WEB GALERIA DE VIDEOJUEGOS

Daniel Garcia Brun

1º DAM

11/12/2024

Contenido

Pagina inicial

Código

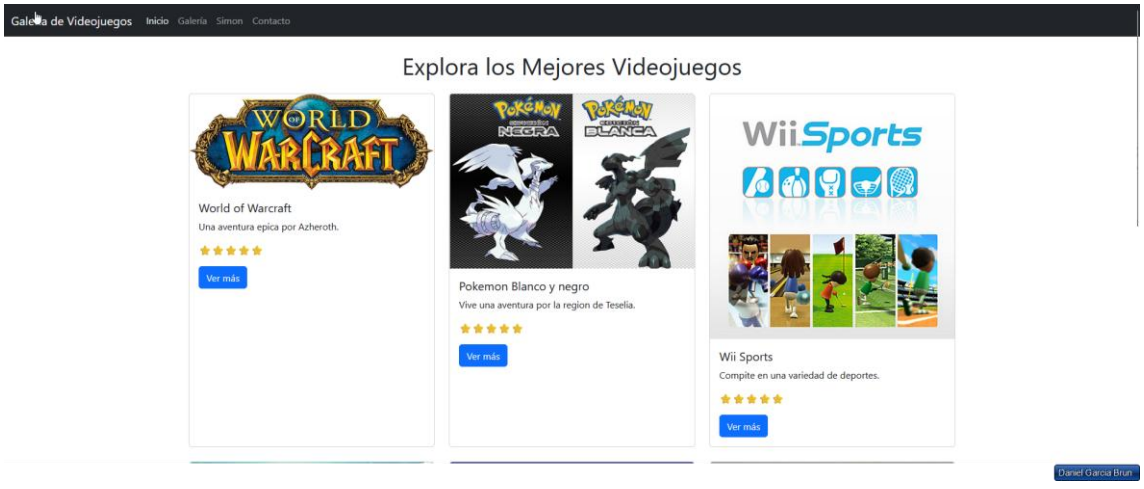
Aquí podemos ver como he puesto todos los modals y cards en el java script el cual tiene un JSON detrás como se puede ver con la función link vinculamos el JavaScript, en el footer tiene en class lo de bg dark para ponerle el color negro

```
<!-- Main Content -->
<main id="main" class="container my-4">
  <h1 class="text-center mb-4">Explora los Mejores Videojuegos</h1>
  <div class="row" id="game-cards-container">
    <!-- Las tarjetas de videojuegos se cargarán aquí -->
  </div>
</main>

<!-- Modals -->
<div id="modals-container">
  <!-- Los modales se cargarán aquí -->
</div>
<!-- Footer -->
<footer id="footer" class="bg-dark text-white text-center py-4">
  <p>&copy; 2024 Galería de Videojuegos. Todos los derechos reservados.</p>
</footer>
<script src="./js/carga.js"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
```

Daniel Garcia Brun

Vista web

Como se puede ver los juegos en todas las vistas de pantalla tiene el mismo formato tres juegos por fila



The legend of Zelda Tears of the kingdom

Enbarcate en una emocionante aventura por Hyrule

★★★★★

Ver más

Minecraft

Deja colar tu imaginacion y construye.

★★★★

Ver más

Crash of the titans

Pelea y controla monstruos

★★★

Ver más

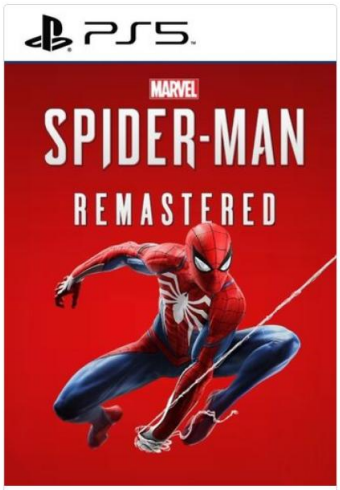
Daniel Garcia Brun

Mario y luigi compañeros en el tiempo

Salva el reino champioñon otra vez.

