

Funktionale Analyse:

Auf einer Landingpage wird der Nutzer begrüßt und die Anwendung vorgestellt. Nach den einleitenden Worten gibt es unten auf der Seite zwei Buttons: Klickt man auf „Anleitung“, so wird man zu einer Seite weitergeleitet, die dem User erklärt, wie er mit der Anwendung interagieren kann. Klickt man auf den „Start“-Button, der sowohl auf der Landingpage als auch der Seite der Anleitung steht, gelangt man direkt zum Editor, wo man ein Bild erstellen, speichern und laden kann.

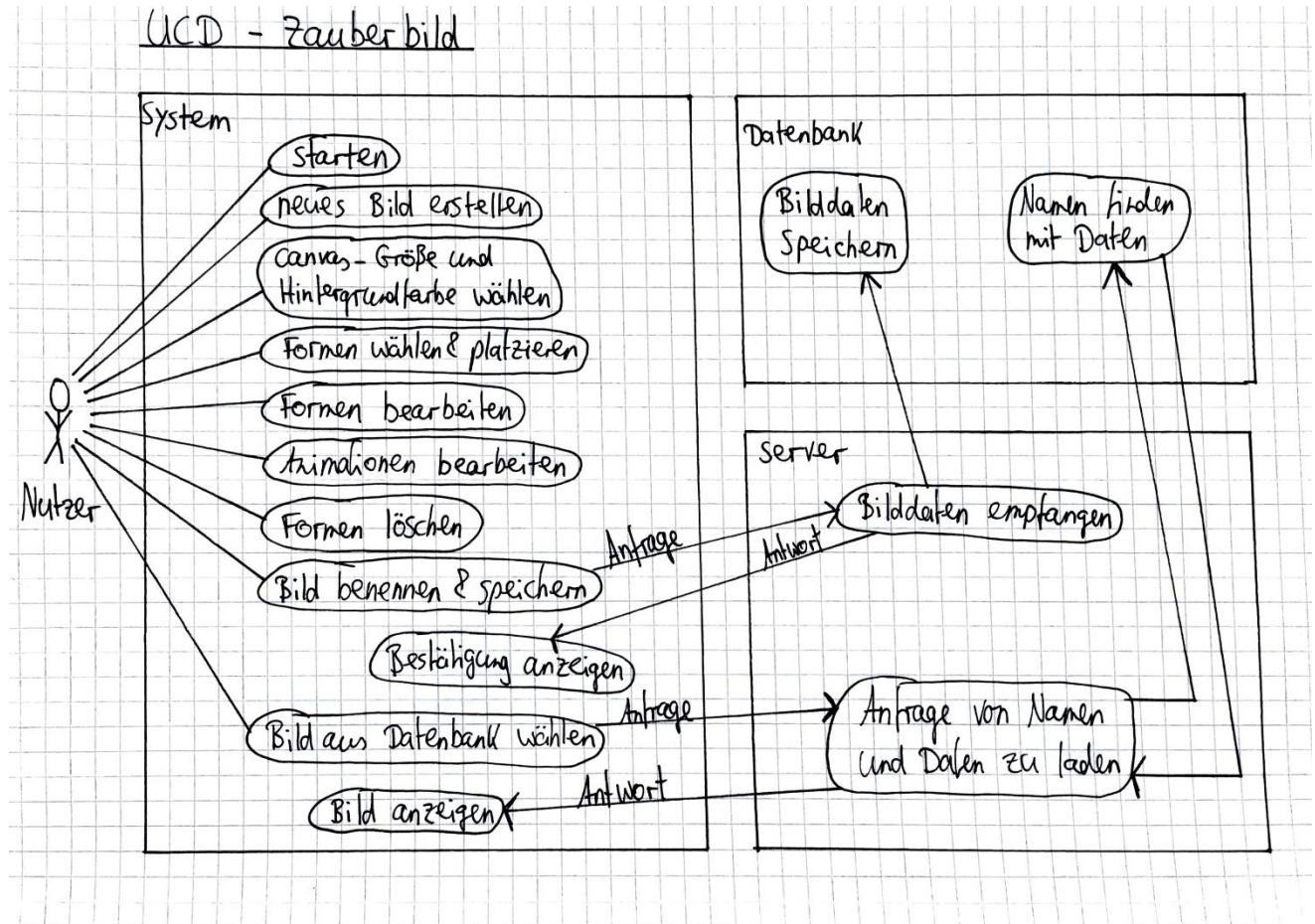
Die Interaktionsmöglichkeiten im Editor im Überblick:

- Der Nutzer kann die Größe (Breite und Höhe) sowie die Hintergrundfarbe seiner Zeichenfläche frei wählen. Um die Dimensionen einzustellen, muss nur in das jeweilige Feld geklickt, ein Wert eingegeben und mit der Entertaste bestätigt werden. Daneben wird es ein Farbfeld geben, mit dem man aus (fast) unendlich vielen Farben eine auswählen kann, die als Hintergrund der Zeichenfläche genutzt werden soll.
- Dem User stehen verschiedene Formen zur Verfügung, die er mit einem Klick auf der Zeichenfläche erscheinen lassen kann.
- Jede Forme kann animiert werden. Dazu muss die Form zunächst in der Übersichtsliste ausgewählt werden. Anschließend können die Skalierung, Position, Farbe und Rotation des Objekts individuell angepasst werden. Außerdem kann die Position einer angewählten Form mit einem Klick auf die Zeichenfläche neu gesetzt werden.
- Um eine Form zu löschen, muss sie angewählt sein. Mit der Entf-Taste kann sie anschließend gelöscht werden.
- Am Ende kann das erstellte Zauberbild mit einem Klick auf den Button „Speichern“ in einer Datenbank gespeichert werden. Hierzu wird der Benutzer aufgefordert, einen Namen für sein Bild einzugeben.
- Ebenso ist es möglich, abgespeicherte Bilder aus der Datenbank zu laden und weiter zu bearbeiten. Dazu muss er auf den Button „Laden“ klicken und ein Bild, aus der darauf erscheinenden Liste, auswählen.

