Informe del Juego de memoria

**Introducción**

Este informe describe el desarrollo del juego de memoria que creamos utilizando tecnologías web como HTML, CSS y JavaScript. El juego está diseñado para desafiar a los jugadores a encontrar pares de imágenes relacionadas con herramientas informáticas. Además, el proyecto incluye la integración de música de fondo controlada por el usuario. A continuación, se presentan los detalles técnicos y explicaciones de cada parte del código.

**Funcionalidades Destacadas:**

* Juego de Memoria: Los jugadores deben emparejar todas las cartas de herramientas informáticas dentro de un límite de tiempo establecido 10min.
* Controles de Música: Permite ajustar el volumen y controlar la reproducción del audio de fondo para personalizar la experiencia del usuario.
* Temporizador: Inicia automáticamente al comenzar el juego y muestra el tiempo restante en minutos y segundos. Se detiene cuando se completan todas las coincidencias o cuando se agota el tiempo disponible.

**Objetivos:**

* Proporcionar una experiencia de usuario interactiva y atractiva.
* Mejorar las habilidades de memoria y concentración de los jugadores.
* Demostrar habilidades en el desarrollo web utilizando HTML, CSS y JavaScript.

**Código HTML**

* El archivo HTML define la estructura básica de la página web del juego de memoria.

<!DOCTYPE html>

* Esta línea indica que el documento es HTML5.

**<**html lang='en'>

* Define el inicio del documento HTML y especifica el lenguaje (en este caso, inglés).

<head>

* Contiene meta-información sobre el documento, como su codificación y el título.

<meta charset='UTF-8'>

* Establece la codificación de caracteres como UTF-8 para admitir caracteres especiales.

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

* Asegura que el diseño sea responsive, ajustándose al tamaño de la pantalla del dispositivo que se esté utilizando.

**<**title>Memory Game - Herramientas Informáticas</title>

* Define el título que aparecerá en la pestaña del navegador.

<link rel='stylesheet' href='styles.css'>

* Incorpora el archivo CSS externo para aplicar estilos a la página.

</head>

<body>

* Contiene todos los elementos visibles en la página web.

<div class='game-container'>

* Este contenedor agrupa todos los elementos principales del juego.

<h1>Juego de Memoria: Herramientas Informáticas</h1>

* Encabezado principal del juego que describe el título del mismo.

<div id='game-board' class='game-board'></div>

* Elemento que contiene las cartas del juego de memoria.

<div id='status-message' class='status-message'></div>

* Muestra mensajes al jugador, como resultados de partidas y errores.

<button id='reset-button' class='reset-button'>Reiniciar Juego</button>

* Botón que permite reiniciar el juego cuando se presiona.

<!-- Controles de música -->

* Sección de controles de música, incluyendo el botón de reproducción y el control de volumen.

<button id='play-pause-btn'>🔊</button>

* Botón que permite al usuario reproducir o pausar la música de fondo.

<input type='range' id='volume-control' min='0' max='1' step='0.1' value='0.5'>

* Control deslizante que permite ajustar el volumen de la música de fondo entre 0 (silencio) y 1 (volumen máximo).

<audio id='background-music' src='audio/Musica1.mp3' loop></audio>

* Etiqueta que carga la música de fondo y permite su reproducción en bucle.

</div>

</body>

</html>

**Código CSS**

El archivo CSS proporciona el estilo visual de la página, definiendo cómo se ven los diferentes elementos del juego, como el tablero de juego, las tarjetas, los botones y los controles de música.

