

CICLO 2

Javascript – TEST 2

Hacer una página web que simule una batalla entre pokemones.

La interfaz gráfica de la página debería ser similar a la siguiente:

INTERFAZ:



DINÁMICA DE JUEGO:

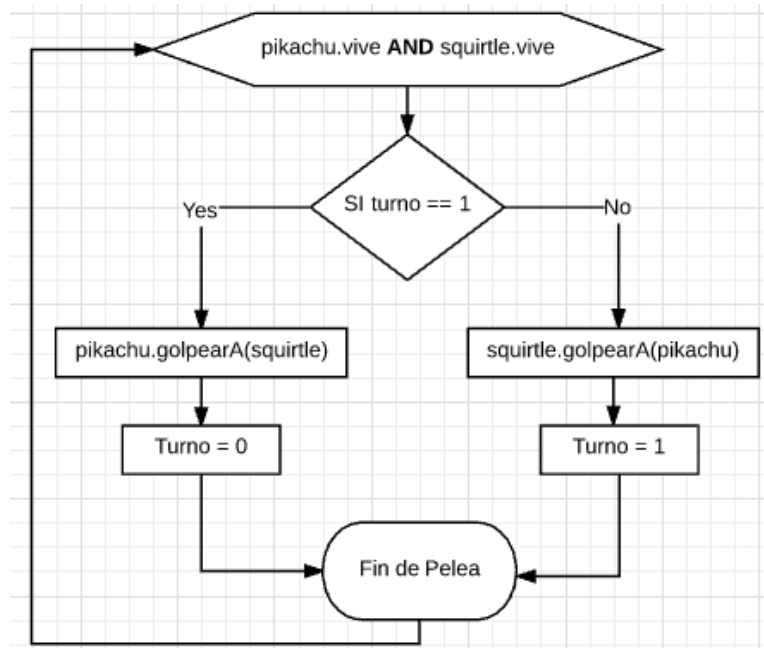
- 🌟 La página empieza sin ninguno de los dos pokemones
- 🌟 Hay dos botones: "MI Pokémon" y "Figth"
- 🌟 El usuario hace clic en el botón "MI Pokémon" y le aparece un Pokémon al azar. El usuario puede presionar el botón cuantas veces quiera, hasta que elija el Pokémon de preferencia. Junto con la imagen del Pokémon debe aparecer las características de este.
- 🌟 Una vez que el usuario escoja su Pokémon puede presionar el botón Figth (este botón no funciona hasta que el usuario haya escogido un Pokémon. Si el usuario no ha escogido uno, entonces muéstrele un mensaje de que debe escoger un Pokémon para poder empezar la pelea) y el sistema escogerá al azar un Pokémon que quede después de 7 segundos de haber presionado el botón "Figth" (durante estos 5 segundos, deben aparecer las imágenes de diferentes pokemones al azar).

Una vez pasado los 5 segundos y haber escogido el Pokémon contrincante, debe fijarse la imagen y las características de este y empieza el algoritmo de batalla.

- El usuario puede mantener su Pokémon durante las batallas que desee sin necesidad de escoger uno nuevo cada vez.

ALGORITMO DE BATALLA

El algoritmo de batalla se define de forma general con el siguiente diagrama de flujo



- La propiedad “vive” de cada Pokémon depende de que la propiedad “HP” sea mayor que cero.
- El turno varía para que se simule quien debe atacar cada vez. Esto se calcula dependiendo de la velocidad de cada Pokémon.

$$var_turno = | MI_Poke.velocidad - CPU_Poke.Velocidad |$$

Si $MI_Poke > CPU_Poke$ entonces MI_Poke atacará var_turno veces, luego de lo cual le tocará el turno al CPU_Poke atacar por una sola vez.

Si no, entonces CPU_Poke atacará var_turno veces, luego de lo cual le tocará el turno al MI_Poke atacar por una sola vez.

- El impacto de cada golpe hará que la propiedad HP disminuya así:

$$HP\ restante = HP\ inicial \cdot e^{-k \cdot ataque \cdot \left(1 - \frac{defensa}{defensa + 100}\right) \cdot \left(1 + \frac{velocidad}{200}\right)}$$

- En cada turno se debe verificar cuál de los dos pokemones sigue vivo y cuando el valor de la propiedad HP de alguno de ellos sea menor o igual a cero, entonces se termina el juego
- Al final del juego se debe mostrar cual es el Pokémon Ganador y con cuanta vida (HP) queda cada uno.

- ✨ Si el usuario mantiene el Pokemon para una siguiente batalla, se recargar nuevamente sus propiedades originales con un incremento en HP de la propiedad `base_experience`.

Formula:

$$hp = \left(\frac{hp}{2}\right) * \left[1 + \left(\frac{base_experience}{100}\right)\right]$$

Por ejemplo, Pikachu empezó con un HP de 35 y si el usuario lo escoge para la siguiente pelea, su nuevo HP es:

$$hp = \frac{35}{2} * \left[1 + \left(\frac{112}{100}\right)\right], \text{ entonces en la siguiente batalla el hp es de 38 y si lo escoge una posterior batalla, el valor de HP seria de 40}$$

El HP incrementa ya que el Pokémon adquiere experiencia en cada batalla siguiente si es elegido con continuidad.

DATOS

Los datos de los puedes encontrar aquí:

- API: <https://pokeapi.co/api/v2/pokemon/{id-pokemon}>
- Rango de Pokemones: [1 – 806]
- Nombre – propiedad: `name`
- Imagen – propiedad: `sprites.other.official-artwork.front_default`
- [Hp, Attack, Defense, Speed] - Name - propiedad: `stats[k].stat.name`
- [Hp, Attack, Defense, Speed] - Valor - propiedad: `stats[k].base_stat`
- Base_experience – valor - propiedad: `base_experience`