<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Happy Birthday Manvik - F1 Racer</title>

<style>

html, body {

margin: 0; padding: 0;

overflow: hidden;

font-family: sans-serif;

background: black;

}

canvas { display: block; }

#overlay {

position: absolute;

top: 10px; left: 10px;

color: white;

font-size: 16px;

z-index: 2;

}

#loading {

position: absolute;

top: 50%; left: 50%;

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

color: white;

font-size: 20px;

z-index: 3;

}

#restart {

position: absolute;

bottom: 10px;

left: 50%;

transform: translateX(-50%);

background: red;

color: white;

padding: 10px 20px;

border: none;

font-size: 18px;

cursor: pointer;

display: none;

}

#touch-controls {

position: absolute;

bottom: 20px;

width: 100%;

display: flex;

justify-content: space-around;

z-index: 2;

}

.btn {

width: 60px;

height: 60px;

background: rgba(255,255,255,0.2);

border-radius: 50%;

}

</style>

</head>

<body>

<div id="loading">Loading...</div>

<div id="overlay">Speed: 0 | Opponents Passed: 0</div>

<button id="restart">Restart Race</button>

<div id="touch-controls">

<div class="btn" id="leftBtn"></div>

<div class="btn" id="rightBtn"></div>

</div>

<script type="module">

import \* as THREE from 'https://cdn.jsdelivr.net/npm/three@0.155.0/build/three.module.js';

import { OrbitControls } from 'https://cdn.jsdelivr.net/npm/three@0.155.0/examples/jsm/controls/OrbitControls.js';

let scene, camera, renderer, car, road, buildings = [], opponents = [], obstacles = [];

let speed = 0.2, moveLeft = false, moveRight = false, passed = 0;

let isCrashed = false;

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

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

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

const touchLeft = document.getElementById('leftBtn');

const touchRight = document.getElementById('rightBtn');

init();

animate();

function init() {

scene = new THREE.Scene();

camera = new THREE.PerspectiveCamera(75, window.innerWidth/window.innerHeight, 0.1, 1000);

camera.position.set(0, 5, -10);

camera.lookAt(0, 0, 0);

renderer = new THREE.WebGLRenderer({ antialias: true });

renderer.setSize(window.innerWidth, window.innerHeight);

document.body.appendChild(renderer.domElement);

window.addEventListener('resize', () => {

camera.aspect = window.innerWidth/window.innerHeight;

camera.updateProjectionMatrix();

renderer.setSize(window.innerWidth, window.innerHeight);

});

createRoad();

createCar();

createBuildings();

createOpponents();

createObstacles();

document.addEventListener('keydown', onKeyDown);

document.addEventListener('keyup', onKeyUp);

restartBtn.onclick = restart;

// Touch

touchLeft.ontouchstart = () => moveLeft = true;

touchLeft.ontouchend = () => moveLeft = false;

touchRight.ontouchstart = () => moveRight = true;

touchRight.ontouchend = () => moveRight = false;

loading.style.display = 'none';

}

function createRoad() {

const geometry = new THREE.PlaneGeometry(10, 1000);

const material = new THREE.MeshStandardMaterial({ color: 0x222222 });

road = new THREE.Mesh(geometry, material);

road.rotation.x = -Math.PI / 2;

scene.add(road);

const light = new THREE.DirectionalLight(0xffffff, 1);

light.position.set(0, 20, 10);

scene.add(light);

scene.add(new THREE.AmbientLight(0x404040));

}

function createCar() {

const group = new THREE.Group();

const body = new THREE.Mesh(

new THREE.BoxGeometry(1, 0.5, 2),

new THREE.MeshStandardMaterial({ color: 0xff0000 })

);

body.position.y = 0.25;

group.add(body);

const wing = new THREE.Mesh(

new THREE.BoxGeometry(1.2, 0.1, 0.3),

new THREE.MeshStandardMaterial({ color: 0x0000ff })

);

wing.position.set(0, 0.35, 1);

group.add(wing);

group.position.set(0, 0, 0);

car = group;

scene.add(car);

}

function createBuildings() {

for (let i = -5; i <= 5; i += 2) {

for (let z = 0; z < 1000; z += 50) {

const height = Math.random() \* 4 + 2;

const box = new THREE.Mesh(

new THREE.BoxGeometry(1, height, 1),

new THREE.MeshStandardMaterial({ color: 0x888888 })

);

box.position.set(i < 0 ? -6 : 6, height/2, -z);

scene.add(box);

buildings.push(box);

}

}

}

function createOpponents() {

for (let i = 0; i < 5; i++) {

const opp = car.clone();

opp.position.set(Math.random()\*4-2, 0, -Math.random()\*100 - 20);

scene.add(opp);

opponents.push(opp);

}

}

function createObstacles() {

for (let i = 0; i < 10; i++) {

const obs = new THREE.Mesh(

new THREE.CylinderGeometry(0.3, 0.3, 1, 12),

new THREE.MeshStandardMaterial({ color: 0xffff00 })

);

obs.position.set(Math.random()\*4-2, 0.5, -Math.random()\*300 - 50);

scene.add(obs);

obstacles.push(obs);

}

}

function onKeyDown(e) {

if (e.key === 'ArrowLeft') moveLeft = true;

if (e.key === 'ArrowRight') moveRight = true;

}

function onKeyUp(e) {

if (e.key === 'ArrowLeft') moveLeft = false;

if (e.key === 'ArrowRight') moveRight = false;

}

function animate() {

requestAnimationFrame(animate);

if (!isCrashed) {

car.position.z -= speed;

if (moveLeft) car.position.x -= 0.1;

if (moveRight) car.position.x += 0.1;

camera.position.z = car.position.z - 10;

camera.position.x = car.position.x;

camera.lookAt(car.position.x, 0, car.position.z);

// Move opponents

for (let opp of opponents) {

opp.position.z += speed \* 0.6;

if (opp.position.z > car.position.z + 10) {

opp.position.z = car.position.z - Math.random()\*100 - 20;

opp.position.x = Math.random()\*4 - 2;

passed++;

}

}

// Check collisions

for (let obs of obstacles) {

if (car.position.distanceTo(obs.position) < 1) {

crash();

}

}

for (let opp of opponents) {

if (car.position.distanceTo(opp.position) < 1) {

crash();

}

}

overlay.textContent = `Speed: ${(speed \* 100).toFixed(0)} | Opponents Passed: ${passed}`;

}

renderer.render(scene, camera);

}

function crash() {

isCrashed = true;

overlay.textContent = "CRASHED! Happy Birthday Manvik!";