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

* Este bloque elimina los márgenes y rellenos predeterminados de todos los elementos en la página.
* Lo que hace es asegura que el diseño sea consistente en todos los navegadores, evitando que los elementos tengan espacios indeseados por defecto.
* **box-sizing: border-box**: Cambia el modelo de caja de los elementos para que el ancho y la altura incluyan el padding y el borde, lo que facilita el cálculo del tamaño de los elementos.

body {

font-family: Arial, sans-serif;

color: #ffffff;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

overflow: hidden; /\* Evita el scroll \*/

position: relative; /\* Necesario para el posicionamiento del video \*/

}

* Establece el estilo general del cuerpo del documento.
  + font-family: Define la fuente del texto.
  + color: Cambia el color del texto a blanco.
  + display: flex: Utiliza Flexbox para alinear elementos hijos.
  + justify-content y align-items: Centran el contenido tanto horizontal como verticalmente en la pantalla.
  + height: 100vh: Hace que el cuerpo ocupe toda la altura de la ventana del navegador.
  + overflow: hidden: Evita el scroll de la página.
  + position: relative: Permite posicionar elementos absolutamente dentro del cuerpo.

#background-video {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover; /\* Asegura que el video cubra toda la pantalla \*/

z-index: -1; /\* Coloca el video detrás del contenido \*/

}

* Estilos aplicados al video de fondo.
  + position: absolute: Permite que el video cubra toda la pantalla.
  + top y left: Posicionan el video en la esquina superior izquierda.
  + width y height: Hacen que el video ocupe el 100% de la pantalla.
  + object-fit: cover: Asegura que el video cubra toda la pantalla sin distorsionarse.
  + z-index: -1: Coloca el video detrás del resto del contenido de la página.

.game-container {

text-align: center;

background-color: rgba(28, 20, 20, 0.8); /\* Color de fondo semi-transparente \*/

padding: 20px;

border-radius: 10px;

}

* Define el estilo del contenedor que alberga el juego.
  + text-align: center: Centra el texto dentro del contenedor.
  + background-color: Establece un fondo semi-transparente oscuro, lo que ayuda a resaltar el contenido frente al video de fondo.
  + padding: Agrega espacio interno alrededor del contenido.
  + border-radius: Redondea las esquinas del contenedor.

h1 {

color: #00BFFF; /\* Azul brillante \*/

}

* Estilo aplicado al elemento <h1>.
* Cambia el color del texto del título a un azul brillante, haciéndolo más atractivo visualmente.

.game-board {

display: grid;

grid-template-columns: repeat(10, 100px); /\* Cambiado a 10 columnas \*/

grid-gap: 10px;

justify-content: center;

margin-top: 20px;

padding: 20px;

border-radius: 10px;

}

* Estilo del tablero del juego.
  + display: grid: Utiliza un sistema de cuadrícula para organizar los elementos dentro del tablero.
  + grid-template-columns: Crea 10 columnas de 100 píxeles cada una.
  + grid-gap: Establece el espacio entre las columnas y filas.
  + justify-content: Centra el tablero en su contenedor.
  + margin-top: Agrega espacio arriba del tablero.
  + padding y border-radius: Mejoran la apariencia visual del tablero.

.card {

background-color: #000000; /\* Tarjetas negras \*/

width: 100px;

height: 120px;

position: relative;

cursor: pointer;

border-radius: 10px;

border: 2px solid #00BFFF; /\* Borde azul \*/

}

* Estilo aplicado a las tarjetas individuales del juego.
  + background-color: Define el fondo de las tarjetas como negro.
  + width y height: Establecen el tamaño de las tarjetas.
  + position: relative: Permite posicionar elementos secundarios dentro de cada tarjeta.
  + cursor: pointer: Cambia el cursor al pasar sobre la tarjeta para indicar que es interactiva.
  + border-radius: Redondea las esquinas de las tarjetas.
  + border: Agrega un borde azul alrededor de las tarjetas.

.card img {

width: 100%;

height: 100%;

border-radius: 10px;

display: none;

}

.flipped img {

display: block;

}

* Estilo para las imágenes dentro de las tarjetas.
  + Las imágenes ocupan el 100% del ancho y alto de la tarjeta.
  + display: none: Oculta las imágenes por defecto.
  + En el estado .flipped, las imágenes se muestran (display: block).

