<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fire Racer</title>

<style>

body {

margin: 0;

overflow: hidden;

background-color: #222;

font-family: Arial, sans-serif;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

#game-container {

position: relative;

width: 400px;

height: 600px;

background-color: #333;

overflow: hidden;

border: 3px solid #ff6600;

border-radius: 8px;

}

#road {

position: absolute;

width: 100%;

height: 100%;

background-color: #444;

}

.lane-marker {

position: absolute;

width: 10px;

height: 60px;

background-color: #fff;

left: 50%;

transform: translateX(-50%);

}

#player {

position: absolute;

width: 60px;

height: 100px;

bottom: 40px;

left: 170px;

z-index: 100;

}

.flint {

position: absolute;

width: 50px;

height: 50px;

border-radius: 5px;

}

.cigar {

position: absolute;

width: 30px;

height: 40px;

border-radius: 5px;

}

#score {

position: absolute;

top: 10px;

left: 10px;

color: #fff;

font-size: 20px;

z-index: 200;

}

#speed {

position: absolute;

top: 40px;

left: 10px;

color: #ff8800;

font-size: 18px;

z-index: 200;

}

#flint-count {

position: absolute;

top: 70px;

left: 10px;

color: #ff5555;

font-size: 18px;

z-index: 200;

}

#game-over {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.7);

display: none;

flex-direction: column;

justify-content: center;

align-items: center;

z-index: 300;

}

#game-over h2 {

color: #ff6600;

font-size: 36px;

margin-bottom: 20px;

}

#game-over button {

padding: 10px 20px;

background-color: #ff6600;

border: none;

color: white;

font-size: 18px;

cursor: pointer;

border-radius: 5px;

}

#game-over button:hover {

background-color: #ff8800;

}

</style>

</head>

<body>

<div id="game-container">

<div id="road"></div>

<div id="score">Score: 0</div>

<div id="speed">Speed: 1x</div>

<div id="flint-count">Max Flints: 4</div>

<div id="player">

<svg width="60" height="100" viewBox="0 0 60 100">

<!-- Red Ferrari body -->

<rect x="10" y="30" width="40" height="70" fill="#cc0000" rx="5" />

<!-- Car details -->

<rect x="15" y="40" width="30" height="20" fill="#880000" rx="2" />

<rect x="15" y="70" width="30" height="20" fill="#880000" rx="2" />

<!-- Wheels -->

<circle cx="10" cy="60" r="8" fill="#222" />

<circle cx="10" cy="60" r="4" fill="#444" />

<circle cx="50" cy="60" r="8" fill="#222" />

<circle cx="50" cy="60" r="4" fill="#444" />

<!-- Flame character head -->

<circle cx="30" cy="25" r="15" fill="#ff8800" />

<!-- Sunglasses -->

<rect x="20" y="20" width="20" height="8" fill="#222" rx="2" />

<!-- Flame on top -->

<path d="M20,15 Q25,5 30,10 Q35,0 40,15 Z" fill="#ff6600" />

<!-- Mouth -->

<path d="M25,30 Q30,32 35,30" stroke="#222" stroke-width="2" fill="none" />

</svg>

</div>

<div id="game-over">

<h2>GAME OVER</h2>

<button id="restart-btn">Play Again</button>

</div>

</div>

<script>

// Game elements

const gameContainer = document.getElementById('game-container');

const road = document.getElementById('road');

const player = document.getElementById('player');

const scoreElement = document.getElementById('score');

const speedElement = document.getElementById('speed');

const flintCountElement = document.getElementById('flint-count');

const gameOverScreen = document.getElementById('game-over');

const restartBtn = document.getElementById('restart-btn');

// Game variables

let score = 0;

let gameSpeed = 5;

let playerX = 170;

let gameRunning = true;

let flints = [];

let cigars = [];

let keys = {};

let laneMarkers = [];

let speedMultiplier = 1;

let maxFlints = 4; // Starting maximum number of flints

let speedIncreaseInterval;

let flintIncreaseInterval; // New interval for increasing maxFlints

let distanceTraveled = 0;

