

MO 6

Desenvolupament

Web Client

UF1 - Projecte Joc del Penjat

Mohamed Bourig Ait Bourig
11-11-2023

ÍNDICE

1 – Objetivos:	2
2 - Variables Globales:	3
3 - Función iniciar():.....	4
4 - Función escogerPalabra(array, min, max):	5
5 - Función dibujarGuionPalabra(palabra):	5
6 - Función introducirLetra():	6
7 - Función comprobarLetras(letra):	7
8 - Función gameOver():.....	8
9 - Función abandonarPartida():	9
10 - Funciones setCookie(cname, cvalue, exdays) y leerCookie(nombre):.....	9

1 – Objectivos:

1. El joc s'inicia en carregar-se la pàgina que el conté.
2. Les paraules per al joc les hem d'introduir al iniciar el joc amb un prompt, d'una única vegada, introduint-les separant cada paraula amb una ',' (coma) i seran emmagatzemades a un Array. Elimineu possibles espais en blanc al principi i/o final del text introduït per l'usuari. D'entre les paraules de l'Array se'n seleccionarà una de forma aleatòria. Si l'usuari no introdueix cap paraula hem d'utilitzar unes definides per nosaltres.
3. La pàgina principal ens ha de mostrar dos botons:
 - 3.1. El primer botó ens permetrà introduir una lletra per comprovar si forma part de la paraula.
 - 3.2. El segon botó ens servirà per abandonar la partida i provar-ho amb una nova paraula.
4. A la pantalla ens apareixerà el dibuix (inicialment net) que s'anirà completant a mida que es fallin lletres.
5. També apareixerà l'estat de la paraula que estem cercant, amb les lletres encertades que sortiran descobertes i la resta amb un '_' (guio baix).
6. Han aparèixer a la pantalla les estadístiques de la partida actual amb les paraules encertades, els abandonaments, i els cops que ens han penjat.
7. Si abandonem esperarem 5 segons per iniciar una nova partida.
8. Si ens pengen tardarem 10 segons en ressuscitar, és a dir, 10 segons per iniciar nova partida.
9. S'ha d'emmagatzemar (amb cookies) les estadístiques globals de les partides guanyades, abandonades i perdudes. Si tanquem la pàgina que conté el joc i la tornem a obrir se'ns han de mostrar aquests camps, tal i com els teníem.
10. El projecte s'ha de fer només amb les funcions i els objectes que s'han vist a classe.
11. És obligatori l'ús d'Arrays per mantenir l'estat de la paraula a descobrir.
12. És obligatori l'ús de console.log, per mostrar a la consola l'estat dels elements de la partida.
13. S'han d'utilitzar funcions per a estructurar el codi, s'han d'utilitzar el menor nombre possible de variables globals i s'han de passar com a paràmetres a les funcions on les utilitzarem.
14. El codi ha d'estar comentat i ben codificat.

2 - Variables Globales:

palabrasAleatorias: Un array de palabras predefinidas.

palabras: Una variable que recoge las palabras introducidas por el usuario.

letrasIntroducidas: Un array para almacenar las letras que el usuario ha introducido.

guiones: Un array para representar las letras a adivinar con guiones bajos.

palabraEscogida: La palabra que se elige para jugar.

contFallos: Contador de fallos del jugador.

palabrasAcertadas: Contador de palabras acertadas.

partidasPerdidas: Contador de partidas perdidas.

partidasAbandonadas: Contador de partidas abandonadas.

```
// variables globales
let palabrasAleatorias = ["Pelota", "Jardin", "Cielo", "Carrera"];
let palabras = prompt("Introduce las palabras con las que desea jugar:").trim();
let letrasIntroducidas;
let guiones;
let palabraEscogida = null;
let contFallos;
let palabrasAcertadas;
let partidasPerdidas;
let partidasAbandonadas;
```

3 - Función iniciar():

Inicializa las variables y obtiene la palabra para la partida.

Si el usuario no introduce palabras, se elige una al azar de las predefinidas.

Si el usuario introduce palabras, se elige una al azar de las proporcionadas.

