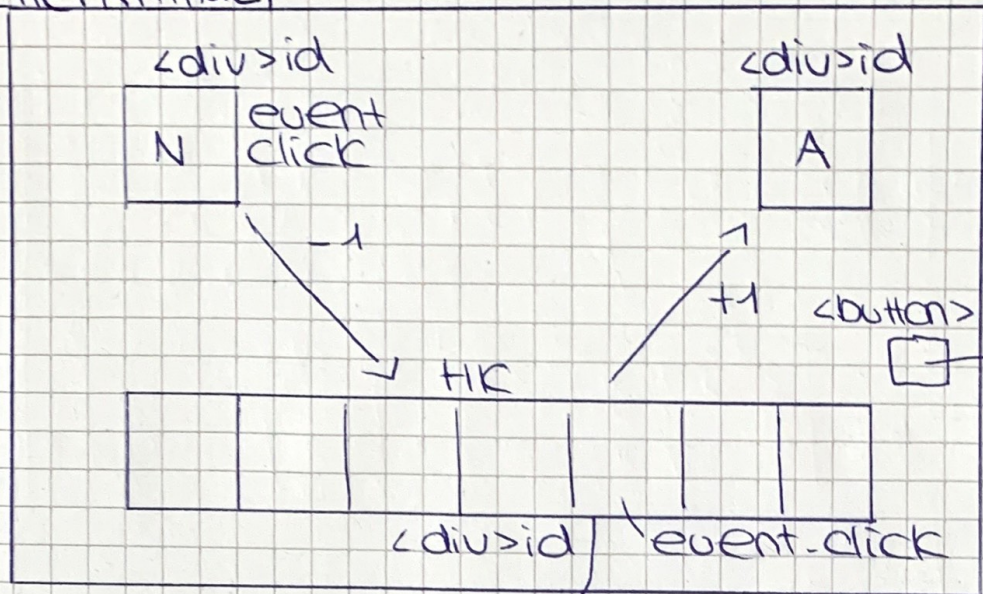


1. Interaktionsmöglichkeiten (Computerseiten (HTML))



Klassen nach Farben
 - green
 - yellow
 - red
 - blue

Spieler

klickt auf Handkarte
 klickt auf button
 klickt auf Nachrichtenbereich

2. Anwendungsfälle

Spieler &

System und

Handkarten < Nachrichtenstapel

Karten aus Nachrichtenstapel holen

Anzeige

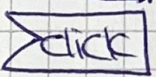
① oder



auf Nachrichtenstapel

Gibt Karte aus NS aus

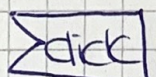
②



auf einzelne Karte aus Handkarte

Karte wird auf Ablagestapel gelegt

③

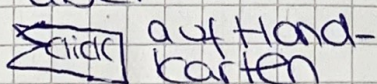


auf button Array.sort

Karten auf Handkarten werden nach Alphabet (Farbe) geordnet

Nachziehbereich leer

kein `click` auf NS mehr möglich aber

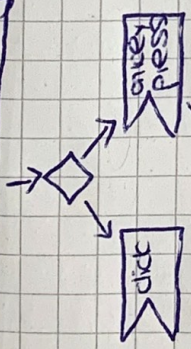


auf Handkarten

Anwendungsstelle bei naldwald
als Architektdiagramme

① nachziehen

handcards: string[]
nachziehen: string[]



random Karte aus nach-
zieh wählen -> math.floor
math.random

Karte in handcards[]
gepusht -> handcards.push(cards(...))

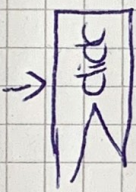
Karte aus nachzieh[] löschen
cards.splice(...)

return handcards
nachziehen



② ablegen

handcards: string[]
ablage: string[]



Karte aus handcards[]
wählen

Karte in ablage[]
gepusht -> ablage.push

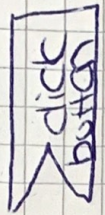
Karte aus handcards[]
löschen cards.splice

return handcards
stapel



③ sortieren

handcards: string



Karten nach
Farbe sortieren
handcards.color.
sort()

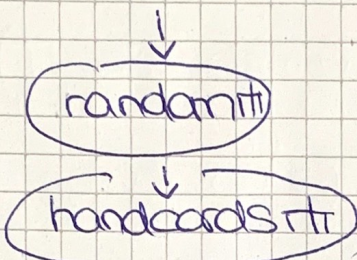
return
handcards




```
interface Card
color: string
value: string
```

```
let colors: string[]
let values: string[]
let cards: string[]
```

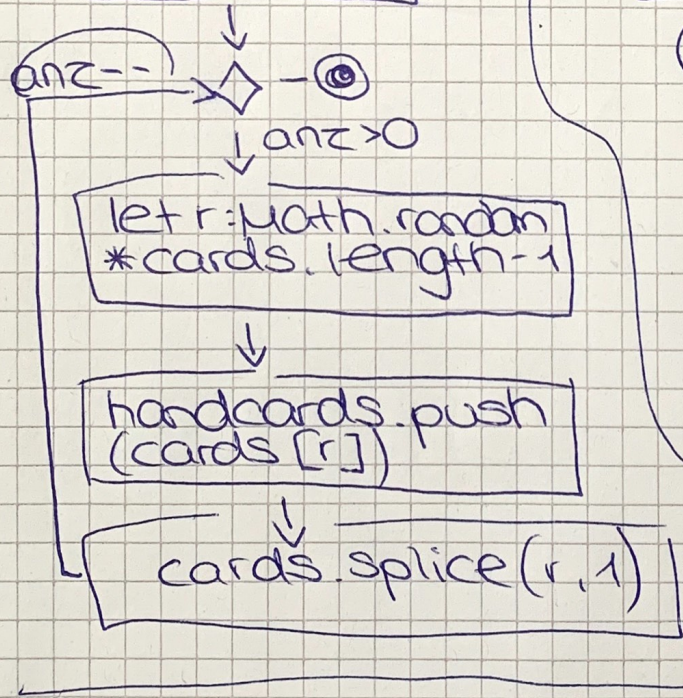
```
farbe: number = 0
```



↳ event für Nachrichtenstapel, Handkarten und sortieren erstellen

random

```
anz: number = x
```



handcards

```
let farbe: number = 0
```

