Piedra Papel Tijera

Ejercicio Práctico

EmDevApps

Índice

1. Descripción de las funciones implementadas en el código.

Inicialización y Preparación

Se inicializan las variables globales necesarias para el funcionamiento del juego, como el total de partidas, el número de partidas jugadas, el nombre del jugador y el historial de partidas. Además, se obtienen referencias a los elementos del DOM que serán utilizados, como las imágenes de los jugadores, los botones y los campos de entrada.

```
// Este array no se puede modificar, es condición del enunciado de la Pac de desarrollo.
var posibilidades = ["piedra", "papel", "tijera"];


// Ponemos a la escucha una función anonima, del evento DOMContentLoaded, es decir cuando el DOM haya cargado en su totalidad.
document.addEventListener("DOMContentLoaded", function() {


  // Inicializamos variables globales
  var totalPartidas = 0;
  var partidasJugadas = 0;
  var nombreJugador = "";
  var historialPartidas = [];


  // A continuación, estamos almacenando en variables propias elementos del DOM, diferentes elementos del dom, a los que
  // accedemos mediante
  // document.querySelectorAll,  getElementById...  hay múltiples formas de acceder a elementos del DOM
  // mediante atributos id, name , class....  Dependiendo de si queremos almacenar 1 único elemento por id
  // o necesitamos una agrupación de elementos por class, o por etiqueta html. <img/>, <div>, <input/>

  var jugadorImgs = document.querySelectorAll("#jugador img");
  var maquinaImg = document.querySelector("#maquina img");
  var botonJugar = document.querySelector("button");
  var botonTirar = document.querySelector("h2 button");
  var spanActual = document.getElementById("actual");
  var spanTotal = document.getElementById("total");
  var nombreInput = document.querySelector("input[name='nombre']");
  var partidasInput = document.querySelector("input[name='partidas']");
  var historialUl = document.getElementById("historial");
  var botonReset = document.querySelector("div button");
```

Función para validar el nombre del jugador (validarNombre)

Esta función comprueba si el nombre ingresado por el jugador es válido. Para ser válido, el nombre debe tener más de tres caracteres y no puede comenzar con un número.

Función para validar el número de partidas (validarPartidas)

Esta función verifica si el número de partidas ingresado por el jugador es válido. Para ser válido, el número debe ser un entero mayor que cero.

```
// Función para comprobar si un nombre es válido
// La variable nombre ha sido inicializada al inicio.
// Tambien verificamos que no sea un numero, queremos trabajar con string.


function validarNombre(nombre) {

    return nombre.length > 3 && isNaN(nombre[0]);
}


//Se especifica en el enunciado de la Pac que debe ser un numero superior a 0 para que sea válido.
// Función para comprobar si el número de partidas es válido

function validarPartidas(partidas) {
    return parseInt(partidas) > 0;
}
```

Función para actualizar el historial de partidas (actualizarHistorial)

Esta función recibe como parámetro el resultado de una partida (por ejemplo, "gana máquina", "empate", "gana + nombreJugador") y agrega un nuevo elemento de lista (li) al historial de partidas en el DOM con este resultado.

```
// Función para actualizar el historial de partidas
// Recibimos por parámetro un resultado, el cual será por ejemplo:  "gana máquina" , "empate", "gana +"nombreJugador"
// Lo que estamos haciendo es es crear un elemento li, le estamos insertando el valor de ese resultado, Y estamos
//insertando como hijo de ese elemento del dom el contenido string del resltado.


function actualizarHistorial(resultado) {
    var li = document.createElement("li");
    li.textContent = resultado;
    historialUl.appendChild(li);
}
```

Función para resetear la partida (resetearPartida)

Esta función restablece todas las configuraciones de la partida a su estado inicial. Habilita los campos de entrada para el nombre y el número de partidas, reinicia los contadores de partidas jugadas, actualiza los marcadores en el DOM, restaura la imagen de la máquina a su estado por defecto y limpia el historial de partidas.

```
// Función para resetear la partida
//Debido a que hemos almacenado en variables de javascript algunos elementos del dom
//podemos acceder directamente a sus propiedades mediante la nomenclatura del punto (.)
//Es decir,  introducimos nombre de la (variable + . + nombre propiedad) y directamente le asignamos un valor (true, false, "0"...)
// Para resetear la partida por ejemplo se resetean algunas propiedades del conjunto de elementos del dom


function resetearPartida() {


    nombreInput.disabled = false;
    partidasInput.disabled = false;
    spanActual.textContent = "0";
    partidasJugadas = 0;
    spanTotal.textContent = totalPartidas;
    maquinaImg.src = "img/defecto.png";
    historialPartidas = [];
    historialUl.innerHTML = "";
}
```

