IES ANTONIO SEQUEROS
Financiado por la Unión Europea NextGenerationEU
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL
Plan de Recuperación, Transformación y Resiliencia
GENERALITAT VALENCIANA Conselleria d'Educació, Cultura i Esport
GVANEXT Fondos Next Generation en la Comunitat Valenciana

Lenguajes de marcas y sistemas de gestión de información.

UT 4. JavaStript.

Práctica final. Conecta 4

1. Introducción

El principal uso (aunque no único) de JavaScript es la manipulación del árbol DOM del navegador. En esta práctica se trabaja con el manejo de este árbol, accediendo, modificando, eliminando y creando nodos.

Se trata de crear el famoso juego Conecta-4 utilizando JavaScript, HTML y CSS. Este posee diferentes escenas o pantallas, cada pantalla ha de tener su propia música:

- Introducción. Una imagen en SVG (se puede usar IA para generar la imagen).
- Configuración. Se introduce el nombre del usuario, y se selecciona la imagen que se colocará en cada una de las fichas, además ha de aparecer el nombre del personaje en la ficha . **El botón de siguiente solo puede estar activo si se tienen los datos para iniciar el juego.** Las imágenes y el nombre del personaje se obtienen realizando una petición con fetch a un servidor externo u obteniendo un JSON propio, por ejemplo:

  https://digi-api.com/

  https://ygoprodeck.com/

  https://rickandmortyapi.com/documentation

  https://pokemontcg.io/

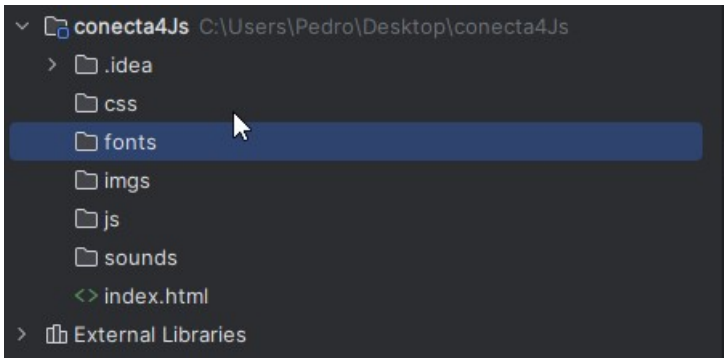
  https://docs.magicthegathering.io/

- Juego. El juego propiamente dicho. Por turnos, se hace click sobre una de las columnas, si es posible se coloca una ficha del jugador de ese turno en la primera celda inferior libre, se evalúa para ver si se ha conseguido 4 en línea (horizontal, vertical y 2 diagonales), en caso de no conseguirlo, se pasa al siguiente.
- Final. Similar a la introducción aparece una imagen es SVG/animación (se puede usar IA para genera la imagen) y los datos del ganador (nombre, imagen). Se debe poder volver al inicio.

2. Recomendaciones de desarrollo

Recomendaciones de cómo afrontar el desarrollo de la práctica, por supuesto se han de desarrolla más métodos, ficheros CSS. HTML

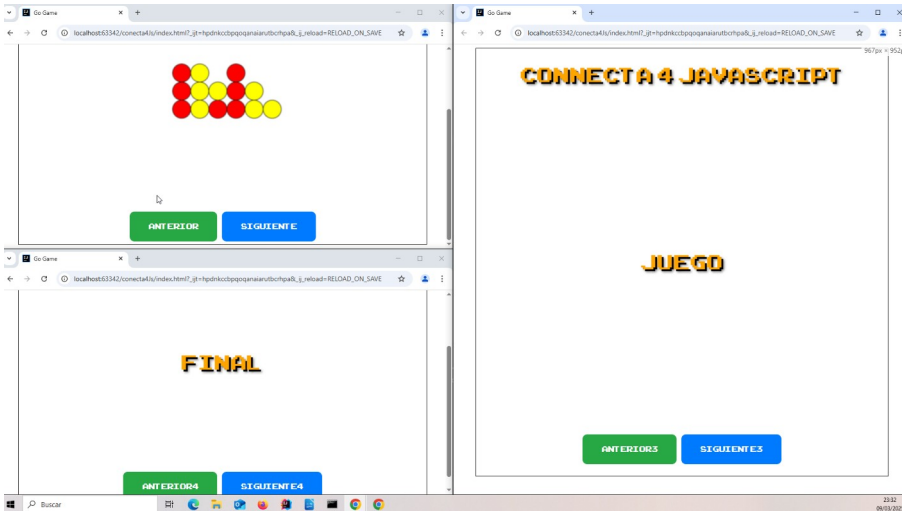
- Crear un proyecto con la estructura básica: css, js, imgs….



