```
import asyncio
import random
import pygame

W, H = 360, 640
BASKET_W, BASKET_H = 90, 22
BASKET_Y = H - 80
R = 16  # item size (radius)

pygame.init()
screen = pygame.display.set_mode((W, H))
pygame.display.set_caption('Catch the Apples')
clock = pygame.time.Clock()
font = pygame.font.Font(None, 34)
big = pygame.font.Font(None, 56)


def draw_item(kind, x, y):
    if kind == 'apple':
        pygame.draw.circle(screen, (220, 30, 30), (x, y), R)
        pygame.draw.line(screen, (30, 120, 40), (x, y - R), (x + 6, y - R - 8), 3)
    elif kind == 'gold':
        pygame.draw.circle(screen, (255, 200, 0), (x, y), R)
        pygame.draw.circle(screen, (255, 245, 150), (x - 5, y - 5), 4)
    else:  # bomb
        pygame.draw.circle(screen, (40, 40, 40), (x, y), R)
        pygame.draw.line(screen, (120, 80, 40), (x, y - R), (x + 6, y - R - 8), 3)
        pygame.draw.circle(screen, (255, 120, 0), (x + 6, y - R - 8), 4)


def say(text, f, color, x, y):
    img = f.render(text, True, color)
    screen.blit(img, img.get_rect(center=(x, y)))


async def main():
    best = 0
    state = {}

    def reset():
        state.update(basket=W / 2, items=[], timer=0.0,
                     score=0, lives=3, over=False)

    reset()
    running = True

    while running:
        dt = clock.tick(60) / 1000

        # ---------- touch / mouse / keyboard ----------
        for event in pygame.event.get():
```

```
    if event.type == pygame.QUIT:
        running = False
    elif event.type == pygame.MOUSEBUTTONDOWN:
        if state['over']:
            reset()
        state['basket'] = event.pos[0]
    elif event.type == pygame.MOUSEMOTION and event.buttons[0]:
        state['basket'] = event.pos[0]
    elif event.type in (pygame.FINGERDOWN, pygame.FINGERMOTION):
        state['basket'] = event.x * W

keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT] or keys[pygame.K_a]:
    state['basket'] -= 400 * dt
if keys[pygame.K_RIGHT] or keys[pygame.K_d]:
    state['basket'] += 400 * dt
state['basket'] = max(BASKET_W / 2, min(W - BASKET_W / 2, state['basket']))

# ---------- game logic ----------
if not state['over']:
    score = state['score']
    state['timer'] += dt
    if state['timer'] > max(0.35, 0.9 - 0.02 * score):
        state['timer'] = 0
        r = random.random()
        kind = 'bomb' if r < 0.2 else 'gold' if r < 0.3 else 'apple'
        state['items'].append(
            {'x': random.randint(R, W - R), 'y': -R, 'kind': kind})

    speed = min(200 + score * 6, 450)
    for item in state['items'][:]:
        item['y'] += speed * dt
        caught = (abs(item['x'] - state['basket']) < BASKET_W / 2 + R / 2
                  and BASKET_Y - R < item['y'] < BASKET_Y + BASKET_H)
        if caught:
            if item['kind'] == 'apple':
                state['score'] += 1
            elif item['kind'] == 'gold':
                state['score'] += 5
            else:
                state['lives'] -= 1      # bomb!
            state['items'].remove(item)
        elif item['y'] > H + R:
            if item['kind'] == 'apple':  # missed a red apple
                state['lives'] -= 1
            state['items'].remove(item)

    if state['lives'] <= 0:
        state['over'] = True
        state['items'] = []
        best = max(best, state['score'])

# ---------- drawing ----------
screen.fill((135, 206, 235))
for item in state['items']:
```

```
            draw_item(item['kind'], int(item['x']), int(item['y']))
        pygame.draw.rect(screen, (150, 75, 20),
                         (state['basket'] - BASKET_W / 2, BASKET_Y, BASKET_W, BASKET_H),
                         border_radius=8)
        say(f"Score: {state['score']}   Lives: {state['lives']}", font, (0, 0, 0), W // 2, 24)

        if state['over']:
            say('GAME OVER', big, (200, 0, 0), W // 2, H // 2 - 40)
            say(f"Score: {state['score']}   Best: {best}", font, (0, 0, 0), W // 2, H // 2 + 5)
            say('Tap to play again', font, (0, 0, 0), W // 2, H // 2 + 45)

        pygame.display.flip()
        await asyncio.sleep(0)  # needed so the game also runs in the browser

    pygame.quit()


asyncio.run(main())
```