

PAC Desarrollo

Iván Sánchez Sáez

Índice

INTRODUCCIÓN.....	2
DESARROLLO	2
PRUEBAS	7
CONCLUSIÓN.....	12
REFERENCIAS BIBLIOGRÁFICAS.....	13

INTRODUCCIÓN

El presente documento contiene la explicación del ejercicio de la PAC de Desarrollo donde se ha utilizado JavaScript para dar la funcionalidad necesaria a una web con la finalidad de desarrollar un juego de papel piedra y tijera.

Para una mayor comprensión, se van a introducir capturas de pantalla directamente del código.

DESARROLLO

En la parte inicial del documento JavaScript se han declarado una serie de variables necesarias para el correcto funcionamiento, como son las partidas, el jugador y el seguimiento de las partidas totales y actuales:

```
1 // Este array no se puede modificar,  
2 var posibilidades = ["piedra", "papel", "tijera"];  
3 //  
4  
5 var partidas = 0;  
6 var jugador = "";  
7 var partidasTotales = document.getElementById("total");  
8 var partidaActual = document.getElementById("actual");
```

En las siguientes líneas se ha añadido una función para inicializar los eventos que incluye la variable del botón de jugar, de tal modo que se ha establecido una escucha al evento click sobre el botón:

```

11 // FUNCION INICIALIZAR EVENTOS
12 function inicializarEventos() {
13     var botonJugar = document.getElementsByTagName("button")[0];
14     console.log(botonJugar);
15     botonJugar.addEventListener("click", jugar);
16 }
17 document.addEventListener("DOMContentLoaded", inicializarEventos);

```

Seguidamente se ha desarrollado la función de jugar para que haga las comprobaciones de los campos jugador y número de partidas para que en el caso de que sean incorrectas se inserte la clase CSS fondoRojo. Además esta función también modifica en el HTML el número de partidas a jugar:

```

20 //FUNCION DE JUGAR
21 function jugar(){
22     jugador = document.getElementsByName("nombre")[0]; // COMPROBACIÓN DEL COMIENZO DE LA PARTIDA
23     console.log(jugador.value);
24
25     partidas = document.getElementsByName("partidas")[0];
26     console.log(partidas.value);
27
28     if(jugador.value.length <= 2 || isNaN(jugador.value.charAt(0))){
29         console.log("nombre no valido");
30         jugador.classList.add("fondoRojo");
31     }
32
33     else {
34         jugador.classList.remove("fondoRojo");
35     }
36
37     if(partidas.value <= 0){
38         console.log("partida no valida");
39         partidas.classList.add("fondoRojo");
40     }
41
42     else {
43         partidas.classList.remove("fondoRojo");
44     }
45
46     partidasTotales.textContent = partidas.value;
47 }
48

```

Para insertar las imágenes que tiene el jugador para hacer su elección y mostrar la clase CSS seleccionado o noSeleccionado se ha tenido que recurrir a un document.querySelectorAll y una función respectivamente.

Ha sido necesario cambiar el source de las imágenes mediante un bucle for que utilizara el array posibilidades, ya que este contenía parte de los nombres de los archivos png.

Por otro lado, también se ha establecido un bucle for para recorrer toda la longitud de imágenes y asignarle primero la clase de noSeleccionado a todas ellas, de manera que solo se le añada la clase seleccionado a la imagen que el usuario seleccione:

```
52 // IMAGENES DEL JUGADOR
53 var imagenes = document.querySelectorAll('#jugador img');
54 for (var i = 0; i < imagenes.length; i++) {
55     imagenes[i].src = 'img/' + posibilidades[i] + 'Jugador' + '.png'; // AÑADIR IMÁGENES
56     imagenes[i].addEventListener('click', function seleccionada() { // EVENTO CLICK ASOCIADO A LA FUNCIÓN SELECCIONADA
57
58         for (var j = 0; j < imagenes.length; j++) { // les pongo a todas la clase no seleccionadas
59             imagenes[j].classList.remove('seleccionado');
60             imagenes[j].classList.add('noSeleccionado');
61         }
62
63         this.classList.remove('noSeleccionado'); //les pongo a la que se selecciona, la clase seleccionada
64         this.classList.add('seleccionado');
65         var indice = Array.from(imagenes).indexOf(this);
66         tiradaJugador = posibilidades[indice];
67         console.log(tiradaJugador);
68     });
69 }
```

Tras esto, hay varias líneas de código que le dan una funcionalidad al botón ¡YA!. En primer lugar se establece la tirada de la máquina generando un índice aleatorio entre 0 y 2 mediante `Math.floor Math.random`. En segundo lugar se va sumando 1 a la variable de número de partidas actuales para que el usuario sepa cuántas le quedan por jugar. De esta manera, cuando el número de partidas jugadas sea igual al número de partidas totales, se deshabilitará el botón ¡YA! para que no pueda jugar más.

