



Diseño de interfaces web

Alberto Ramírez Salas
22/04/2022

Indice

1- Crear un desplegable para seleccionar el número de participantes.....	2
2- Crear un array con tantas posiciones como participantes se hayan seleccionado....	2
3- Todos los coches deben situarse uno debajo del otro con un "margin-left" de 0px....	2
4- Indicar un valor fijando una línea de meta.....	3
5- Se trata de ir incrementando el margen desplazando los coches hacia la derecha (con el método animate) hasta rebasar la línea de meta (valor indicado).....	3
6- Se crearán dos botones, "Iniciar" y "Reiniciar". Al cargar la página, solamente debe mostrarse el botón inicio, reiniciar debe estar oculto. Sin embargo, cada vez que pulsamos uno de ellos, desaparece y se muestra el otro.....	3
7- Al pulsar el botón inicio, los coches empezarán a avanzar. En cambio, si se pulsa el botón "Reiniciar" los coches tendrán que volver al puesto inicial.....	4
8- El avance de cada coche se realiza añadiendo de forma aleatoria valores entre 1 y 10.....	4
9- El primer coche que llegue al valor fijado como línea de meta, será el ganador.....	4
10- Mostrar una tabla con las posiciones de cada participante.....	5

CONTENIDO

1. Crear un **desplegable** para seleccionar el número de participantes.

El numero de participantes disponibles esta asociado a un input select con las opciones posibles.

```
<select name="cars" id="nCars">
  <option selected value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
  <option value="5">5</option>
  <option value="6">6</option>
  <option value="7">7</option>
  <option value="8">8</option>
  <option value="9">9</option>
</select>
```

El id esta asociado a un listener que al cambiar el valor actualiza los datos y pinta los coches participantes

```
params.nCars.addEventListener("change", () => {
  CarController.loadCars();
})
```

2. Crear un **array** con tantas posiciones como participantes se hayan seleccionado.

Al cambiar el valor del input de participantes se prepara una array con una instancia de las clase Car para controlar los coches

```
createCars() {
  for (let i = 1; i <= params.nCars.value; i++) {
    const car = new Car(i);
    this.cars = [...this.cars, car];
  }
},
```

La clase coche se encarga de almacenar los datos referentes a cada coche y a moverlos por la carretera.

3. Todos los coches deben situarse uno debajo del otro con un **"margin-left" de 0px**.

Los coches están en una etiqueta img dentro de un div, de manera que quedan unos debajo de otros y posicionados en la izquierda.

```
const carComponent = (car) => {  
  return `<div class="road">  
      
  </div>`;   
};
```

4. Indicar un valor fijando una línea de meta.

La línea de meta queda fijada con el 100% width del div de la carretera

5. Se trata de ir incrementando el margen desplazando los coches hacia la derecha (con el método **animate**) hasta rebasar la línea de meta (valor indicado).

Se crea un setInterval que se ira ejecutando la función de la clase Car de correr animando el margin-left hasta que los coches lleguen a la meta.

```
run() {
  if (this.distance < 100) {
    const dist = getRandom();
    document.querySelector(`#car_${this.id}`).animate(
      {
        marginLeft: [`${this.distance}%`,
`${(this.distance + dist) < 100 ? this.distance + dist : 100}%`],
      },
      {
        duration: params.speed.value * 250,
        easing: "linear",
        iterations: 1,
        fill: "both"
      }
    );
    this.distance += dist;
    if (this.distance >= 100) this.finish();
  }
}
```

6. Se crearán **dos botones**, "Iniciar" y "Reiniciar". Al cargar la página, solamente debe mostrarse el botón inicio, reiniciar debe estar oculto. Sin embargo, cada vez que pulsamos uno de ellos, desaparece y se muestra el otro.

El botón de reiniciar al principio esta oculto con un display none y solo esta visible el de Iniciar. Al hacer clic en cualquiera de los botones se cambia el display de cada uno para que se oculte el que estaba visible y se oculte el visible

```
params.start.addEventListener("click", () => {
  params.start.style.display = 'none';
  params.reset.style.display = 'block';
  params.nCars.disabled = true;
  params.speed.disabled = true;
  startRace();
});
params.reset.addEventListener("click", () => {
  CarController.loadCars();
  params.start.style.display = 'block';
  params.reset.style.display = 'none';
  params.nCars.disabled = false;
  params.speed.disabled = false;
  clearInterval(params.interval);
  params.table.innerHTML = '';
});
```

7. Al pulsar el botón inicio, los coches empezarán a avanzar. En cambio, si se pulsa el botón "Reiniciar" los coches tendrán que volver al puesto inicial.

Al pulsar el botón de inicio se activa el setInterval que ejecuta el movimiento de cada coche

```
const startRace = () => {
  params.interval = setInterval(() => {
    CarController.cars.forEach((car) => {
      car.run();
    })
    if (params.positions.length == params.nCars.value) {
      clearInterval(params.interval);
      params.table.innerHTML = tableComponent();
    }
  }, params.speed.value * 250)
}
```

Al pulsar el botón de stop se para el setInterval y se vuelven a pintar los coches en el inicio de la carrera

```
params.reset.addEventListener("click", () => {
  CarController.loadCars();
  params.start.style.display = 'block';
  params.reset.style.display = 'none';
  params.nCars.disabled = false;
  params.speed.disabled = false;
  clearInterval(params.interval);
  params.table.innerHTML = '';
});
```

8. El avance de cada coche se realiza añadiendo de forma aleatoria valores entre 1 y 10.

Para ir sumando distancia a cada coche se usa la función para obtener un numero aleatorio y se multiplica por 10 para obtener el rango máximo, se aplica la función floor para redondear y quitar los decimales

```
const getRandom = () => (Math.floor(Math.random() * 10) + 1);
```

9. El primer coche que llegue al valor fijado como línea de meta, será el ganador.

A medida que llegan los coches a la meta se van añadiendo los coches a la array de posiciones de manera que queda ordenada por orden de llegada.

```
finish() {
  params.positions.push(this.id);
  this.position = params.positions.length;
}
```

10. Mostrar una tabla con las posiciones de cada participante.

Al llegar todos los coches a la meta se pinta la tabla con las posición en la que ha quedado cada coche.

```
const tableComponent = () => {
  let tableHtml = `<div class="container-table">
<div class="table">`;
  const carSort = sortByKeyDesc(CarController.cars, "position");
  carSort.forEach((row) => {
    tableHtml += `
      <div class="row ${getClassPosition(row.position)}">
        <div class="column"><img src='${row.image}' /></div>
        <div class="column
position">${positionComponent(row.position)}</div>
      </div>
    `;
  });
  tableHtml += `</div></div>`;
  return tableHtml;
};
```

CONTRAPORTADA

Actividad de desarrollo (UF2) de Diseño de interfaces web.