y = 100
17 | vitesse = 5
18 |
19 | # Taille du joueur (automatique via l'image)
20 | player_largeur = joueur_image.get_width()
21 | player_hauteur = joueur_image.get_height()
22 |
23 | # Carré doré (pièce)
24 | carre_x = 300
25 | carre_y = 200
26 | taille_carre = 20
27 | couleur_carre = (255, 215, 0)
28 |
29 | # Créer le rectangle du carré
30 | carre_rect = pygame.Rect(carre_x, carre_y, taille_carre, taille_carre)
31 |
32 | # Boucle de jeu
33 | clock = pygame.time.Clock()
34 | running = True
35 | while running:
36 |     for event in pygame.event.get():
37 |         if event.type == pygame.QUIT:
38 |             running = False
39 |
40 |     # Gérer les touches
41 |     touches = pygame.key.get_pressed()
42 |     if touches[pygame.K_LEFT]:
43 |         player_x -= vitesse
44 |     if touches[pygame.K_RIGHT]:
```

```

45     player_x += vitesse
46     if touches[pygame.K_UP]:
47         player_y -= vitesse
48     if touches[pygame.K_DOWN]:
49         player_y += vitesse
50
51     # Créer le rectangle du joueur (à jour à chaque boucle)
52     player_rect = pygame.Rect(player_x, player_y, player_largeur, player_hauteur)
53
54     # Vérifier la collision
55     if player_rect.colliderect(carre_rect):
56         print("Le joueur touche la pièce !")
57
58     # Affichage
59     screen.fill((0, 0, 0)) # fond noir
60
61     # Dessiner le carré doré
62     pygame.draw.rect(screen, couleur_carre, carre_rect)
63
64     # Dessiner le joueur
65     screen.blit(joueur_image, (player_x, player_y))
66
67     pygame.display.flip()
68     clock.tick(60)
69
70     pygame.quit()
71     sys.exit()

```



À retenir

- Tu peux détecter une collision entre un **rectangle et une image** grâce à `pygame.Rect()`
- La méthode `.colliderect()` renvoie `True` si deux rectangles **se touchent**
- Cette logique fonctionne même si le joueur est une **image** !



Prochaine étape

Tu es prêt à passer à :

- plusieurs carrés à l'écran
- supprimer un carré touché
- compter les points