**CODE FOR HIT AND DIE WEB GAME DEVELOPMENT**

// HTML Code

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>iDragon - JavaScript Game</title>

    <link href="https://fonts.googleapis.com/css2?family=Ubuntu:wght@300&display=swap" rel="stylesheet">

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

    <script src="script.js"></script>

</head>

<body>

    <div class="gameContainer">

        <div class="gameOver" style="font-family:cursive; color:darkblue; font-size: 78px;" ><b>Welcome to iDragon Adventures </b><br>Created by Umar.  </div>

        <!-- <div class="gameOver">Welcome to iDragon - Created by Umar</div> -->

        <div class="dinoo"></div>

        <div id="scoreCont">Your Score: 0</div>

        <div class="obstacle obstacleAni"></div>

    </div>

</body>

</html>

//CSS Code

\*{

    margin: 0;

    padding:0;

}

body{

    background-color: red;

    overflow: hidden;

}

.gameContainer{

    background-image: url(wall.png.png);

    background-repeat: no-repeat;

    background-size: 100vw 100vh;

    width: 100%;

    height: 100vh;

}

.dinoo{

    background-image: url(dinoo.png);

    background-repeat: no-repeat;

    background-size: cover;

    width: 233px;

    height: 114px;

    position: absolute;

    bottom:0;

    left: 52px;

}

.obstacle{

    width: 166px;

    height: 113px;

    background-image: url(dragon.png);

    background-size: cover;

    position: absolute;

    bottom: 0;

    left: 44vw;

}

.animateDino{

    animation: dino 0.6s linear;

}

.obstacleAni{

    animation: obstacleAni 5s linear infinite;

}

.gameOver{

    position: relative;

    top:63px;

    font-size: 53px;

    text-align: center;

    font-family: 'Ubuntu', sans-serif;

}

#scoreCont{

    font-size: 25px;

    color: #54212f;

    font-weight: bold;

    position: absolute;

    right: 45px;

    top: 31px;

    border: 2px solid black;

    padding: 10px;

    font-family: 'Ubuntu', sans-serif;

    border-radius: 10px;

}

@keyframes dino{

    0%{

        bottom: 0;

    }

    50%{

        bottom: 422px;

    }

    100%{

        bottom: 0;

    }

}

@keyframes obstacleAni{

    0%{

        left: 100vw;

    }

    100%{

        left: -10vw;

    }

}

//JavaScript Code

score = 0;

cross = true;

audio = new Audio('music.mp3');

audiogo = new Audio('gameover.mp3');

setTimeout(() => {

    audio.play()

}, 1000);

document.onkeydown = function (e) {

    console.log("Key code is: ", e.keyCode)

    if (e.keyCode == 38) {

        dino = document.querySelector('.dinoo');

        dino.classList.add('animateDino');

        setTimeout(() => {

            dino.classList.remove('animateDino')

        }, 700);

    }

    if (e.keyCode == 39) {

        dino = document.querySelector('.dinoo');

        dinoX = parseInt(window.getComputedStyle(dino, null).getPropertyValue('left'));

        dino.style.left = dinoX + 112 + "px";

    }

    if (e.keyCode == 37) {

        dino = document.querySelector('.dinoo');

        dinoX = parseInt(window.getComputedStyle(dino, null).getPropertyValue('left'));

        dino.style.left = (dinoX - 112) + "px";

    }

}

setInterval(() => {

    dinoo = document.querySelector('.dinoo');

    gameOver = document.querySelector('.gameOver');

    obstacle = document.querySelector('.obstacle');

    dx = parseInt(window.getComputedStyle(dino, null).getPropertyValue('left'));

    dy = parseInt(window.getComputedStyle(dino, null).getPropertyValue('top'));

    ox = parseInt(window.getComputedStyle(obstacle, null).getPropertyValue('left'));

    oy = parseInt(window.getComputedStyle(obstacle, null).getPropertyValue('top'));

    offsetX = Math.abs(dx - ox);

    offsetY = Math.abs(dy - oy);

    // console.log(offsetX, offsetY)

    if (offsetX < 73 && offsetY < 52) {

        gameOver.innerHTML = "Game Over - Reload to Play Again"

        obstacle.classList.remove('obstacleAni')

        audiogo.play();

        setTimeout(() => {

            audiogo.pause();

            audio.pause();

        }, 1000);

    }

    else if (offsetX < 145 && cross) {

        score += 1;

        updateScore(score);

        cross = false;

        setTimeout(() => {

            cross = true;

        }, 1000);

        setTimeout(() => {

            aniDur = parseFloat(window.getComputedStyle(obstacle, null).getPropertyValue('animation-duration'));

            newDur = aniDur - 0.1;

            obstacle.style.animationDuration = newDur + 's';

            console.log('New animation duration: ', newDur)

        }, 500);

    }

}, 10);

function updateScore(score) {

    scoreCont.innerHTML = "Your Score: " + score

}