

## PROYECTO FINAL MAQUINA TRAGAPERRAS ERVIGIO

Para realizar el proyecto lo primero que he hecho a sido analizar las funcionalidades que debía tener para separar en funciones las diferentes acciones a llevar a cabo.

El proyecto consta de:

- Al cargar la página dispondremos de un crédito inicial para poder realizar las tiradas. Este será indispensable para ello, ya que sin crédito los botones no deberán hacer nada. Se encargará de ello: **creditoInicial()**
- Un botón principal que efectúa una tirada general y esta genera aleatoriamente unas imágenes de frutas que estarán ubicadas en tres cajas en pantalla. Se encargará de ello: **tiradaPrincipal()**. Una vez finalizada se llamará a la función que comprobara si las imágenes son iguales: **comprobarPremio()** y la función de **actualizarMonedas()** que restara o sumara según convenga.
- Tres ventanas en las cuales se colocarán las imágenes de las frutas. En caso de ser iguales las tres imágenes se otorgará un premio de monedas que se aumentaran al monedero.
- Tres botones que harán que la ventana que esta sobre ellos cambie de imagen aleatoriamente. Estos funcionaran siempre que haya crédito, pero solo tras haberse realizado la tirada general.  
Los movimientos los realiza la función **avance()** que actuara sobre los botones y las ventanas y posteriormente la función **comprobarPremio()** se encargara de compararlas para ver si hay premio. Nuevamente se llamará a la función de **actualizarMonedas()** para restar o sumar monedas.
- Un mensaje modal que aparecerá cuando no haya crédito o cuando se dé un premio. Se encargará la función **velo()** que generará una ventana emergente con un mensaje y unas imágenes. Dicha ventana esta presente siempre, pero oculta. Para cerrar dicha ventana se crea un evento onclick llamado **cerroVelo()** sobre la imagen de la cruz que ocultara de nuevo la ventana.
- Sonidos para cada uno de los eventos. En la función **sonidos()** se cargará una ruta y se reproducirá un archivo de sonido especificado como parámetro.

### 1. Función inicio():

En esta función se preparan los eventos onclick de los botones. Los 3 botones de volver a tirar están dentro de un bucle for. Aquí se inicializa la variable mp3 para la reproducción del sonido definiendo el id que tiene en el HTML.

Fuera de la función he creado 3 arrays: uno para almacenar las monedas (Strings), otro para almacenar las imágenes de frutas (strings con el nombre del archivo png) y un tercero llamado ventanas que siempre tendrá un tamaño de 3 posiciones que serán las ventanas donde se colocan las frutas. Cada vez que haya una fruta en una ventana se sobrescribirá este array, así leyendo cualquiera de sus posiciones tengo controlada que fruta hay en pantalla.

Se inicializarán dos variables: una booleana para el control de los botones de volver a tirar y otra llamada mp3 para el archivo de audio.

### 2. Función creditoInicial:

Esta función se cargará al iniciar la página, invocada dentro de la función inicio(). Aquí he creado una variable cuyo valor es el redondeo de la función Math.random que generara un numero aleatorio en un rango del 0 al 15. Este número será el total de monedas que dispondremos. Este valor será el length de un array llamado monedas que se va a llenar con Strings con el nombre de las imágenes png de una moneda (“moneda.png”). Usare un condicional IF para asegurarme que el numero aleatorio sea mayor que 0, en el caso contrario tomara un valor de 1 por defecto. Mediante un bucle for y el método .push() lleno el array con los strings. Posteriormente mediante un innerHTML inyectare en el div con una estructura de <img> el numero de monedas totales. El numero random también me servirá para mediante otro innerHTML escribir el número total de monedas.

### 3. Función tiradaPrincipal:

En esta función en primer lugar se comprueba con un If si tenemos monedas para tirar. Esto lo controlo con el array de monedas y su longitud. En caso de no tener aparece el velo, que es una función a la cual le paso parámetros de mensaje “sin crédito” y nombre del archivo de audio.