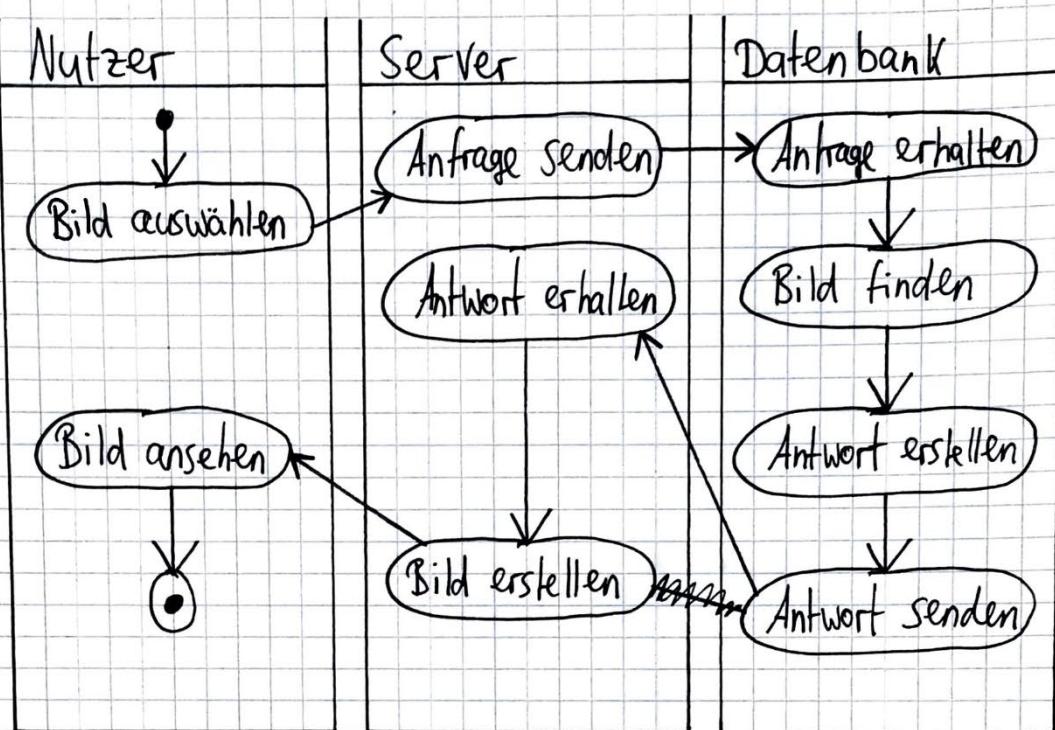
Die vielen Interaktionsmöglichkeiten sollen dem Nutzer das Gefühl vermitteln, etwas Besonderes und Einzigartiges zu erschaffen. Die Individualität eines jeden Zauberbilds steht im Mittelpunkt und der User soll sich als Künstler verwirklichen können. Um das magische Gefühl bei der Benutzung zu verstärken, wird anstelle des Cursors ein kleiner Zauberstab erscheinen. Akustisch wird man zudem von der Titelmusik aus *Harry Potter* begleitet, während man im Editor „zaubert“.

Die Anwendung wird für PC umgesetzt. Der Grund hierfür ist die geplante Bedienung mit Tastatur und Maus. Wenn auch einige Elemente, wie die Einstellung der Hintergrundgröße oder das Auswählen von Formen, mit Touch-Events umgesetzt werden könnten, so stößt die Anwendung spätestens beim Verschieben der Elemente oder der Farbauswahl an ihre Grenzen. Hier kann mit einer Maus wesentlich präziser interagiert werden als mit dem Finger auf einem (ohnehin schon) kleineren Display. Ein weiterer Grund ist, dass Formen mit einem Klick auf die Entf-Taste gelöscht werden sollen. Dies soll verhindern, dass aus Versehen ein Symbol gelöscht wird. Diese Taste gibt es bekanntermaßen auf Tastaturen von mobilen Endgeräten nicht.

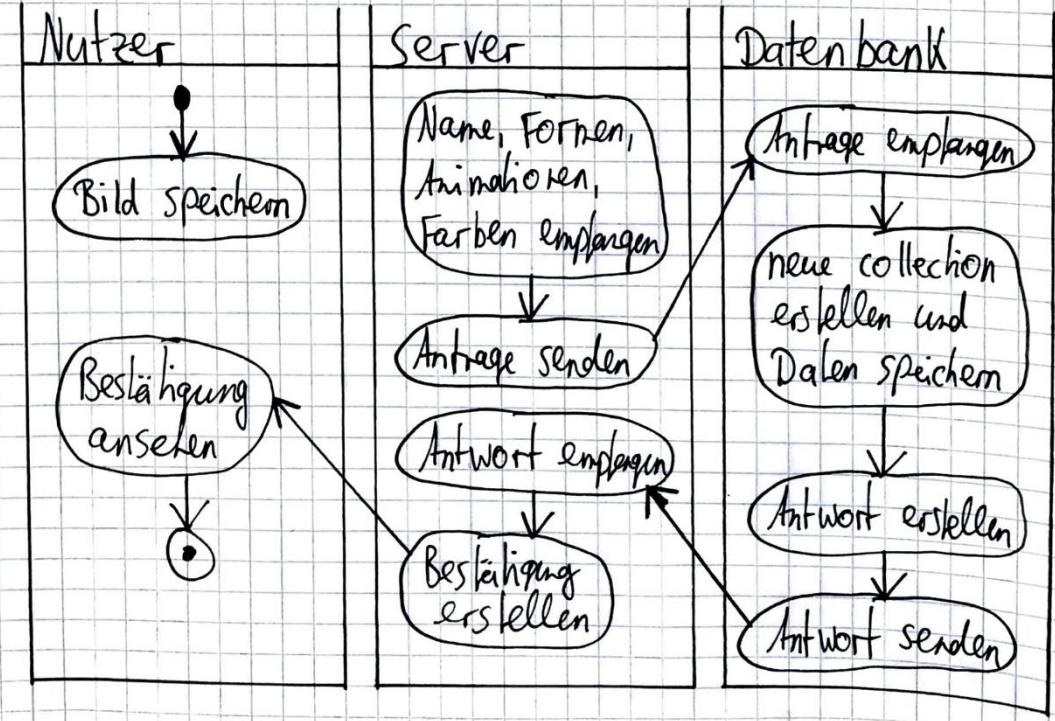
Bei allen Freiheiten, die der User haben soll, müssen bestimmte Interaktionen natürlich unterbunden werden. So soll es z.B. nicht möglich sein, zwei Bilder mit demselben Namen in der Datenbank abzuspeichern, um Verwechslungen vorzubeugen. Im Falle, dass ein Zauberbild mit einem Namen gespeichert werden soll, der schon existiert, wird der Nutzer aufgefordert, einen anderen Namen einzugeben.



