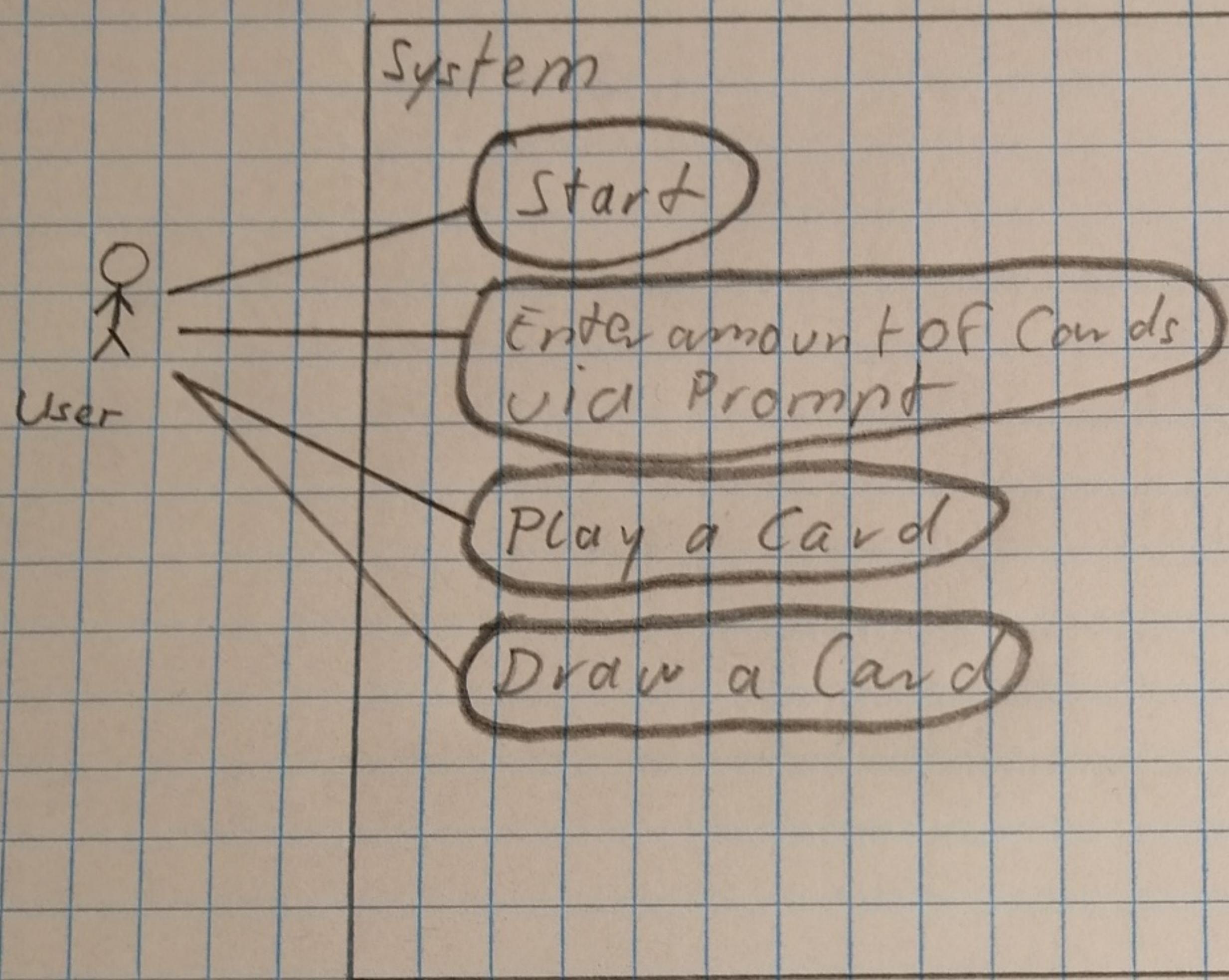
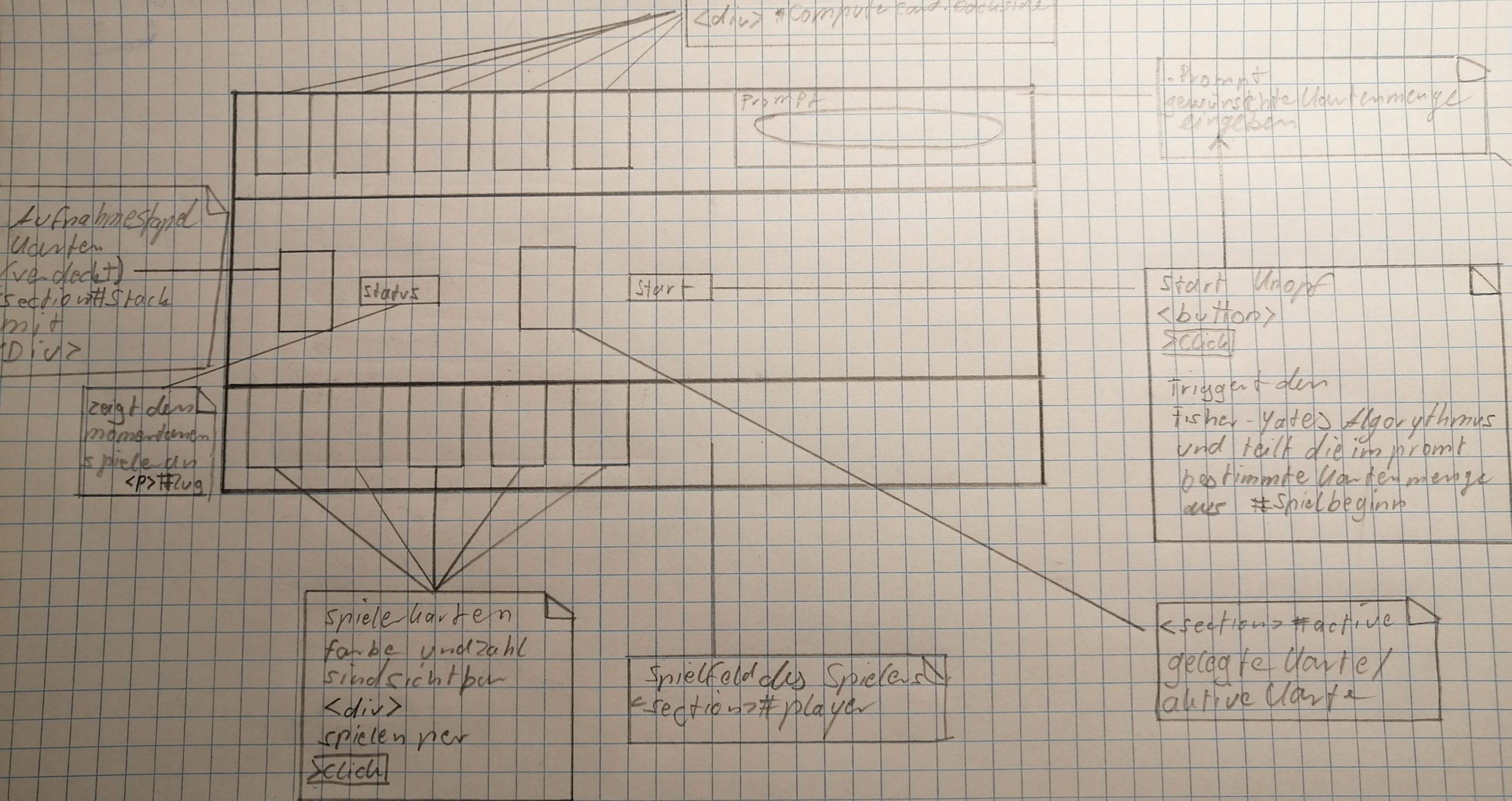


