<!DOCTYPE html>

<html lang=”em”>

<head>

<meta charset=”UTF-8”>

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

<title>Meu Jogo</title>

<link rel=”stylesheet” href=”style.css”>

</head>

<body>

<h1>Bem-vindo ao Meu Jogo!</h1>

<div id=”gameArea”>

<!—O jogo será inserido aqui 🡪

</div>

<script src=”script.js”></script>

</body>

</html>

Body {

Font-family: Arial, sans-serif;

Text-align: center;

}

#gameArea {

Margin: 20px auto;

Width: 600px;

Height: 400px;

Border: 2px solid black;

}

Let canvas = document.createElement(“canvas”);

Let ctx = canvas.getContext(“2d”);

Document.getElementById(“gameArea”).appendChild(canvas);

Canvas.width = 600;

Canvas.height = 400;

Let snake = [{ x: 10, y: 10 }];

Let direction = { x: 1, y: 0 };

Let food = { x: 15, y: 15 };

Let score = 0;

Function gameLoop() {

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

// Desenha a cobra

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

Ctx.fillStyle = “green”;

Ctx.fillRect(snake[i].x \* 20, snake[i].y \* 20, 18, 18);

}

// Move a cobra

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

Snake.unshift(head);

// Checa se a cobra comeu a comida

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

Food = { x: Math.floor(Math.random() \* 30), y: Math.floor(Math.random() \* 20) };

Score++;

} else {

Snake.pop();

}

// Desenha a comida

Ctx.fillStyle = “red”;

Ctx.fillRect(food.x \* 20, food.y \* 20, 18, 18);

// Verifica colisão com bordas ou consigo mesma

If (head.x < 0 || head.x >= 30 || head.y < 0 || head.y >= 20 || snake.slice(1).some(seg => seg.x === head.x && seg.y === head.y)) {

Alert(“Game Over! Score: “ + score);

Snake = [{ x: 10, y: 10 }];

Direction = { x: 1, y: 0 };

Score = 0;

}

}

Function changeDirection(event) {

If (event.keyCode === 37 && direction.x !== 1) { // Esquerda

Direction = { x: -1, y: 0 };

} else if (event.keyCode === 38 && direction.y !== 1) { // Cima

Direction = { x: 0, y: -1 };

} else if (event.keyCode === 39 && direction.x !== -1) { // Direita

Direction = { x: 1, y: 0 };

} else if (event.keyCode === 40 && direction.y !== -1) { // Baixo

Direction = { x: 0, y: 1 };

}

}

Document.addEventListener(“keydown”, changeDirection);

setInterval(gameLoop, 100);

<!DOCTYPE html>

<html lang=”pt-br”>

<head>

<meta charset=”UTF-8”>

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

<title>Caça aos Círculos</title>

<link rel=”stylesheet” href=”style.css”>

</head>

<body>

<h1>Jogo: Caça aos Círculos</h1>

<div id=”gameArea”>

<!—O jogo será inserido aqui 🡪

</div>

<div id=”score”>Pontuação: 0</div>

<button id=”startButton”>Iniciar Jogo</button>

<script src=”script.js”></script>

</body>

</html>

Body {

Font-family: Arial, sans-serif;

Text-align: center;

}

#gameArea {

Margin: 20px auto;

Width: 600px;

Height: 400px;

Border: 2px solid black;

Position: relative;

Background-color: lightblue;

}

.circle {

Width: 50px;

Height: 50px;

Border-radius: 50%;

Background-color: red;

Position: absolute;

Cursor: pointer;

}

#score {

Font-size: 20px;

Margin-top: 20px;

}

#startButton {

Padding: 10px 20px;

Font-size: 16px;

Margin-top: 10px;

Cursor: pointer;

}

Let gameArea = document.getElementById(‘gameArea’);

Let scoreDisplay = document.getElementById(‘score’);

Let startButton = document.getElementById(‘startButton’);

Let score = 0;

Let gameInterval;

Let circleInterval;

Function startGame() {

// Reseta o jogo

Score = 0;

updateScore();

// Inicia o intervalo que cria círculos

circleInterval = setInterval(createCircle, 1000);

// Desabilita o botão de início enquanto o jogo roda

startButton.disabled = true;

// O jogo dura 30 segundos

gameInterval = setTimeout(endGame, 30000);

}

Function createCircle() {

// Cria um novo círculo

Let circle = document.createElement(‘div’);

Circle.classList.add(‘circle’);

// Define posição aleatória dentro da área de jogo

Let x = Math.random() \* (gameArea.clientWidth – 50);

Let y = Math.random() \* (gameArea.clientHeight – 50);

Circle.style.left = `${x}px`;

Circle.style.top = `${y}px`;

// Adiciona evento de clique ao círculo

Circle.addEventListener(‘click’, () => {

Score++;

updateScore();

gameArea.removeChild(circle);

});

// Remove o círculo após 1,5 segundos se não for clicado

setTimeout(() => {

if (gameArea.contains(circle)) {

gameArea.removeChild(circle);

}

}, 1500);

// Adiciona o círculo à área de jogo

gameArea.appendChild(circle);

}

Function updateScore() {

scoreDisplay.innerText = `Pontuação: ${score}`;

}

Function endGame() {

// Para de criar círculos

clearInterval(circleInterval);

// Habilita o botão de início novamente

startButton.disabled = false;

alert(`Fim de Jogo! Sua pontuação final foi: ${score}`);

}

// Inicia o jogo quando o botão for clicado

startButton.addEventListener(‘click’, startGame);