Ein Bild laden (Schwimm bahn en)



Ein Bild speichern (Schwimm bahn en)



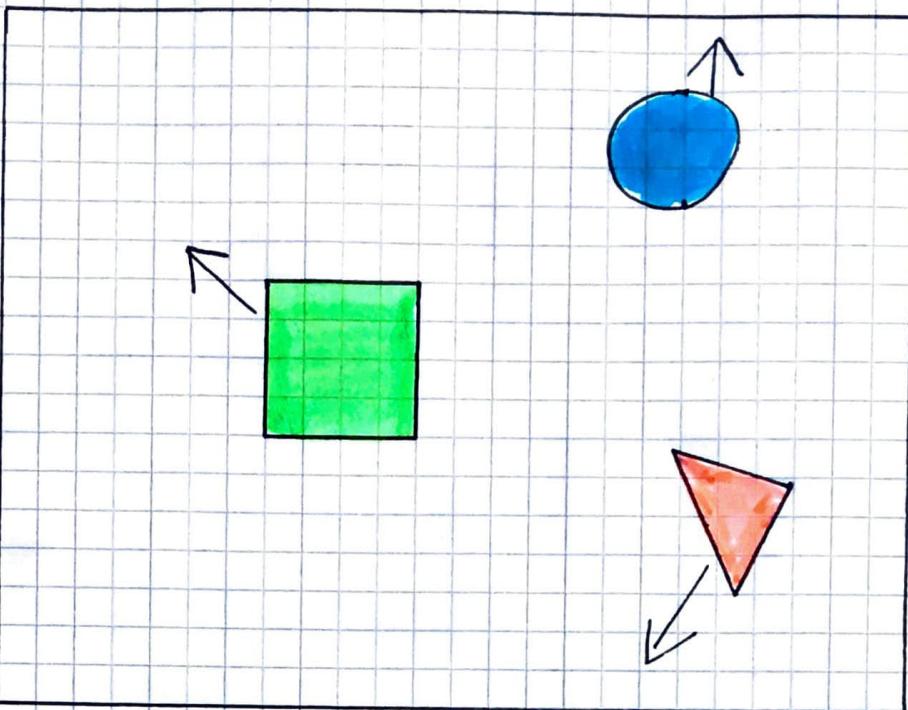


Bild 1:

Formen bewegen sich im Raum.
Die Pfeile zeigen die (zufällige) Bewegungsrichtung an.

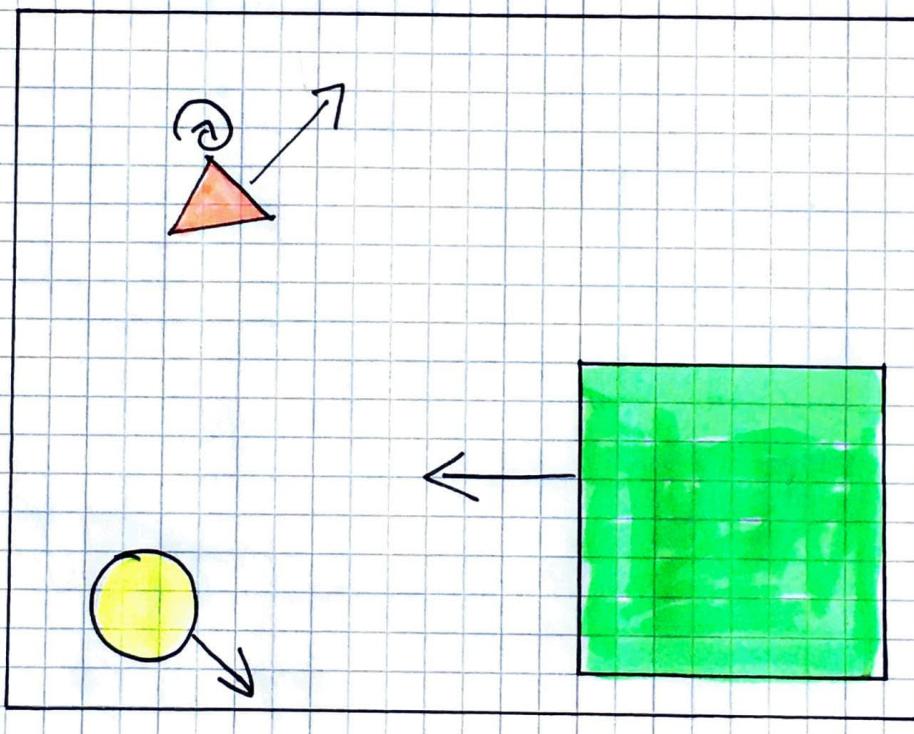
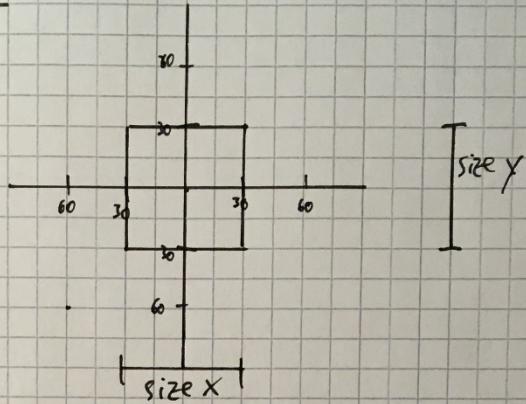