★★★★★

Ver más



Spider-Man

Vuelvete Spiderman y balanceate por la ciudad

★★★★★

Ver más

Tetris

Apila bloques

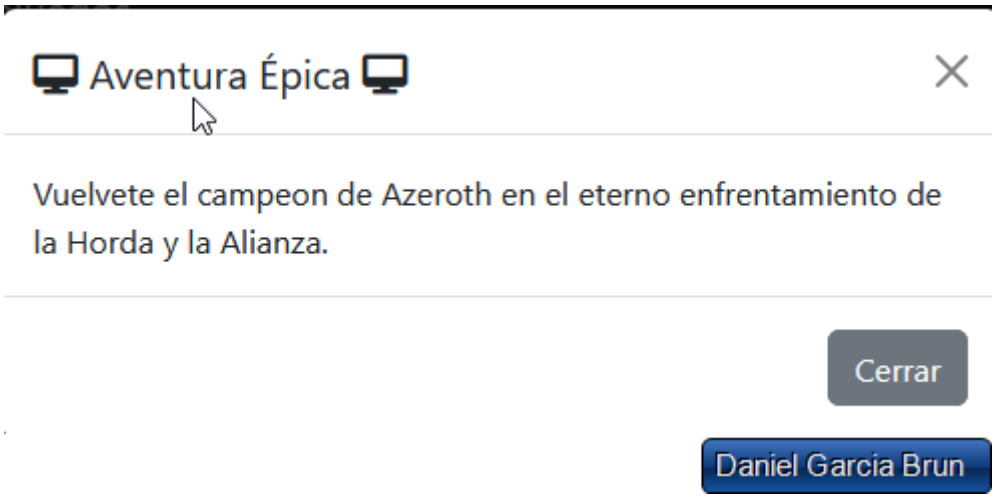
★★★★

Ver más

Daniel Garcia Brun

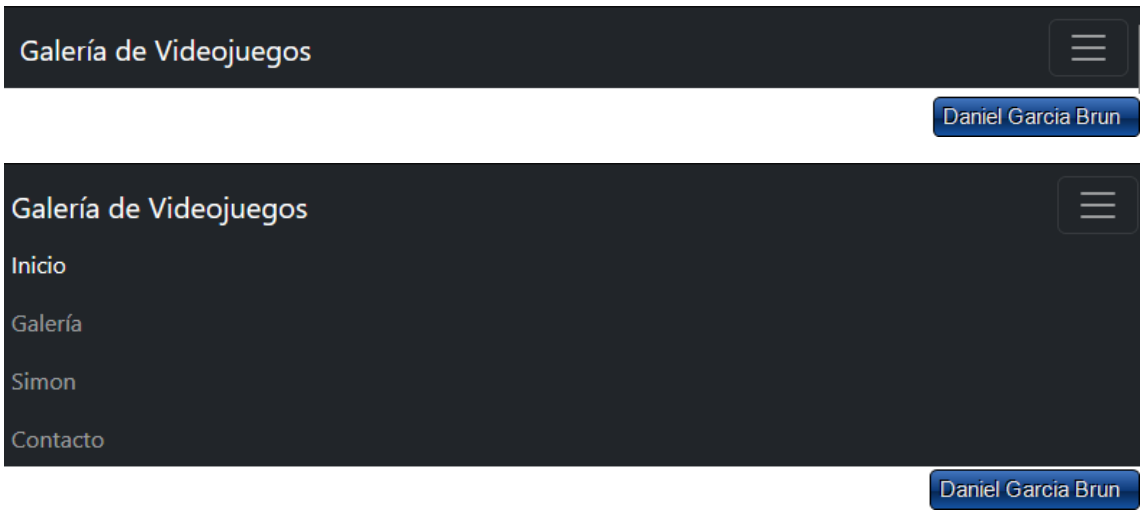
Modal

Aquí se puede ver que es lo que se abre cuando le damos a ver más, en todas las vistas se conserva la cuadricula, también se nos implementa el botón cerrar o una cruz para volver a ver los juegos



