Tic Tac Toe Game in Pygame

RAJKUMAR
UID:CS24NY339750

Constants

- PLAYER_X: Represents Player X ('X')
- PLAYER_O: Represents Player O ('O')
- EMPTY: Represents an empty cell on the board (' ')
- GRID_SIZE: Size of the Tic Tac Toe grid (3x3)
- CELL_SIZE: Size of each cell in the grid (100 pixels)
- SCREEN_WIDTH and SCREEN_HEIGHT: Dimensions of the game window (300x300 pixels)

- WHITE, LINE_COLOR, CIRCLE_COLOR, CROSS_COLOR, BUTTON_COLOR, BUTTON_HOVER_COLOR, TEXT_COLOR: Color constants for UI elements

Functions

Initialize Board

initialize_board()

- Returns: A 3x3 grid with all cells initialized to EMPTY

Check Winner

check_winner(board)

- Parameters: board (3x3 grid)
- Returns: The winner of the game ('X', 'O', or 'Draw') or None if the game is not over

Draw Grid

draw_grid()

- Draws the 3x3 grid on the screen

Draw Move (X/O)

draw_move(row, col, player)

- Parameters: row, col (cell coordinates), player ('X' or 'O')
- Draws the player's symbol (X or O) at the specified cell

Handle Player Move

handle_move(row, col, board, current_player)

- Parameters: row, col (cell coordinates), board (3x3 grid), current_player ('X' or 'O')
- Returns: True if the move is valid and made, False otherwise

AI Move (Easy/Hard)

random_move(board) (Easy)

best_move(board) (Hard)

- Parameters: board (3x3 grid)
- Returns: The AI's chosen move (Easy: random, Hard: using Minimax algorithm)

Display Difficulty Screen

display_difficulty_screen()

- Displays the difficulty selection screen
- Returns: The chosen difficulty level ('Easy' or 'Hard')

Display End Screen

display_end_screen(winner)

- Parameters: winner ('X', 'O', or 'Draw')
- Displays the game over screen with the winner and options to restart or exit

Game Loop

game_loop(difficulty)

- Parameters: difficulty ('Easy' or 'Hard')
- Runs the main game loop until the game is over

Main Execution

1. Display the difficulty selection screen using display_difficulty_screen().
2. Run the game loop with the chosen difficulty using game_loop(difficulty).

Code Documentation

```
# Import necessary libraries
import pygame
import random
import sys

# Constants
PLAYER_X = 'X'  # Represents Player X
```

```
PLAYER_O = 'O'  # Represents Player O
EMPTY = '  # Represents an empty cell on the board
GRID_SIZE = 3  # Size of the Tic Tac Toe grid (3x3)
CELL_SIZE = 100  # Size of each cell in the grid (100 pixels)
SCREEN_WIDTH = CELL_SIZE * GRID_SIZE  # Dimensions of the game window
(300x300 pixels)
SCREEN_HEIGHT = CELL_SIZE * GRID_SIZE
WHITE = (255, 255, 255)  # Color constants for UI elements
LINE_COLOR = (0, 0, 0)
CIRCLE_COLOR = (242, 85, 96)
CROSS_COLOR = (28, 170, 156)
BUTTON_COLOR = (28, 170, 156)
BUTTON_HOVER_COLOR = (28, 128, 120)
TEXT_COLOR = (255, 255, 255)

# Initialize Pygame
pygame.init()
screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Tic Tac Toe")
font = pygame.font.Font(None, 36)

### Functions ###

def initialize_board():
    """Returns a 3x3 grid with all cells initialized to EMPTY"""
    return [[EMPTY for _ in range(GRID_SIZE)] for _ in
range(GRID_SIZE)]

def check_winner(board):
    """Checks for a winner in the game"""
    #... (function implementation)

def draw_grid():
    """Draws the 3x3 grid on the screen"""
    #... (function implementation)

def draw_move(row, col, player):
    """Draws the player's symbol (X or O) at the specified cell"""
    #... (function implementation)
```

```
def handle_move(row, col, board, current_player):
    """Handles the player's move"""
    #... (function implementation)

def random_move(board):
    """Returns the AI's random move (Easy difficulty)"""
    #... (function implementation)

def best_move(board):
    """Returns the AI's best move using Minimax algorithm (Hard
difficulty)"""
    #... (function implementation)

def display_difficulty_screen():
    """Displays the difficulty selection screen"""
    #... (function implementation)

def display_end_screen(winner):
    """Displays the game over screen with the winner and options to
restart or exit"""
    #... (function implementation)

def game_loop(difficulty):
    """Runs the main game loop until the game is over"""
    #... (function implementation)

### Main Execution ###

def main():
    difficulty = display_difficulty_screen()
    game_loop(difficulty)

if __name__ == "__main__":
    main()
```

Thank You for Using Our Tic Tac Toe Game Documentation!