Générer plusieurs carrés d'or sur l'écran

Voici un article sur la **génération de plusieurs carrés d'or** à l'écran.

Tu vas d'abord afficher des objets avec lesquels le joueur pourra interagir. Ici, on va afficher des **carrés d'or** sous forme de **rectangles jaunes**. Cela permet de se concentrer sur la logique sans devoir gérer des images.

🎯 Objectif

Afficher plusieurs carrés d'or à l'écran à des positions fixes ou aléatoires, sous forme de rectangles.

🧠 Ce que tu dois déjà savoir faire

✓ Créer une fenêtre PyGame
✓ Afficher un personnage (même en rectangle)
✓ Utiliser des coordonnées `(x, y)` ✓ Afficher un carré à l'écran

🧩 Étape 2 : Plusieurs carrés à l'écran

On crée une **liste de rectangles**, chaque rectangle représentant un carré.

```
liste_carres = [
    pygame.Rect(100, 150, taille_carre, taille_carre),
    pygame.Rect(300, 220, taille_carre, taille_carre),
    pygame.Rect(500, 350, taille_carre, taille_carre)
]
```

Et dans la boucle de jeu, on les affiche tous :

```
for carre in liste_carres:
    pygame.draw.rect(screen, couleur_carre, carre)
```

Le fait de travailler avec une liste, que l'on verra plus tard, permet de travailler de façon systématique et automatique sur plusieurs objets. "En même temps".

🎲 Étape 3 : Génération aléatoire

On peut aussi générer des carrés à des positions aléatoires :

```
import random

```

```
liste_carres = []
for _ in range(5):  # 5 carrés aléatoires
    x = random.randint(0, largeur_ecran - taille_carre)  # largeur_ecran - largeur du carré pc
    y = random.randint(0, hauteur_ecran - taille_carre)  # hauteur_ecran - hauteur du carré pc
    rect_carre = pygame.Rect(x, y, taille_carre, taille_carre)
    liste_carres.append(rect_carre)
```

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.
- Commence avec 2-3 carrés fixes pour tester avant d'ajouter du hasard.

Exemple complet

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

```
import pygame
import random
import sys

# Initialisation de PyGame
pygame.init()

# Définir la taille de la fenêtre
largeur_ecran = 800
hauteur_ecran = 600
screen = pygame.display.set_mode((largeur_ecran, hauteur_ecran))
pygame.display.set_caption("Pièces d'or")

# Définir la couleur et la taille des carrés (pièces)
couleur_carre = (255, 215, 0)  # Doré
taille_carre = 20

# Génération aléatoire de 5 carrés
liste_carres = []
for _ in range(5):
    x = random.randint(0, largeur_ecran - taille_carre)
    y = random.randint(0, hauteur_ecran - taille_carre)
    rect_carre = pygame.Rect(x, y, taille_carre, taille_carre)
    liste_carres.append(rect_carre)

# Boucle de jeu
clock = pygame.time.Clock()
running = True
while running:
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
            running = False

    # Effacer l'écran
    screen.fill((0, 0, 0))

    # Dessiner tous les carrés de la liste
    for carre in liste_carres:
        pygame.draw.rect(screen, couleur_carre, carre)

    # Mettre à jour l'affichage
    pygame.display.flip()
    clock.tick(60)

# Quitter proprement
pygame.quit()
sys.exit()
```

La suite

Dans le prochain article, tu apprendras à :

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