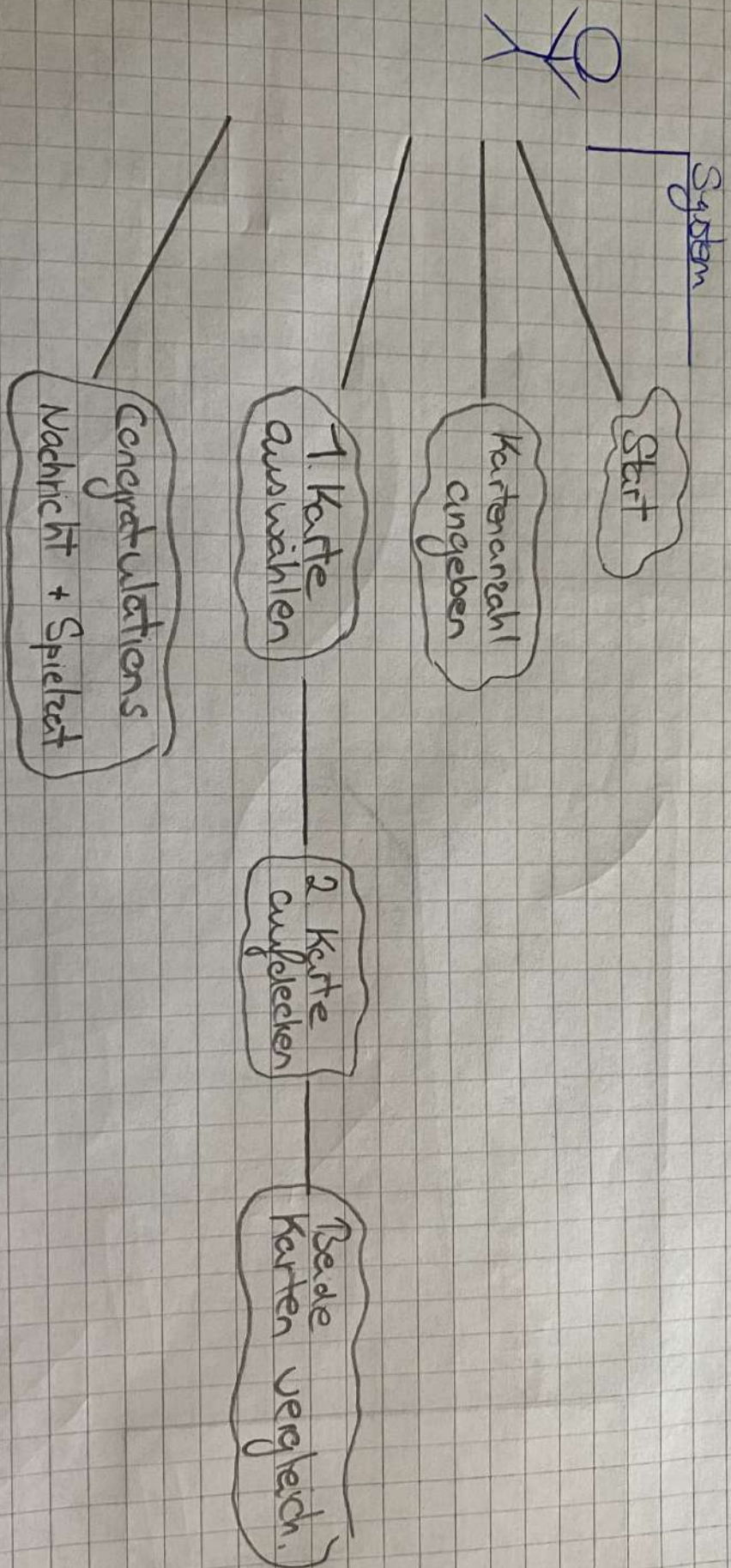


# Use-Case Diagram



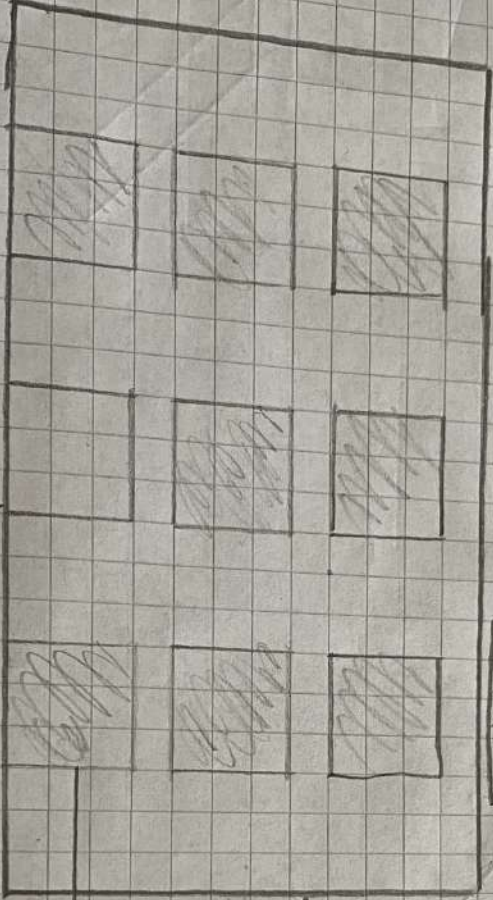


# USER INTERFACE

①

<h1>  
<h2>

Memory von Tim



<h1>  
color = black

<div>  
class = "aufgedeckt"  
<div>  
class = "verdeckt"  
back-c: green black  
Housevert

class = "korrekt"  
=> visibility = hidden

<div>  
class = "verdeckt"  
back-c: green black  
Housevert

<div>  
id = playarea  
Housevert

<select>

congratulations  
"00:00"