// Create lane markers

function createLaneMarkers() {

const numMarkers = 10;

const spacing = 600 / numMarkers;

for (let i = 0; i < numMarkers; i++) {

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

marker.className = 'lane-marker';

marker.style.top = (i \* spacing) + 'px';

road.appendChild(marker);

laneMarkers.push(marker);

}

}

// Move lane markers to create illusion of movement

function moveLaneMarkers() {

laneMarkers.forEach(marker => {

let y = parseInt(marker.style.top);

y += gameSpeed;

if (y > 600) {

y = -60;

}

marker.style.top = y + 'px';

});

}

// Create a flint on the road

function createFlint() {

if (!gameRunning) return;

// Check if we've reached the flint limit

if (flints.length >= maxFlints) {

return;

}

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

flint.className = 'flint';

// Random lane position

const lane = Math.floor(Math.random() \* 3);

let xPos;

if (lane === 0) xPos = 80;

else if (lane === 1) xPos = 175;

else xPos = 270;

// Place flint at random position on road

flint.style.left = xPos + 'px';

flint.style.top = (-200 - Math.random() \* 800) + 'px'; // Position above road, will appear as player moves

// Add flint SVG

const flintSVG = document.createElementNS('http://www.w3.org/2000/svg', 'svg');

flintSVG.setAttribute('width', '50');

flintSVG.setAttribute('height', '50');

flintSVG.setAttribute('viewBox', '0 0 50 50');

// Random flint color

const stoneColors = ['#888888', '#777777', '#666666', '#999999', '#555555'];

const stoneColor = stoneColors[Math.floor(Math.random() \* stoneColors.length)];

// Create a jagged rock shape

const flintPath = document.createElementNS('http://www.w3.org/2000/svg', 'path');

// Make each flint shape slightly different

const randomPoints = [

[10 + Math.random() \* 5, 20 + Math.random() \* 5],

[5 + Math.random() \* 5, 30 + Math.random() \* 5],

[15 + Math.random() \* 5, 40 + Math.random() \* 5],

[30 + Math.random() \* 5, 45 + Math.random() \* 5],

[40 + Math.random() \* 5, 35 + Math.random() \* 5],

[45 + Math.random() \* 5, 20 + Math.random() \* 5],

[35 + Math.random() \* 5, 10 + Math.random() \* 5],

[20 + Math.random() \* 5, 5 + Math.random() \* 5]

];

// Create path string from points

let pathData = `M${randomPoints[0][0]},${randomPoints[0][1]}`;

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

pathData += ` L${randomPoints[i][0]},${randomPoints[i][1]}`;

}

pathData += ' Z';

flintPath.setAttribute('d', pathData);

flintPath.setAttribute('fill', stoneColor);

flintPath.setAttribute('stroke', '#333');

flintPath.setAttribute('stroke-width', '2');

// Add highlights/cracks to make flint more realistic

const highlight = document.createElementNS('http://www.w3.org/2000/svg', 'path');

highlight.setAttribute('d', `M${15 + Math.random() \* 10},${15 + Math.random() \* 10}

L${25 + Math.random() \* 10},${20 + Math.random() \* 10}`);

highlight.setAttribute('stroke', '#aaa');

highlight.setAttribute('stroke-width', '1.5');

highlight.setAttribute('fill', 'none');

flintSVG.appendChild(flintPath);

flintSVG.appendChild(highlight);

flint.appendChild(flintSVG);

gameContainer.appendChild(flint);

flints.push(flint);

// Remove flints that are completely offscreen (below)

flints = flints.filter(v => {

const rect = v.getBoundingClientRect();

const containerRect = gameContainer.getBoundingClientRect();

if (rect.top > containerRect.bottom + 100) {

v.remove();

return false;

}

return true;

});

}

// Create a cigar on the road

function createCigar() {

if (!gameRunning) return;

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

cigar.className = 'cigar';