Muestras estadísticas y resetea valores.

```
function iniciar() {
  let arrayPalabras = [];
  letrasIntroducidas = [];
  guiones = [];
  letra.value = ""; // limpia el input text
  let min = 0;
  contFallos = 0;
  palabrasAcertadas = leerCookie("cookiePalabrasAcertadas");
  partidasPerdidas = leerCookie("cookiePartidasPerdidas");
  partidasAbandonadas = leerCookie("cookiePalabrasAbandonas");

  // si en el prompt le damos a cancelar o aceptar sin introducir palabras entonces nos escogera las palabras predefinidas
  if (palabras == null || palabras == "") {
    palabraEscogida = escogerPalabra(palabrasAleatorias, min, (palabrasAleatorias.length - 1));
    dibujarGuionPalabra(palabraEscogida);
  } else {
    // en caso contrario separaremos las palabras por la coma y escogemos una con la funcion escoger alabra
    arrayPalabras = palabras.split(",");
    palabraEscogida = escogerPalabra(arrayPalabras, min, (arrayPalabras.length - 1));
    dibujarGuionPalabra(palabraEscogida);
  }

  // reseteo los valores de la imagen, las palabras introducidas y el mensaje de que ha ganado la partida
  document.getElementById("mostrarLetrasIntroducidas").textContent = "";
  document.getElementById("estadoPartida").textContent = "";
  document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_0.png";

  // muestro las estadísticas con los valores de las cookies
  document.getElementById("mostrarPalabrasAcertadas").textContent = "Palabras acertadas: " + leerCookie("cookiePalabrasAcertadas");
  document.getElementById("mostrarPartidasAbandonadas").textContent = "Partidas abandonadas: " + leerCookie("cookiePalabrasAbandonas");
  document.getElementById("mostrarPartidasPerdidas").textContent = "Partidas perdidas: " + leerCookie("cookiePartidasPerdidas");
  console.log(palabraEscogida);
  console.log(guiones);
}
```

El ahorcado

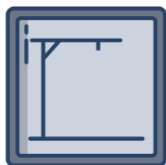
Introduce una letra

— · — · — · — · —

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 13



4 - Función escogerPalabra(array, min, max):

Devuelve una palabra aleatoria de un array dado y dentro de un rango específico.

```
function escogerPalabra(array, min, max) {  
  // devuelve un numero random entre 0 y el tamaño de una palabra ejemplo cielo (5)  
  return array[Math.floor(Math.random() * (max - min + 1) + min)];  
}
```

```
Jardin  
  ▶ Array(6)  
Cielo  
  ▶ Array(5)  
Jardin  
  ▶ (6) ['_', '_', '_', '_', '_', '_']  
Cielo  
  ▶ (5) ['_', '_', '_', '_', '_']  
Pelota  
  ▶ (6) ['_', '_', '_', '_', '_', '_']  
Carrera  
  ▶ (7) ['_', '_', '_', '_', '_', '_']
```

5 - Función dibujarGuionPalabra(palabra):

Inicializa el array de guiones para representar la palabra a adivinar.

```
function dibujarGuionPalabra(palabra) {  
  // recorro la longitud de la palabra y por cada letra dibujo una _  
  for (let i = 0; i < palabra.length; i++) {  
    guiones[i] = "_";  
  }  
  document.getElementById("mostrarPalabraHtml").textContent = guiones;  
}
```

```
Carrera  
  ▶ (7) ['_', '_', '_', '_', '_', '_']
```

6 - Función introducirLetra():

Verifica la letra introducida por el usuario.

Muestra mensajes de error si la entrada es incorrecta.

Llama a comprobarLetras() si la entrada es válida.

```
function introducirLetra() {  
  // guardo el valor de la letra introducida en una variable  
  let letras = letra.value;  
  // si cumple alguna de estas condiciones entonces muestra un mensaje de error  
  if (letras == null || letras == " " || letras == "_" || letras == "") {  
    inputErroneo.innerHTML = "<p><strong> El campo de texto no acepta caracteres y tampoco puede estar vacío</strong></p>";  
  } else {  
    // si no llamara al metodo que comprueba las letras, y ira mostrando las letras introducidas y la palabra con las letras acertadas  
    comprobarLetras(letras.charAt(0)); // solo coge la primera letra por si introducimos varias letras  
    letrasIntroducidas.push(letras.charAt(0));  
    document.getElementById("mostrarLetrasIntroducidas").textContent = "Letras introducidas: " + letrasIntroducidas;  
    console.log(guiones);  
  }  
}
```

