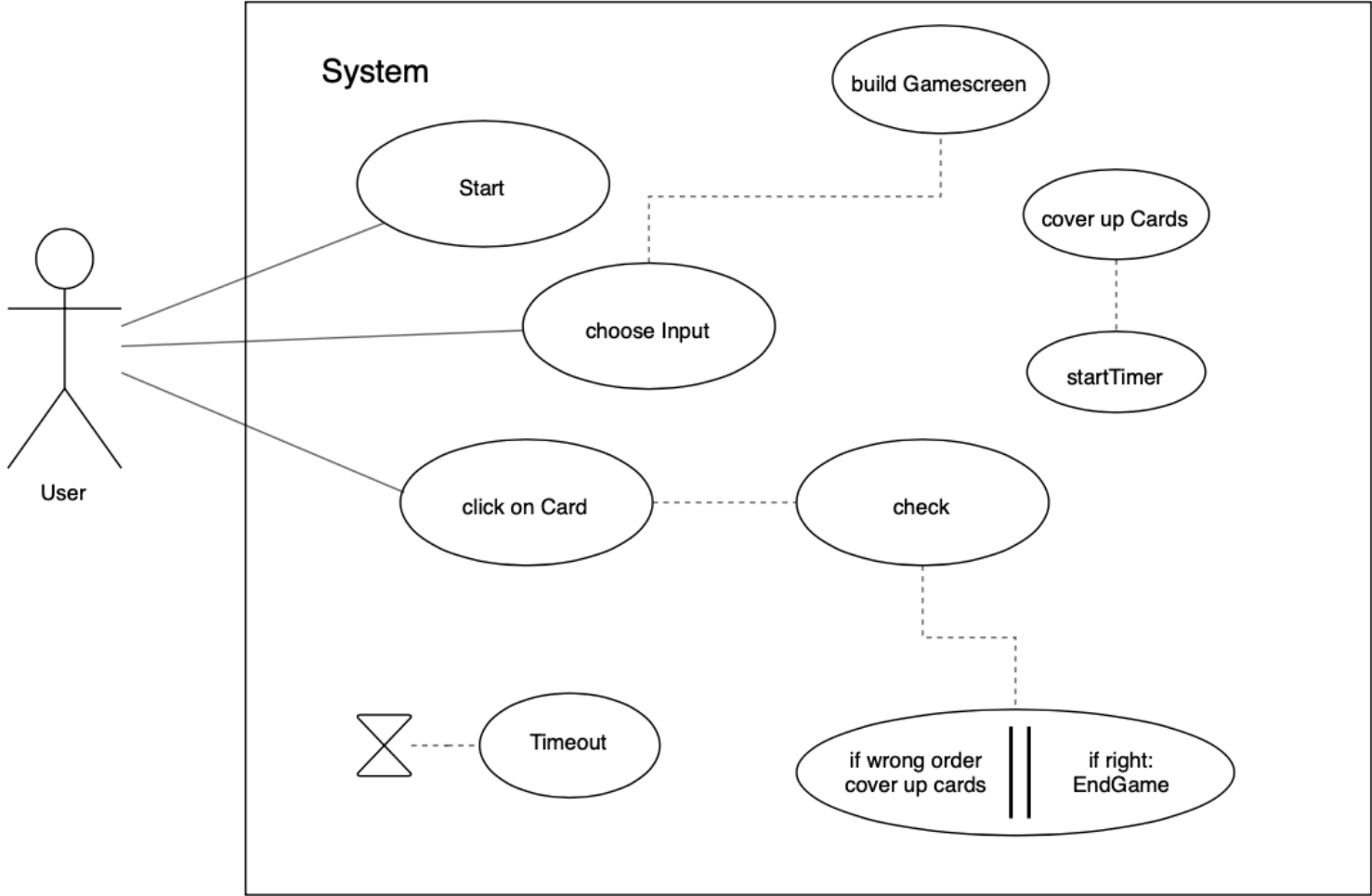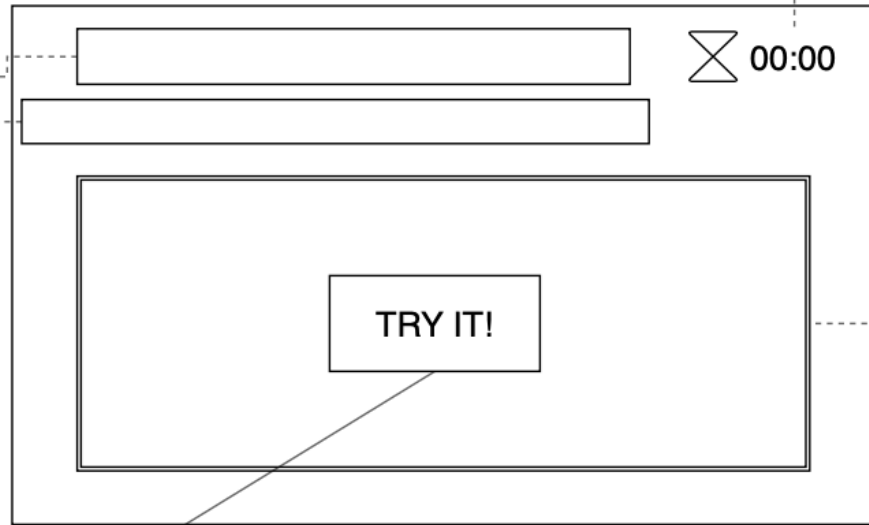# L02.2_Sequenzmemory: Use-Case-Diagram