// Random position

const xPos = 80 + Math.floor(Math.random() \* 3) \* 95; // Align with lanes better

cigar.style.left = xPos + 'px';

cigar.style.top = (-100 - Math.random() \* 600) + 'px'; // Position above road

// Add cigar SVG

const cigarSVG = document.createElementNS('http://www.w3.org/2000/svg', 'svg');

cigarSVG.setAttribute('width', '30');

cigarSVG.setAttribute('height', '40');

cigarSVG.setAttribute('viewBox', '0 0 30 40');

// Cigar body (brown cylinder)

const cigarBody = document.createElementNS('http://www.w3.org/2000/svg', 'rect');

cigarBody.setAttribute('x', '5');

cigarBody.setAttribute('y', '10');

cigarBody.setAttribute('width', '20');

cigarBody.setAttribute('height', '25');

cigarBody.setAttribute('fill', '#8B4513');

cigarBody.setAttribute('rx', '3');

// Cigar band (red band in middle)

const cigarBand = document.createElementNS('http://www.w3.org/2000/svg', 'rect');

cigarBand.setAttribute('x', '5');

cigarBand.setAttribute('y', '18');

cigarBand.setAttribute('width', '20');

cigarBand.setAttribute('height', '4');

cigarBand.setAttribute('fill', '#B22222');

// Cigar tip (ash/glow)

const cigarTip = document.createElementNS('http://www.w3.org/2000/svg', 'circle');

cigarTip.setAttribute('cx', '15');

cigarTip.setAttribute('cy', '10');

cigarTip.setAttribute('r', '5');

cigarTip.setAttribute('fill', '#FF4500');

// Smoke

const smoke1 = document.createElementNS('http://www.w3.org/2000/svg', 'path');

smoke1.setAttribute('d', 'M15,5 Q18,3 17,1');

smoke1.setAttribute('stroke', '#DDDDDD');

smoke1.setAttribute('stroke-width', '1.5');

smoke1.setAttribute('fill', 'none');

const smoke2 = document.createElementNS('http://www.w3.org/2000/svg', 'path');

smoke2.setAttribute('d', 'M13,5 Q10,2 12,0');

smoke2.setAttribute('stroke', '#DDDDDD');

smoke2.setAttribute('stroke-width', '1.5');

smoke2.setAttribute('fill', 'none');

cigarSVG.appendChild(cigarBody);

cigarSVG.appendChild(cigarBand);

cigarSVG.appendChild(cigarTip);

cigarSVG.appendChild(smoke1);

cigarSVG.appendChild(smoke2);

cigar.appendChild(cigarSVG);

gameContainer.appendChild(cigar);

cigars.push(cigar);

// Remove cigars that are completely offscreen (below)

cigars = cigars.filter(f => {

const rect = f.getBoundingClientRect();

const containerRect = gameContainer.getBoundingClientRect();

if (rect.top > containerRect.bottom + 50) {

f.remove();

return false;

}

return true;

});

}

// Update the positions of all elements to create illusion of player movement

function updatePositions() {

// Move flints

flints.forEach(flint => {

let y = parseInt(flint.style.top) || -50;

y += gameSpeed;

flint.style.top = y + 'px';

// Check for collision

if (checkCollision(player, flint)) {

gameOver();

}

});

// Move cigars

for (let i = cigars.length - 1; i >= 0; i--) {

const cigar = cigars[i];

let y = parseInt(cigar.style.top) || -40;

y += gameSpeed;

cigar.style.top = y + 'px';

// Check for cigar collection

if (checkCollision(player, cigar)) {

cigar.remove();

cigars.splice(i, 1);

score += 10;

scoreElement.textContent = `Score: ${score}`;

}

}

// Track distance traveled

distanceTraveled += gameSpeed;

// Add new objects periodically based on distance

if (distanceTraveled % 200 < gameSpeed && flints.length < maxFlints) {

createFlint();

}

if (distanceTraveled % 300 < gameSpeed) {

createCigar();