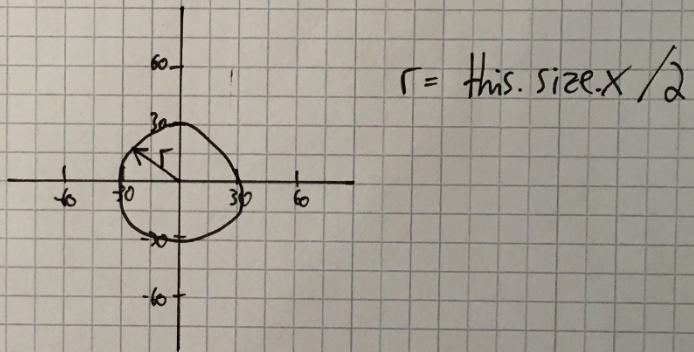
Bild 2:

Die Formen wurden verändert.
Das Quadrat wurde skaliert,
die Farbe vom Kreis geändert
und das Dreieck hat eine
Animation in Form einer Parabolion erhalten.

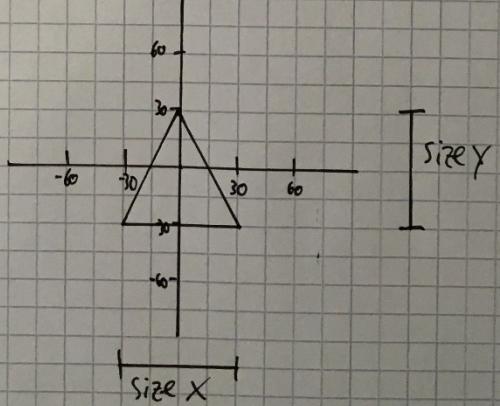
Quadrat:



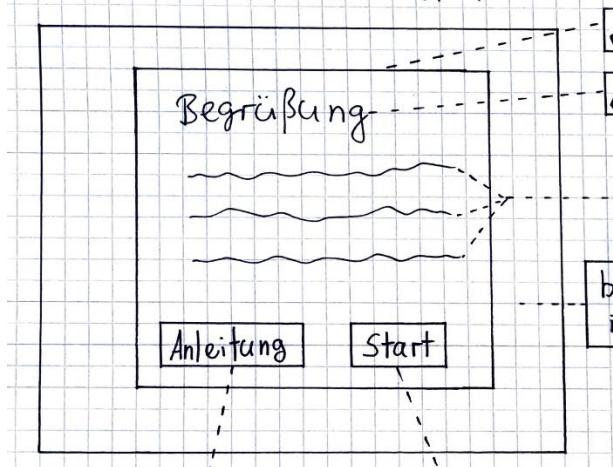
Kreis:



Triangle:



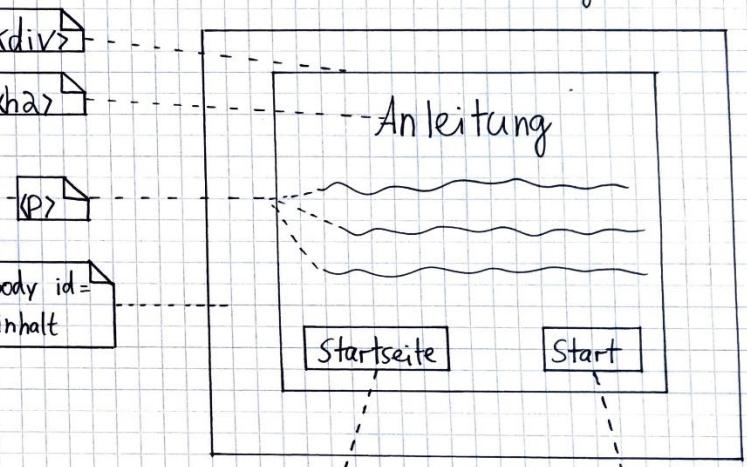
User Interface - Landing page



`button `

`button `

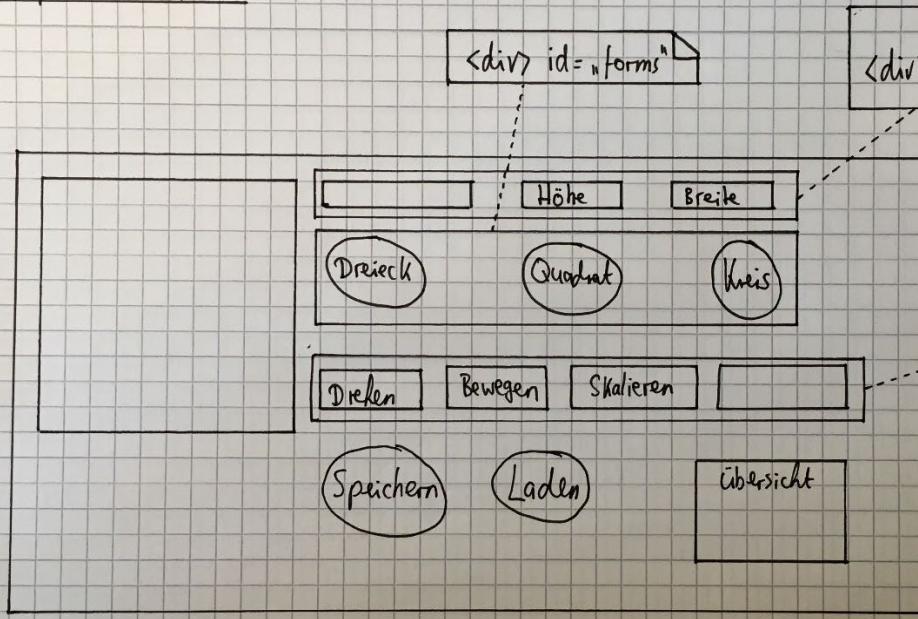
User Interface - Anleitung



`button `

`button `

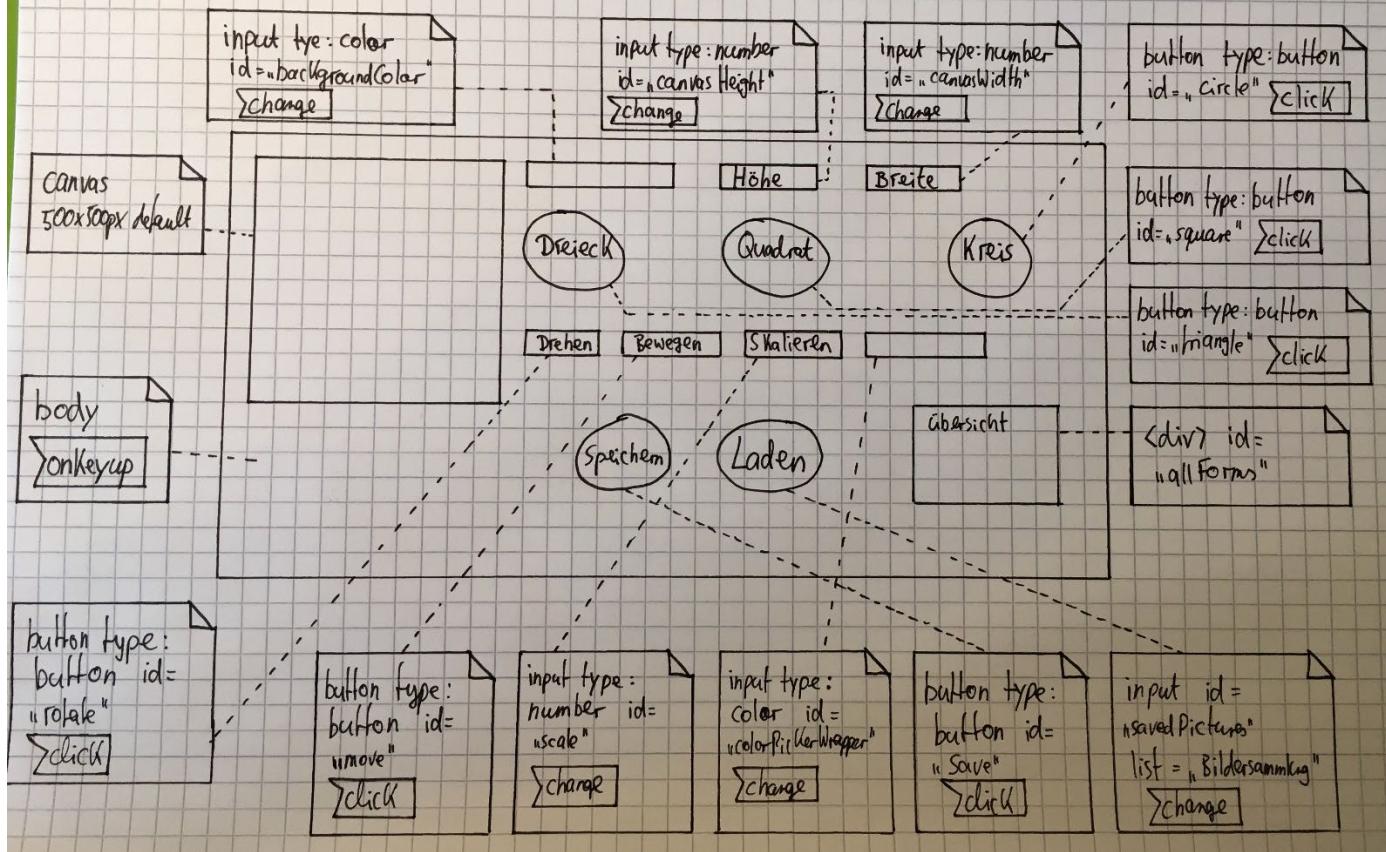
User Interface - Editor



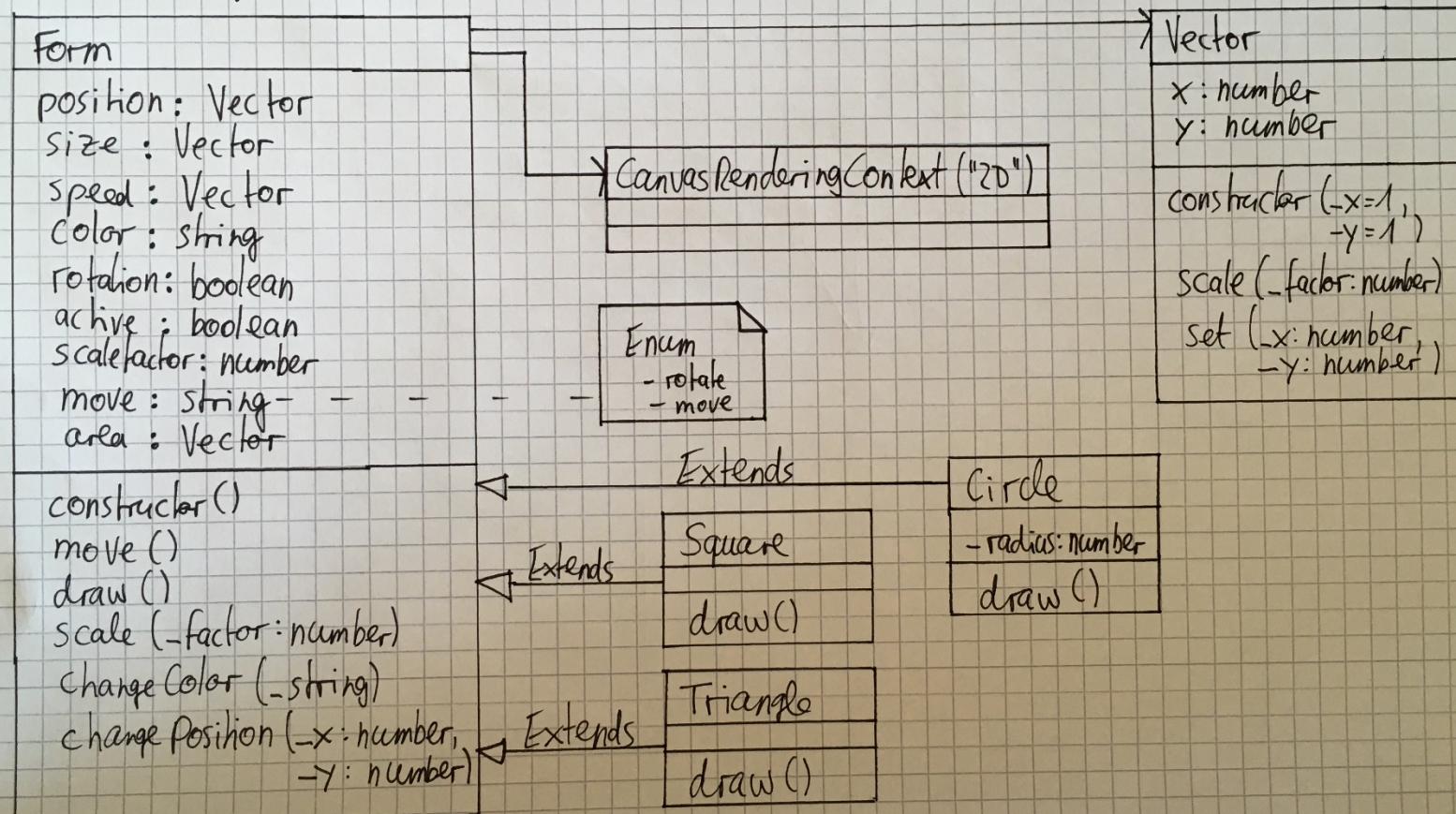
`<div id='canvasstyle'>`

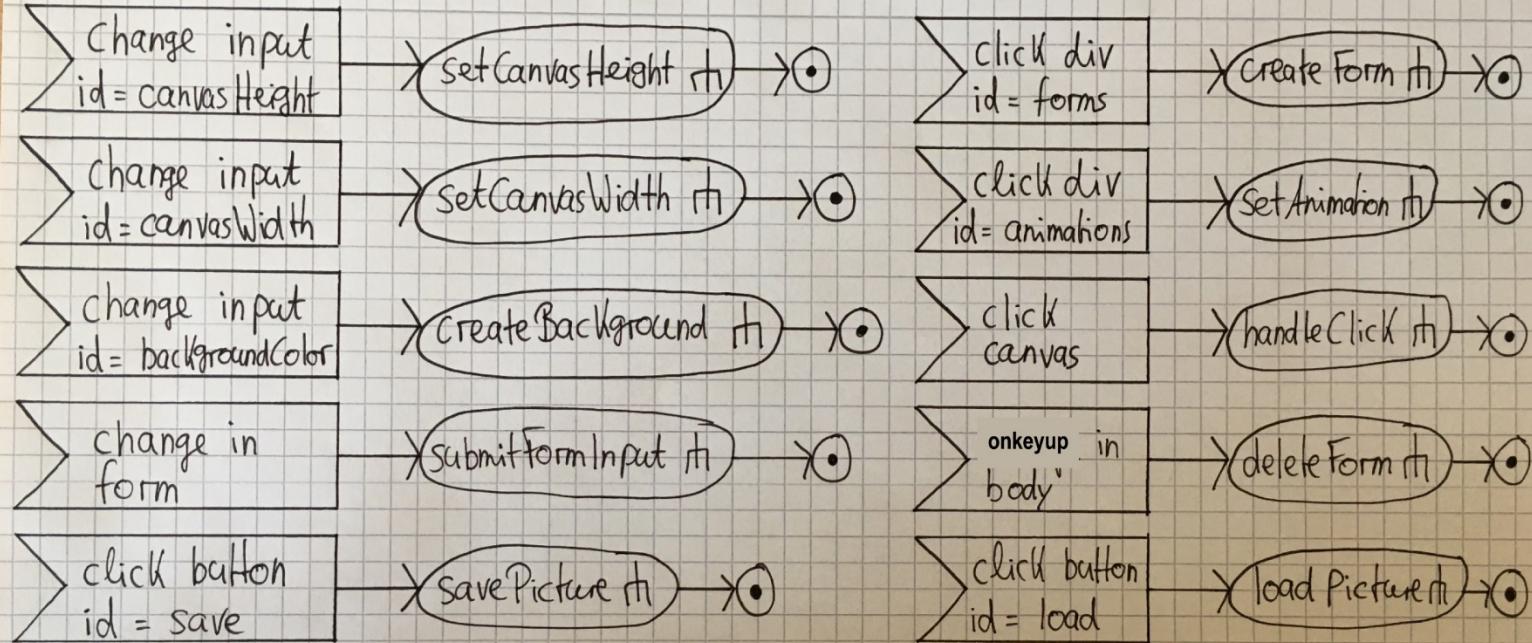
`<div id='animations'>`

User Interface - Editor



Klassendiagramm





```
canvas : HTMLCanvasElement  
Crc2 : CanvasRenderingContext2D("2D")  
height = canvas.height  
width = canvas.width  
figures : Form[]  
backgroundImageData : ImageData
```

load listener installieren
mit handleLoad fn

changePosition

```
-x : number  
-y : number
```

this.PositionX = -x
this.PositionY = -y

scale

```
-factor : number
```

this.size = (-factor)