- Avanzar poco a poco, realizando pruebas
- En caso de no entender algo o tener dudas sobre decisiones a tomar, preguntar al profesor, debatir con los compañeros... (no usar la solución fácil de usar la IA)

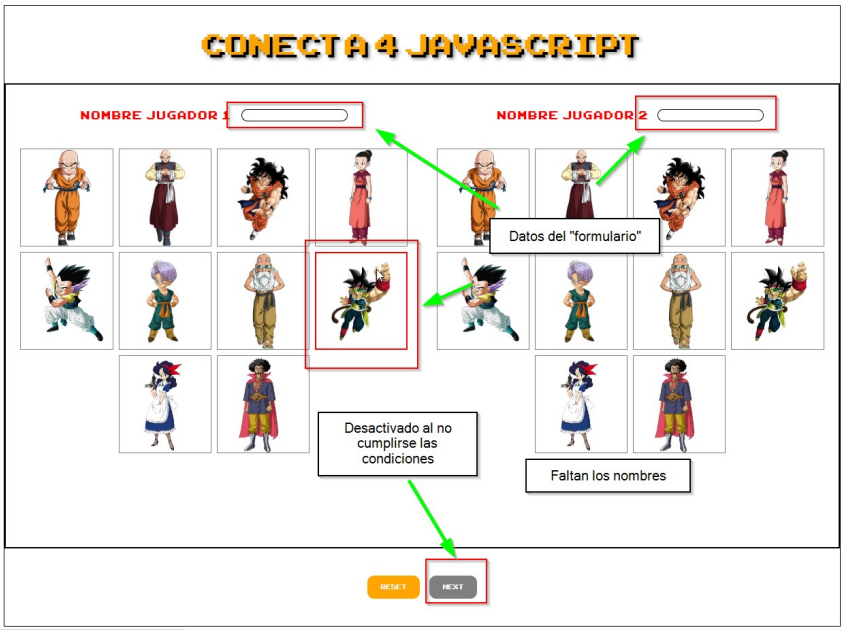
2.1. Creando las escenas.

1. Comenzar definiendo la clase abstracta escena.
2. Definir las 4 clases que heredan de escenas (introducción, configuración, juego, final).
3. Crear la clase Game que contendrá las 4 escenas en un array/lista entre otros elementos
4. Añadir el código HTML con las cuatro escenas. Por ejemplo 4 div con identificadores, y dentro de cada uno de ellos 2 botones, anterior y siguiente.
5. En index, importar el código JS, crear una variable de tipo Game que tome una etiqueta HTML para construir las escenas.
6. Añadir el código necesario dentro de las clases de forma que se pueda navegar usando los botones anteriores entre las escenas (anterior y siguiente), por ejemplo, pasando una lambda en el constructor.
7. Probar.



2.2. Configuración

1. Crear un método que realice una petición a algún servicio web (o un fichero JSON/XML creado por los alumnos, junto con las imágenes) que devuelva un conjunto de imágenes y nombres, y a partir de este crear la interfaz gráfica (incluido los escuchadores), permita seleccionar la imagen de cada jugador. **También es necesario indicar su nombre.**
2. Definir un botón para "resetear" la configuración y otro para confirmar la configuración, este solo ha de estar activo si todo es correcto.
3. Hacer que el botón de siguiente se encuentre activo solo si se cumplen los requisitos.



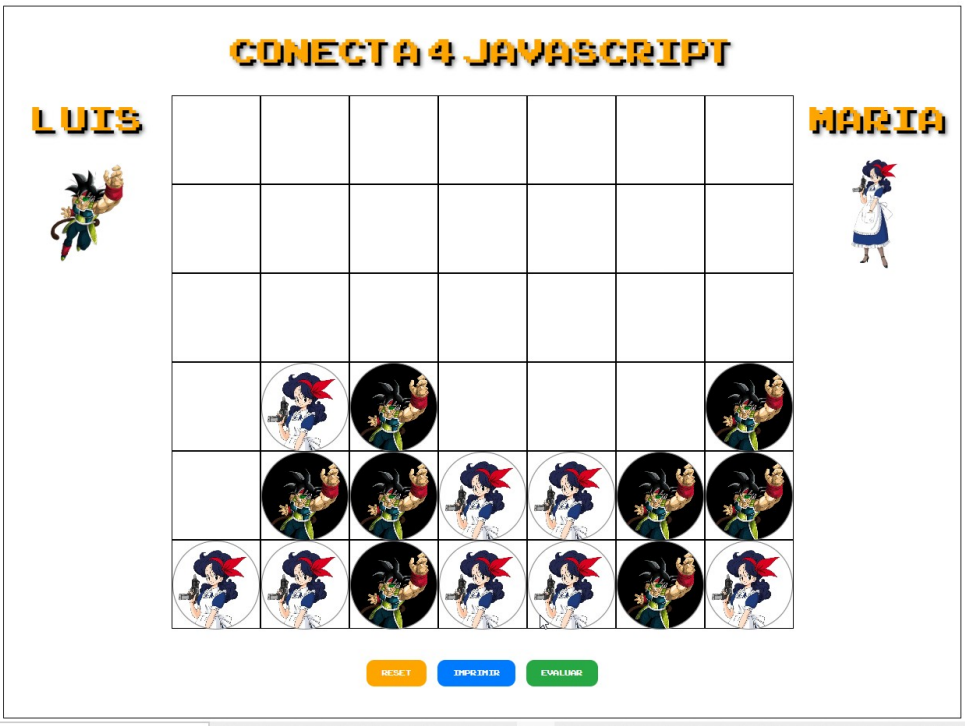
Una propuesta de los posibles métodos principales de la clase:

Método	Tipo	Descripción
`#createList()`	Privado	Obtiene imágenes de una API y las agrega a las listas de selección de jugadores.
`#selectImg1()`	Privado	Selecciona una imagen para el jugador 1, la resalta con un borde rojo y verifica si se puede activar el botón "Next".
`#selectImg2()`	Privado	Similar a `#selectImg1()`, pero para el jugador 2.
`#reset()`	Privado	Reinicia los campos de nombres y deselecciona

Método	Tipo	Descripción
		las imágenes de los jugadores.
`#enableNext()`	Privado	Activa el botón "Next" si los nombres y las imágenes han sido seleccionados.
`start()`	Público	Reproduce el sonido si está disponible.
`stop()`	Público	Pausa el sonido si está disponible.
`restart()`	Público	Método vacío (sin implementación actual).

2.3. Juego

1. Definir y crear el área, por ejemplo, dividir la pantalla en 3, 2 áreas con info de cada jugador y área central con el tablero.
2. Crear una matriz de forma dinámica de 6x7 que representará el tablero, añadiendo a cada celda escuchadores de eventos en caso de pulsación.
3. Definir un método que cree una ficha y la inserte en el lugar adecuado.
4. Codificar un método que evalué si se ha conseguido el conecta 4.



Una propuesta de los posibles métodos de la clase (falta métodos):

Método	Tipo	Descripción
`#evalBoard()`	Privado	Evalúa el tablero en busca de combinaciones ganadoras y aplica animaciones si se encuentran.
`#evalCell(row, column)`	Privado	Evalúa una celda y busca conexiones en todas las direcciones (horizontal, vertical y diagonales).
`#isEmpty(cell)`	Privado	Verifica si una celda está vacía.
`toggleTurn()`	Público	Alterna el turno entre los jugadores.
`#setInColumn(column)`	Privado	Inserta una ficha en la columna indicada, en la fila más baja disponible.
`#createNode(row, column)`	Privado	Crea un nodo de celda y le asigna un evento de clic para jugar.
`#print()`	Privado	Imprime el estado del tablero en la consola.
`#createBoard()`	Privado	Construye visualmente el tablero y lo inicializa con fichas de prueba.
`#reset()`	Privado	Reinicia el tablero eliminando todas las fichas.
`start()`	Público	Reinicia y crea el tablero, además de reproducir sonido si está disponible.
`stop()`	Público	Pausa el sonido si está reproduciéndose.
`restart()`	Público	Reinicia el tablero y vuelve a generarlo.

3. Evaluación.

1. Se define la clase Scene abstracta, junto con las clases de los 4 niveles, permitiendo la navegación entre ellas (CE b, c, d, f) (1,5 puntos)
2. Se gestiona correctamente la música (CE b, c, d) (0,5 puntos)
3. La escena de configuración crea de forma correcta las configuraciones de los jugadores (1 puntos). (CE b, c, d, e, f)
4. Se puede “resetear” la configuración (0,5 puntos) (CE f)
5. Se crean las diferentes secciones del juego (área de los jugadores, tablero...) de forma dinámica. (2 puntos). (CE b, c, d, f)
6. El juego es jugable, creando los nodos de forma correcta cuando se pulsa sobre una columna, además posee un botón para reiniciar el tablero (2,5 punto) (CE b, c, d, f)
7. El algoritmo de evaluación de conecta 4 es correcto, detectando que se han conseguido conectar 4 fichas correctamente. (2 puntos) (CE b, c, f)

La corrección se realizará de forma presencia, teniendo el alumno que responder a preguntas sobre la práctica y/o realizar pequeñas modificaciones para comprobar la autoría y originalidad de la práctica.

4. Entrega

En Aules.

1. Las memorias e informes en formato PDF. **Si se encuentran en otro formato no se corregirán.**
2. Memorias con solo capturas de pantalla no se corregirán.
3. Carpeta con los ficheros creados.
4. Fichero comprimido con el formato TUSINCIALES.zip.
5. No se admiten retrasos.

5. Resultado de aprendizaje y criterios de evaluación

RA 3. Accede y manipula documentos web utilizando lenguajes de script de cliente.

CRITERIO EVALUACIÓN.

a) Se han identificado y clasificado los lenguajes de script de cliente relacionados con la web y sus diferentes versiones y estándares.

b) Se ha identificado la sintaxis básica de los lenguajes de script de cliente.

c) Se han utilizado métodos para la selección y acceso de los diferentes elementos de un documento web.

d) Se han creado y modificado elementos de documentos web.

e) Se han eliminado elementos de documentos web

f) Se han realizado modificaciones sobre los estilos de un documento web.