He creado una variable que mediante un querySelector de la clase ventana detecta cuantas hay para saber cuántas imágenes random ha de generar. Con un for lleno el array con una imagen aleatoria siendo el total de interacciones la cantidad de ventanas y luego las inyecto con innerHTML.

Se invoca la función de generar sonido.

Una vez fuera del If, ejecutara en primer lugar el consumo de una moneda, con monedas.pop() que elimina la ultima posición. Acto seguido se comprueba si hay premio en esa primera tirada mediante la función comprobarPremio() y después se actualizara el contador de monedas con la función actualizarMonedas(). Por último, para desbloquear los botones de volver a tirar que estaban deshabilitados desde el inicio se pone la variable avanceDisponible en true.

### 4. Función comprobarPremio():

Esta función mediante un if comprueba que las imágenes que hay en las 3 ventanas son iguales. En caso afirmativo, con otra secuencia de if comprueba el valor de una de las posiciones y la contrasta con los nombres de las frutas. He añadido una variable llamada premio y según que fruta sea le aplicará un valor u otro, asimismo mediante un .push() añadirá imágenes (strings) al array de monedas con un bucle for con longitud la variable premio.

### 5. Función actualizarMonedas():

En esta función lo que hago es volver a sobrescribir con innerHTML todo el div que contiene el “monedero” de la misma forma que en la función creditoInicial() con la diferencia que esta vez lo que hago primero es vaciar el div y luego leo el array de monedas para llenar de nuevo el div.

En caso de que no queden monedas, la cual cosa compruebo al final con un if y el length del array monedas, aparece el velo indicando que se ha quedado sin crédito y bloquea los botones de avance para que no sigan tirando. También podía haber bloqueado el botón de tirada inicial, pero como cuando pulsas aparece que no hay crédito y ya no hace nada más, me ha parecido suficiente.

### 6. Función avance():

Esta es la función que mas me ha costado, ya que no sabía cómo hacerla. Lo que he hecho a sido en primer lugar es mediante un if decidir si puede tirar o no. Podrá realizarse una tirada extra siempre que la variable booleana sea cierta (cambia a cierta después de cada tirada principal) y siempre que tengamos monedas (compruebo que la longitud del array monedas sea mayor que 0).

Una vez pasada esta comprobación lo que hago es generar una imagen aleatoria y según el id del botón pulsado lo asocio a una de las ventanas. Esto lo capturo con un this y comparando si el valor del id de ese botón es igual a "b0, b1 o b2" y entonces le asigno un numero que es la posición (0,1,2).

Esta parte me ha costado por que no acabo de saber que puedo realizar con el this. Me hubiese gustado directamente saber que posición está ocupando el botón clicado (0,1,2) sin necesidad de hacer esta comprobación.

Para evitar que la imagen random vuelva a ser la misma que la que hay he añadido un bucle while que lo que hace es que mientras la imagen que hay en la ventana y la imagen que se acaba de generar sea la misma repita el random de obtener imagen. Después se llama la función de reproducir sonido con un parámetro con el nombre del archivo, se elimina una moneda consumida por la tirada con monedas.pop() y finalmente se inyecta de nuevo la imagen en la ventana con innerHTML y a su vez la posición del array ventanas se actualiza con la imagen que hay ahora. De esta forma tengo un array llamado ventanas que siempre tiene los tres valores de las ventanas actualizados.

Por ultimo se vuelve a comprobar si hay premio en esa tirada y se actualizan las monedas para restar la consumida o el premio otorgado en caso de que lo haya.

### 7. Función velo():

En esta función tengo tres parámetros que voy pasando siempre que la llamo y son dos strings: un mensaje según sea "no tienes crédito" o "has ganado", y el otro con el nombre del archivo de audio que se ha de reproducir. Este ultimo lo usare para llamar a la función de sonido dentro de esta función y pasarlo como parámetro. Lo primero que se hace en esta función es mostrar el div del velo con un display flex. Acto seguido con innerHTML se generan los div hijos con el mensaje o el premio recibidos. He añadido dentro un div extra que tendrá las monedas (iconos) igual que en el monedero. Este se inyecta también con innerHTML y un bucle que genera tantas veces como premio se haya obtenido. Si no ha habido premio también se genera, pero como el valor es 0 no se ve en pantalla. Estas monedas las he alineado con un estilo en línea en el css.

