Car Racing Game (Pygame) – Code Documentation

Overview

This document explains a simple **Car Racing Game** built using **Python and Pygame**. The player controls a car that moves left and right to avoid enemy cars. Each enemy avoided increases the score. A collision ends the game.

Requirements

- Python 3.x
- Pygame library

Install Pygame using:

```
pip install pygame
```

Modules Used

- **pygame**: For game development (graphics, events, sounds)
- **random**: To generate random enemy positions
- **sys**: To exit the game safely

Game Configuration

- **Screen Size**: 400 × 600 pixels
- **Frame Rate**: 60 FPS
- **Player Speed**: 5
- **Enemy Speed**: 6

Game Components

1. Player Car

- Represented using a black rectangle
- Moves left and right using arrow keys
- Positioned near the bottom of the screen

2. Enemy Car

- Represented using a red rectangle
- Moves from top to bottom
- Respawns at a random x-position after leaving the screen

3. Score System

- Score increases when the enemy car successfully passes the player
- Displayed at the top-left corner

4. Collision Detection

- Detects overlap between player and enemy cars
- Triggers game over when collision occurs

5. Game Over Screen

- Displays "GAME OVER" text
- Waits for 2 seconds before closing the game

Source Code

```
import pygame
import random
import sys

# Initialize pygame
pygame.init()

# Screen size
WIDTH = 400
HEIGHT = 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Car Racing Game")

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

# Clock
clock = pygame.time.Clock()
speed = 5

# Fonts
```

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

# Player car
player_width = 50
player_height = 80
player_x = WIDTH // 2 - player_width // 2
player_y = HEIGHT - 100

# Enemy car
enemy_width = 50
enemy_height = 80
enemy_x = random.randint(0, WIDTH - enemy_width)
enemy_y = -100
enemy_speed = 6

# Score
score = 0

# Draw score
def show_score(value):
    text = font.render(f"Score: {value}", True, BLACK)
    screen.blit(text, (10, 10))

# Game Over
def game_over():
    text = font.render("GAME OVER", True, RED)
    screen.blit(text, (100, 250))
    pygame.display.update()
    pygame.time.delay(2000)
    pygame.quit()
    sys.exit()

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

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()

    # Controls
    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT] and player_x > 0:
        player_x -= speed
    if keys[pygame.K_RIGHT] and player_x < WIDTH - player_width:
        player_x += speed
```

```
    # Enemy movement
    enemy_y += enemy_speed
    if enemy_y > HEIGHT:
        enemy_y = -enemy_height
        enemy_x = random.randint(0, WIDTH - enemy_width)
        score += 1

    # Collision detection
    if (player_x < enemy_x + enemy_width and
        player_x + player_width > enemy_x and
        player_y < enemy_y + enemy_height and
        player_y + player_height > enemy_y):
        game_over()

    # Draw cars
    pygame.draw.rect(screen, BLACK, (player_x, player_y, player_width,
player_height))
    pygame.draw.rect(screen, RED, (enemy_x, enemy_y, enemy_width, enemy_height))

    show_score(score)

    pygame.display.update()
    clock.tick(60)
```

Conclusion

This mini project is ideal for beginners learning **Pygame**, game loops, collision detection, and basic game mechanics. You can enhance it further by adding: - Background images - Sound effects - Multiple enemy cars - Levels and increasing difficulty

---

Author: Praveen Kumar