

1 fast leere HTML-Seite

Memory Board $\hat{=}$ Flexbox
deklariert nur in CSS

ts erzeugen von 3 divs
 $\hat{=}$ 3 Buttons .createElement
EASY AVERAGE HARD

- Eventlistener für jeden Button
- alle 3 Buttons sollen wieder verschwinden "hidden"
 - jeweiliges Kartendeck soll erscheinen
 - jedoch jedes Mal in einer anderen Reihenfolge

Funktion zum Shufflen eines Arrays
(Fisher Yates Algorithmus)
(! Es sollen Pöchen erscheinen!)

erzeugen eines Memory Boards
 $\hat{=}$ einer Flexbox .createElement

.createElement \rightarrow divs
 $\hat{=}$ Karten, Kartengröße zuvor in CSS,
Funktion, die entweder 8, 16
oder 32 Karten / divs erstellt &
das Memory Board erstellt

Aufbau
des Spieles

1

alle divs / Karten
haben mehrere
Eigenschaften / types
Object: cards (32 cards)

text:
color:
picture:
background:

wederum in
CSS gesetzt
(Textgröße,
Bildgröße)

Vergleichsoperatoren
passen die Karten ism?

if (card 1 == card 2)
• verschwinden beide
• score ++
else
• setTimeout
• Karten umdrehen

Karten werden
klickbar EventListener,
2 ausgesuchte Karten
kommen in ein Array
selected [], background
wird "hidden"

gehören die Karten
NICHT zusammen,
drehen sie sich wieder
um
setTimeout

1. Zwischenstopp:
man kann alleine spielen

2

neues Array
Karten, die sich auf der Spielfläche befinden []
(8, 16 oder 32)

function, die 2 Karten aus diesem Array aussucht
(Math.random) ! dürfen nicht dieselben sein

mit setTimeout werden die 2 random Karten
umgedreht: background = "hidden"

Vergleichs-funktion vergleicht
if (card 1 === card 2) { Karten verschwinden
• splice() }
computer funktion wieder wiederum, nochmals
aufgerufen
else { wieder umdrehen }

Nutzer-funktion wird wieder aufgerufen /
User kann wieder klicken / aufdecken

Gewinner funktion

if (computerscore > userscore) {
... }
else { ... }

1 Text soll erscheinen

um spiel neu zu starten:

• reload } einfach die
Seite neu
laden

2. Funktion des Gegners
kommt hinzu +
Gewinnerberechnung &
reload-funktion

3