.status-message {

margin-top: 20px;

font-size: 1.2em;

color: #FFD700;

}

.timer {

color: #FFD700;

font-size: 1.5em;

margin-top: 10px;

}

* Estilos para el mensaje de estado y el temporizador.
  + margin-top: Agrega espacio por encima de estos elementos.
  + color: Cambia el color del texto a dorado.
  + font-size: Ajusta el tamaño de la fuente para hacerlos más visibles.

.reset-button {

margin-top: 5px;

padding: 15px 25px;

background-color: #00BFFF; /\* Botón azul \*/

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

}

.reset-button:hover {

background-color: #1e90ff;

}

* Estilo del botón que reinicia el juego.
  + margin-top: Agrega espacio encima del botón.
  + padding: Define el espacio interno del botón.
  + background-color: Establece el color del botón.
  + border: Elimina el borde predeterminado.
  + cursor: pointer: Cambia el cursor al pasar sobre el botón.
  + En :hover, cambia el color del botón al pasar el mouse sobre él.

.music-controls {

display: flex;

align-items: center;

justify-content: center;

margin-top: 20px;

}

#play-pause-btn {

background-color: #00BFFF;

color: white;

border: none;

border-radius: 5px;

padding: 10px;

font-size: 1.5em;

cursor: pointer;

position: absolute; /\* Para permitir posicionarlo en cualquier parte \*/

top: 10px; /\* Ajusta según sea necesario \*/

left: 10px; /\* Ajusta según sea necesario \*/

}

#play-pause-btn:hover {

background-color: #1e90ff;

}

#volume-control {

margin-left: 10px;

width: 150px;

position: absolute; /\* Para permitir posicionarlo en cualquier parte \*/

top: 10px; /\* Ajusta según sea necesario \*/

left: 10px; /\* Ajusta según sea necesario \*/

opacity: 0;

}

* Estilos para los controles de música, incluyendo el botón de reproducción/pausa y el control de volumen.
  + .music-controls: Usa Flexbox para centrar los controles.
  + #play-pause-btn: Estilo del botón de reproducción/pausa, similar al botón de reinicio.
  + `#volume 4o mini

**Código JavaScript**

El archivo JavaScript contiene la lógica del juego de memoria. A continuación, se describe el código y su funcionalidad.

**Evento** DOMContentLoaded

document.addEventListener("DOMContentLoaded", () => {

// Todo el código para controles de música y juego de memoria se encuentra dentro de este bloque.

});

* Espera a que todo el contenido HTML de la página se haya cargado antes de ejecutar el resto del código JavaScript, asegurando que todos los elementos necesarios estén disponibles.

**Controles de Música**

* **Elementos**:
  + playPauseBtn: Botón para reproducir/pausar la música.
  + volumeControl: Control deslizante para ajustar el volumen.
  + backgroundMusic: Elemento de audio que contiene la música de fondo.
* **Funcionalidad**:
  + **Botón Reproducir/Pausar**:

playPauseBtn.addEventListener('click', () => {

if (backgroundMusic.paused) {

backgroundMusic.play();

playPauseBtn.textContent = '⏸️'; // Cambia a ícono de pausa

} else {

backgroundMusic.pause();

playPauseBtn.textContent = '🎵'; // Cambia a ícono de música

}

});

* + - Reproduce o pausa la música de fondo y cambia el ícono del botón de acuerdo al estado (reproduciendo o en pausa).
  + **Control de Volumen**:

volumeControl.addEventListener('input', () => {

backgroundMusic.volume = volumeControl.value;

});

* + - Cambia el volumen de la música de fondo según el valor del control deslizante.
  + **Inicio Automático**:

backgroundMusic.volume = 1;

backgroundMusic.play();

* + - Configura un volumen inicial del 100% y reproduce la música automáticamente.

