

Proyecto Máquina Tragaperras

Los pasos que he seguido para desarrollar el programa han sido los siguientes:

1. He creado el fichero index.js. El HTML lo he modificado para hacer referencia a este fichero.
2. He declarado las siguientes variables globales al inicio del fichero.
 - a. La variable **crédito**, que es tipo var y es el crédito que tiene el jugador. En el inicio se calcula como un número al azar entre 10 y 15.
 - b. He declarado la variable **otraTirada**, que es de tipo booleana. Puede ser true o false, según si el jugador puede pulsar los botones “Volver a tirar”. Al inicio tiene valor false, ya que sólo se puede “Tirar”, no “Volver a tirar”
 - c. La variable **imágenes** es un array donde están las imágenes. Este array es tipo const, ya que no se modificará nunca. Las imágenes están en la carpeta img.
 - d. La variable **imagenesJuegos**, que es un array donde se almacenan las tres imágenes. Al inicio, como aún no se ha tirado, el array está formado por texto en blanco.
3. En la función **inicio()** he indicado las funciones y eventos que se ejecutan en el programa cuando se carga la página:
 - a. Al principio llamo a la función actualizarCredito(), que es la función encargada de mostrar el crédito disponible y muestra las imágenes de las monedas. Esta función también se llamará en el resto del código cuando sea necesario.
 - b. En el evento de hacer clic en el botón lanzar, se ejecuta la función tirar():
 - c. En el evento de hacer clic en cualquiera de los botones “Volver a tirar”, llamo a la función volvertirar(). He hecho un bucle con una única función para no tener que hacer una función para cada botón.
 - d. También he añadido una función para el evento de cerrar el cuadro de mensaje. En este caso, se ejecuta la función cerrar_mensaje().
4. En la función **tirar()** he añadido el código que se ejecuta al pulsar Tirar. Esta función hace lo siguiente:
 - a. Comprueba si hay crédito disponible o no. Si no lo hay, se llama a la función sin_credito() que es la que muestra el mensaje avisando.
 - b. Si hay crédito, en primer lugar, se reproduce el sonido correspondiente.
 - c. Luego se ejecuta un bucle, donde para cada una de las tres imágenes se busca una al azar dentro del array de imágenes, y se muestra en la ventana que corresponde.
 - d. Seguidamente, compruebo si las tres imágenes son iguales mediante la función comprobar_iguales(). Si lo son, se da un premio (un número de monedas al azar entre 3 y 5) y muestra un mensaje.

5. En la función **volveratirar()** he realizado el código que se ejecuta al pulsar cualquiera de los botones “Volver a tirar”. Realiza lo siguiente:
 - a. Se busca cual de los tres botones se ha pulsado, utilizando “this” y el “parentNode” y “children” para que se miren uno por uno todos los botones y ver cuál está activo.
 - b. Se ejecuta el código, comprobando si hay crédito suficiente y si está permitido volver a tirar.
 - c. Se reproduce el sonido correspondiente.
 - d. Se busca una imagen al azar y se comprueba con un bucle do/while si es igual a la que ya existente. Si lo fuera, se busca otra imagen al azar.
 - e. Se muestra la imagen en pantalla y se comprueba si las tres imágenes son iguales mediante la función comprobar_iguales().

Dificultades encontradas

Las dificultades más importantes que he tenido han sido las siguientes.

1. La primera dificultad ha sido estructurar el código y saber cómo empezar, y ha requerido invertir tiempo en ver como organizar el código y por dónde empezar.
2. La segunda dificultad ha sido escribir el crédito en pantalla, ya que al principio se escribía solo el número, pero desaparecía el símbolo del euro (€). Con las indicaciones de clase, se ha modificado el código y se ha añadido código HTML en el javascript para mostrar el símbolo:

```
document.querySelector("#dinero").innerHTML = credito + `<span class="euros">€</span>`;
```

3. La tercera dificultad ha sido que se mostraran los mensajes en el programa. Al principio se han hecho con “alerts” pero no se visualizaba bien. Siguiendo las indicaciones de clase, se ha conseguido hacer modificando las propiedades del CSS desde Javascript. Se ha creado una función que se llama mostrarMensaje, que se utiliza para todos los mensajes que se muestran en el programa y desde donde se abre la ventana del mensaje con la información que corresponde:

```
document.querySelector("#velo").style.display = "flex";  
document.querySelector("#mensaje").innerHTML = mensaje;
```

4. La cuarta dificultad ha sido unificar las funciones de los tres botones “Volver a tirar” en una sola función. Para realizarlo, se han seguido las indicaciones dadas en la clase de bucles para saber cuál es el botón que está activo, y se ha preparado el el siguiente código:

```
let hijosBoton= this.parentNode.children;  
let activo = 0;  
for (let k=0; k<hijosBoton.length;k++) {  
    if (this===hijosBoton[k]) {  
        activo = k;  
    }  
}
```

Comentario final

Este proyecto me ha parecido muy útil para aplicar todos los conocimientos del curso, ya que he utilizado declaración de variables, condicionales, funciones, bucles, etc. y lo he podido poner en práctica.

El proyecto me ha parecido que ha tenido bastante dificultad, especialmente en comparación con el resto de las prácticas, ya que no había una guía con los pasos a seguir, sino que he tenido que plantear como realizar el funcionamiento de la máquina. Esto ha requerido que haya tenido que invertir mucho más tiempo, pero también ha favorecido mi aprendizaje.