# Mini Uno Use Case - Diagrams



# Mini Uno Ui - Scribble



Kun

Inter-Card  
Farbe : string  
zahl : number

Wertverzeich : Werte [ ]  
(array aller existierende  
Werte)

function  
KartenGenerieren()

playerCards [ ]

Karten [ ]

gespielte Werte [ ]

spielbare Werte : Montage

momentanerSpieler : Boolean

Werte genommen : Boolean = false

function Spielbeginn (click)

if

Boolean  
momspieler

true

false

Function  
playThisCard

click

rti

Function  
Opponent

rti

if player.cards.length  
== 0

else

funktion

Karten generieren

(let i = 0 (number))

let hfarbe : string : void;

if i ==

hfarbe == purple

i++

let i = 0 = " "

let newCard: Karte

farbe : hfarbe = " "

zahl : s = " "

j++

Ukartendek.push newCard

↓

if i <= 9

false ↓ true

Funktion  
spielbeginn

gesamtes Blattendekk  
mit dem Fisher - Yates  
Algorithmus mischen

Prompt  
wie viele Karten  
sollen ausgeteilt  
werden

i = " "

i--

Playercards.push  
kiKarten.push (gegnerkarten)

if i > 0

true

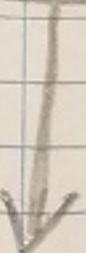
false

function  
updateHtml (playercards)  
Update Html (Gegnerkarten)