3. **Juego de Memoria**

* **Variables Importantes**:
  + images: Array que contiene las rutas de las imágenes para las tarjetas.
  + cardArray: Contiene una copia duplicada de las imágenes (para crear pares) y las mezcla.
  + gameBoard, statusMessage, resetButton: Referencias a los elementos HTML donde se muestra el tablero de juego, el mensaje de estado y el botón de reinicio.
  + firstCard, secondCard: Almacenan las cartas seleccionadas para verificar si forman un par.
  + lockBoard: Bloquea el tablero mientras se revisan las cartas, evitando clics adicionales.
  + matches: Lleva la cuenta de los pares encontrados.
  + timeLeft y timerInterval: Controlan el temporizador del juego.
* **Funciones**:
  + **createBoard**:

function createBoard() { /\* código para crear el tablero de juego \*/ }

* + - Inicializa el tablero de juego creando cartas a partir del cardArray y las añade al gameBoard. También inicia el temporizador.
  + **flipCard**:

function flipCard() { /\* código para voltear la carta \*/ }

* + - Voltea una carta seleccionada y, si ya hay otra seleccionada, llama a checkForMatch para verificar si coinciden.
  + **checkForMatch**:

function checkForMatch() { /\* código para verificar si las cartas coinciden \*/ }

* + - Compara las dos cartas seleccionadas. Si son iguales, llama a disableCards para desactivarlas; si no, llama a unflipCards para devolverlas a su estado inicial.
  + **disableCards**:

function disableCards() { /\* código para desactivar cartas coincidentes \*/ }

* + - Desactiva las cartas que han coincidido, evitando que puedan seleccionarse nuevamente. Si se encuentran todos los pares, detiene el temporizador y muestra un mensaje de victoria.
  + **unflipCards**:

function unflipCards() { /\* código para desvoltear cartas no coincidentes \*/ }

* + - Desvuelve las cartas que no coinciden a su estado original después de un breve tiempo para que el jugador pueda verlas antes de ocultarlas.
  + **resetBoard**:

function resetBoard() { /\* código para reiniciar variables de selección de cartas \*/ }

* + - Reinicia las variables firstCard y secondCard para que el jugador pueda seleccionar nuevas cartas.
  + **resetGame**:

function resetGame() { /\* código para reiniciar el juego completo \*/ }

* + - Reinicia el juego barajando nuevamente las cartas, restableciendo el temporizador y el mensaje de estado.
  + **shuffle**:

function shuffle(array) { /\* código para mezclar el array de cartas \*/ }

* + - Mezcla aleatoriamente el array de cartas usando Math.random() para obtener un orden diferente cada vez.
  + **startTimer**:

function startTimer() { /\* código para iniciar el temporizador \*/ }

* + - Inicia un temporizador de cuenta regresiva de 10 minutos, actualizando el tiempo restante en pantalla cada segundo. Si el tiempo se agota, muestra un mensaje de "Se acabó el tiempo" y bloquea el tablero.
  + **showTemporaryMessage**:

function showTemporaryMessage(message) { /\* código para mostrar mensaje temporal \*/ }

* + - Muestra un mensaje temporal en la interfaz durante un corto tiempo, útil para indicar al jugador si una selección fue incorrecta.
* **Evento de Reinicio**:

resetButton.addEventListener('click', resetGame);

* + Llama a resetGame al hacer clic en el botón de reinicio, reestableciendo el juego.

4. **Botón de Salida**

document.getElementById('exit-button').addEventListener('click', function() {

window.location.href = '/Plays/index.html';

});

* Al hacer clic en el botón de salida, el navegador redirige a la página principal (/Plays/index.html).

**Conclusión**

Este juego de memoria está diseñado para entretener y desafiar a los usuarios mientras ponen a prueba su capacidad de recordar y emparejar imágenes relacionadas con herramientas informáticas. Está implementado en un entorno web utilizando tecnologías como HTML, CSS y JavaScript, proporcionando una experiencia interactiva y visualmente atractiva.