Header

Si ponemos la vista pequeña, el menú nav se mete en esas tres rayas y cuando le das clic se expande



Java Script

Aquí podemos ver como cargamos la info del JSON al html, primero lo buscamos por eso la ruta ./JSON/ganes.json, una vez echo esto escribimos que queremos que se ponga en el html, para esto le decimos que cree un elemento con creareElement y con className le ponemos nombre para encontrarlo en caso de añadirle un css, con innertHTML escribimos como si estuviésemos en el Html y el ${} ponemos la variable del json que queremos que se ponga en este caso game.title

```
document.addEventListener('DOMContentLoaded', function () {
    const gameCardsContainer = document.getElementById('game-cards-container');
    const modalsContainer = document.getElementById('modals-container');

    if (!gameCardsContainer || !modalsContainer) {
        console.error('No se encontraron los contenedores necesarios en el DOM.');
        return;
    }

    // Ruta correcta al JSON según tu estructura
    fetch('./JSON/games.json')
        .then(response => {
            if (!response.ok) {
                throw new Error('Error al cargar el archivo JSON');
            }
            return response.json();
        })
        .then(data => {
            // Asegurarse de acceder a "games"
            const games = data.games;

            games.forEach(game => {
                // Crear tarjeta
                const card = document.createElement('div');
                card.className = 'col-md-4 mb-4';
                card.innerHTML = `
                    <div class="card h-100">
                        <img src="${game.image}" class="card-img-top" alt="${game.title}">
                        <div class="card-body">
                            <h5 class="card-title"> ${game.title} </h5>
                            <p class="card-text">${game.description}</p>
```

Daniel Garcia Brun

Una vez escrito eso haremos los modals de la misma forma

```
                        <p class="card-text">${game.calificacion}</p>
                        <button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#${game.modal.id}">V
                    </div>
                </div>
            `;
            gameCardsContainer.appendChild(card);

            // Crear modal
            const modal = document.createElement('div');
            modal.className = 'modal fade';
            modal.id = game.modal.id;
            modal.tabIndex = -1;
            modal.setAttribute('aria-labelledby', `${game.modal.id}Label`);
            modal.setAttribute('aria-hidden', 'true');
            modal.innerHTML = `
                <div class="modal-dialog">
                    <div class="modal-content">
                        <div class="modal-header">
                            <h5 class="modal-title" id="${game.modal.id}Label"><i class="fa-solid fa-desktop"></i>
                            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></bu
                        </div>
                        <div class="modal-body">
                            ${game.modal.body}
                        </div>
```

Daniel Garcia Brun

```
                        </div>
                        <div class="modal-footer">
                            <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</button
                        </div>
                    </div>
                </div>
            `;
            modalsContainer.appendChild(modal);
        });
    })
    .catch(error => console.error('Error al cargar los datos:', error));
});
```

Daniel Garcia Brun

JSON

Aquí podemos ver el JSON cuya lo que le proporciona información al modal y a las tarjetas e el html

```
{
    "games": [
        {
            "title": "World of Warcraft",
            "description": "Una aventura epica por Azheroth.",
            "image": "../Fotos/Cards Juegos/World of Warcraft.png",
            "calificacion": "⭐⭐⭐⭐⭐",
            "modal": {
                "id": "modal1",
                "title": "Aventura Épica",
                "body": "Vuelvete el campeon de Azeroth en el eterno enfrentamiento de la Horda y la Alianza. "

            }
        },
```

