

Générer un carré d'or sur l'écran

Voici un article sur la **génération d'un carré d'or** à l'écran, première étape avant la détection de collisions. L'objectif est de te préparer à gérer des interactions simples entre objets dans PyGame.

Avant d'apprendre à détecter les collisions, il faut d'abord afficher des objets avec lesquels le joueur pourra interagir. Ici, on va afficher un **carré d'or** sous forme de **rectangle jaune**. Cela permet de se concentrer sur la logique sans devoir gérer des images.

Objectif

Afficher un carré d'or à l'écran à une positions fixe ou aléatoire.

Ce que tu dois déjà savoir faire

- ✓ Créer une fenêtre PyGame
- ✓ Afficher un personnage (même en rectangle)
- ✓ Utiliser des coordonnées (x, y)

Étape 1 : Un seul carré

On commence avec un carré de 20x20 pixels, couleur dorée $(255, 215, 0)$:

```
1 | taille_carre = 20
2 | carre_x = 300
3 | carre_y = 200
4 | couleur_carre = (255, 215, 0)
```

Dans la boucle de jeu, on dessine le carré :

```
1 | pygame.draw.rect(screen, couleur_carre, (carre_x, carre_y, taille_carre, taille_carre))
```

Étape 3 : Génération aléatoire

On peut aussi générer le carré à une position aléatoire :

```
1 | import random
2 |
```

```
3 | carre_x = random.randint(0, largeur_ecran - taille_carre) # largeur_ecran - largeur du carré
4 | carre_y = random.randint(0, hauteur_ecran - taille_carre) # hauteur_ecran - hauteur du carré
```

On laisse une marge pour que la carré ne sorte pas de l'écran.



Bonnes pratiques

- Utilise des `pygame.Rect` dès le départ : c'est pratique pour afficher ET détecter les collisions.
- Garde la même taille pour toutes les carrés pour simplifier les calculs.

Exemple complet

Voici le code complet qui te permet de générer un carré au hasard en reprenant le code utilisé dans les exemples précédents.

```
1 | import pygame
2 | import random
3 | import sys
4 |
5 | # Initialisation de PyGame
6 | pygame.init()
7 |
8 | # Définir la taille de la fenêtre
9 | largeur_ecran = 800
10 | hauteur_ecran = 600
11 | screen = pygame.display.set_mode((largeur_ecran, hauteur_ecran))
12 | pygame.display.set_caption("Pièces d'or")
13 |
14 | # Définir la couleur et la taille des carrés (pièces)
15 | couleur_carre = (255, 215, 0) # Doré
16 | taille_carre = 20
17 |
18 | # Génération du carré
19 | carre_x = random.randint(0, largeur_ecran - taille_carre) # largeur_ecran - largeur du carré
20 | carre_y = random.randint(0, hauteur_ecran - taille_carre) # hauteur_ecran - hauteur du carré
21 |
22 | # Boucle de jeu
23 | clock = pygame.time.Clock()
24 | running = True
25 | while running:
26 |     for event in pygame.event.get():
27 |         if event.type == pygame.QUIT:
28 |             running = False
29 |
30 |     # Effacer l'écran
31 |     screen.fill((0, 0, 0))
32 |
33 |     # Dessiner le carré
34 |     pygame.draw.rect(screen, couleur_carre, (carre_x, carre_y, taille_carre, taille_carre))
35 |
```

```
36 | # Mettre à jour l'affichage
37 |     pygame.display.flip()
38 |     clock.tick(60)
39 |
40 | # Quitter proprement
41 | pygame.quit()
42 | sys.exit()
```



La suite

Dans les prochains articles, tu apprendras à :

- vérifier si le **joueur touche le carré** (`collidect`)
 - **faire disparaître** le carré touché
 - **augmenter un score**
-