No tenia muy claro cómo podía cerrar el velo al clicar la imagen de la cruz. Lo que he hecho ha sido tratarlo como si fuese un botón y darle una función onclick, la cual simplemente se encarga de volver a poner el div velo en display none.

Se reproduce el audio que se ha pasado por parámetro "final.mp3" o "ganar.mp3".

## 8. Función sonidos():

Esta función inicialmente no estaba prevista, y había añadido los sonidos en cada función. Para tratar de no repetir el código he pensado en pasar como parámetros el nombre del audio que debía sonar y con el crear una ruta específica. Esto lo hago en la variable ruta que luego asigno a la variable mp3.src. En la función inicio se prepara la variable mp3 dándole el id del HTML.

Después de la variable lo que hago es reproducir el audio preparado con mp3.play. La razón por la que tengo la variable playPromise y el bucle if es porque me estaba dando error (en el inspector del navegador, aunque todo funcionaba correctamente). *Uncaught (in promise) DOMException: The play() request was interrupted by a new load request.* El propio navegador me sugería un enlace a esta solución.

## 9. Función cerrarVelo():

Esta función que es simplemente un querySelector y style.display none, me hubiese gustado incluirla en la misma función velo(), pero no he sabido implementarla. Lo único que hace es volver a ocultar el div.

He mantenido la apariencia (a excepción del fondo) y funcionalidades como en el ejemplo.

Aunque posiblemente tenga mucho código innecesario o repetitivo, creo que he hecho un buen trabajo en general y que he aprendido bastante en el curso.

Es posible que abuse del “innerHTML” pero no conozco otra forma de implementar esas partes sin repetir código y mucho menos cuando el contenido del mismo no es igual como ha sido el caso de muchas actividades. Esta es una de las partes que mas me ha costado del curso, las concatenaciones de estructuras con variables (vamos que no se partir los div)

Los conceptos de programación tales como variables, bucles, etc. ya los conocía debido a mi formación en lenguaje Java con lo cual no me ha costado entenderlos, aunque a veces me cuesta horrores ponerlos en práctica.

Estoy bastante contento con el curso y me gustaría seguir ampliándolo con mas materia (objetos, etc.) y sobre todo con ayuda de los Reyes Godos.

Adjunto todo el código JavaScript que he realizado con comentarios en casi cada línea. Esto, aunque se que es una mala practica ya que un buen código se ha de entender sin comentarios, pero lo hago para yo mismo acordarme de donde salen las cosas y por si en un futuro debo refrescar algo.

En el código que voy a entregar lo voy a eliminar para que no parezca demasiado largo.

```
window.onload = inici;

//array donde guardar las monedas disponibles
var monedas = [];
//array para guardar los nombres de las imagenes
var imagenes = ["cerezas.png", "fresa.png", "limon.png", "naranja.png",
"platanos.png", "sandia.png"];
//array que guarda las imagenes de las ventanas para luego compararlas
var ventanas = [];
//variable para el audio
var mp3;
//bandera para activar los 3 botones extras
var avanceDisponible = false;

function inici() {
    //boton tirar
    document.querySelector("#lanzar").onclick = tiradaPrincipal;
    //botones volver a tirar
    for (let i = 0; i < 3; i++) {
        document.querySelectorAll(".boton")[i].onclick = avance;
    }
    //inicializacion de variable audio
    mp3 = document.getElementById("audio");
    //generar credito inicial
    creditoInicial();
}

//funcion para añadir al inicio un numero de monedas aleatorio
function creditoInicial() {
    //se aplica la formula random para obtener un maximo total de 15 monedas
    let monedasRandom = Math.floor(Math.random() * 15);
    //el numero obtenido por el random sera el total de monedas y el total
    del tamaño del array que guardara imagenes de monedas
    var length = monedasRandom;
    //si el numero obtenido aleatorio es mayor que el 0 llenara el array
    siendo ese mismo su longitud
    if (monedasRandom > 0) {
        for (let i = 0; i < length; i++) {
            monedas.push("moneda.png");
        }
    }
    else {
        monedasRandom = 1;
        monedas.push("moneda.png");
    }
}
```

