

Hungry Worm Game Code Documentation

Overview

This documentation explains the JavaScript code used to create "Hungry Worm," a simple browser-based snake game.

Game Code

Initialization

The game starts when the DOM content is fully loaded. Key elements, such as the snake (**snakeElement**) and food (**foodElement**), are retrieved from the DOM. Initial configurations, including the snake's starting position, movement direction, and the initial food position, are set.

Game Logic

Constants

Snake: Snake initially appears near the center of the canvas and moves on its own

Direction: Snake is initially facing right. Will change directions with user inputs.

Food: Food appears in random positions on the canvas

The game logic is organized into functions:

- **getRandomPosition():** Generates a random position for the food element.
- **updateGame():** Calls functions to move the snake, check for collisions, check for food consumption, and render the game.
- **moveSnake():** Updates the snake's position based on the current direction and handles food consumption.
- **checkCollision():** Checks for collisions with walls or itself and resets the game if a collision is detected.
- **checkSelfCollision():** Checks if the snake collides with itself.
- **checkFood():** Checks if the snake consumes food and updates the snake and food accordingly.
- **render():** Renders the snake and food elements based on their current positions.

Event Listeners

The game responds to user input using arrow keys for controlling the snake's direction.

Game Loop

The game loop is implemented using **setInterval()** to repeatedly call the **updateGame()** function at a specified interval, controlling the game's speed. To change the speed of the snake, change the integer value of **setInterval()**.

How to Play

The snake will start moving automatically. Use the arrow keys (up, down, left, right) to change the snake's direction. The objective is to consume the food items that appear on the screen. The game continues until the snake collides with the walls or itself. Upon collision, the game will prompt the user to play again.