

Dodge the Falling Blocks - Code Explanation

1. Introduction

This document explains the code for the game 'Dodge the Falling Blocks', which is built using the Pygame library. The goal of the game is to control a blue block that moves left and right to avoid falling red blocks. The player earns points by surviving longer. If a collision occurs, the game ends.

2. Setup and Initialization

1. ****Import Libraries****: The ``pygame`` and ``random`` libraries are imported. Pygame is used for graphics, and random is used to generate falling block positions.
2. ****Initialize Pygame****: ``pygame.init()`` initializes all Pygame modules.
3. ****Screen Setup****: The screen size is set to 800x600 pixels, and the game window title is set to 'Dodge the Falling Blocks!'.
4. ****Colors****: Colors are defined using RGB values for white, red, blue, and black.
5. ****Player Settings****: The player is a blue block, and its initial position and speed are set.
6. ****Block Settings****: Blocks are red and fall from the top of the screen at a constant speed.
7. ****Font and Clock****: A font is set to display the score, and a clock is created to control the frame rate.

3. Functions

- **1. draw_player(x, y)****: Draws a blue block (the player) at position (x, y).

Dodge the Falling Blocks - Code Explanation

****2. draw_block(x, y)**:** Draws a red block (the obstacle) at position (x, y).

****3. display_score(score)**:** Displays the current score at the top of the screen.

****4. check_collision(player_x, player_y, block_x, block_y)**:** Checks if the player collides with a falling block.

If the player's position overlaps with a block's position, the game ends.

4. Game Loop

The ``game_loop()`` function is the core of the game. It runs continuously until the player loses. Here's what happens in each frame:

1. ****Event Handling**:** The game listens for events, like key presses and window close requests.
2. ****Player Movement**:** The player moves left or right when the corresponding arrow key is pressed.
3. ****Block Generation**:** Randomly generates blocks at the top of the screen.
4. ****Block Movement**:** Moves each block down the screen. If it moves off-screen, it's removed, and the player's score increases.
5. ****Collision Detection**:** If the player touches a block, the game ends.
6. ****Drawing Elements**:** The player, blocks, and score are drawn on the screen in each frame.

Dodge the Falling Blocks - Code Explanation

7. **Frame Rate**: The game runs at 60 frames per second (FPS).

5. Full Code Walkthrough

```
import pygame
```

```
import random
```

```
# Initialize pygame
```

```
pygame.init()
```

```
# Set up screen dimensions
```

```
screen_width = 800
```

```
screen_height = 600
```

```
screen = pygame.display.set_mode((screen_width, screen_height))
```

```
# Colors
```

```
WHITE = (255, 255, 255)
```

```
RED = (255, 0, 0)
```

```
BLUE = (0, 0, 255)
```

```
BLACK = (0, 0, 0)
```

```
# Player settings
```

```
player_width = 50
```

Dodge the Falling Blocks - Code Explanation

```
player_height = 50

player_x = screen_width // 2 - player_width // 2

player_y = screen_height - player_height - 10

player_speed = 5


# Block settings

block_width = 50

block_height = 50

block_speed = 10

block_frequency = 40


# Clock to control frame rate

clock = pygame.time.Clock()


# Functions for player, block, and collision

def draw_player(x, y):

    pygame.draw.rect(screen, BLUE, [x, y, player_width, player_height])


def draw_block(x, y):

    pygame.draw.rect(screen, RED, [x, y, block_width, block_height])


def check_collision(player_x, player_y, block_x, block_y):

    if (player_x < block_x + block_width and

        player_x + player_width > block_x and
```

Dodge the Falling Blocks - Code Explanation

```
player_y < block_y + block_height and  
player_y + player_height > block_y):  
  
    return True  
  
    return False
```

```
def game_loop():  
  
    player_x = screen_width // 2 - player_width // 2  
  
    score = 0  
  
    blocks = []  
  
    game_over = False  
  
    while not game_over:  
  
        for event in pygame.event.get():  
  
            if event.type == pygame.QUIT:  
  
                game_over = True  
  
        keys = pygame.key.get_pressed()  
  
        if keys[pygame.K_LEFT] and player_x > 0:  
  
            player_x -= player_speed  
  
        if keys[pygame.K_RIGHT] and player_x < screen_width - player_width:  
  
            player_x += player_speed  
  
        if random.randint(1, block_frequency) == 1:  
  
            block_x = random.randint(0, screen_width - block_width)  
  
            blocks.append([block_x, 0])  
  
        for block in blocks[:]:  
  
            block[1] += block_speed
```

Dodge the Falling Blocks - Code Explanation

```
if block[1] > screen_height:

    blocks.remove(block)

if check_collision(player_x, player_y, block[0], block[1]):

    game_over = True

draw_block(block[0], block[1])

draw_player(player_x, player_y)

pygame.display.update()

clock.tick(60)

pygame.quit()

game_loop()
```

6. Conclusion

Congratulations! You've learned how the 'Dodge the Falling Blocks' game is created from start to finish. You now understand key concepts like Pygame initialization, drawing objects, detecting collisions, and managing game loops. You can enhance this game by adding sounds, better graphics, or increasing difficulty as the score increases.