}

}

// Check collision between two elements

function checkCollision(element1, element2) {

const rect1 = element1.getBoundingClientRect();

const rect2 = element2.getBoundingClientRect();

return !(rect1.right < rect2.left ||

rect1.left > rect2.right ||

rect1.bottom < rect2.top ||

rect1.top > rect2.bottom);

}

// Move player horizontally

function movePlayer() {

if (keys.ArrowLeft) {

playerX -= 5;

}

if (keys.ArrowRight) {

playerX += 5;

}

// Keep player on screen

playerX = Math.max(0, Math.min(gameContainer.offsetWidth - player.offsetWidth, playerX));

player.style.left = playerX + 'px';

}

// Increase game speed

function increaseSpeed() {

if (!gameRunning) return;

speedMultiplier += 0.5;

gameSpeed = 5 \* speedMultiplier;

speedElement.textContent = `Speed: ${speedMultiplier}x`;

// Make the player's flame character appear more intense with speed

const playerSVG = player.querySelector('svg');

const flamePath = playerSVG.querySelector('path');

// Update flame height based on speed

const flameHeight = Math.min(15, 5 + speedMultiplier \* 2);

flamePath.setAttribute('d', `M20,15 Q25,${15-flameHeight} 30,10 Q35,${5-flameHeight} 40,15 Z`);

// Update flame color

const r = Math.min(255, 255);

const g = Math.max(0, Math.min(255, Math.floor(136 - speedMultiplier \* 10)));

const b = Math.max(0, Math.min(255, Math.floor(0)));

flamePath.setAttribute('fill', `rgb(${r},${g},${b})`);

}

// Increase maximum number of flints

function increaseMaxFlints() {

if (!gameRunning) return;

maxFlints += 1;

// Cap at a reasonable limit to prevent game from becoming impossible

if (maxFlints > 20) {

maxFlints = 20;

}

flintCountElement.textContent = `Max Flints: ${maxFlints}`;

// Create new flints if we're below the new max

if (flints.length < maxFlints) {

createFlint();

}

}

// Game loop

function gameLoop() {

if (!gameRunning) return;

movePlayer();

updatePositions();

moveLaneMarkers();

requestAnimationFrame(gameLoop);

}

// Start the game

function startGame() {

gameRunning = true;

score = 0;

gameSpeed = 5;

speedMultiplier = 1;

playerX = 170;

distanceTraveled = 0;

maxFlints = 4; // Reset max flints

scoreElement.textContent = `Score: ${score}`;

speedElement.textContent = `Speed: ${speedMultiplier}x`;

flintCountElement.textContent = `Max Flints: ${maxFlints}`;

gameOverScreen.style.display = 'none';

// Clear existing elements

flints.forEach(flint => flint.remove());

flints = [];

cigars.forEach(cigar => cigar.remove());

cigars = [];

// Clear any existing timers

clearInterval(speedIncreaseInterval);

clearInterval(flintIncreaseInterval);

// Setup lane markers

laneMarkers.forEach(marker => marker.remove());

laneMarkers = [];

createLaneMarkers();

// Create initial obstacles

for (let i = 0; i < 2; i++) {

createFlint();

}

for (let i = 0; i < 3; i++) {

createCigar();

}

// Increase speed every 5 seconds

speedIncreaseInterval = setInterval(increaseSpeed, 5000);

// Increase max flints every 2 seconds

flintIncreaseInterval = setInterval(increaseMaxFlints, 2000);

gameLoop();

}

// Game over

function gameOver() {

gameRunning = false;

clearInterval(speedIncreaseInterval);

clearInterval(flintIncreaseInterval);

gameOverScreen.style.display = 'flex';

}

// Event listeners

document.addEventListener('keydown', (e) => {

keys[e.key] = true;

});

document.addEventListener('keyup', (e) => {

keys[e.key] = false;

});

restartBtn.addEventListener('click', startGame);

// Start the game when the page loads

window.onload = startGame;

</script>

</body>

</html>