Función principal para jugar una partida (jugarPartida)

Esta función se ejecuta cada vez que el jugador hace clic en el botón "TIRAR". Primero, genera una elección aleatoria para la máquina y determina la opción seleccionada por el jugador. Luego, calcula el resultado de la partida comparando las opciones del jugador y de la máquina. Actualiza el historial de partidas, los contadores de partidas jugadas y el marcador en el DOM. Finalmente, muestra la opción elegida por la máquina y verifica si se alcanzó el límite de partidas.

```
function jugarPartida() {

    var resultado;
    var opcionJugador;
    var opcionMaquina = posibilidades[Math.floor(Math.random() * posibilidades.length)];

    // Obtener opción seleccionada por el jugador
    // De la lista de imagenes, la imagen seleccionada tiene un índice [i], este índice es utilizado para saber en función dela imagen
    // seleccionada, saber qué posibilidad del array de posibilidades se debe asignar. var posibilidades = ["piedra", "papel", "tijera"];
    // En funcionamiento sería algo así:  El jugador selecciona una imagen, y la imagen seleccionada se relaciona con una posibilidad
    // de las 3 del array.


    for (var i = 0; i < jugadorImgs.length; i++) {

        if (jugadorImgs[i].classList.contains("seleccionado")) {
            opcionJugador = posibilidades[i];
            break;
        }
    }

    // Comprobar resultado de la partida
    // Vamos a comprobar la posibilidad obtenida por la maquina del array: var posibilidades = ["piedra", "papel", "tijera"]
    // comparada por la posibilidad obtenida por el jugador  var posibilidades = ["piedra", "papel", "tijera"]


    if (opcionJugador === opcionMaquina) {

        resultado = "Empate";

        //a continuación vamos a "jugar con los indices"  del array posibilidades. Sabemos que piedra gana a tijera, sabemos que
        //papel gana a piedra  y sabemos que tijera gana a papel.

        // conociendo el orden de nuestro array var posibilidades = ["piedra", "papel", "tijera"];
        //Podemos realizar el siguiente análisis.

        // la posición de piedra es  posibilidades[0]
        // la posición de papel es  posibilidades[1]
        // la posición de tijera es  posibilidades[2]
```

```
    //Ponemos un ejemplo
    // si el jugador saca tijera: el indice de piedra es 2
    // si la máquina saca papel: el indice de papel es 1
    // Por lo tanto   2-1 = 1
    // Luego ese 1 le sumamos la posibilidades, que son 3.
    // tenemos como resultado 4
    // Si hacemos la operación 4%3    (módulo de 4 dividido 3)
    // Y se obtiene como resultado 1, debido a que 1 === 1 ----->  gana Jugador.

    //Pongamos otro ejemplo
    // si el jugador saca piedra: el indice de piedra es 0
    // si la máquina saca papel: el indice de papel es 1
    // Por lo tanto   0-1 = -1
    // Luego ese -1 le sumamos la posibilidades, que son 3.
    // tenemos como resultado 2
    // Si hacemos la operación 2%3    (módulo de 2 dividido 3)   cuyo resultado es diferente de 1.
    // (módulo de 2 dividido 3) es diferente de 1 ----->  gana máquina porque se sale de la condición y entra en el else.



} else if ((posibilidades.indexOf(opcionJugador) - posibilidades.indexOf(opcionMaquina) + posibilidades.length) %
 posibilidades.length === 1) {


    resultado = "Gana " + nombreJugador;

} else {
    resultado = "Gana la máquina";


}
```

Función actualizarHistorial();

```
// Actualizar historial y contadores
//En este momento la partida ha finalizado y tenemos que aplicar la función   actualizarHistorial(resultado);  pasándole
//por parámetro el resultado, (empate, gana maquina, gana+nombrejugador)
//Tambien tenemos en el DOM  spanActual,  vamos a su propiedad textContent para asignarle el valor del contador de partidas.



actualizarHistorial(resultado);


partidasJugadas++;
spanActual.textContent = partidasJugadas;
```

