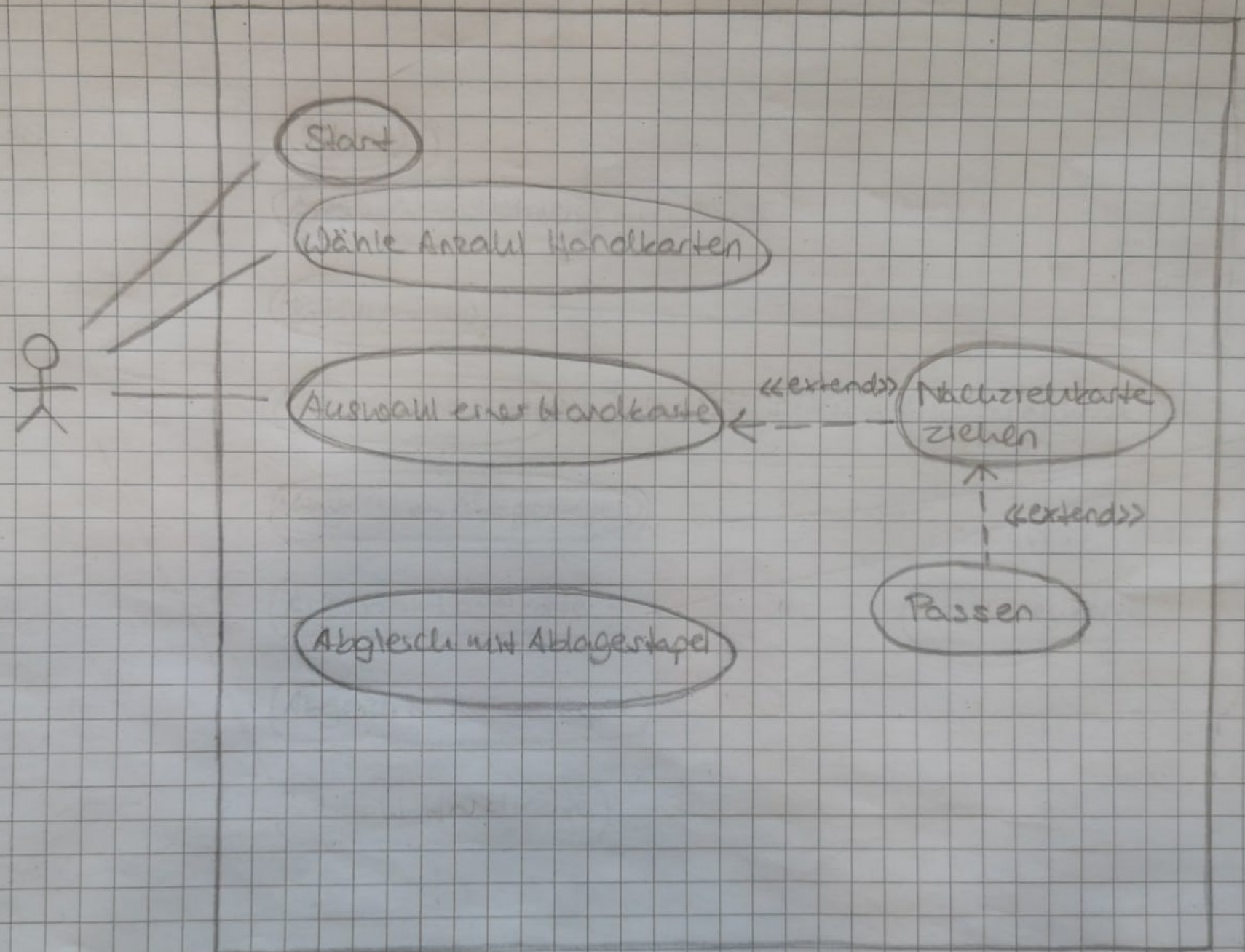
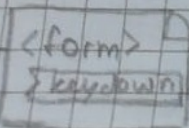
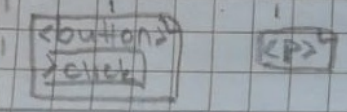
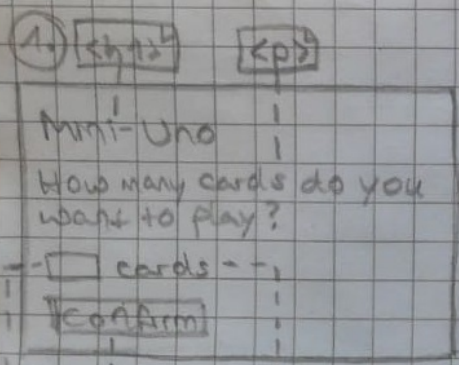
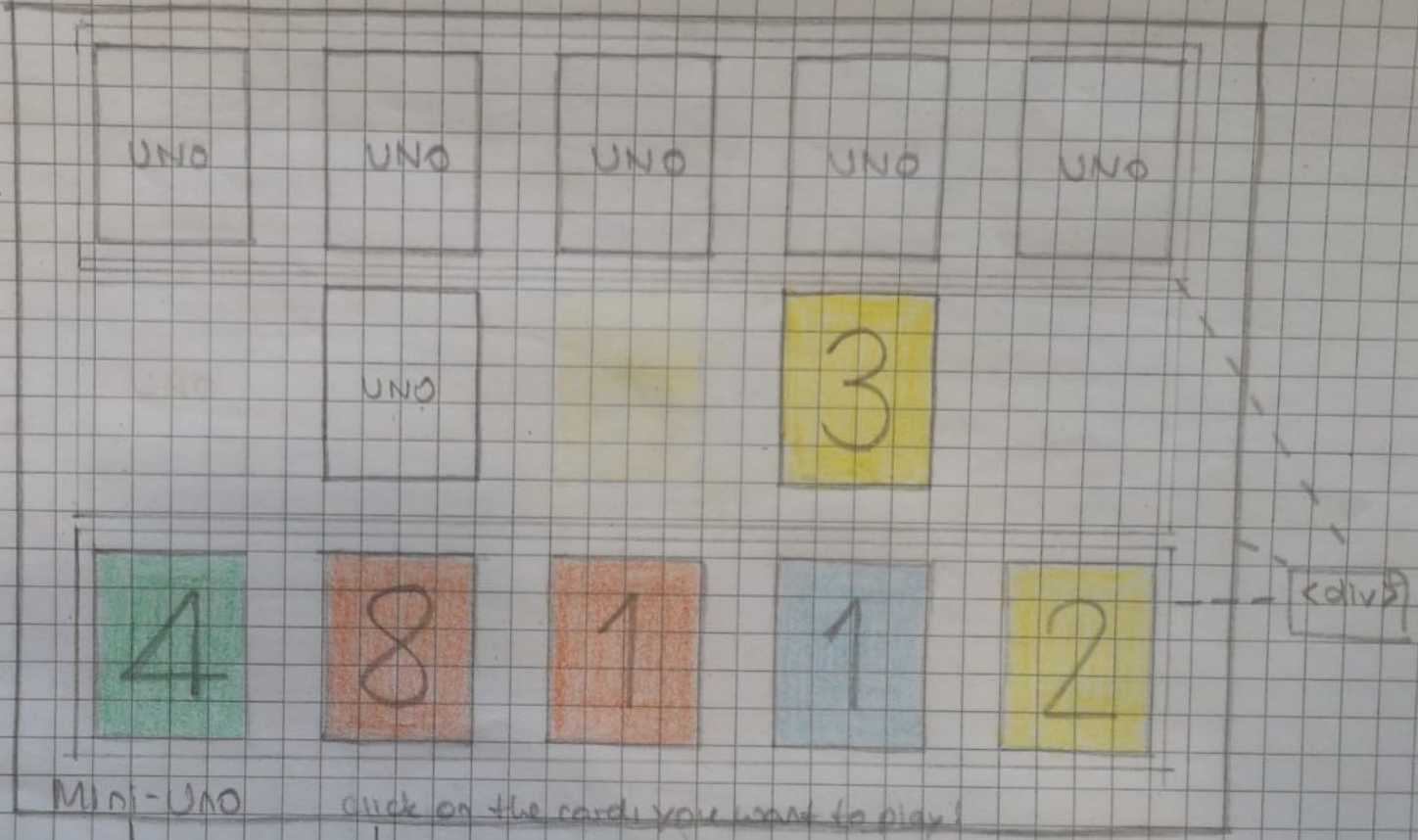