Daniel Garcia Brun

Galería

Código

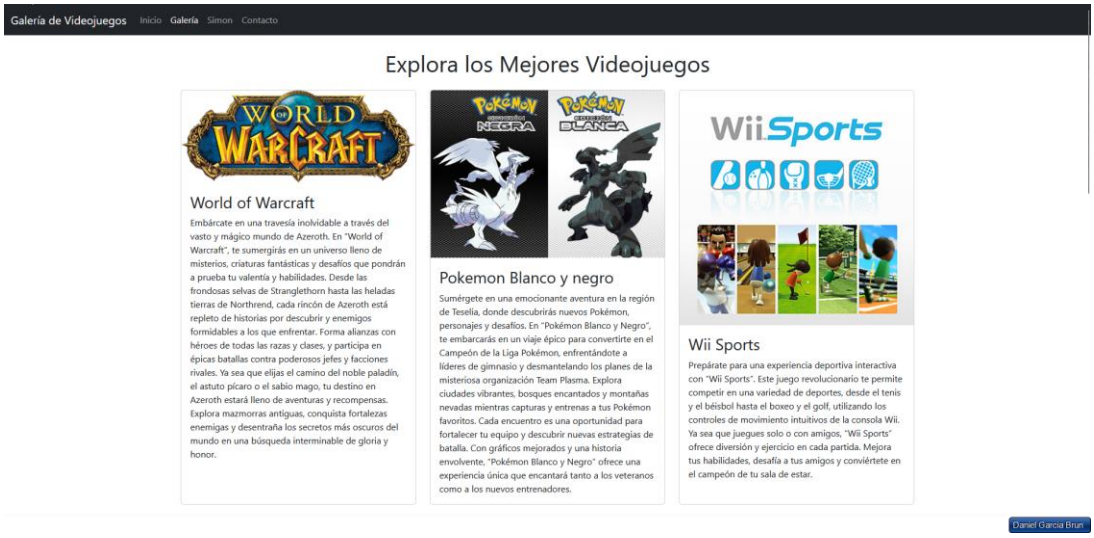
Aquí he optado por hacerlo de la forma mas sencilla ya que no me dio tiempo a pasarlo a otro Json, en esta pagina vemos la portada del juego en la misma disposición que la anterior, pero tiene mucha información añadida del videojuego

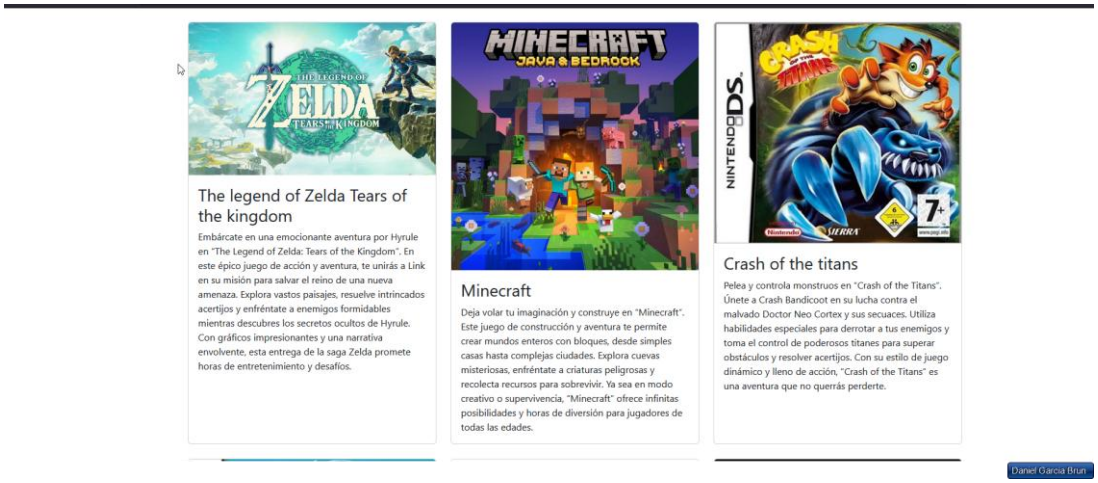
```
<!-- Main Content -->
<main id="main" class="container my-4">
  <h1 class="text-center mb-4">Explora los Mejores Videojuegos</h1>
  <div class="row">
    <!-- Video Game Card 1 -->
    <div class="col-md-4 mb-4">
      <div class="card h-100">
        <img src="../Fotos/Cards Juegos/World of Warcraft.png" class="card-img-top" alt="Aventura Épica">
        <div class="card-body">
          <h3>World of Warcraft</h3>
          <p class="card-text">Embárcate en una travesía inolvidable a través del vasto y mágico mundo de Azeroth. En
            Forma alianzas con héroes de todas las razas y clases, y participa en épicas batallas contra poderosos je
        </div>
      </div>
    </div>
```