Bucle for

Este bucle realiza un procedimiento esencial, que es asignar la class (seleccionado o noSeleccionado, a jugadorImgs, para saber cual es la imagen seleccionada.

```
    //La partida ha finalizado en este punto del código, por lo tanto debemos de resetear  jugadorImgs, para añadirle un
    //noSeleccionado.
    // Resetear selección del jugador


    for (var j = 0; j < jugadorImgs.length; j++) {
        jugadorImgs[j].classList.remove("seleccionado");
        jugadorImgs[j].classList.add("noSeleccionado");
    }

    // Mostrar opción de la máquina
    maquinaImg.src = "img/" + opcionMaquina + "Ordenador.png";

    // Comprobar fin del juego
    if (partidasJugadas >= totalPartidas) {
        botonTirar.disabled = true;
        botonJugar.textContent = "Nueva partida";
    }
}
```

Evento al pulsar el botón "JUGAR" (botonJugar.addEventListener)

Cuando el jugador hace clic en el botón "JUGAR", se ejecuta una función anónima que valida el nombre y el número de partidas ingresados por el jugador. Si ambos son válidos, se deshabilitan los campos de entrada, se actualiza el nombre del jugador y el total de partidas, se actualiza el marcador en el DOM y se muestran las imágenes correspondientes del jugador. Si hay errores en la entrada, se resaltan los campos correspondientes en rojo.

```
    botonJugar.addEventListener("click", function() {


     //eliminamos los espacios en nombre y partida

        var nombre = nombreInput.value.trim();
        var partidas = partidasInput.value.trim();


// llamamos a las funciones que hemos creado más arriba, para validar el value de los input.
// dependiendo de los resultados que nos entreguen, vamos a acceder a la class de cada elemento para modificar la propiedad
// css que nos permite añadir un fondo rojo al input.


// classList.add("fondoRojo") añadimos
// classList.remove("fondoRojo");  quitamos


    if (validarNombre(nombre) && validarPartidas(partidas)) {



      nombreJugador = nombre;
      totalPartidas = parseInt(partidas);
      spanTotal.textContent = totalPartidas;
      nombreInput.disabled = true;
      partidasInput.disabled = true;
      botonJugar.textContent = "JUGAR";
      botonTirar.disabled = false;


      // Mostrar imágenes correctas del jugador al inicio


      jugadorImgs.forEach(function(img, index) {
          img.src = "img/" + posibilidades[index] + "Jugador.png";
      });
  } else {
```

```
      if (!validarNombre(nombre)) {
          nombreInput.classList.add("fondoRojo");
      } else {
          nombreInput.classList.remove("fondoRojo");
      }

      if (!validarPartidas(partidas)) {
          partidasInput.classList.add("fondoRojo");
      } else {
          partidasInput.classList.remove("fondoRojo");
      }
  }


      });
```

Evento al seleccionar una opción de jugador (jugadorImgs.forEach)

Este evento se activa cuando el jugador hace clic en una de las imágenes de las opciones (piedra, papel o tijera). Se modifica la clase de las imágenes para indicar cuál fue seleccionada por el jugador.

```
// Evento al seleccionar una opción de jugador
// aquí es donde vamos a modificar la class de las imagenes, de seleccionado  a noSeleccionado
// de esa forma sabremos luego que imagen fue seleccionada, porque tendrá un determinado valor la clase.



jugadorImgs.forEach(function(img, index) {
    img.addEventListener("click", function() {
        jugadorImgs.forEach(function(img) {
            img.classList.remove("seleccionado");
            img.classList.add("noSeleccionado");
        });
        img.classList.remove("noSeleccionado");
        img.classList.add("seleccionado");
    });
});
```