L02.2_Sequenzmemory: UI-Scribble

1.
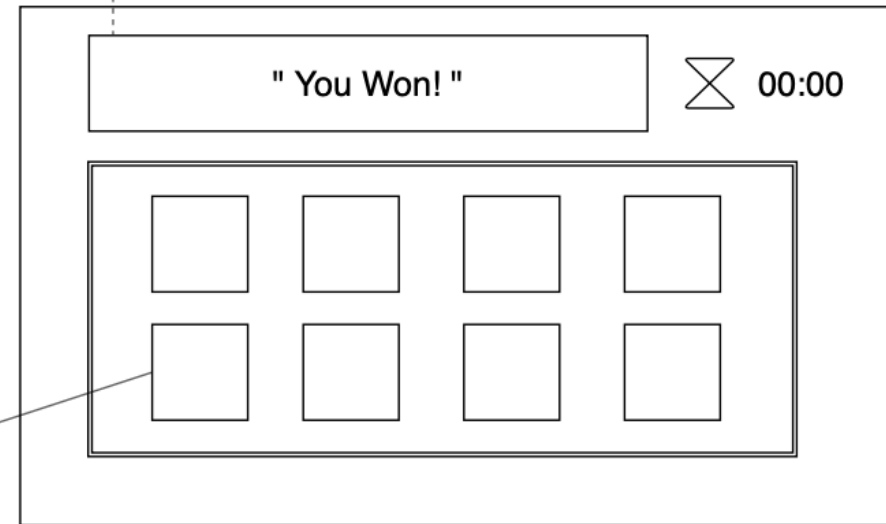
<span> id= timer

<h1>

<p>

00:00

TRY IT!

<div> id= memory (css)

<span> click

<h2> window.alert

2.

" You Won! "

00:00

<span> click

L02.2_Sequenzmemory: Activity Diagram

load

handleLoad

click on button

chooseInput

click on Card

clickCard

clickCard

_event: Pointer Event

- turn choosenCard and show letter
- install eventListener on the Span-Element
- startTimer
- checkCards

let chooseInout: string []
let input: string []

install loadListener

handleLoad

install clickListener on button

chooseInput

_event: Pointer Event

create a prompt box which ask the user for a message

set class for a <div> element with id= "memory" and style id in css --> doc.getElementbyId...

createGame

createGame

createCards:
let content: string[]
let symbol: string[]
let color: string[]
let cardNumber: number

make an Array out of String --> const myArray= input.split("");

Shuffle Array with Math.random()-0,5 + split String into an Array

create Cards(HTMLElements) and assign them a value. --> create Span-Element + set class Name + add id="memory" to doc. as child + add selectedInput[I] as innerHTML and also as a value or class for comparison

place Cards on field and show shuffledCards with input

startTimer for 3 sec., when time is up, turn over Cards --> setTimeout

clickCard

checkCard

else / **[if(input==myArray.length]**

- flipCards
- endTimer
- loseMessage -->window.alert

- check if input is equal to the length of Array with the letters
- remove Card
- create a query to see whether the card was the last card

[if(!=lastCard)] → clickCard

- endTimer
- winMessage -->window.alert

startTimer

let timer: document.createElement("Span");
timer.className("countdown");

- create a countdown, counting down with setInterval

- when the time is up
  --> counter=0
  -->window.alter("You lost!")

- clickCard