Seguidamente, para mostrar por pantalla la imagen que corresponde con la tirada de la máquina se vuelve a cambiar el source con el array posibilidades pero esta vez la posición del array la determinará la variable `indiceAleatorio`.

```

75 document.addEventListener("DOMContentLoaded", function() {
76     // Seleccionar el botón "¡YA!" mediante la etiqueta h2
77     var botonYa = document.querySelector('h2 button');
78
79     // Agregar un evento de clic al botón
80     botonYa.addEventListener('click', function(){
81         // Obtener la imagen id "maquina"
82         var imagenMaquina = document.querySelector('#maquina img');
83
84         // Generar un número aleatorio entre 0 y 2 para seleccionar una imagen aleatoria
85         var indiceAleatorio = Math.floor(Math.random() * 3);
86
87         // Determinar la opción de la máquina según el índice aleatorio
88         var tiradaMaquina = posibilidades[indiceAleatorio];
89         console.log(tiradaMaquina);
90
91         // INFORMAR DE CUÁNTAS PARTIDAS VAN QUEDANDO
92         var partidaActualNumero = parseInt(partidaActual.textContent) + 1;
93         partidaActual.textContent = partidaActualNumero;
94
95         if (partidaActualNumero >= parseInt(partidasTotales.textContent)) {
96             botonYa.disabled = true; //DESHABILITAR EL BOTON YA PARA QUE NO PUEDA JUGAR MAS
97         }
98
99         var resultadoHistorial = resultado(tiradaJugador, tiradaMaquina);
100         console.log(resultadoHistorial);
101         // Cambiar el src de la imagen según el índice aleatorio
102         imagenMaquina.src = 'img/' + posibilidades[indiceAleatorio] + "Ordenador" + '.png';
103     });

```

El código continúa para mostrar en el historial si has ganado o perdido en la lista HTML, para lo que ha sido necesario determinar una función que muestre el resultado siguiendo la lógica booleana del juego de piedra papel o tijera:

```

104 // Mostrar si has ganado o perdido en el historial
105 var historial = document.getElementById("historial");
106 var li = document.createElement("li");
107 li.textContent = resultadoHistorial;
108 historial.appendChild(li);
109
110 function resultado(){
111     if (tiradaJugador === tiradaMaquina) {
112         return "Empate";
113     }
114     else if (
115         (tiradaJugador === "piedra" & tiradaMaquina === "tijera") ||
116         (tiradaJugador === "papel" & tiradaMaquina === "piedra") ||
117         (tiradaJugador === "tijera" & tiradaMaquina === "papel")
118     ){
119         return "Ganas";
120     }
121     else {
122         return "Pierdes";
123     }
124 };
125
126 });
127 });

```

Finalmente, para el botón RESET, que está dentro de una etiqueta div en el HTML, se reinician todos los valores necesarios para la partida, como es el número de partidas actuales y totales, el nombre del jugador, las imágenes y volver a habilitar el botón de ¡YA!

```

132 document.addEventListener("DOMContentLoaded", function() {
133     var botonReset = document.querySelector("div button"); // Seleccionar el botón RESET
134
135     botonReset.addEventListener("click", function() { // Agregar un evento de clic al botón R
136
137     botonReset.addEventListener('click', () => {
138         historial.innerHTML = '';
139     });
140
141     // Reiniciar el valor del número de partidas jugadas
142     var partidaActual = document.getElementById("actual");
143     partidaActual.textContent = "0";
144
145     // Reiniciar el valor del input de partidas
146     var partidasTotales = document.getElementById("total");
147     partidasTotales.textContent = "0";
148
149     // Reiniciar el valor del input de nombre del jugador
150     var jugador = document.getElementsByName("nombre")[0];
151     jugador.value = "";
152     var partidas = document.getElementsByName("partidas")[0];
153     partidas.value = 0;
154
155     // Reiniciar el estado de las imágenes del jugador
156     var imagenes = document.querySelectorAll('#jugador img');
157     for (var i = 0; i < imagenes.length; i++) {
158         imagenes[i].classList.remove('seleccionado');
159         imagenes[i].classList.add('noSeleccionado');
160     }
161
162     // Reiniciar el estado de la imagen de la máquina
163     var imagenMaquina = document.querySelector('#maquina img');
164     imagenMaquina.src = 'img/defecto.png';
165
166     // Habilitar el botón ;YA!
167     var botonYa = document.querySelector('h2 button');
168     botonYa.disabled = false;
169
170 });

```

Activar Windows

PRUEBAS

Se expone a continuación con imágenes las pruebas numeradas que requiere el ejercicio.

1. Introducción de usuario con datos no válidos

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

1iv

2. Introducción de cantidad de partidas con datos no válidos.

¿Cuántas partidas quieres jugar?

-1

3. Acceso a la aplicación don datos válidos.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Jugando la partida 0 de 3.



4. Seleccionar una de las opciones y jugar al menos 5 partidas

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Jugando la partida 5 de 5.



Piedra, papel o tijera....



Historial de partidas

- Pierdes
- Ganas
- Empate
- Pierdes
- Pierdes

5. Pulsar el botón RESET y jugar al menos 3 partidas

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

¡JUGAR!

Jugando la partida 0 de 0.



Piedra, papel o tijera....



Historial de partidas

- Pierdes
- Ganas
- Empate
- Pierdes
- Pierdes

RESET

CONCLUSIÓN

A modo de conclusión, resaltar que las funcionalidades se han desarrollado según los requisitos del ejercicio, apreciándose, por ejemplo, en la comprobación del nombre o del número de partidas a jugar, que se limita a que el nombre no empiece por número y sea igual o mayor a 3 caracteres o a que el número de partidas no sea negativo.

Además, es importante saber que se ha acudido a las fuentes que se establecen a continuación como referencias bibliográficas durante todo el desarrollo de la actividad.

REFERENCIAS BIBLIOGRÁFICAS

W3Schools. (s.f.). JavaScript Tutorial. Recuperado de <https://www.w3schools.com/js/>

Libro de la asignatura.