UD2E3 Juego de memoria “Simon”

Crea el video juego de memoria “*Simon*”.

La estructura de la interfaz tiene las siguientes características:

- La página se divide en 2 columnas, la izquierda para el juego y la derecha para información y control.
- La columna de juego tiene una tabla 2x2 en la cada celda será un botón de color.
- La columna derecha tiene:
    - Un campo de entrada para los milisegundos que se iluminan los botones.
    - Un campo de entrada para los milisegundos de espera entre encendidos de los botones (para poder diferenciar cuando se encienda un color varias veces seguidas).
    - Un botón para comenzar el juego.
    - Información con el número alcanzado en la mejor racha.
    - Información con el número de aciertos de la partida actual. Por ejemplo “5 de 7”.

La estructura del código tiene las siguientes características.

- La lógica del juego se define en una función autoinvocada denominada “simon”. Que expone las siguientes características:
    - Un método para obtener el array con la secuencia de colores de la iteración actual.
    - Un método para establecer el último color pulsado. Recibe el color pulsado.
    - Un método para obtener la mejor racha.
    - Un método para obtener la posición a comprobar de la racha actual.
    - Un método para iniciar el juego.
    - Un método para obtener el estado del juego, posibles valores: “parado” y “jugando”.
- La función “simon” internamente contara con el código necesario para gestionar el estado del juego y generar colores aleatoriamente.
- Los colores del juego son: verde, rojo, azul y amarillo.
- El código del juego está separado del código encargado de la interfaz.
    - Una función auxiliar para presentar la secuencia de colores con la siguiente lógica:

        - Apagar todas las luces.
        - Esperar tiempo entre encendidos.
        - Iluminar luz actual.
        - Esperar tiempo de iluminación y pasar al siguiente valor de la secuencia.
        - Gestionar el ultimo valor de la secuencia.
        - Muy recomendable definir una variable para saber si se está representando la secuencia de colores, te ayudará a descartar eventos.
    - Una función auxiliar para arrancar el bucle del juego con las siguientes tareas:
        - Si el estado del juego es “jugando” no hacer nada.
        - Iniciar juego en “simon”.
        - Mostrar la nueva secuencia después de una ronda exitosa.
    - Una función manejadora para la pulsación de los botones de color.
        - Si el juego está en estado parado o se está mostrando la secuencia de colores no hacer nada.
        - Indicarle a “simon” el color pulsado, iluminar el botón, actualizar la información. Si se pierde la partida indicarlo.
- En el evento “load” de la ventana se asocian los eventos click.

Criterios de codificación.

- Separa en archivos el html, js y css.
- Los botones de colores sólo muestran su color el tiempo que están activos, después tienen un color genérico (con la idea de facilitar el seguimiento de la secuencia).
- Si te resulta fácil emplea identificadores para los botones y <table> para la estructura del juego.

Consideraciones

Entérate de lo que tienes que construir, organízate y trocea los problemas. Después resuelve y prueba cada problema de uno en uno.

- Puedes hacerte un esquema en papel.
- Puedes crear comentarios sobre las funciones que vas a necesitar y lo que deben hacer.
- Piensa en el formato de las estructuras de datos. Que el formato que elijas facilite la posterior codificación de las distintas funcionalidades.

Define primero el html de los elementos de la interfaz, después emplea este código a modo de ejemplo para crear las funciones auxiliares.

Vas a necesitar depurar para arrancar la aplicación, los temporizadores serán un incordio. Puede ser útil usar el console.log (o debug más correctamente) para trazar las invocaciones y los estados de los objetos.

En la teoría hay muchas cosas que te facilitan la vida, si no la has leído no sabrás que existen.