El ahorcado

Introduce una letra

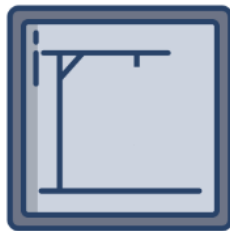
_ , _ , R, R, _ , R, _

Letras introducidas: r

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 13



7 - Función comprobarLetras(letra):

Comprueba si la letra introducida está en la palabra escogida.

Actualiza el array de guiones con las letras acertadas.

Gestiona los fallos y muestra imágenes del ahorcado.

Si todas las letras se adivinan, se muestra un mensaje de victoria y se reinicia el juego.

Si se alcanza el número máximo de fallos, se muestra un mensaje de game over y se reinicia el juego.

```
function comprobarLetras(letra) {
    // guardo el valor de la palabra escogida en mayusculas en una variable
    let palabraMayusculas = palabraEscogida.toUpperCase();
    // guardo el valor de la letra introducida en mayusculas en una variable
    let letraMayuscula = letra.toUpperCase();
    // en el bucle le indico la longitud de la palabra, ira comprobando letra por letra y si coincide la añadira en un array llamado guiones
    for (let i = 0; i < palabraEscogida.length; i++) {
        if (palabraMayusculas[i].includes(letraMayuscula)) {
            guiones[i] = letraMayuscula;
        }
    }

    // muestro el array con las letras, encontradas, si no pues aparecera _
    document.getElementById("mostrarPalabraHtml").textContent = guiones;
    // si la letra introducida no coincide con ninguna de la letra de la palabra, entonces se sumara un fallo
    if (!guiones.includes(letraMayuscula)) {
        contFallos++;
    }
}
```

```
// segun los fallos mostrara una foto del personaje ahorcandose
if (contFallos == 1) {
    document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_1.png";
} else if (contFallos == 2) {
    document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_2.png";
} else if (contFallos == 3) {
    document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_3.png";
} else if (contFallos == 4) {
    document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_4.png";
} else if (contFallos >= 5) {
    document.getElementById("imagenAplicacion").src = "/imagenes/ahorcado_5.png";
    // al haber llegado al maximo de intentos, en 0,5seg se llamara a una funcion game over para ostrar una imagen con game over
    setTimeout(gameOver, 500);
    // se aumentara 1 en partidas perdidas
    partidasPerdidas++;
    // creo cookie con nombre, valor, expira en 60 dias y la muestro
    setCookie("cookiePartidasPerdidas", partidasPerdidas, 60);
    document.getElementById("mostrarPartidasPerdidas").textContent = "Partidas perdidas: " + leerCookie("cookiePartidasPerdidas");
    // reseteo las letras introducidas
    letrasIntroducidas = [];
    document.getElementById("mostrarLetrasIntroducidas").textContent = "";
    // y en 10 seg llama a la funcion iniciar
    setTimeout(iniciar, 10000);
}
```

```
if (!guiones.includes("_")) {
    palabrasAcertadas++; // le sumo 1 a palabras acertadas
    document.getElementById("estadoPartida").textContent = "PALABRA ACERTADA"; // muestro el mensaje de palabra acertada
    setCookie("cookiePalabrasAcertadas", palabrasAcertadas, 60); // creo cookie con nombre, valor y expira en 60 dias

    // reseteo las letras introducidas
    letrasIntroducidas = [];
    document.getElementById("mostrarLetrasIntroducidas").textContent = "";

    // muestro las palabras acertadas
    document.getElementById("mostrarPalabrasAcertadas").textContent = "Palabras acertadas: " + leerCookie("cookiePalabrasAcertadas");
    // y en 2 seg llama a la funcion iniciar
    setTimeout(iniciar, 2000);
}
```


El ahorcado

Introduce una letra

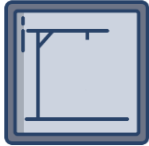
C, _ ,R,R, _ ,R, _

Letras introducidas: r,c

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 13



[Iniciar partida](#) [Introducir letra](#) [Abandonar partida](#)