changeColor

```
-newColor : string
```

this.color = -newColor

handle Load

```
Canvas = get canvas  
CFC2 = canvas.getContext ("2D")
```

click- und change listener
installieren und Input-
felder erstellen

500x500 Rechteck
erschaffen

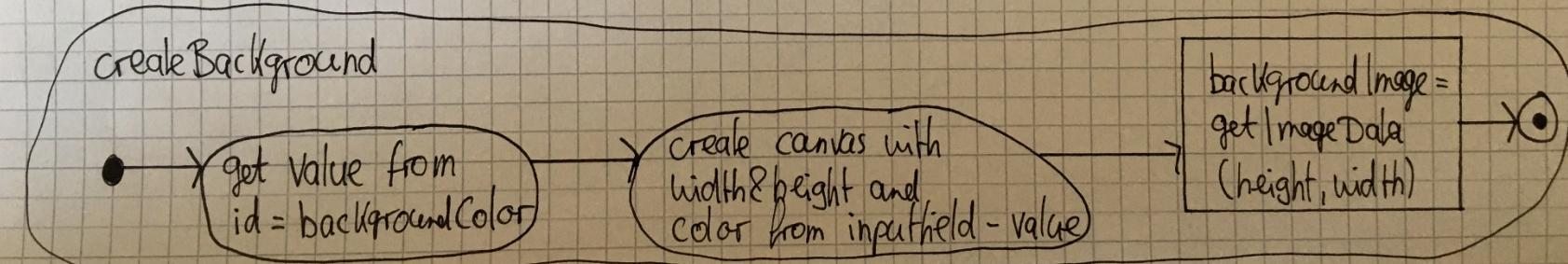
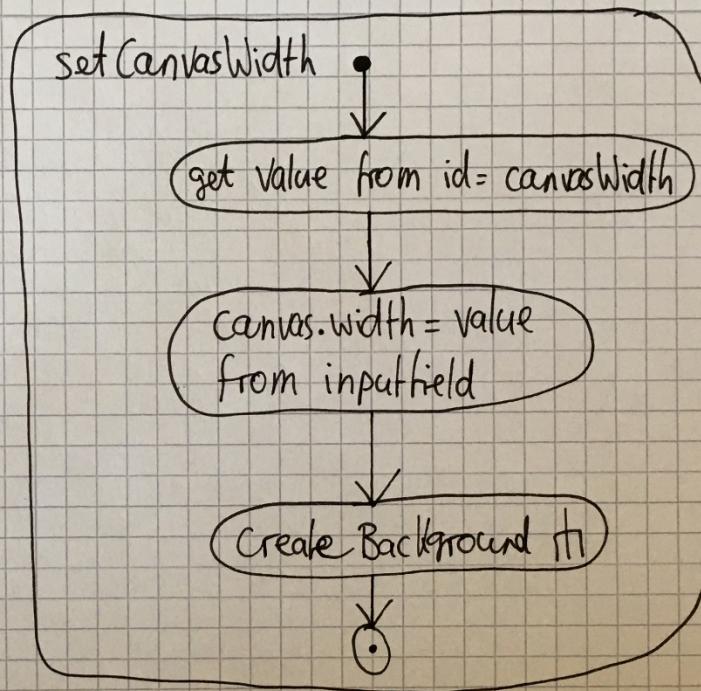
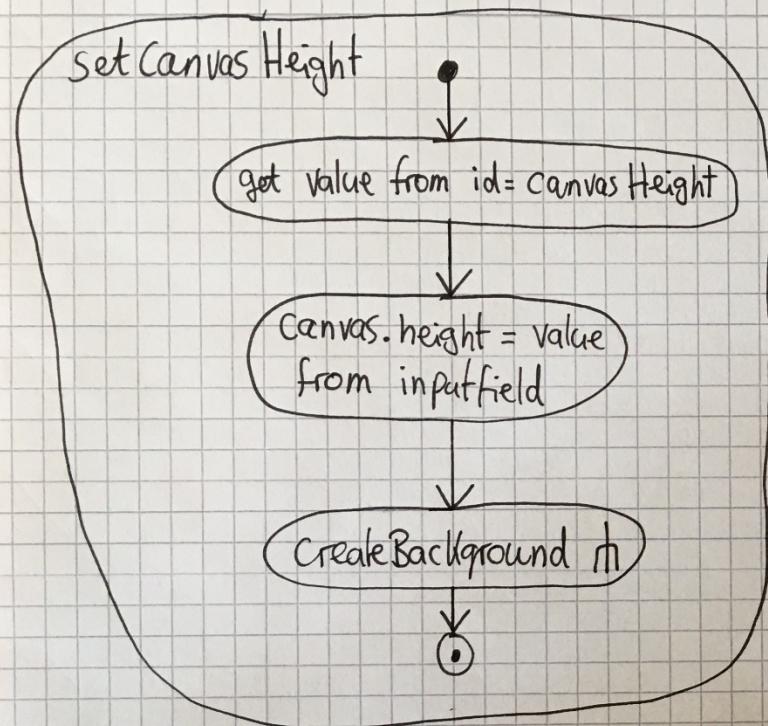
animate it