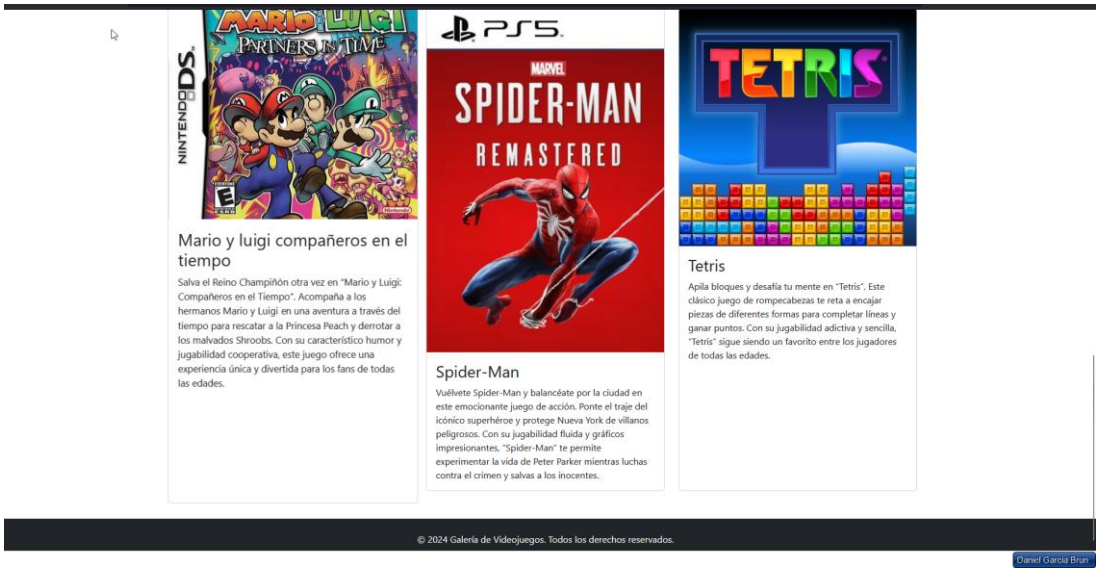
Daniel García Brun

Vista Web

Aquí podemos ver a lo que me refería, Tenemos un total de 9 cartas de videojuegos ordenadas en filas de 3 cartas







