**Lista tareas EMOJIS:**

Hechas Pendientes Canceladas

GitHub y compartirlo

**1-HTML:**

**Asignado a Hugo**

* **Página principal - Flex(header-body-footer**)
* **Header - Flex (botón iniciar partida- título del juego**- **contador de intentos(cambio de ubicación**)
* **Body(section(grid 4x4 hueco por carta) - reglas del juego)**
* **Crear cartas (opción 3)-introducir imaxes back/front**
* **Footer(Nombres-Fecha-HAB)**

**2-CSS**

* **Definir aspecto básico (colores, fuentes, tamaño de letra, estética de video juego)**

**Asignado Juan**

* **Aspecto página principal**
* **Asignado Juan**
* **Aspecto cartas ( aspecto boca abajo** y **boca arriba)**
* **Efecto girar cartas**

**3- JAVA-SCRIPT**

**Distribución de cartas:**

* Opción 1: Crear las cartas en JS
* Opción 2: Crear las cartas en HTML y poner el icono mediante las clases en CSS
* **Opción 3: Crear las cartas con los iconos en HTML y modificar el grid área desde JS**

**Lógica de aleatoriedad:**

Opción 1:

* Creamos arrayComparación =[ ]
* Creamos arrayCartas mediante element.children
* Iteramos mediante forEach el arrayCartas
* En la iteración creamos números aleatorios y comprobamos que no esté en arrayComparación, en caso de estar repetimos creación de número y comparación, en caso de no estar se le asigna al gridArea del elemento y se añade al arrayCOmparación mediante push()

**Opción 2:**

* **Creamos un set y metemos dentro números aleatorios hasta que su length sea 16.**
* **Creamos el arrayCartas mediante element.children**
* **Asignamos el valor del primer elemento del Set de números aleatorios al gridArea del primer elemento del arrayCartas, repetimos en todo el array (arrayCartas[i].gridArea=arrayNumeroRandom[i]**)
* También podemos modificar el gridAreaTemplate del contenedor.

**Botón de inicio de juego:**

* Cortina sobre las cartas
* Opción desde el JS variar la propiedad visivility de las cartas y el botón desde el mismo Evento
* Deshacer los eventos removeEventListener()
* Reiniciar las cartas a todas dadas la vuelta

**Botón de reset:**

* **Devolver las cartas a dadas la vuelta**
* **Volver a colocar las cartas de manera aleatoria**
* **Contador de intentos a 0**

**Lógica destapar dos cartas:**

* **Limitar las cartas a destapar de dos en dos**
* **Sumar uno al contador intentos cuando se destapen dos cartas**
* **Comparar los valores/una propiedad tras destapar dos cartas (crear alguna propiedad o asignar algún valor que compartan ambos elementos de la pareja)**
* **Tapar o mantener destapadas las cartas tras la comparación**.
* **No hay límite de intentos en el contador**
  + **sumar un intento en cada xogada.**
    - **P.ex.-Array de intentos**
* Acabar el juego al destapar todas las cartas

-P.ex-Array de puntuación

-if (lenth.flipped===16)

* Mensaje final al terminar el juego (Felicidades + Número de intentos)
* **No se puede interactuar con las cartas levantada*s***

**Reuniones:**

* 12/03/2022: Distribución de las primeras tareas y análisis de la aplicación
* 14/03/2022: FECHA SIGUIENTE REUNIÓN
* 15/03/2022

**Hitos:**

* 16/03/2022:
  + **HTML terminado**
  + **CSS aspectos básicos terminados**
  + **CSS página principal terminada**
  + **CSS cartas terminado**
  + **Análisis individual de las funciones identificadas en la reunión 1.**

**e se sobra tempo…**

1. Crear usuario e gardar puntuación
2. Alternativas a repetición de código(crear cartas automaticamente, segun o número de parellas que queira xogar o usuario)(podese reducir o tamaño dos arrays, facendo unha duplicacion dos elementos)
   1. crear array de emojis, con lonxitude máxima, igual ó número de cartas máximo que se podan xogar.Os obxecto deste array, teran o emoji e o valor comparativo para saber se son iguais
   2. determinar o número de cartas a xogar
   3. aleatorio SET entre 1 e o numero de cartas que se queran xogar.
   4. crear un array de cartas(sen emoji) con lonxitude igual ó número de cartas que sen queran xogar.(BARALLA).
   5. combinar os tres arrays(emojis,baralla, e aleatorio)
      1. aleatorio–posicion no grid
      2. emojis-imaxe+valor comparador para igualdade
      3. baralla-cartas a xogar/plantilla de carta
   6. modificar o grid do taboleiro según o número de cartas xogadas
3. incluir temporizador(console.time()//console.timeEnd())
4. axustar mellor a accesibilidade. Excede o campo que tenta abarcar o exercicio en si.(consultar con Samuel, preferiblemente facer os outros puntos da lista)
5. **modularización/cleanCode**

**HTML**

* ~~Vincular <script> (Posicionamento e nome do arquivo<script>)Modularización;estructuración de código~~
* ~~Vincular CSS(nome do arquivo <link>)~~
* ~~<html lang=”es”>~~
* ~~Aria-label(accesibilidade páxina)~~

**En ppio o screen reader, lee as figuras das cartas sen problema.Hai que destapar e volver a clickar(posibilidade de facer cun só click).ó ppio ,cando se carga a páxina, lee todas as cartas(comprobar cando están ocultas(botón start)).Posibilidade de incluir a posición no aria-label, e cando se destape, activar o evento que añada o aria -label,para saber que figura e en que posición está.**

* ~~“icon” páxina~~
* ~~Botón de reset(resetear partida en calquera momento)~~
* ~~Icono “HAB”~~
  + ~~Descargar~~
  + ~~Cartas~~
  + **~~Páxina (é demasiado grande e non queda ben ou non son capaz de facer que quede ben)~~**
* ~~cambio posición marcador~~
* **~~“literatura”(contido do xogo e do documento)-Pendiente versión Juan~~**
* **~~elemento de emparexamento/propiedade do obxecto(class en html, comparar valores en JavaScript)Hugo~~**

**CSS(cartas)**

* ~~marxes,posicionamento,tamaño(según necesidades grid)~~
* ~~media queries~~
* ~~marcador para puntuación~~
* ~~footer diferenciado del resto pagina~~
* ~~style de los botones~~