Primer taller pre-parcial

16 de Octubre de 2020

Indicaciones generales

- Este taller sirve para la preparación del primer exámen de la asignatura desarrollo web, no es de obligatoria elaboración
- En e-aulas hay documentación necesaria para el desarrollo del taller, sin embargo, siéntase libre de consultar la documentación de javascript
- Las dudas del taller serán resueltas por slack nombre espacio de trabajo: desarrollo2020espacio.slack.com por el canal de dudas.

En este segundo taller preparatorio para el parcial podrás practicar todos los conceptos vistos en clase, que serán evaluados en el segundo corte.

- CSS
- HTML
- Scripting usando javascritp y querySelector
- Web Components
- Callbacks

El siguiente repositorio en github contiene un minijuego muy sencillo que te ayudará a comprender mejor los conceptos vistos en clase:

`https://github.com/DoritaSuarez/segundoPreparcial.git`

El juego consiste en una cuadricula con casillas de colores que cambian cada dos segundos y un color de referencia. El color de referencia cambia cada 2 segundos también, en este tiempo el jugador debe buscar entre 5 posibles casillas las que tengan el mismo color. Por cada coincidencia, se añade 1 punto al marcador de puntuaciones. Si la persona comete un error en su elección el marcador de puntuaciones se reinicia.

[[Parte 1 - CSS y Web Components]]

`Lo primero que vamos a hacer es construir las casillas, si no eres experto en el sistema de cuadriculas (grid-layout) de CSS sigue estos pasos, de lo contrario, puedes pasar directamente al paso 4. Te recomiendo tener abierta las diapositivas de clase y la documentación de grid-layout para que puedas consultar rápidamente cómo hacer cada uno de los siguientes pasos:`

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout
https://www.w3schools.com/css/css_grid.asp
https://developer.mozilla.org/es/docs/Web/Web_Components

- Crea un div que represente una de las casillas cuya clase sea 'casilla', añade un fondo y fija un tamaño para poder visualizarla en nuestro html.
- Usar el sistema grid-layout de CSS para construir una cuadricula de 2x2, cada elemento debe tener la misma clase que creaste en el paso 1, por lo cual deberás agregar la propiedad *display: grid;* al estilo de la casilla.
- Usar el sistema grid-layout de CSS para construir una cuadricula de 4x4, cada casilla debe ser un *div* de la clase con la que creaste la casilla. Añade espacio entre las casillas (tendrás que copiar la casilla 16 veces para poderla visualizar).
- Crear un elemento llamado "tablero-juego"que tenga la misma clase que el sistema de grid anterior y construir la misma cuadricula de 6x6 pero ahora utilizando javascript con el método *document.createElement()*.
- Crear un **web component** que represente cada una de las casillas a través de la clase Card-Game y cuyo nombre de etiqueta sea *casilla-gatito*
- Hacer lo mismo que en el paso 3. pero ahora crear las casillas a través de la etiqueta *casilla-gatito*
- Añade los detalles de estilo que consideres necesarios hasta que las casillas se vean adecuadamente en los espacios del grid-layout que creaste, selecciona 4 colores que serán los que correspondan a las casillas de color y si quieres añade alguna imagen a las casillas.
- Acomoda el espacio de juego creando un contenedor para las casillas, para el nombre del usuario, la puntuación y la carta de referencia (es decir, diseña la página donde el usuario va a interactuar).
- Crea la carta del color de referencia a tavés de un div cuyo *id* sea *reference-card* y con dos clases: *reference-card* y la misma clase de la casilla que creaste y ubicala en el espacio que destinaste en el paso 8.
- Añade los detalles que consideres necesarios.

[[**Parte 2 - Modificando atributos e interacciones**]]

En esta segunda parte del taller, vamos a hacer que la interactividad con el usuario funcione correctamente, para ello puedes realizar los siguientes pasos:

1. 1. Al web component que le da vida a las casillas, le debemos agregar una función asociada al evento çlick", llamada *changeCard()* para ello, en el callback *connectedCallback* (usa el método de *addEventListener* en el elemento)

2. En el mismo web component define la función *changeCard* y deja que tenga un mensaje en la consola *console.log(Çlick a la carta")* y comprueba en el navegador que funcione

3. En el web componet registra un atributo que te permita interactuar con el color de la carta, llamado çolorz usando el callback de *observedAttributes* y en el callback *attributeChangedCallback*, cada que el usuario de click se debe almacenar el color seleccionado en *LastClickColor*.

[[**Parte 3 - Creando las casillas cada 2 segundos**]]

`Esta parte tiene un comportamiento que vamos a aprovechar para crear callabacks con comportamientos asíncronos. Para ello, vas a crear una función que se llame 'aparecer()' esta función se va a encargar de varias cosas (yo creé un objeto con todos los colores, y demás pero basta con hacer un array con los colores):`

1. 1. Crear un numero aleatorio de casilla

2. Crear un numero aleatorio de color (posición)

3. Usando *querySelectorAll* seleccionar todas las casillas por su clase

4. Seleccionar el elemento correspondiente a la casilla aleatoria de la casilla 1 y modificar el color del fondo al color aleatorio creado en el paso 2.

5. Utilizando la función *setTimeout* retornar a los valores originales de la casilla modficando el atributo de çolor"de nuestro web component y esto debe pasar cada 2 segundos

6. Ya que tenemos la función *aparecer()* para crear un callback llamado start, este callback debe ejecutar la operación de *aparecer()* 5 veces (se modifican 5 casillas simultáneamente) cada 2 segundos.

7. Hacer que la función se ejecute infinitamente

8. Hacer lo mismo que hicimos en el paso 3 a 6 con la casilla de referencia, pero esta se ejecuta una vez cada 2 segundos, el color actual se debe almacenar en la variable *LastCardColor*.

9. Volver al web - component y modificar los elementos para terminar hacer que si *LastClickColor* es igual a *LastCardColor* se sume un punto, o si no que sea cero. Usando la propiedad *innerHTML* cambiar el contenido del espacio que asignaste para mostrar el marcador al usuario.