<h2>  
color = green

②



# \* Anmerkungen

```

let numcardpairs: number
let cardIcon: number = 0
let cardArray: number[] = []
let cardArray: number[] = []
let cardArray: number[] = []
let checkResult: boolean = 0

```

install load-Listener

create memoryCard

createElement("div")

push memoryCard in "card Array"

> load

handleLoad

\* cardUnreve  
\* cardsUnreveArray

handleLoad

numcardpairs

"choose pair of cards 525"

create gamefield

gamefield

create memoryCard

random cards

(math.random)

add card to  
→ gamefield

start timer

> click on ~~memoryCard~~

click Handler

event Event

let target: HTMLDivElement  
= event.target

\* Berechnung aus IF

compare Cards (if/else)

if else

if Same  
→ cardIcon

.classList.add("korrekt")

checkResult()

.classList.remove("aufgedeckt")

classList.add("verdeckt")

cardsOpenArray = []  
cardsOpen = 0  
check Result

checkResult

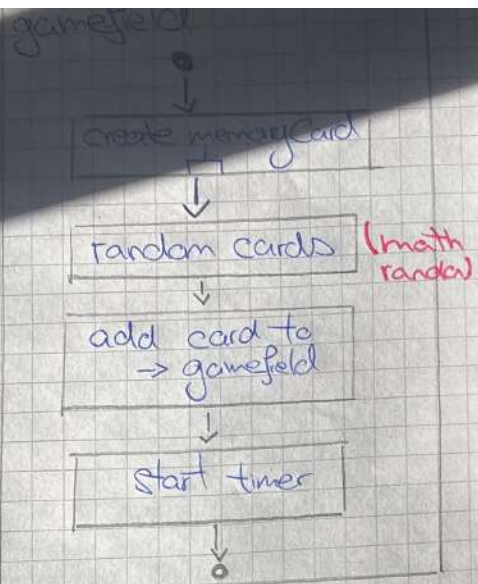
if (checklist.length = 0)

alert with message

reset

Timer  
→ vergess  
\*  
stop timer



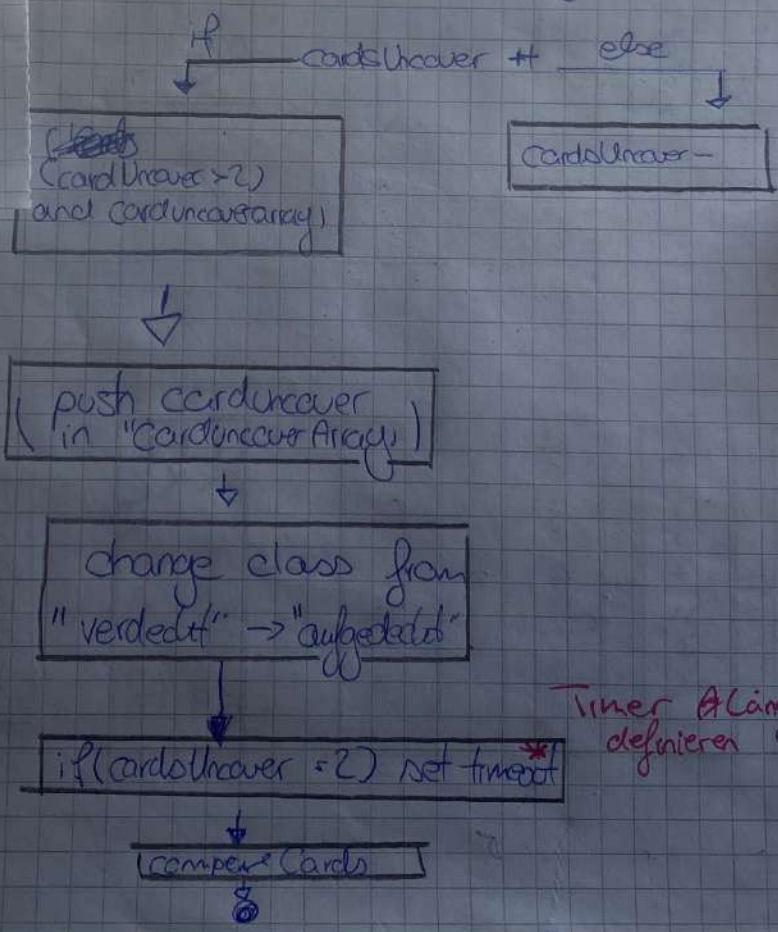


yCard

ent

### clickHandler

let target: HTMLInputElement =  
\_event.target



gt = 0)

message

\*

Timer  
vergessen

Stop timer

Timer A Länge  
definieren

\*