Mini-Uno: Use-Case-Diagram



Mini-Uno: UI-Skizze

2.



Mini-Uno

click on the cards you want to play!

`<h1>`

`<p>`

LO2-Elemente Aktivitäts-Diagramm Min-Uro

vorbereitung.html

main
(vorbereitung.html)

Erzeuge HTML-
Dokument

Installiere einen
Click-Listener auf
den Start-Button
im HTML-Dokument



click

Eingegebene Zahl in
Local Storage speichern

a href auf HTML-
Dokument (spiel.html)



main
(spiel.html)

Erzeuge HTML-
Document (Spielbrett)

Spiel vorbereiten

Spiel spielen

Spiel vorbereiten

interface Karte { Wert=
number, Farbe= string };

Stapel: Karte[] = [(0, "grün"),
(1, "grün")];
ablage: Karte[] = [];
spielerKarten: Karte[] = [];
gegnerKarten: Karte[] = [];

ClickListener wird auf die
Spielbrett Div installiert
(Leg, Karten, Stapel und Button)

Spielkarten ausstellen

Spielkarten ausstellen

spielerKarten und gegner-
Karten-Arrays mit zu
Beginn angegebener
Kartenzahl befüllen
(for-Schleifen mit
math.random)

+
ablage-Array mit einer
zufälligen Karte befüllen
(math.random-Zahl
für stapel[i])

Spiel spielen

Spielerzug

Karte wählen

Karte von Stapel
ziehen r1

Passen

Karte wählen

Karte legen

Meldung:
Gewonnen!

Gegnerzug

Jede Karte wird einzeln
durchgegangen und
überprüft

Karte von Stapel
ziehen r1

Passen

Karte legen

Meldung:
verloren

spiel.html

(Karte von Stapel
ziehen)



Jeweiliger Karten Array
um eine weitere Karte
aufstocken
(math.random Zahl
für Stapel [i])

