Capstone Starter Pack

Paste-into-Trinket skeleton + a gallery of cartoon sprites you can drop straight into YOUR game.

How to use this pack

1. Open a brand-new pygame Trinket. 2. Delete the starter code. 3. Copy the skeleton on the next two pages into it. 4. It runs as-is - press SPACE to start, dodge red squares, press R when you die. 5. Replace the TODOs with your own colours, sprites, and rules.

Then flip to the Sprite Gallery (pages 4-5) and grab whichever cartoon character matches your theme. Paste it in, then call it from your draw loop instead of **draw_hero** or **draw_foe**.

The skeleton · part 1 of 2 (setup + helpers)

```
# ■■■ CAPSTONE STARTER · MY GAME (part 1 of 2) ■■■■■■■■■
# Replace "MY GAME" with your title. Fill in every TODO.

import pygame
import random

# 1. SCREEN
pygame.init()
WIDTH, HEIGHT = 800, 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("MY GAME")
clock = pygame.time.Clock()

# 2. COLOURS — TODO: pick your 3
BG    = (0, 0, 50)      # background
HERO  = (0, 230, 100)   # player colour
FOE   = (255, 0, 0)     # enemy colour
WHITE = (255, 255, 255)

# 3. SPRITE HELPERS — TODO: cartoonify draw_hero / draw_foe
def draw_hero(s, x, y):
    pygame.draw.rect(s, HERO, (x, y, 50, 50))

def draw_foe(s, e):
    pygame.draw.rect(s, FOE, e)

# 4. STATE + DATA
state  = "MENU"             # MENU / PLAYING / GAME_OVER
player = pygame.Rect(370, 500, 50, 50)
enemies = [pygame.Rect(random.randint(0, WIDTH-50),
                       random.randint(-300, 0), 50, 50)
           for _ in range(5)]
score, lives = 0, 3
font     = pygame.font.SysFont("Arial", 28)
big_font = pygame.font.SysFont("Arial", 64, bold=True)

def draw_text(text, x, y, color=WHITE, big=False):
    f = big_font if big else font
    screen.blit(f.render(text, True, color), (x, y))

def reset_game():
    global score, lives, state
    score, lives, state = 0, 3, "PLAYING"
    player.x = 370
    for e in enemies:
        e.y = random.randint(-300, -50)
        e.x = random.randint(0, WIDTH-50)

# ■■ the game loop continues on the next page ■■
```

The skeleton · part 2 of 2 (the game loop)

```
# ■■■ CAPSTONE STARTER · part 2 of 2 ■■■■■■■■■■■■■■■■■■■■

# 5. GAME LOOP
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: running = False
        if event.type == pygame.KEYDOWN:
            if state == "MENU" and event.key == pygame.K_SPACE:
                state = "PLAYING"
            if state == "GAME_OVER" and event.key == pygame.K_r:
                reset_game()

    keys = pygame.key.get_pressed()
    screen.fill(BG)

    if state == "MENU":
        draw_text("MY GAME", WIDTH//2 - 140, 220, color=HERO, big=True)
        draw_text("Press SPACE to start", WIDTH//2 - 130, 320)

    elif state == "PLAYING":
        if keys[pygame.K_LEFT]  and player.x > 0:          player.x -= 5
        if keys[pygame.K_RIGHT] and player.x < WIDTH - 50: player.x += 5

        for e in enemies:
            e.y += 3
            if e.y > HEIGHT:
                e.y = random.randint(-200, -50)
                e.x = random.randint(0, WIDTH-50)
                lives -= 1
                if lives <= 0: state = "GAME_OVER"
            if player.colliderect(e):
                state = "GAME_OVER"

        draw_hero(screen, player.x, player.y)
        for e in enemies: draw_foe(screen, e)
        draw_text("Score: " + str(score), 10, 10)
        draw_text("Lives: " + str(lives), WIDTH - 130, 10)

    elif state == "GAME_OVER":
        draw_text("GAME OVER", WIDTH//2 - 150, 220, color=FOE, big=True)
        draw_text("Press R to restart", WIDTH//2 - 110, 320)

    pygame.display.flip()
    clock.tick(60)

pygame.quit()
```

Sprite Gallery (paste & remix)

Drop any of these helpers into your skeleton (above the game loop). Then call them in the draw block - e.g. draw_turtle(screen, player.x, player.y). Recolour, resize, redesign - this is YOUR game.

