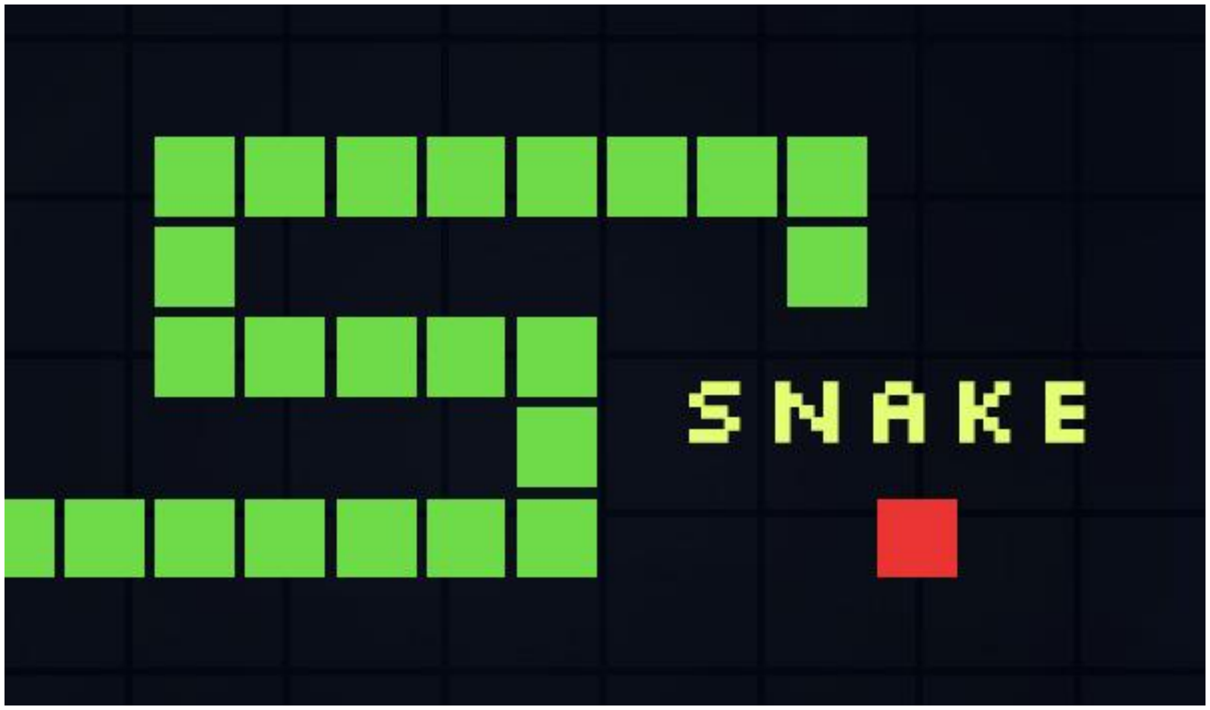

PYTWO Project

David Durães Valadares

12/09/2024

Contents

Snake Game Documentation

1. Overview

This is a simple Snake Game implemented using Python and the Pygame library. The game involves controlling a snake that grows longer as it consumes food. In addition to food, poison blocks appear randomly on the screen, which the snake must avoid. If the snake collides with a poison block or its own body, the game ends.

2. External Libraries Used

1. **Pygame**

- Purpose: Pygame is used for creating video games in Python. It provides functionality for drawing graphics, handling user input, and controlling game time. It simplifies tasks like handling key presses, drawing to the screen, and managing game loops.

- **Key Features Used:**
    - **Rendering:** Drawing the snake, food, poison blocks, and background.
    - **Input Handling:** Capturing key events (arrow keys) to control the snake's movement.
    - **Timing:** Controlling the game's frame rate (using pygame.time.Clock()).
    - **Fonts:** Displaying the score and messages using Pygame's font system.
- **Installation:**
  To install Pygame, run the following command in your terminal or command prompt:

```
pip install pygame
```

2. **Random**

- Purpose: The random module is used for generating random numbers. In this game, it is primarily used to:
    - Spawn food and poison blocks at random positions on the screen.
    - Randomize the intervals at which poison blocks spawn.
    - Randomize the lifespan of poison blocks.
- **Installation:**
  Random is part of Python's standard library, so no installation is required.

3. **Time**

- **Purpose:** The time module is used to manage time-based operations in the game, such as:
    - Tracking the lifespan of poison blocks (using time.time()).
    - Controlling the intervals at which poison blocks spawn.
- **Installation:**
  time is also part of Python's standard library, so no installation is required.

