Dodge the Enemies - Game Code with Explanation

import pygame  
import random  
import sys

Import necessary modules. `pygame` is used for the game, `random` for random enemy positions, and `sys` to exit cleanly.

# Initialize Pygame  
pygame.init()

Initialize the Pygame library.

# Screen dimensions  
WIDTH, HEIGHT = 600, 400  
screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
pygame.display.set\_caption("Dodge the Enemies")

Set up the game window size and caption.

# Colors  
WHITE = (255, 255, 255)  
BLUE = (0, 100, 255)  
RED = (255, 0, 0)  
BLACK = (0, 0, 0)

Define RGB values for colors used in the game.

# Clock  
clock = pygame.time.Clock()  
FPS = 60

Create a clock to control game speed and set frames per second.

# Player setup  
player\_width = 60  
player\_height = 15  
player\_x = WIDTH // 2 - player\_width // 2  
player\_y = HEIGHT - 40  
player\_speed = 5

Set the size, starting position, and movement speed of the player.

# Enemy setup  
enemy\_width = 30  
enemy\_height = 30  
enemy\_x = random.randint(0, WIDTH - enemy\_width)  
enemy\_y = -enemy\_height  
enemy\_speed = 5

Set size and initial position of the enemy block, with a random x-axis start.

# Score  
score = 0  
font = pygame.font.SysFont("comicsans", 24)

Initialize the score counter and set the font for displaying the score.

# Game loop  
running = True  
while running:  
 screen.fill(WHITE)

Main game loop begins and clears the screen each frame.

# Handle events  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 running = False

Handle quit event to close the game window.

# Move player with mouse  
 mouse\_x = pygame.mouse.get\_pos()[0]  
 player\_x = mouse\_x - player\_width // 2

Update the player's position based on the current mouse x-coordinate.

# Keep player in bounds  
 if player\_x < 0:  
 player\_x = 0  
 elif player\_x > WIDTH - player\_width:  
 player\_x = WIDTH - player\_width

Prevent the player from moving outside the window boundaries.

# Move enemy  
 enemy\_y += enemy\_speed

Move the enemy downward each frame.

# Reset enemy and increase score  
 if enemy\_y > HEIGHT:  
 enemy\_y = -enemy\_height  
 enemy\_x = random.randint(0, WIDTH - enemy\_width)  
 score += 1

If the enemy moves off-screen, reset its position and increase the score.

# Collision detection  
 player\_rect = pygame.Rect(player\_x, player\_y, player\_width, player\_height)  
 enemy\_rect = pygame.Rect(enemy\_x, enemy\_y, enemy\_width, enemy\_height)  
  
 if player\_rect.colliderect(enemy\_rect):  
 print("Game Over! Final Score:", score)  
 running = False

Detect collision between player and enemy. If collision occurs, end the game.

# Draw player and enemy  
 pygame.draw.rect(screen, BLUE, player\_rect)  
 pygame.draw.rect(screen, RED, enemy\_rect)

Render the player and enemy blocks on the screen.

# Draw score  
 score\_text = font.render(f"Score: {score}", True, BLACK)  
 screen.blit(score\_text, (10, 10))

Display the current score at the top-left corner.

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

Update the screen and control the frame rate.

pygame.quit()  
sys.exit()

Cleanly exit the game and close the Pygame window.