Sprite 1 · Sea Turtle

```
# ■ SEA TURTLE — for Ocean Adventure
def draw_turtle(s, x, y):
    pygame.draw.ellipse(s, (40, 140, 80),  (x, y, 50, 38))           # shell
    pygame.draw.ellipse(s, (20, 90, 50),   (x + 5, y + 5, 40, 28))   # darker shell
    pygame.draw.circle (s, (40, 140, 80),  (x + 50, y + 19), 10)      # head
    pygame.draw.circle (s, (0, 0, 0),      (x + 53, y + 16), 2)       # eye
    pygame.draw.ellipse(s, (40, 140, 80),  (x - 4, y + 25, 14, 8))   # flipper
```

Sprite 2 · Dragon

```
# ■ DRAGON — for Dragon Quest
def draw_dragon(s, x, y):
    pygame.draw.ellipse(s, (180, 30, 50),  (x, y + 10, 60, 35))     # body
    pygame.draw.polygon(s, (180, 30, 50),
        [(x + 60, y + 15), (x + 80, y), (x + 80, y + 45)])           # head/snout
    pygame.draw.polygon(s, (255, 140, 0),
        [(x + 80, y + 20), (x + 95, y + 28), (x + 80, y + 30)])     # flame
    pygame.draw.circle (s, (255, 255, 0), (x + 75, y + 15), 3)       # eye
    pygame.draw.polygon(s, (180, 30, 50),
        [(x + 10, y + 10), (x + 25, y - 5), (x + 30, y + 12)])      # wing
```

Sprite 3 · Pizza Slice

```
# ■ PIZZA SLICE — for Pizza Panic
def draw_pizza(s, x, y):
    pygame.draw.polygon(s, (240, 200, 100),
        [(x, y), (x + 50, y), (x + 25, y + 45)])                    # crust + base
    pygame.draw.polygon(s, (210, 80, 50),
        [(x + 5, y + 3), (x + 45, y + 3), (x + 25, y + 38)])        # tomato
    pygame.draw.circle (s, (255, 255, 220), (x + 18, y + 12), 4)     # cheese blob
    pygame.draw.circle (s, (255, 255, 220), (x + 32, y + 14), 4)
    pygame.draw.circle (s, (160, 30, 30),   (x + 25, y + 22), 3)     # pepperoni
```

Sprite 4 · Football

```
# ■ FOOTBALL — for Penalty Pro
def draw_ball(s, x, y):
    pygame.draw.circle(s, (255, 255, 255), (x + 20, y + 20), 20)     # white
    # black pentagons (just dots for simplicity)
    pygame.draw.circle(s, (0, 0, 0), (x + 20, y + 20), 5)
    pygame.draw.circle(s, (0, 0, 0), (x + 10, y + 14), 3)
    pygame.draw.circle(s, (0, 0, 0), (x + 30, y + 14), 3)
    pygame.draw.circle(s, (0, 0, 0), (x + 10, y + 28), 3)
    pygame.draw.circle(s, (0, 0, 0), (x + 30, y + 28), 3)
```

Design your own

Every sprite above follows the same recipe: 3-5 stacked shapes (polygon / circle / ellipse), wrapped in a tiny def draw_X(s, x, y): function. To design your own:

1. Sketch your character on paper as 3-5 simple shapes.
2. Pick a colour for each shape.
3. Write def draw_yourname(s, x, y): and add one pygame.draw.* call per shape, positioning each shape relative to (x, y).
4. Call draw_yourname(screen, player.x, player.y) in your game loop.
5. Save your sprite somewhere - you'll reuse it in your next game!