Evento al hacer clic en el botón "RESET" y botón YA

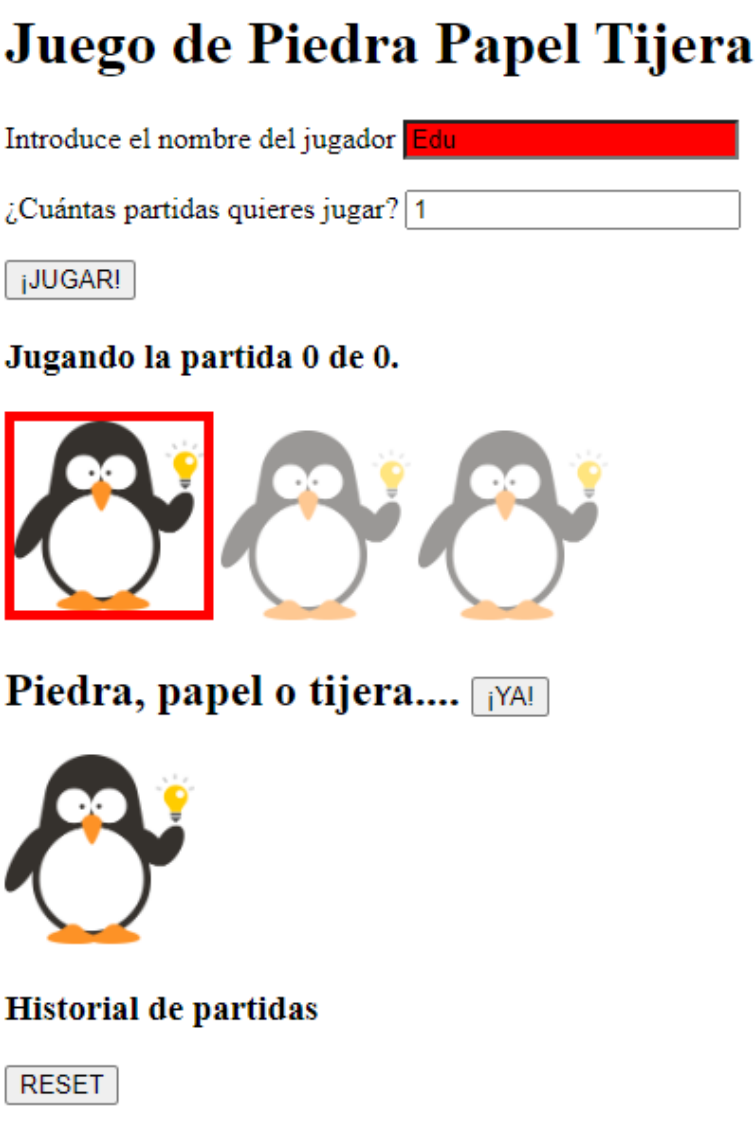
Cuando el jugador hace clic en el botón "RESET", se ejecuta la función resetearPartida, que restablece todas las configuraciones de la partida a su estado inicial. **(botonReset.addEventListener)**

```
    // Cuando hacemos clic en YA  ponemos la funcion jugarPartida a hacer el trabajo

    botonTirar.addEventListener("click", jugarPartida);

    // Cuando hacemos clic en RESET  ponemos la funcion resetearPartida a hacer el trabajo

    botonReset.addEventListener("click", resetearPartida);


});
```

2. Pruebas

2.1 Introducción de usuario con datos no válidos.

Usuario no válido



Partidas y Usuario no válidos

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Edu

¿Cuántas partidas quieres jugar? 0

¡JUGAR!

Jugando la partida 0 de 0.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

RESET

2.2 Introducción de cantidad de partidas con datos no válidos.

Partidas no válidas

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 0

¡JUGAR!

Jugando la partida 0 de 0.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

RESET

2.3. Acceso a la aplicación con datos válidos.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

JUGAR

Jugando la partida 0 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

RESET

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

Partida 1.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

JUGAR

Jugando la partida 1 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

- Empate

RESET

Partida 2.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

JUGAR

Jugando la partida 2 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

- Empate
- Gana la máquina

RESET

Partida 3.

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

JUGAR

Jugando la partida 3 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

- Empate
- Gana la máquina
- Gana Eduardo

RESET

Partida 4

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

JUGAR

Jugando la partida 4 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

- Empate
- Gana la máquina
- Gana Eduardo
- Gana la máquina

RESET

Partida 5

Juego de Piedra Papel Tijera

Introduce el nombre del jugador Eduardo

¿Cuántas partidas quieres jugar? 5

Nueva partida

Jugando la partida 5 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

- Empate
- Gana la máquina
- Gana Eduardo
- Gana la máquina
- Empate

RESET

Reset

Juego de Piedra Papel Tijera

Introduce el nombre del jugador

¿Cuántas partidas quieres jugar?

Nueva partida

Jugando la partida 0 de 5.

Piedra, papel o tijera.... ¡YA!

Historial de partidas

RESET