Snake Game (Python Tkinter)

Project Overview

This is a classic arcade-style Snake game built using Python and the tkinter graphical user interface (GUI) library. The player controls a snake that grows in length as it consumes food, while avoiding collisions with the walls or its own body.

Features

- **Real-time Gameplay:** Smooth movement and controls using Python's event handling.
- **Score Tracking:** Displays the current score in real-time.
- **Randomized Food:** Food spawns at random coordinates within the grid.
- **Collision Detection:** accurate detection for wall impacts and self-collisions.
- **Game Over Screen:** distinct end-game state when the player loses.

Prerequisites

To run this project, you need:

- **Python 3.x** installed on your system.
- **Tkinter** (Usually included with standard Python installations).

How to Run

1. Ensure you have the snake_game.py file saved in a directory.
2. Open your terminal or command prompt.
3. Navigate to the directory containing the file.
4. Run the following command:
   python snake_game.py

Controls

- **Up Arrow:** Move Up
- **Down Arrow:** Move Down
- **Left Arrow:** Move Left
- **Right Arrow:** Move Right

Configuration (Customization)

You can easily tweak the game settings by modifying the constants at the top of the snake_game.py file:

Constant	Description	Default Value
GAME_WIDTH	Width of the game window in pixels	700
GAME_HEIGHT	Height of the game window in pixels	600
SPEED	Game speed (lower is faster)	100
SPACE_SIZE	Size of the grid cells/snake body parts	50
BODY_PARTS	Initial length of the snake	3
SNAKE_COLOR	Hex color code for the snake	#0BE70B (Green)
FOOD_COLOR	Hex color code for the food	#FF0000 (Red)
BACKGROUND_COLOR	Hex color code for the background	#000000 (Black)

Code Structure

Classes

1. **Snake**:
    - Initializes the snake's body parts and starting coordinates.
    - Manages the graphical squares representing the snake on the canvas.
2. **Food**:
    - Generates a random coordinate for food spawn.
    - Ensures food spawns within the grid boundaries.
    - Renders the food object (circle) on the canvas.

Key Functions

- **next_turn(snake, food)**: The main game loop. It calculates the new coordinates based on direction, handles movement, checks if food is eaten, and calls itself recursively using window.after.
- **change_direction(new_direction)**: Handles user input to change the snake's direction, ensuring the snake cannot immediately reverse into itself (e.g., cannot go DOWN if currently going UP).

- **check_collisions(snake)**: Verifies if the head of the snake has hit the window boundaries or any part of its own body.
- **game_over()**: Clears the canvas and displays the "Game Over" text.