**SNAKE GAME**

<!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>

    <style>

        body {

            display: flex;

            flex-direction: column;

            align-items: center;

            background-color: #f0f0f0;

            font-family: Arial, sans-serif;

        }

        h1 {

            margin-bottom: 20px;

        }

        canvas {

            border: 1px solid #000;

            background-color: #fff;

        }

    </style>

</head>

<body>

    <h1>Snake Game</h1>

    <canvas id="gameCanvas" width="400" height="400"></canvas>

    <script>

        const canvas = document.getElementById('gameCanvas');

        const ctx = canvas.getContext('2d');

        const box = 20;

        let snake = [{ x: 9 \* box, y: 9 \* box }];

        let direction = 'RIGHT';

        let food = spawnFood();

        let score = 0;

        document.addEventListener('keydown', changeDirection);

        function spawnFood() {

            return {

                x: Math.floor(Math.random() \* (canvas.width / box)) \* box,

                y: Math.floor(Math.random() \* (canvas.height / box)) \* box

            };

        }

        function changeDirection(event) {

            if (event.key === 'ArrowLeft' && direction !== 'RIGHT') {

                direction = 'LEFT';

            } else if (event.key === 'ArrowUp' && direction !== 'DOWN') {

                direction = 'UP';

            } else if (event.key === 'ArrowRight' && direction !== 'LEFT') {

                direction = 'RIGHT';

            } else if (event.key === 'ArrowDown' && direction !== 'UP') {

                direction = 'DOWN';

            }

        }

        function collision(head) {

            return (

                head.x < 0 || head.x >= canvas.width ||

                head.y < 0 || head.y >= canvas.height ||

                snake.some(segment => segment.x === head.x && segment.y === head.y)

            );

        }

        function draw() {

            ctx.clearRect(0, 0, canvas.width, canvas.height);

            for (let i = 0; i < snake.length; i++) {

                ctx.fillStyle = i === 0 ? 'green' : 'lightgreen';

                ctx.fillRect(snake[i].x, snake[i].y, box, box);

                ctx.strokeStyle = 'darkgreen';

                ctx.strokeRect(snake[i].x, snake[i].y, box, box);

            }

            ctx.fillStyle = 'red';

            ctx.fillRect(food.x, food.y, box, box);

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

            if (direction === 'LEFT') head.x -= box;

            if (direction === 'UP') head.y -= box;

            if (direction === 'RIGHT') head.x += box;

            if (direction === 'DOWN') head.y += box;

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

                score++;

                food = spawnFood();

            } else {

                snake.pop();

            }

            if (collision(head)) {

                clearInterval(game);

                alert('Game Over! Your score: ' + score);

                return;

            }

            snake.unshift(head);

        }

        const game = setInterval(draw, 100);

    </script>

</body>

</html>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated