import pygame

import random

import sys

# Initialize

pygame.init()

# Screen

WIDTH, HEIGHT = 800, 600

screen = pygame.display.set\_mode((WIDTH, HEIGHT))

pygame.display.set\_caption("Balloon Pop Game")

# Colors

WHITE = (255, 255, 255)

RED = (255, 50, 50)

BLUE = (50, 100, 255)

BLACK = (0, 0, 0)

GREEN = (0, 200, 0)

# Fonts

font = pygame.font.SysFont("Arial", 30)

large\_font = pygame.font.SysFont("Arial", 50)

# Clock

clock = pygame.time.Clock()

FPS = 60

# Shooter class

class Shooter:

    def \_\_init\_\_(self):

        self.width = 40

        self.height = 20

        self.x = WIDTH // 2

        self.y = HEIGHT - 20

        self.speed = 7

    def move(self, keys):

        if keys[pygame.K\_LEFT]:

            self.x -= self.speed

        if keys[pygame.K\_RIGHT]:

            self.x += self.speed

        self.x = max(0, min(WIDTH - self.width, self.x))

    def draw(self, surface):

        pygame.draw.rect(surface, BLACK, (self.x, self.y, self.width, self.height))

# Dart class

class Dart(pygame.sprite.Sprite):

    def \_\_init\_\_(self, x):

        super().\_\_init\_\_()

        self.image = pygame.Surface((5, 20))

        self.image.fill(BLUE)

        self.rect = self.image.get\_rect(midbottom=(x, HEIGHT - 20))

        self.speed = -10

    def update(self):

        self.rect.y += self.speed

        if self.rect.bottom < 0:

            self.kill()

# Balloon class

class Balloon(pygame.sprite.Sprite):

    def \_\_init\_\_(self, level):

        super().\_\_init\_\_()

        self.image = pygame.Surface((40, 60), pygame.SRCALPHA)

        pygame.draw.ellipse(self.image, RED, [0, 0, 40, 60])

        self.rect = self.image.get\_rect(

            center=(random.randint(50, WIDTH - 50), random.randint(50, 200))

        )

        self.speed = random.choice([-2, 2]) \* (1 + (level - 1) \* 0.5)

    def update(self):

        self.rect.x += self.speed

        if self.rect.left <= 0 or self.rect.right >= WIDTH:

            self.speed \*= -1

# Groups

dart\_group = pygame.sprite.Group()

balloon\_group = pygame.sprite.Group()

# Game variables

score = 0

level = 1

balloons\_popped = 0

shooter = Shooter()

level\_start\_time = pygame.time.get\_ticks()

def spawn\_balloons(level):

    for \_ in range(5 + level):

        balloon\_group.add(Balloon(level))

def show\_level\_summary(level, popped, score, success):

    screen.fill(WHITE)

    title = f"Level {level} {'Complete!' if success else 'Failed'}"

    msg = f"Balloons Popped: {popped} / {level \* 30}"

    score\_text = f"Score: {score}"

    screen.blit(large\_font.render(title, True, GREEN if success else RED), (WIDTH // 2 - 180, 150))

    screen.blit(font.render(msg, True, BLACK), (WIDTH // 2 - 140, 230))

    screen.blit(font.render(score\_text, True, BLACK), (WIDTH // 2 - 90, 270))

    screen.blit(font.render("Press any key to continue...", True, BLACK), (WIDTH // 2 - 170, 340))

    pygame.display.flip()

    wait\_for\_key()

def wait\_for\_key():

    while True:

        for event in pygame.event.get():

            if event.type == pygame.QUIT:

                pygame.quit()

                sys.exit()

            elif event.type == pygame.KEYDOWN:

                return

# Start

spawn\_balloons(level)

running = True

while running:

    clock.tick(FPS)

    screen.fill(WHITE)

    current\_time = pygame.time.get\_ticks()

    elapsed\_time = (current\_time - level\_start\_time) // 1000

    time\_limit = level \* 60

    time\_left = max(0, time\_limit - elapsed\_time)

    target\_pops = level \* 30

    keys = pygame.key.get\_pressed()

    shooter.move(keys)

    # Events

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            running = False

        if event.type == pygame.KEYDOWN and event.key == pygame.K\_SPACE:

            dart\_group.add(Dart(shooter.x + shooter.width // 2))

    # Update

    dart\_group.update()

    balloon\_group.update()

    # Collisions

    hits = pygame.sprite.groupcollide(dart\_group, balloon\_group, True, True)

    balloons\_popped += len(hits)

    score += len(hits) \* 10

    if len(balloon\_group) == 0:

        spawn\_balloons(level)

    # Draw

    dart\_group.draw(screen)

    balloon\_group.draw(screen)

    shooter.draw(screen)

    # HUD

    screen.blit(font.render(f"Level: {level}", True, BLACK), (10, 10))

    screen.blit(font.render(f"Score: {score}", True, BLACK), (10, 40))

    screen.blit(font.render(f"Time Left: {time\_left}s", True, BLACK), (10, 70))

    screen.blit(font.render(f"Popped: {balloons\_popped} / {target\_pops}", True, BLACK), (10, 100))

    # End of level

    if elapsed\_time >= time\_limit:

        level\_success = balloons\_popped >= target\_pops

        show\_level\_summary(level, balloons\_popped, score, level\_success)

        if level\_success:

            level += 1

            balloons\_popped = 0

            level\_start\_time = pygame.time.get\_ticks()

            balloon\_group.empty()

            dart\_group.empty()

            spawn\_balloons(level)

        else:

            # Retry same level

            balloons\_popped = 0

            level\_start\_time = pygame.time.get\_ticks()

            balloon\_group.empty()

            dart\_group.empty()

            spawn\_balloons(level)

    pygame.display.flip()

pygame.quit()

sys.exit()