import pygame

import sys

import random

# Initialize Pygame

pygame.init()

# Constants

WIDTH, HEIGHT = 800, 600

PLAYER\_SIZE = 40

GROUND\_COLOR = (0, 255, 0)

FPS = 60

# Create the game window

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

pygame.display.set\_caption("Platformer with Score")

# Player attributes

player\_image = pygame.image.load("/home/pi/Downloads/Police.png") # Replace with your player image

player\_image = pygame.transform.scale(player\_image, (PLAYER\_SIZE, PLAYER\_SIZE))

player\_x = WIDTH // 2

player\_y = HEIGHT // 2

player\_vel\_x = 0

player\_vel\_y = 0

jumping = False

# Enemy attributes

enemy\_images = []

for i in range(1, 5):

enemy\_image = pygame.image.load("/home/pi/Downloads/Spyware.png") # Replace with your enemy images

enemy\_image = pygame.transform.scale(enemy\_image, (PLAYER\_SIZE, PLAYER\_SIZE))

enemy\_images.append(enemy\_image)

enemies = []

enemy\_vel\_x = -3

# Additional enemy attributes

enemy\_attributes = [

{"images": enemy\_images, "velocity": -3},

{"images": enemy\_images, "velocity": -2},

{"images": enemy\_images, "velocity": -4},

{"images": enemy\_images, "velocity": -5},

]

# Score

score = 0

# Game state

game\_over = False

well\_done = False

# Restart button

restart\_button\_rect = pygame.Rect(WIDTH // 2 - 100, HEIGHT // 2 + 60, 200, 40)

# Max number of enemies

MAX\_ENEMIES = 2

# Game loop

clock = pygame.time.Clock()

def restart\_game():

global player\_x, player\_y, player\_vel\_x, player\_vel\_y, jumping, enemies, score, game\_over, well\_done

player\_x = WIDTH // 2

player\_y = HEIGHT // 2

player\_vel\_x = 0

player\_vel\_y = 0

jumping = False

enemies = []

score = 0

game\_over = False

well\_done = False

while not well\_done:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if not game\_over:

keys = pygame.key.get\_pressed()

# Player movement

if keys[pygame.K\_LEFT]:

player\_vel\_x = -5

elif keys[pygame.K\_RIGHT]:

player\_vel\_x = 5

else:

player\_vel\_x = 0

# Jumping

if not jumping:

if keys[pygame.K\_SPACE]:

player\_vel\_y = -10

jumping = True

# Apply gravity

player\_vel\_y += 0.5

# Update player position

player\_x += player\_vel\_x

player\_y += player\_vel\_y

# Collision with ground (player)

if player\_y >= HEIGHT - PLAYER\_SIZE:

player\_y = HEIGHT - PLAYER\_SIZE

player\_vel\_y = 0

jumping = False

# Spawn enemies (limited to MAX\_ENEMIES)

if len(enemies) < MAX\_ENEMIES and random.random() < 0.02:

attributes = random.choice(enemy\_attributes)

enemy\_image = random.choice(attributes["images"])

enemy\_velocity = attributes["velocity"]

enemy = {"x": WIDTH, "y": HEIGHT - PLAYER\_SIZE, "image": enemy\_image, "velocity": enemy\_velocity}

enemies.append(enemy)

# Update enemy positions and check for collisions

for enemy in enemies:

enemy["x"] += enemy["velocity"]

if (

player\_x < enemy["x"] + PLAYER\_SIZE

and player\_x + PLAYER\_SIZE > enemy["x"]

and player\_y < enemy["y"] + PLAYER\_SIZE

and player\_y + PLAYER\_SIZE > enemy["y"]

):

game\_over = True

# Collision with ground (enemies)

if enemy["y"] >= HEIGHT - PLAYER\_SIZE:

enemy["y"] = HEIGHT - PLAYER\_SIZE

# Remove off-screen enemies

enemies = [enemy for enemy in enemies if enemy["x"] > -PLAYER\_SIZE]

# Increment the score

# Well done screen

if score > 100:

game\_over = True

well\_done = True

# Fill the screen

screen.fill((0, 0, 0))

# Draw the player

screen.blit(player\_image, (player\_x, player\_y))

# Draw the enemies

for enemy in enemies:

screen.blit(enemy["image"], (enemy["x"], enemy["y"]))

# Draw the ground

pygame.draw.rect(screen, GROUND\_COLOR, (0, HEIGHT - 20, WIDTH, 20))

# Display score

font = pygame.font.Font(None, 36)

text = font.render(f"Score: {score}", True, (255, 255, 255))

screen.blit(text, (10, 10))

pygame.display.update()

clock.tick(FPS)

if well\_done:

screen.fill((0, 0, 0))

font = pygame.font.Font(None, 72)

text = font.render("Well Done!", True, (0, 255, 0))

screen.blit(text, (WIDTH // 2 - 120, HEIGHT // 2 - 36))

# Restart button

pygame.draw.rect(screen, (0, 255, 0), restart\_button\_rect)

font = pygame.font.Font(None, 36)

text = font.render("Restart", True, (0, 0, 0))

screen.blit(text, (WIDTH // 2 - 50, HEIGHT // 2 + 60))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x, y = event.pos

if restart\_button\_rect.collidepoint(x, y):

restart\_game()