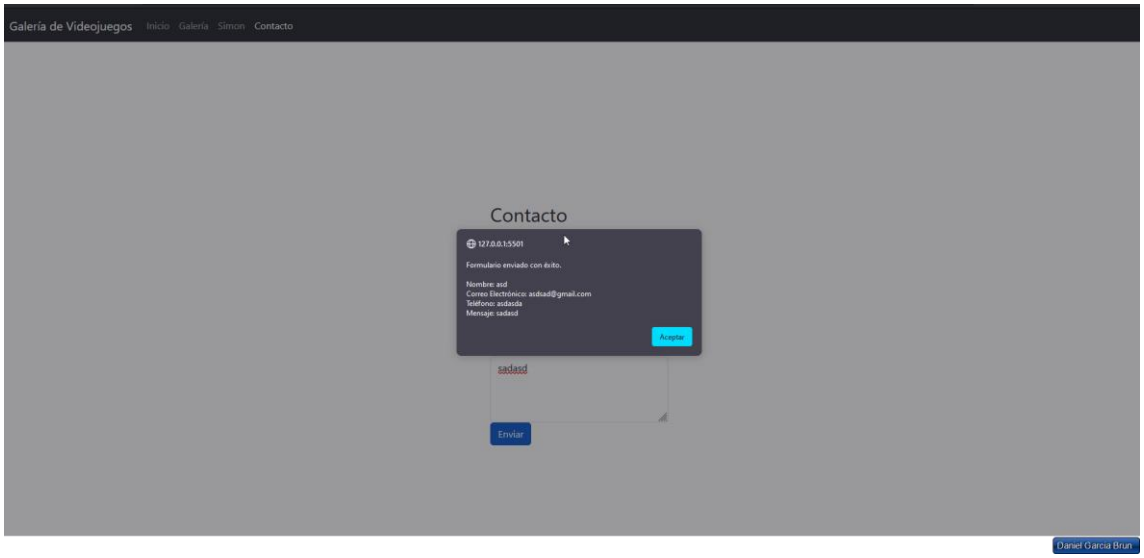
Contacto

Código

La pagina de contacto es igual a la del resto de webs, he hecho un formulario vinculado a un JavaScript que cuando le das a enviar parece una ventana emergente con los datos introducidos, se puede ver el comando form y los inputs para introducir datos

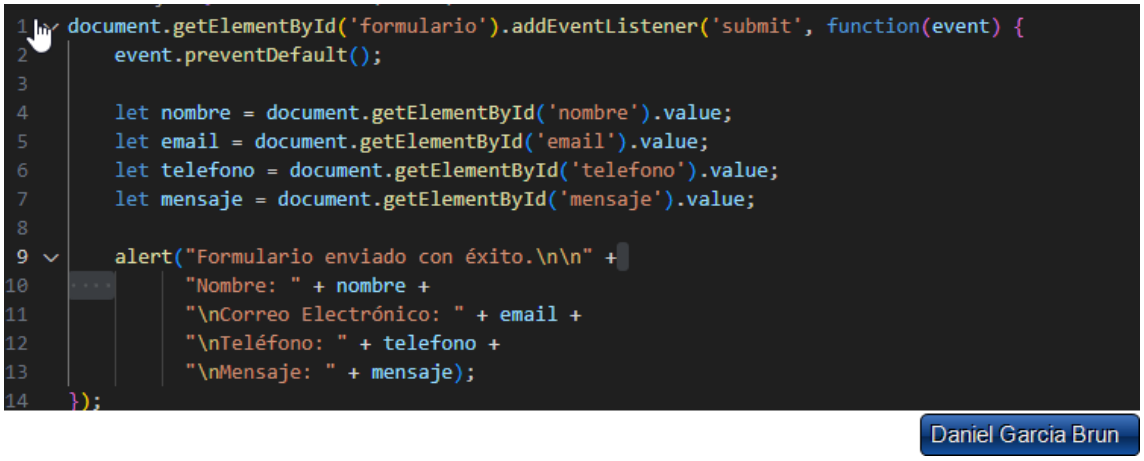
Vista web

Aquí podemos ver que una vez enviado el formulario pone lo que has introducido, el diseño de la pagina de contacto me gusta hacerlo minimalista ya que solo pongo un formulario así que opto por no hacer algo muy grande que quede muy abrumador



JavaScript

Aquí podemos ver el JavaScript que recoge los datos introducidos en los id u lo printea por pantalla al darle clic al botón de enviar del formulario esto es posible gracias al alert



Simón

Código HTML

Esto es un juego simon, el juego resalta una ficha y le das si fallas en el orden pierdes, aquí en el html solo hay para empezar el juego ya que el resto esta en el JavaScript, como podemos ver las id están colocadas pero no tenemos contenido

```
</header>
  <!-- Contenido del juego Simon -->
  <div class="container text-center mt-5">
    <h1 id="heading">Simon Game</h1>
    <div id="info" class="hidden mt-3 d-flex justify-content-center align-items-center" style="height: 50px;">
      <!-- Este contenedor ahora está centrado vertical y horizontalmente -->
    </div>
    <!-- Contenedor de las fichas -->
    <div id="tileContainer" class="d-flex justify-content-center flex-wrap gap-3 my-4 unclickable">
      <div data-tile="red" data-sound="red-sound" class="tile" ></div>
      <div data-tile="green" data-sound="green-sound" class="tile" ></div>
      <div data-tile="blue" data-sound="blue-sound" class="tile"></div>
      <div data-tile="yellow" data-sound="yellow-sound" class="tile" ></div>
    </div>
    <!-- Botón de inicio -->
    <button id="startButton" class="btn btn-primary">Start Game</button>
  </div>

  <!-- Footer -->
<footer id="footer" class="bg-dark text-white text-center py-4 footer">
      <p>&copy; 2024 Galería de Videojuegos. Todos los derechos reservados.</p>
    </footer>

  <script src="../js/siomon.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
```

