**SNAKE GAME**

****

AUTHOR: MRA.HASEN AL BANNA

***HTML FILE***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Snake Game</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="game-board"></div>

<button id="retry-button" onclick="startGame()">Retry</button>

<script src="script.js"></script>

</body>

</html>

***CSS FILE***

body {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

height: 100vh;

margin: 0;

background-color: #000;

}

#game-board {

width: 400px;

height: 400px;

background-color: #111;

position: relative;

}

.snake {

width: 20px;

height: 20px;

background-color: #0f0;

position: absolute;

}

.food {

width: 20px;

height: 20px;

background-color: #f00;

position: absolute;

}

#retry-button {

margin-top: 20px;

padding: 10px 20px;

font-size: 16px;

cursor: pointer;

background-color: #fff;

border: none;

border-radius: 5px;

display: none; /\* Hidden by default \*/

}

**JAVA SCRIPT**

const board = document.getElementById('game-board');

const retryButton = document.getElementById('retry-button');

const boardSize = 400;

const tileSize = 20;

let snake = [{ x: 100, y: 100 }];

let direction = { x: 0, y: 0 };

let food = { x: 200, y: 200 };

let gameOver = false;

document.addEventListener('keydown', changeDirection);

function changeDirection(event) {

const key = event.keyCode;

if (gameOver) return;

switch(key) {

case 37: // left arrow

if (direction.x === 0) direction = { x: -tileSize, y: 0 };

break;

case 38: // up arrow

if (direction.y === 0) direction = { x: 0, y: -tileSize };

break;

case 39: // right arrow

if (direction.x === 0) direction = { x: tileSize, y: 0 };

break;

case 40: // down arrow

if (direction.y === 0) direction = { x: 0, y: tileSize };

break;

}

}

function moveSnake() {

const head = { x: snake[0].x + direction.x, y: snake[0].y + direction.y };

snake.unshift(head);

if (head.x === food.x && head.y === food.y) {

generateFood();

} else {

snake.pop();

}

}

function drawSnake() {

board.innerHTML = '';

snake.forEach(segment => {

const snakeElement = document.createElement('div');

snakeElement.style.left = `${segment.x}px`;

snakeElement.style.top = `${segment.y}px`;

snakeElement.classList.add('snake');

board.appendChild(snakeElement);

});

}

function generateFood() {

food = {

x: Math.floor(Math.random() \* (boardSize / tileSize)) \* tileSize,

y: Math.floor(Math.random() \* (boardSize / tileSize)) \* tileSize

};

drawFood();

}

function drawFood() {

const foodElement = document.createElement('div');

foodElement.style.left = `${food.x}px`;

foodElement.style.top = `${food.y}px`;

foodElement.classList.add('food');

board.appendChild(foodElement);

}

function checkCollision() {

const head = snake[0];

if (head.x < 0 || head.y < 0 || head.x >= boardSize || head.y >= boardSize) {

return true;

}

for (let i = 1; i < snake.length; i++) {

if (snake[i].x === head.x && snake[i].y === head.y) {

return true;

}

}

return false;

}

function startGame() {

snake = [{ x: 100, y: 100 }];

direction = { x: 0, y: 0 };

gameOver = false;

retryButton.style.display = 'none';

generateFood();

gameLoop();

}

function gameLoop() {

if (gameOver) return;

moveSnake();

drawSnake();

drawFood();

if (checkCollision()) {

gameOver = true;

retryButton.style.display = 'block'; // Show retry button

alert('Game Over!');

} else {

setTimeout(gameLoop, 100);

}

}

// Start the game initially

startGame();