

PRÁCTICA. En busca de los Pokémon perdidos

La empresa **VegaGame** nos ha solicitado desarrollar un juego interactivo en el que el jugador compita contra el ordenador, la descripción completa se encuentra a continuación:

Descripción del juego

1. **Tablero:**
 - El tablero será de **4x4** (16 casillas).
 - Todas las casillas estarán inicialmente ocultas.
 - **5 Pokémon** estarán escondidos aleatoriamente debajo de 5 casillas.
2. **Turnos por rondas:**
 - El jugador **siempre inicia** la ronda, eligiendo una casilla.
 - Si no encuentra un Pokémon, el turno pasa al ordenador.
 - Cada vez que una casilla se descubre:
 - Si tiene un Pokémon, se suma un punto y queda marcada.
 - Si no tiene un Pokémon, también queda marcada y no puede volver a seleccionarse.
3. **Condición de victoria:**
 - Gana el primer jugador (usuario u ordenador) que encuentre **3 Pokémon**.
 - El juego termina inmediatamente cuando esto sucede.

Requerimientos funcionales adicionales

1. **Botón para ganar un turno extra:**
 - Al seleccionar una casilla, se activará un botón que permite al usuario intentar ganar un turno adicional.
 - Este botón funcionará como un **dado**:
 - Generará un número aleatorio entre 1 y 6.
 - Si el resultado es **6**, el usuario podrá seleccionar otra casilla antes de pasar el turno al ordenador.
2. **Botón para reiniciar el juego:**
 - Incluir un botón que permita **resetear el estado completo del juego** sin necesidad de recargar la página (F5).
 - El botón reiniciará:
 - El tablero (casillas ocultas).
 - Los contadores de puntos y tiradas.
 - Los turnos.
3. **Histórico de jugadores:**
 - Al finalizar el juego:
 - Si **gana el usuario**, se le pedirá que introduzca su nombre.
 - Si **gana el ordenador**, se registrará automáticamente el nombre del ganador como "CPU".
 - Mostrar un **histórico** de jugadores con el número de tiradas realizadas por cada uno. Ejemplo:

HISTÓRICO DEL VEGAGAME:

Fernando → 4 tiradas

Víctor → 5 tiradas
Fran → 3 tiradas
CPU → 8 tiradas

- El histórico debe actualizarse dinámicamente tras cada partida.

Notas técnicas y de implementación

1. Uso de eventos (addEventListener):

- **Clic en casillas:** Escuchar los clics en el tablero para descubrir casillas y alternar turnos.
- **Botón de turno extra:** Asociar el evento para lanzar el "dado" que determine si el jugador gana una tirada extra.
- **Botón de reinicio:** Escuchar el evento de clic para restablecer el juego completo.

2. Creación dinámica de nodos:

- Generar el tablero (casillas) y los botones de forma dinámica en el DOM.
- Crear y actualizar el historial de jugadores en pantalla al finalizar cada partida.

3. Turnos y lógica de puntuación:

- Alternar turnos de manera ordenada: jugador → CPU → jugador, y así sucesivamente.
- Contar las tiradas realizadas por cada jugador y mostrarlas al final.

4. Condición de fin del juego:

- Cuando un jugador (usuario o CPU) alcance 3 puntos:
 - Detener la interacción con el tablero.
 - Mostrar al ganador y actualizar el historial.

Estructura de la interfaz

HTML

- **Tablero:**
 - Casillas ocultas (4x4).
- **Botones:**
 - Botón para ganar un turno extra.
 - Botón para reiniciar el juego.
- **Área de información:**
 - Puntos de cada jugador.
 - Contador de tiradas.
 - Histórico de jugadores.

JavaScript

1. Inicio del juego:

- Crear dinámicamente el tablero (16 casillas).
- Colocar aleatoriamente los 5 Pokémon debajo de casillas ocultas.
- Inicializar las puntuaciones y tiradas.

2. Turnos:

- Alternar entre jugador y CPU de forma automática.
- Marcar casillas descubiertas y actualizar el puntaje.

3. **Botón de turno extra:**

- Asociar el evento de clic para generar un número aleatorio y, si es 6, otorgar un turno adicional al jugador.

4. **Reinicio:**

- Resetear el tablero, puntuaciones, tiradas y turnos.

5. **Histórico:**

- Al finalizar:
 - Pedir el nombre del jugador si gana y registrar sus tiradas.
 - Si gana la CPU, registrar automáticamente "CPU" como ganador.

Ejemplo de la interfaz visual.



Ilustración 2.- Tablero inicial



Ilustración 1.- Tablero final