## PROYECTO FINAL MAQUINA TRAGAPERRAS ERVIGIO

```
//en caso de salir el cero en el random, el valor siempre sera un 1 y
llenara el array de monedas con una imagen
//se llenara el div "dinero" con el numero total de monedas y el
simbolo del euro (esto ultimo tengo que mejorar el sistema...)
document.querySelector("#dinero").innerHTML = monedasRandom + `<span class
="euros">€</span>`;
//se llenara el div "monedas" con el contenido del array monedas (que son imag
enes de monedas)
for (let coin of monedas) {
    document.querySelector("#monedas").innerHTML += ``;
}
}

//funcion que generara la tirada general
function tiradaPrincipal() {
//si hay monedas permite tirar. Esto se comprueba con la longitud del
array monedas
    if (monedas.length > 0) {
        //variable para saber cuantas ventanas hay en el juego (son 3)
        let totalDivs = document.querySelectorAll(".ventana").length;
        for (let i = 0; i < totalDivs; i++) {
            //en cada posicion del array coloco una imagen random
            ventanas[i] = imagenes[Math.floor(Math.random() * imagenes.length)
];

            //añado las imagenes al div "ventana"
            document.querySelectorAll(".ventana")[i].innerHTML = ``;
        }
        //añado el sonido de lanzar
        sonidos("lanzar.mp3");
    }
    else {
        //en caso de que en la inicial solo tuviese una moneda y se acabase el
credito
        velo("Te has quedado sin credito", "final.mp3");
    }
    //resto al array monedas una moneda por la tirada hecha
    monedas.pop();
    //llamada a la funcion que comprueba si hay coincidencias en las imagenes
para otorgar premio
    comprobarPremio();
    //llamada a la funcion que resta una moneda y actualiza las imagenes de mo
nedas
    actualizarMonedas();
    //la variable se pone en true para que permita realizar tiradas extras
    avanceDisponible = true;
}
```