El ahorcado

Introduce una letra

C, _ ,R,R, _ ,R, _

Letras introducidas: r,c,k,k,k,k,k

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 13



[Iniciar partida](#) [Introducir letra](#) [Abandonar partida](#)

El ahorcado

Introduce una letra

C, _ ,R,R, _ ,R, _

Letras introducidas: k

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 14



[Iniciar partida](#) [Introducir letra](#) [Abandonar partida](#)

8 - Función gameOver():

Muestra una imagen de "game over" cuando se pierde la partida.

```
function gameOver() {  
    // muestro una de game over  
    document.getElementById("imagenAplicacion").src = "/imagenes/game-over.jpg"  
}
```

El ahorcado

Introduce una letra

C, _ ,R,R, _ ,R, _

Letras introducidas: k

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 14



[Iniciar partida](#) [Introducir letra](#) [Abandonar partida](#)

9 - Función abandonarPartida():

Incrementa el contador de partidas abandonadas.

Muestra estadísticas actualizadas y reinicia el juego después de 5 segundos.

```
function abandonarPartida() {  
    partidasAbandonadas++; // sumo 1 a partidas abandonadas  
    setCookie("cookiePalabrasAbandonas", partidasAbandonadas, 60) // creo cookie con nombre, valor y expira en 60 días  
  
    // muestro las partidas abandonadas  
    document.getElementById("mostrarPartidasAbandonadas").textContent = "Partidas abandonadas: " + leerCookie("cookiePalabrasAbandonas");  
  
    // reseteo las letras introducidas  
    letrasIntroducidas = [];  
    document.getElementById("mostrarLetrasIntroducidas").textContent = "";  
  
    // y en 5 seg llama a la funcion iniciar  
    setTimeout(iniciar, 5000);  
}
```

Palabras acertadas: 9

Partidas abandonadas: 3

Partidas perdidas: 14

10 - Funciones setCookie(cname, cvalue, exdays) y leerCookie(nombre):

Estas funciones se utilizan para gestionar cookies que almacenan estadísticas del juego, para las palabras acertadas, partidas perdidas y partidas abandonadas.

```
function setCookie(cname, cvalue, exdays) {  
    // Crear una nueva instancia de la clase Date para obtener la fecha actual  
    const d = new Date();  
  
    // Calcular la fecha de expiración sumando el número de días especificado en exdays  
    d.setTime(d.getTime() + (exdays * 24 * 60 * 60 * 1000));  
  
    // Construir la cadena de expiración en formato UTC  
    let expires = "expires=" + d.toUTCString();  
  
    // Configurar la cookie en el documento con el nombre, valor, fecha de expiración y la ruta "/"  
    document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";  
}
```

```
function leerCookie(nombre) {

    // Clave de búsqueda. Es el nombre de la cookie
    let clave = nombre + "=";

    // Obtengo todas las cookies y las separo con ;
    let cookies = document.cookie.split(';');

    // Recorro todas las cookies
    for (let i = 0; i < cookies.length; i++) {

        // Obtengo una cookie individual
        let c = cookies[i];

        // Eliminar espacios en blanco al principio de la cookie
        while (c.charAt(0) == ' ') c = c.substring(1, c.length);

        // Verificar si la cookie actual comienza con la clave de búsqueda
        if (c.indexOf(clave) == 0) {
            // Si se encuentra la cookie devolvera el valor decodificado de la cookie
            return decodeURIComponent(c.substring(clave.length, c.length));
        }
    }

    // Si no se encuentra la cookie devolvera 0
    return 0;

}
```

Filtros										
Mostrar solo cookies que tengan un problema										
Nombre	Valor	Domain	Path	Expires / Max-Age	Tama...	HttpOnly	Secure	SameSite	Partition...	Priority
cookiePartidasPerdidas	14	127.0.0.1	/	2024-01-11T18:11:23.000Z	24					Medium
cookiePalabrasAcertadas	9	127.0.0.1	/	2024-01-11T00:58:45.000Z	24					Medium
cookiePalabrasAbandonas	3	127.0.0.1	/	2024-01-11T00:52:58.000Z	24					Medium