

UD6 – Actividad Evaluable

- Mei Mezquíriz, Paloma Sureda, Javier García

Organización del trabajo

Iniciamos este proyecto con la intención de avanzar en conjunto los programas. Generamos toda la estructura de archivos y la estructura de la carpeta del site pero al final por conflictos de horarios durante las vacaciones, el trabajo ha tenido que ser distribuido.

Resultando en que Mei ha hecho la mayor parte del quizz, Javier ha hecho la mayor parte del ejercicio drag & drop y Paloma, el ejercicio extra.

El menú fue la última página que hicimos, usando el ejercicio de drag & drop como fuente del diseño

Estructura de la web

Todos los juegos son accesibles mediante la página principal “menú.html”. Se redirige al usuario al html correspondiente del juego seleccionado.

Todos los archivos existen en la misma carpeta y no existen sub-carpetas. Tampoco se ha requerido guardar otra clase de archivos como imágenes localmente.

Explicación de los juegos

- **Quizz (quizz.html)**

Una serie de preguntas simples sobre JavaScript con 4 opciones posibles. Hay un límite de 10 segundos por pregunta y la puntuación aumenta por cada acierto en función del tiempo restante, si se falla la pregunta o se termina el tiempo, no cambia la puntuación.

Después de que el usuario interactúa con un botón, el temporizador se detiene inmediatamente y se muestra la respuesta correcta en verde, también se marca si el usuario ha pulsado una respuesta incorrecta en rojo.

Finalmente, cuando se terminan las preguntas, al usuario se le presenta su puntuación final y su cantidad de aciertos y errores.

- **Figuras y formas (dragNdrop.html)**

El juego de Drag & Drop tiene como función principal arrastrar y soltar cada figura en su correspondiente silueta, hay un contador de puntos, cada figura correcta suma 100 y cada errónea resta 50, si llegas a 0 pts, pierdes. Al igual con el contador, partes de 1 minuto para completar el puzzle si no consigues hacerlo a tiempo pierdes.

- **La caza del tesoro (wildcard.html)**

El juego Caza del Tesoro tiene como función principal hacer clic en los emojis que aparecen aleatoriamente en la pantalla. Existen tres tipos: el tesoro (💰) que suma 1

punto, la trampa (💀) que resta 1 punto y quita una vida, y el bonus (⭐) que agrega 5 segundos al contador.

El jugador comienza con 3 vidas y 30 segundos en el reloj. Si el tiempo se acaba o se pierden todas las vidas, el juego termina. Los emojis aparecen cada vez más rápidos, lo que aumenta la dificultad con el paso del tiempo. Al finalizar, se muestra un mensaje con los puntos obtenidos y se puede reiniciar la partida.

Quizz responsive

El quizz se adapta a las pantallas por la combinación de varias acciones:

1. Uso de flex para el temporizador y el marcador, haciendo que se ajusten solos al tamaño que recibe su contenedor
2. Uso de min-width y max-width en un contenedor que engloba todo, para evitar que la página se deforme demasiado ante cambios de tamaño
3. Uso de grid junto con media queries en las respuestas, alineándolas de forma diferente para adaptarlas a pantallas más pequeñas.

Estructura del código

- **Quizz**

El html incluye todas las partes de antemano y se manejan alterando su display a “none” o su tipo correspondiente con JavaScript.

Las preguntas están guardadas en un array de objetos que contiene el texto de la pregunta, un array con las posibles respuestas y un indicador de la respuesta correcta.

El código está distribuido en funciones que se llaman entre si para mantener el flujo del juego cambiando la visibilidad de las piezas, controlando el temporizador usando setInterval y clearInterval y . Se mantiene una cuenta de la pregunta actual y se compara con el tamaño del array para saber si debe empezar el círculo de funciones de nuevo o salir a la pantalla final.

- **Figuras y formas**

Se divide en funciones para cosas como el temporizador, los puntos el inicio y el final del juego. Pero la mayor parte del juego sucede alrededor de objetos hechos “draggable” en el html, las zonas donde se pueden soltar (llamadas “dropzones” en nuestro caso) y eventListeners que atienden a cuando un objeto que estaba siendo arrastrado se suelta.

Se evalúa si el objeto ha sido soltado en una zona y si esta es la correspondiente. Entonces se llama a las funciones correspondientes para manejar el incremento de la puntuación.

Cada vez que se actualiza la puntuación se revisa si el juego debe terminar comprobando si se han llenado todos los espacios para dar la victoria, o si se acaba el tiempo para dar la derrota.

- **La caza del tesoro**

El html tiene los contenedores vacíos para las diferentes piezas y es el JavaScript quien se encarga de rellenarlas.

Como el resto, las diferentes partes del juego han sido divididas en funciones que se encargan de iniciar y mantener el juego hasta que una condición llama al final, en este caso un contador de vidas y un temporizador.

Mientras que el juego sigue, es una función la que se encarga de generar los emojis, se decide aleatoriamente usando el resultado de un “`math.random()`” el tipo de emoji que tiene que ser y se coloca en una posición aleatoria del contenedor designado para el juego. Mediante un `eventListener` se atiende a si el emoji se clicka y en función del tipo que se le haya dado cambia el resultado, lo que puede aumentar tu puntuación, el tiempo restante o reducir una de tus vidas.

Al perder la última vida o el temporizador llegar a 0, el juego se finaliza y se le muestra al usuario la opción de reiniciar el juego

Problemas encontrados

No hemos encontrado obstáculos muy grandes durante la realización de este ejercicio, mayoritariamente han sido problemas en css o errores mínimos de sintaxis en JavaScript que se pudieron localizar rápidamente con el uso de la consola.

Es mencionable que, en la realización del ejercicio de formas, Javier no supo como hacer una estrella con css. El problema se solucionó usando links a imágenes csv.

Posibles mejoras

La mejora más evidente es la disparidad del diseño entre las páginas. Podríamos haber ideado un css genérico del que partir antes de programar, pero teníamos nuestras mentes puestas en priorizar la funcionalidad.

Otro un poco más sutil es el viaje entre páginas, a pesar de que el menú puede llevar a todos los juegos sin problema, pero los juegos no tienen implementado ninguna forma de regresar.