## PROYECTO FINAL MAQUINA TRAGAPERRAS ERVIGIO

```
//funcion que comprueba si hay coincidencia en las tres imagenes
function comprobarPremio() {
    //si son las tres imagenes del array ventanas iguales lanzara la alerta
    if (ventanas[0] == ventanas[1] && ventanas[1] == ventanas[2]) {
        //Como las tres imagenes son iguales compruebo una de ellas y
        //dependiendo de la fruta que salga el premio sera mayor o menor
        //lo que hago es leer el array y comparar los nombres, dando x monedas y
        //añadiendo al array monedero una imagen
        var premio;
        if (ventanas[1] == "cerezas.png") {
            premio = 1;
        } else if (ventanas[1] == "fresa.png") {
            premio = 2;
        } else if (ventanas[1] == "limon.png") {
            premio = 3;
        } else if (ventanas[1] == "naranja.png") {
            premio = 4;
        } else if (ventanas[1] == "platanos.png") {
            premio = 5;
        } else if (ventanas[1] == "sandia.png") {
            premio = 6;
        }
        //lleno el array con monedas segun el premio
        for(let i=0;i<premio;i++){
            monedas.push("moneda.png");
        }
        //una vez comprobado el premio aparece el velo de ganador al que le paso los
        //parametros
        velo("Has ganado", "ganar.mp3", premio);
        //se bloquean los tres botones de avance hasta que se tira de nuevo
        avanceDisponible = false;
    }
}

//funcion para actualizar las monedas cada vez que se tira
function actualizarMonedas() {
    //Vuelvo a cargar el div con la cantidad de dinero en numero que hay ahora
    document.querySelector("#dinero").innerHTML = monedas.length + `<span clas
s="euros">€</span>`;
    //Para mostrar la cantidad de monedas que hay ahora vuelvo a cargar el div con
    //las imagenes
    //Primero elimino las que estan ahora
    document.querySelector("#monedas").innerHTML = "";
    for (let coin of monedas) {
        //Inserto de nuevo las monedas
        document.querySelector("#monedas").innerHTML += ``;
    }
    //cuando no quedan monedas avisa mediante mensaje
}
```

```

        if (monedas.length == 0) {
            velo("Te has quedado sin credito", "final.mp3", 0);
            //se bloquean los tres botones de avance
            avanceDisponible = false;
        }
    }

    //funcion que se llamara al pulsar cualquiera de los botones de "volver a
    tirar"
    function avance() {
        //si la variable booleana es cierta y hay credito dejara realizar las tiradas
        if (avanceDisponible && monedas.length > 0) {
            //genero una nueva imagen random
            let imagenNueva = imagenes[Math.floor(Math.random() * imagenes.length)
];
            //obtengo el id del boton clicado y le asigno un numero para saber que ventana
            debe cambiar de imagen
            let posicion;
            if (this.id == "b0") {
                posicion = 0;
            } else if (this.id == "b1") {
                posicion = 1;
            } else if (this.id == "b2") {
                posicion = 2;
            }
            //si la imagen que hay en la ventana es igual que la nueva, genero otra random
            hasta que sea diferente
            while (ventanas[posicion] == imagenNueva) {
                imagenNueva = imagenes[Math.floor(Math.random() * imagenes.length)
];
            }
            //una vez asegurado que la imagen no va a ser igual continuo la secuencia
            //reproducir audio de boton volver a tirar
            sonidos("otra.wav");
            //resto al array monedas una moneda y llamo a la funcion de actualizar el div
            de imagenes de monedas
            monedas.pop();
            //añado las imagenes nuevas al div "ventana" correspondiente segun el boton
            que se ha pulsado
            document.querySelectorAll(".ventana")[posicion].innerHTML = `<img src=
            "img/${imagenNueva}">`;
            //en el array sustituyo la imagen que habia por la nueva, asi se en todo
            momento que imagenes hay en las casillas
            ventanas[posicion] = imagenNueva;
            //comprueba si hay premio despues de cada tirada y actualiza el monedero
            comprobarPremio();
            actualizarMonedas();
        }
    }
}

```



```
//funciones para llenar los velos
function velo(mensaje, archivoMp3, premio) {
    //muestro el div de velo
    document.querySelector("#velo").style.display = "flex";
    //inserto el div con los mensajes
    document.querySelector("#velo").innerHTML = `<div id="cuadro_mensaje">
    
    <div id="mensaje">${mensaje} ${premio}€</div>
    <div id="premio"></div>
    </div>`;
    //segun el premio se añaden monedas en el div premio
    for (let i = 0; i < premio; i++) {
        document.querySelector("#premio").innerHTML += ``;
    }
    //preparo el boton de los velos para cerrarlos
    document.getElementById("cruz").onclick = cerrarVelo;
    //Se reproduce el sonido correspondiente
    sonidos(archivoMp3);
}

//funcion de reproducir audio
function sonidos(nombreAudio) {
    //con el parametro le paso el nombre del audio y con el creo la ruta para hacer los cambios
    let ruta = "../audios/" + nombreAudio;
    mp3.src = ruta;
    //El audio me estaba dando problemas:
    //Uncaught (in promise) DOMException: The play() request was interrupted by a new load request.
    //con lo cual he tenido que añadir esta solucion de google:
    //https://developers.google.com/web/updates/2017/06/play-request-was-interrupted
    var playPromise = mp3.play();
    if (playPromise !== undefined) {
        playPromise.then(_ => {
            mp3.play();
        })
        .catch(error => {
        });
    }
}

//funcion del boton del velo
function cerrarVelo() {
    document.querySelector("#velo").style.display = "none";
}
```