HTML

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sabiomem: Desafío Matemático Espacial</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <!-- PANTALLA DE INICIO -->
    <div id="start-screen">
        <div class="card">
            <h1>SABIOMEM  </h1>
            <p>Misión de Aprendizaje Interactivo</p>
```

```html
            <select id="user-role-select">
                <option value="" disabled selected>Selecciona tu Rol</option>
                <option>Estudiante</option>
                <option>Maestro</option>
            </select>
```

```html
            <input type="text" id="username-input" placeholder="Nombre del Piloto"
maxlength="12">
```

```html
            <select id="difficulty-select">
                <option value="" disabled selected>Nivel de Desafío</option>
                <option value="1">Fácil</option>
                <option value="2">Medio</option>
                <option value="3">Difícil</option>
            </select>
```

```html
            <select id="game-time-select">
                <option value="" disabled selected>Tiempo de Misión</option>
                <option value="60">1 Minuto</option>
                <option value="120">2 Minutos</option>
                <option value="180">3 Minutos</option>
            </select>
```

```html
            <button id="start-button">INICIAR MISIÓN</button>
        </div>
    </div>
```

```html
    <!-- PANTALLA DE JUEGO -->
    <div id="game-wrapper" class="hidden">
        <div class="hud">
            <div>Piloto: <span id="display-name"></span></div>
            <div id="timer-display">Tiempo: 0s</div>
            <div id="score-display">Puntos: 0</div>
        </div>
```

```html
        <div id="game-area">
            <div id="player"></div>
            <div id="obstacle"></div>
        </div>
```

```html
        <div id="ranking-container">
            <h3>  TOP 5</h3>
            <ul id="ranking-list"></ul>
        </div>
    </div>

    <!-- MODAL MATEMÁTICO -->
    <div id="math-modal" class="hidden">
        <div class="modal-content">
            <h2>  SISTEMAS DAÑADOS</h2>
            <p>Resuelve el cálculo:</p>
            <h1 id="math-question"></h1>
            <input type="number" id="math-answer">
            <button id="submit-answer">REPARAR</button>
            <p id="math-feedback"></p>
        </div>
    </div>

    <script src="script.js"></script>
</body>
</html>
```

CSS

```css
body, html {
    margin: 0;
    height: 100%;
    font-family: Arial, sans-serif;
    background: url('sabiomem.jpeg') no-repeat center/cover;
    color: white;
    overflow: hidden;
}

.hidden { display: none !important; }
```

```css
/* Inicio */
#start-screen {
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    background: rgba(0,0,0,0.7);
}
```

```css
.card {
    background: rgba(255,255,255,0.15);
    padding: 30px;
    border-radius: 20px;
    text-align: center;
    width: 350px;
    backdrop-filter: blur(10px);
}
```

```css
input, select, button {
    width: 100%;
```

```css
    padding: 12px;
    margin: 8px 0;
    border-radius: 8px;
    border: none;
}

button {
    background: #00ffcc;
    font-weight: bold;
    cursor: pointer;
}

/* Juego */
.hud {
    display: flex;
    justify-content: space-around;
    background: rgba(0,0,0,0.85);
    padding: 10px;
}

#game-area {
    width: 800px;
    height: 300px;
    margin: 40px auto;
    position: relative;
    background: rgba(0,0,0,0.4);
    border-radius: 10px;
}

#player {
    width: 60px;
    height: 35px;
    background: silver;
    position: absolute;
    bottom: 0;
    left: 50px;
}

#obstacle {
    width: 45px;
    height: 45px;
    background: red;
    position: absolute;
    bottom: 0;
    right: -60px;
}

/* Modal */
#math-modal {
    position: fixed;
    inset: 0;
    background: rgba(0,0,0,0.9);
    display: flex;
    justify-content: center;
    align-items: center;
}

.modal-content {
    background: white;
    color: black;
    padding: 30px;
```

```css
    border-radius: 15px;
    text-align: center;
}

/* Ranking */
#ranking-container {
    width: 300px;
    margin: auto;
    background: rgba(255,255,255,0.15);
    padding: 10px;
    border-radius: 10px;
}
```

JS

```javascript
let score = 0;
let timeLeft = 0;
let isJumping = false;
let obstacleSpeed = 8;
let difficulty = 1;
let gameInterval, timerInterval;
let currentCorrectAnswer;

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

// Iniciar
document.getElementById('start-button').addEventListener('click', () => {
    const name = document.getElementById('username-input').value;
    const time = document.getElementById('game-time-select').value;
    const diff = document.getElementById('difficulty-select').value;

    if(!name || !time || !diff) {
        alert("Completa todos los campos");
        return;
    }

    difficulty = parseInt(diff);
    obstacleSpeed = difficulty === 1 ? 8 : difficulty === 2 ? 12 : 16;
    timeLeft = parseInt(time);

    document.getElementById('display-name').innerText = name;
    document.getElementById('start-screen').classList.add('hidden');
    document.getElementById('game-wrapper').classList.remove('hidden');

    startGame();
});

function startGame() {
    score = 0;
    updateScore();
    startTimer();
```

```javascript
        moveObstacle();
}

// Salto
document.addEventListener('keydown', e => {
    if((e.code === "Space" || e.code === "ArrowUp") && !isJumping) {
        isJumping = true;
        player.style.bottom = "180px";
        setTimeout(() => {
            player.style.bottom = "0";
            isJumping = false;
        }, 500);
    }
});

function moveObstacle() {
    let pos = 800;
    gameInterval = setInterval(() => {
        pos -= obstacleSpeed;
        if(pos < -60) {
            pos = 800;
            score += 10 * difficulty;
            updateScore();
        }
        obstacle.style.right = (800 - pos) + "px";

        let playerBottom = parseInt(getComputedStyle(player).bottom);
        if(pos < 110 && pos > 50 && playerBottom < 40) {
            collision();
        }
    }, 20);
}

function collision() {
    clearInterval(gameInterval);
    document.getElementById('math-modal').classList.remove('hidden');
    generateMath();
}

function generateMath() {
    let n1 = Math.floor(Math.random()*10)+5;
    let n2 = Math.floor(Math.random()*10)+5;
    currentCorrectAnswer = n1 + n2;
    document.getElementById('math-question').innerText = `${n1} + ${n2}`;
}

document.getElementById('submit-answer').addEventListener('click', () => {
    const ans = parseInt(document.getElementById('math-answer').value);
    if(ans === currentCorrectAnswer) {
        document.getElementById('math-modal').classList.add('hidden');
        document.getElementById('math-answer').value = "";
        moveObstacle();
    } else {
        document.getElementById('math-feedback').innerText = "Respuesta incorrecta";
    }
});

function startTimer() {
    timerInterval = setInterval(() => {
        timeLeft--;
        document.getElementById('timer-display').innerText = `Tiempo: ${timeLeft}s`;
```

```
        if(timeLeft <= 0) endGame();
    }, 1000);
}
```

```
function endGame() {
    clearInterval(gameInterval);
    clearInterval(timerInterval);
    alert(`Juego terminado. Puntaje: ${score}`);
    location.reload();
}
```

```
function updateScore() {
    document.getElementById('score-display').innerText = `Puntos: ${score}`;
}
```

- URL del repositorio con el commit y el push realizado de lo solicitado.

https://github.com/ralvaradadocontrera/Proyectosabiomen