Daniel García Brun

Vista Web

Aquí se puede ver que simplemente es un start game y 4 colores , esto esta echo que cuando le das clic a start game uno de ellos aumentara de amaño con scale 1.1 y tienes x segundos para darle clic si el tiempo pasa o fallas perderás.

CSS

Este es el Css el cual solo contiene el color de cada ficha, y que cuando se actore la ficha para iniciar la secuencia crezca en una escala 1.2, aparte de añadirle el border solid para que las fichas se vean bien

```css
[data-tile="red"] {
    background-color: red;
}

[data-tile="green"] {
    background-color: green;
}

[data-tile="blue"] {
    background-color: blue;
}

[data-tile="yellow"] {
    background-color: yellow;
}

/* Estilo activado */
.activated {
    transform: scale(1.2);
    opacity: 0.7;
    box-shadow: 0 0 20px white;
}

/* Estilo de las fichas */
#tileContainer div {
    width: 100px;
    height: 100px;
    margin: 10px;
    display: inline-block;
    border: 2px solid black;
    border-radius: 10px;
```

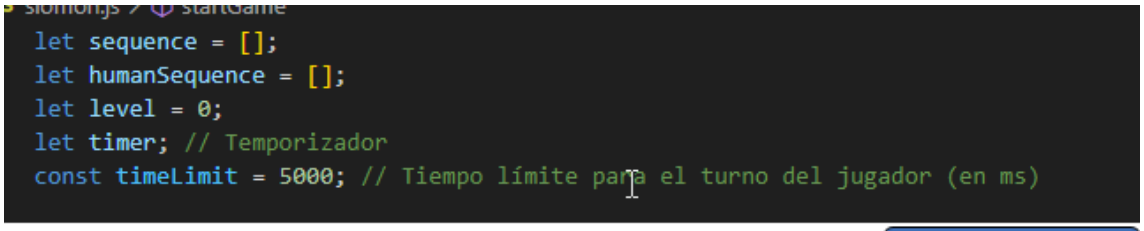
Daniel Garcia Brun

Java Script

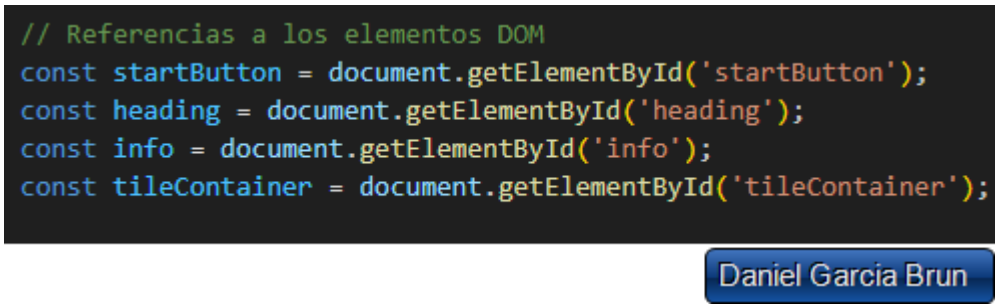
Primero de todo podemos observar variables vacías, en sequence es donde se almacena que pasos a procedido a hacer y human sequencie es lo mismo, pero de lo que hemos clicado nosotros, el sequence se almacena hasta que acaba el juego y el de los humanos se borra a cada ronda.

El let level es para aumentar la ronda, la dificultad y saber por qué ronda vamos.