3. Key Variables and Settings

- **Screen Dimensions**:
  - width = 640 pixels
  - height = 480 pixels
- **Colors**:
  - **black**: Snake color (0, 0, 0)
  - **white**: Background color (255, 255, 255)
  - **red**: Poison block color (200, 0, 0)
  - **green**: Food color (34, 139, 34)
  - **dark_gray**: Game screen background color (60, 60, 60)
- **Fonts**:
  - **font_style**: Used to display messages and game-over texts.
  - **score_font**: Used to display the player's score.
- **Game Settings**:
  - snake_block: Size of each block in the snake (20 pixels).
  - start_speed: Initial speed of the snake (10).
  - speed_increase: Speed increment for each food item consumed (0.3).
  - max_speed: Maximum speed the game can reach (20).
- **Poison Settings**:
  - poison_blocks: List storing the coordinates of active poison blocks.
  - poison_block_timer: Dictionary holding the expiration time for each poison block.
  - poison_interval_range: Random interval range for spawning poison blocks (5, 20).
  - poison_lifespan_range: Lifespan of each poison block in seconds (10, 30).

4. Core Functions

your_score(score)

- **Purpose**: Displays the current score at the top of the screen.
- **Parameters**:
  - score: The current score to display.
- **Implementation**: Uses the score_font to render the score and places it at the top-left corner of the screen.

our_snake(snake_block, snake_list)

- **Purpose**: Draws the snake on the screen.
- **Parameters**:
  - snake_block: The size of each snake block.
  - snake_list: A list of coordinates representing the snake's body.
- **Implementation**: Iterates through snake_list and draws each segment using pygame.draw.rect().

message(msg, color, y_offset=0)

- **Purpose**: Displays a message at the center of the screen.
- **Parameters**:
  - msg: The message to display.
  - color: The color of the message text.
  - y_offset: Vertical offset for the message (default is 0).
- **Implementation**: Renders the message using the font_style and adjusts the position to center it horizontally and vertically (with an optional offset).

generate_food_or_poison()

- **Purpose**: Generates a random position for food or poison blocks.
- **Implementation**: Randomly generates x and y coordinates within the screen bounds and aligns them to the snake block size.

spawn_poison_block()

- **Purpose**: Spawns a poison block at a random position on the screen.
- **Implementation**: Adds a new poison block to the poison_blocks list and sets its expiration time using time.time().

game_loop()

- **Purpose**: The main game loop that controls the flow of the game.
- **Implementation**:
  - Initializes the game state and handles player input (keyboard).
  - Manages the snake's movement, collision detection, and screen updates.
  - Checks for collisions with the snake's body, food, and poison blocks.
  - Increases the snake's length when it eats food and speeds up the game.
  - Spawns poison blocks at random intervals and handles their removal when expired.
  - Ends the game if the snake collides with a poison block or itself.
  - Displays the game-over screen with options to restart or quit.

Main Game Loop:

The game continually runs the game_loop() function. After each game ends, the player is asked if they wish to restart. The game quits when the player chooses to exit.

5. Algorithms and Logic

Movement and Collision Detection:

- The snake moves in a direction based on keyboard input (LEFT, RIGHT, UP, DOWN).
- The screen wraps around when the snake moves off one side, reappearing on the opposite side.
- The snake's head is compared with every segment of its body to check for self-collision.
- The snake also checks for collision with poison blocks. If it collides with one, the game ends.

Poison Block Logic:

- Poison blocks are spawned at random intervals and persist for a random lifespan between 10 and 30 seconds.
- Each poison block is drawn on the screen, and expired blocks are removed from the list.
- The game checks if the snake's head collides with any active poison block, leading to game-over if true.

Speed and Growth:

- Every time the snake eats food, its length increases, and the game speed slightly increases.
- The game has a maximum speed limit that prevents the snake from becoming too fast.

6. Game Flow

1. **Starting Screen**: Displays a prompt for the player to press any arrow key to start.
2. **Game Loop**:
    - The snake moves based on keyboard input.
    - Food and poison blocks appear randomly on the screen.
    - The snake grows and the game speed increases as food is consumed.
    - The player loses if the snake collides with its body or a poison block.
3. **Game Over Screen**: Displays a "Game Over" message with options to restart (R) or quit (Q).
4. **Restart/Exit**: The game loop either restarts or exits based on player input.