Desarrollo de un Script para Entrenamiento de Pokémon

Índice

Enunciado

Deberás implementar el archivo `script.ts` para una aplicación web que permite entrenar Pokémon. El objetivo es gestionar los datos de los Pokémon y permitir que el usuario seleccione uno para entrenarlo, incrementando una de sus habilidades (ataque, defensa, velocidad, hp o ataque especial). La aplicación ya cuenta con una interfaz HTML, un servidor backend para gestionar los datos y un archivo JSON que contiene los datos de los Pokémon.

Instala el proyecto: **npm install**
Compilar fichero .ts: **npx tsc**
Ejecuta el servidor con: **npm run start**
Visualiza la aplicación accediendo a **http://localhost:5500/**

Estructura del proyecto

- **index.html:** Contiene la estructura básica de la aplicación web, que incluye **un selector de Pokémon** (`<select id="pokemon-list"></select>`) y un **botón para iniciar el entrenamiento** (`<button id="start-training">Start Training</button>`).
- **pokemon_data.json:** Archivo que contiene la información de los Pokémon, como su ataque, defensa, velocidad, etc.
- **server.js:** Servidor en Node.js que gestiona la obtención y actualización de los datos.
- **script.ts:** Archivo TypeScript que deberás implementar.

Tarea a realizar

Tu tarea es desarrollar el archivo `script.ts` utilizando TypeScript y jQuery. Deberás implementar clases, interfaces y herencia donde sea necesario, para hacer el código modular y organizado.

Detalles específicos de la tarea

1. **Clase `Pokemon`**
   1.1 Implementa una **clase `Pokemon`** que represente a un Pokémon. Esta clase debe implementar una **interfaz `IPokemon`** con los atributos: `id`, `name`, `attack`, `defense`, `speed`, `hp`, y `special_attack`.

1.2 La clase debe tener un **método `train()`** que permita entrenar una habilidad del Pokémon. Este método recibirá el tipo de habilidad (attack, defense, speed, etc) a entrenar y un porcentaje para incrementar su valor (pongamos un 10% sobre el valor que tenga).
1.3 El valor actualizado debe ser redondeado a un valor entero.

2. Clase `PokemonTrainer`

2.1 Implementa una clase `PokemonTrainer` que se encargue de gestionar el entrenamiento de los Pokémon.
2.2 Esta clase deberá manejar la carga de los datos desde `pokemon_data.json` (a través del servidor) y llenar la lista desplegable en el HTML (`<select id="pokemon-list"></select>`).
2.3 La clase también debe tener un método para iniciar el entrenamiento de un Pokémon seleccionado por el usuario.
2.4 Debes simular un tiempo de entrenamiento aleatorio entre 1 y 10 segundos antes de incrementar la habilidad seleccionada.

3. Entrenamiento

Cuando el entrenamiento finalice, deberás mostrar un mensaje en la pantalla indicando:
3.1 El resultado del entrenamiento.
3.2 Y preguntar al usuario si desea actualizar los puntos de la habilidad entrenada.
3.3 Si el usuario acepta, se debe actualizar el archivo JSON en el servidor a través de una solicitud `POST`.

4. Interacción con el HTML usando jQuery

4.1 Usa jQuery para interactuar con los elementos del HTML. Asegúrate de que los Pokémon se carguen en el selector (`<select id="pokemon-list"></select>`) y que el botón de entrenamiento (`<button id="start-training">Start Training</button>`) inicie el proceso de entrenamiento.
4.2 El entrenamiento debe mostrar un mensaje mientras está en proceso (`Entrenando a [nombre] en [habilidad] durante [tiempo] segundos.`) y al finalizar (`El Pokémon [nombre] ha subido de nivel en [habilidad] con [nuevos puntos]. ¿Desea actualizarlos?`).

Al pulsar aceptar se actualizará los puntos en el json (haz uso del método ya creado en el proyecto facilitado). Utiliza jQuery para manipular el DOM y mostrar estos mensajes.

5. Optimización del Código

5.1 Utiliza TypeScript junto con jQuery para reducir la complejidad del manejo del DOM.
5.2 Asegúrate de que las clases y métodos estén bien organizados y sigan las buenas prácticas de programación orientada a objetos.