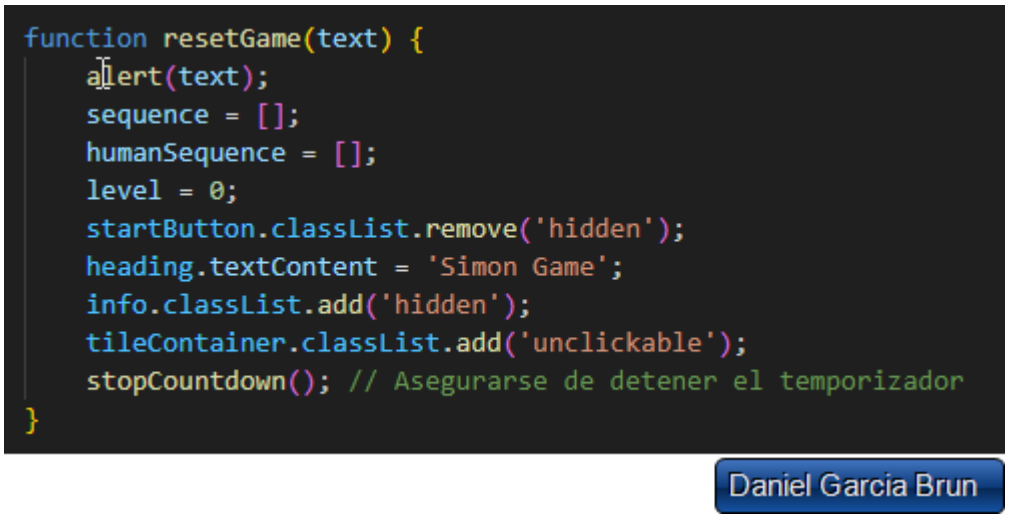
Por último, está el timer esto es un toque personal para que en 5000ms o mejor dicho 5 segundos el jugador pierda instantáneamente.



Aquí podemos ver que asignamos acciones del html al java script



Como podemos ver aquí está preparado para cuando le damos a empezar juego el texto desaparece y inicializamos las funciones antes mencionadas pero vacías, aparte de que cuando acabemos un nivel el contador se para y se reinicia



Aquí podemos ver que una vez acaba el Bot de hacer su secuencia inicia el contador y se inicializa para que empecemos a repetirla,.

```
function humanTurn(level) {
    tileContainer.classList.remove('unclickable');
    startCountdown(); // Inicia la cuenta regresiva
}
```

Daniel Garcia Brun

Aquí podemos ver la función para que los colores se inicien, le asignamos math random y que como máximo sea la cantidad de titles que haya

```
function nextStep() {
    const tiles = ['red', 'green', 'blue', 'yellow'];
    const randomIndex = Math.floor(Math.random() * tiles.length);
    return tiles[randomIndex];
}
```

Daniel Garcia Brun

Aquí podemos ver que creamos la funcio de que si comete un error el juego pare

```
function handleMistake(reason = '¡Juego terminado! Has cometido un error.') {
    stopCountdown(); // Detener temporizador
    resetGame(reason);
}
```

Daniel Garcia Brun

Aunque aun hay mas funciones vamos a ir a cómo funciona esto

Aquí podemos ver que las funciones se juntan para hacer que el juego funcione, empezamos que si le damos a start buton el juego inicializa,podemos ver que tenemos el if y un else el if es en caso de que fallemos llama a la función del fallo y el else es en caso de que el if no se cumpla haz esto otro que eria básicamente seguir sumando niveles

```
// Listeners
startButton.addEventListener('click', startGame);
tileContainer.addEventListener('click', event => {
    const { tile } = event.target.dataset;
    if (!tile) return;

    const correctTile = sequence[humanSequence.length];
    humanSequence.push(tile);
    activateTile(tile);

    if (tile !== correctTile) {
        handleMistake();
        return;
    }

    if (humanSequence.length === sequence.length) {
        handleSequenceCompletion();
    }
});
```

Daniel Garcia Brun