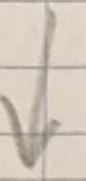
m

updateHtml (erste Karte))

Funktion updateHtml



(Div erstellen)



nummen und  
farbe aus  
array zuweisen

<P>  
background

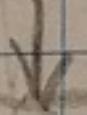
Funktion PlayThisCard



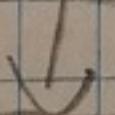
if CardToPlay.farbe == sphante.farbe ||

CardToPlay.zahl == Sphante.zahl

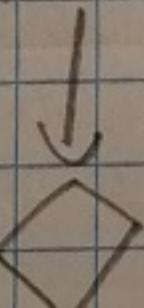
X → Funktion static Card



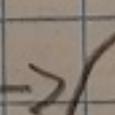
Gespielte Karten.push (CardToPlay)



if player.cards[i] == 0



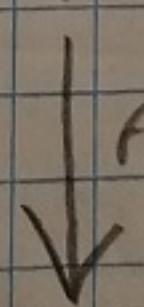
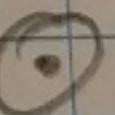
true



Status

<P>

Gewonnen!



false

Function Opponent

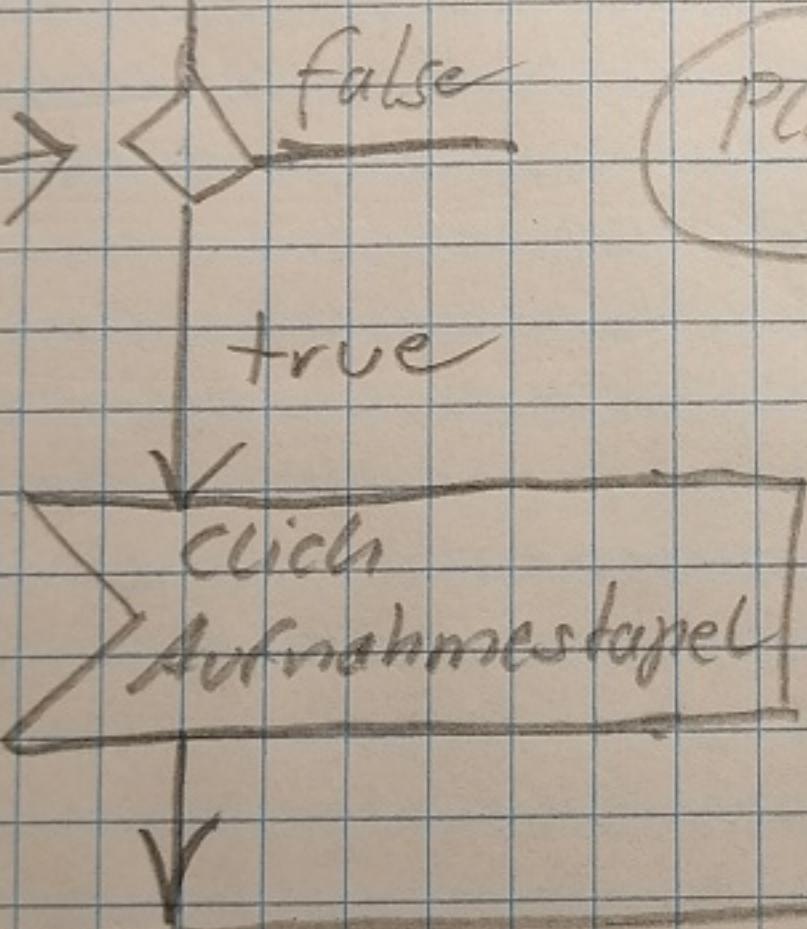
## function takecard

IF : Boolean = karte genommen = false

&&

Playercards.farbe! = sPlauste.farbe

Playercards.Zahl != sPlauste.Zahl



Playercards.push (Ukartendeck.length - 1)  
Ukartendeck.splice (Ukartendeck.length - 1

update HTML (playercards)

karte genommen = true

Funktion Opponent

Let i = 0

Let couldplay : boolean = false

$\rightarrow i < \text{gegnerkarten.length}$

false

If Gegnerkarten[i].Farbe == Spielkarte.Farbe

Gegnerkarte[i].zahl == Spielkarte.zahl

false

couldplay : boolean = true

Spielkarte = Gegnerkarten[i]

gespieltekarte.push(Spielkarte)

gegnerkarten.splice(i, 1)

funktion

updateHtml(gespieltekarten);

updateHtml(gegnerkarten);

Boolean  
momspielen  
true

gegnerkarten.push(

Ukartendeck[Ukartendeck.length - 1]);

Ukartendeck.splice(Ukartendeck.length, 1);

