<!DOCTYPE html>

<html lang="en">

<head>

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

<meta charset="UTF-8">

<title>NPS - Net Promoter Snake</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: Arial, sans-serif;

background-color: #1a2a3b;

color: white;

display: flex;

flex-direction: row;

justify-content: center;

align-items: flex-start;

height: 100vh;

}

header {

display: flex;

justify-content: space-between;

align-items: center;

width: 100%;

background-color: #112233;

padding: 10px;

position: fixed;

top: 0;

}

header img {

height: 40px;

}

header h1 {

flex: 1;

text-align: center;

font-size: 24px;

color: #00b2cc;

}

.score {

margin: 20px 0;

font-size: 20px;

color: #00b2cc;

text-align: center;

}

.score-history {

position: fixed;

top: 80px;

left: 20px;

padding: 10px;

background-color: #112233;

border-radius: 10px;

color: #00b2cc;

font-size: 16px;

max-height: 80%;

overflow-y: auto;

}

#gameContainer {

display: flex;

flex-direction: column;

align-items: center;

margin-top: 100px; /\* Space for the fixed header \*/

}

canvas {

background-color: #0a1a2a;

display: block;

border: 4px solid #00b2cc;

position: relative;

}

#startButton {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

padding: 10px 20px;

font-size: 18px;

cursor: pointer;

background-color: #00b2cc;

border: none;

color: white;

border-radius: 5px;

display: block;

}

.game-over-message {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

background-color: rgba(0, 0, 0, 0.8);

padding: 20px;

border-radius: 10px;

color: white;

font-size: 24px;

text-align: center;

display: none;

}

.game-over-message button {

margin-top: 20px;

padding: 10px;

font-size: 16px;

cursor: pointer;

background-color: #00b2cc;

border: none;

color: white;

border-radius: 5px;

}

.controls {

display: flex;

justify-content: center;

margin-top: 10px;

}

.control-button {

width: 40px;

height: 40px;

margin: 5px;

background-color: #00b2cc;

border: none;

border-radius: 5px;

color: white;

font-size: 18px;

cursor: pointer;

}

</style>

</head>

<body>

<!-- Histórico de Pontuações -->

<div class="score-history">

<h3>Histórico de Pontuações:</h3>

<ul id="scoreList"></ul>

</div>

<div id="gameContainer">

<header>

<img src="Logo CX Day.png" alt="Logo CX Day"> <!-- Certifique-se de que este arquivo está na mesma pasta do HTML -->

<h1>NPS - Net Promoter Snake</h1>

</header>

<!-- Texto de contagem de pontos em tempo real -->

<div class="score">Qtd de Detratores convertidos em Promotores: <span id="currentPoints">0</span></div>

<!-- Canvas do jogo e botão de iniciar -->

<div style="position: relative;">

<canvas id="gameCanvas" width="300" height="300"></canvas>

<button id="startButton" onclick="startGame()">Iniciar jogo</button>

</div>

<!-- Mensagem de fim de jogo -->

<div id="gameOverMessage" class="game-over-message">

Fim de Jogo!<br>Você converteu <span id="points">0</span> Detratores em Promotores.

<button onclick="resetGame()">Jogar novamente</button>

</div>

<!-- Botões de controle para dispositivos móveis -->

<div class="controls">

<button class="control-button" id="up">▲</button>

</div>

<div class="controls">

<button class="control-button" id="left">◄</button>

<button class="control-button" id="down">▼</button>

<button class="control-button" id="right">►</button>

</div>

</div>

<script>

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

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

const gameOverMessage = document.getElementById("gameOverMessage");

const pointsElement = document.getElementById("points");

const currentPointsElement = document.getElementById("currentPoints");

const startButton = document.getElementById("startButton");

const scoreList = document.getElementById("scoreList");

const gridSize = 15;

const tileCount = canvas.width / gridSize;

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

let food = { x: 5, y: 5 };

let dx = 1;

let dy = 0;

let speed = 100; // Velocidade aumentada

let points = 0;

let isGameOver = false;

let attempt = 1;

function startGame() {

// Esconde o botão "Iniciar jogo" e começa o loop do jogo

startButton.style.display = "none";

gameLoop();

}

function gameLoop() {

if (!isGameOver) {

update();

draw();

setTimeout(gameLoop, speed);

}

}

function update() {

// Move the snake

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

snake.unshift(head);

// Check if snake eats food

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

points++;

updateScore();

placeFood();

} else {

snake.pop(); // Remove tail if no food eaten

}

// Check collision with walls or itself

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

endGame();

}

}

function updateScore() {

// Atualiza o número de pontos na tela

currentPointsElement.innerText = points;

}

function draw() {

// Clear the canvas

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

// Draw the snake

snake.forEach((part, index) => {

if (index === 0) {

// Draw the head of the snake with "CX"

ctx.fillStyle = "#00944A"; // Cor da cabeça da cobrinha

ctx.fillRect(part.x \* gridSize, part.y \* gridSize, gridSize, gridSize);

ctx.fillStyle = "#FFFFFF"; // Cor do texto

ctx.font = "bold 10px Arial";

ctx.fillText("CX", part.x \* gridSize + 3, part.y \* gridSize + 12);

} else {

// Draw the rest of the snake

ctx.fillStyle = "#00944A"; // Cor do corpo da cobrinha

ctx.fillRect(part.x \* gridSize, part.y \* gridSize, gridSize, gridSize);

}

});

// Draw the food

ctx.fillStyle = "#CC0000"; // Cor da comida

ctx.fillRect(food.x \* gridSize, food.y \* gridSize, gridSize, gridSize);

}

function placeFood() {

food.x = Math.floor(Math.random() \* tileCount);

food.y = Math.floor(Math.random() \* tileCount);

}

function checkCollision(head) {

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

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

return true;

}

}

return false;

}

function endGame() {

isGameOver = true;

pointsElement.innerText = points;

gameOverMessage.style.display = "block";

addScoreToHistory();

}

function addScoreToHistory() {

// Adiciona a pontuação atual à lista de histórico de pontuações

const li = document.createElement("li");

li.innerText = `${attempt}ª Tentativa: ${points}`;

scoreList.appendChild(li);

attempt++;

}

function resetGame() {

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

dx = 1;

dy = 0;

points = 0;

updateScore();

isGameOver = false;

gameOverMessage.style.display = "none";

gameLoop();

}

// Control the snake

window.addEventListener("keydown", e => {

switch (e.key) {

case "ArrowUp":

if (dy === 0) {

dx = 0;

dy = -1;

}

break;

case "ArrowDown":

if (dy === 0) {

dx = 0;

dy = 1;

}

break;

case "ArrowLeft":

if (dx === 0) {

dx = -1;

dy = 0;

}

break;

case "ArrowRight":

if (dx === 0) {

dx = 1;

dy = 0;

}

break;

}

});

// Mobile controls

document.getElementById("up").addEventListener("click", () => {

if (dy === 0) {

dx = 0;

dy = -1;

}

});

document.getElementById("down").addEventListener("click", () => {

if (dy === 0) {

dx = 0;

dy = 1;

}

});

document.getElementById("left").addEventListener("click", () => {

if (dx === 0) {

dx = -1;

dy = 0;

}

});

document.getElementById("right").addEventListener("click", () => {

if (dx === 0) {

dx = 1;

dy = 0;

}

});

</script>

</body>

</html>