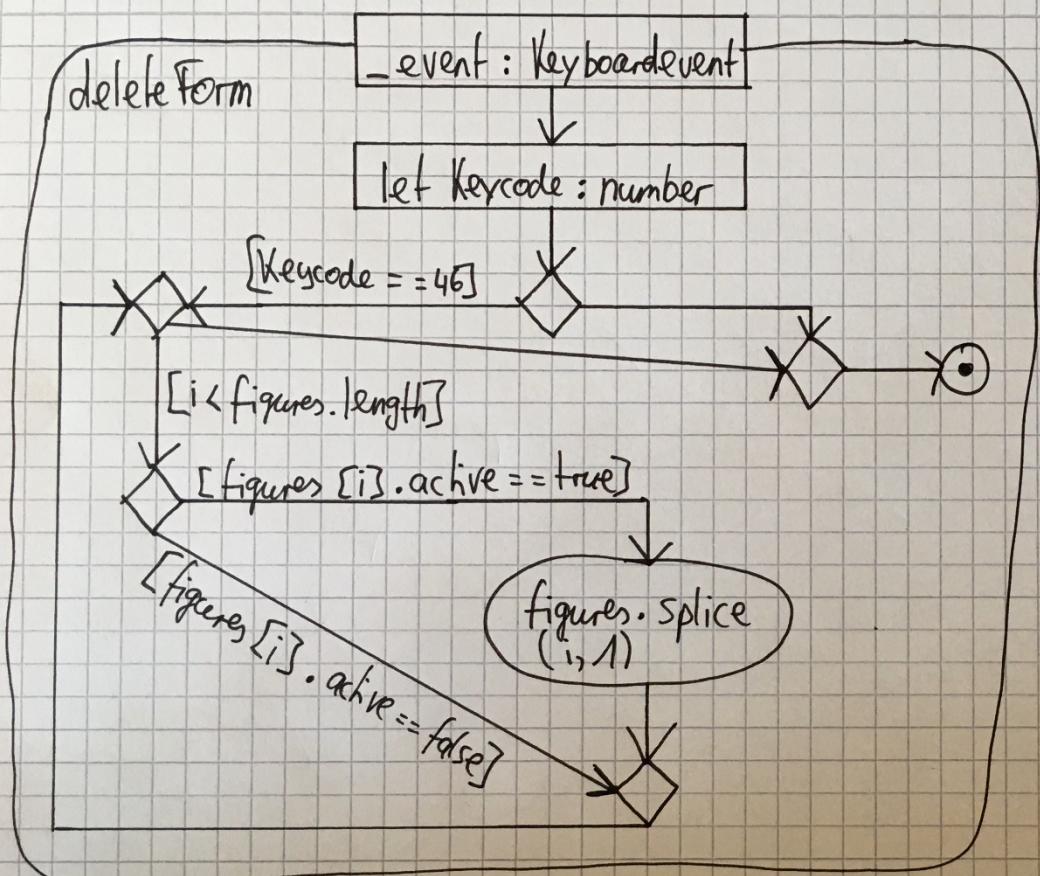
load Picture

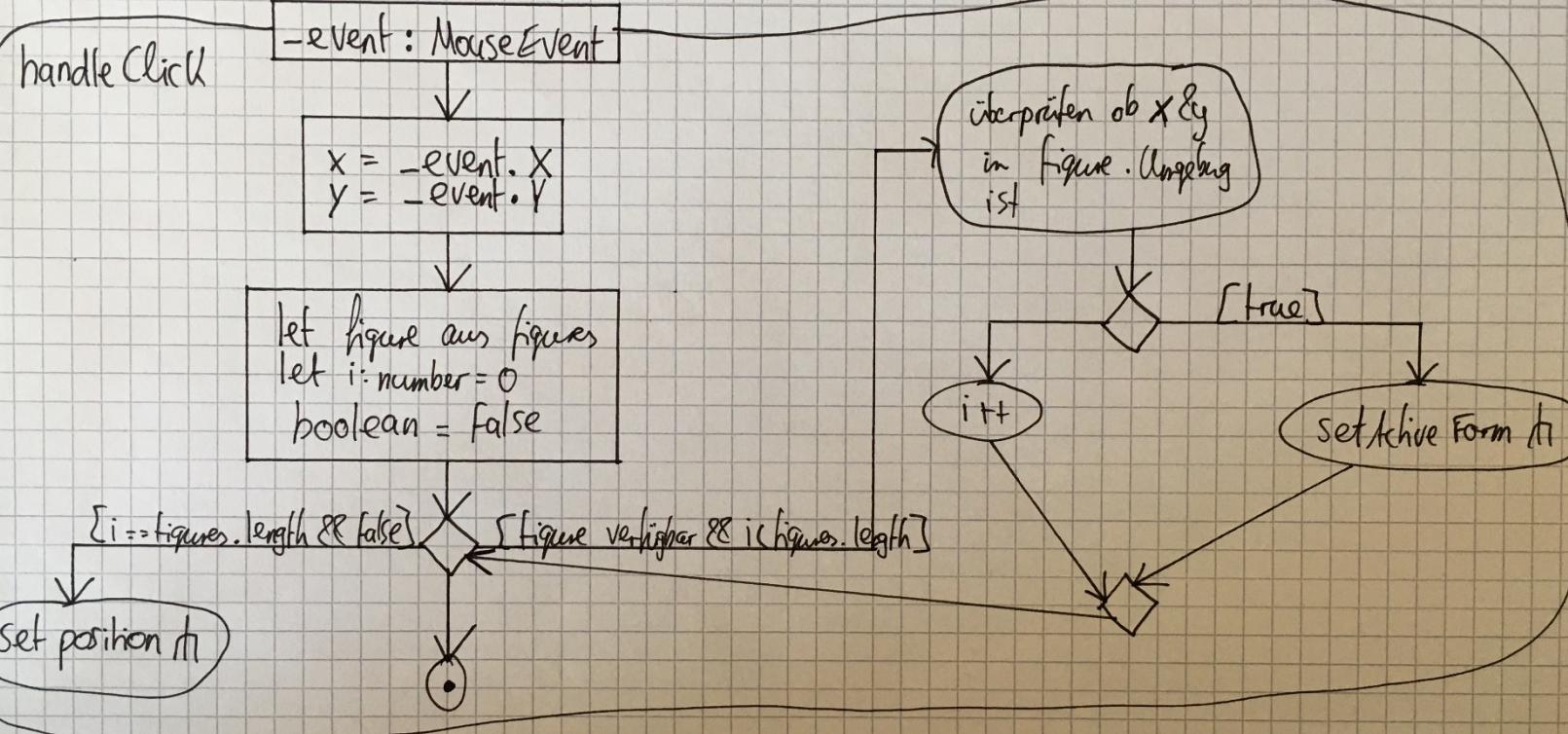
-event: Event

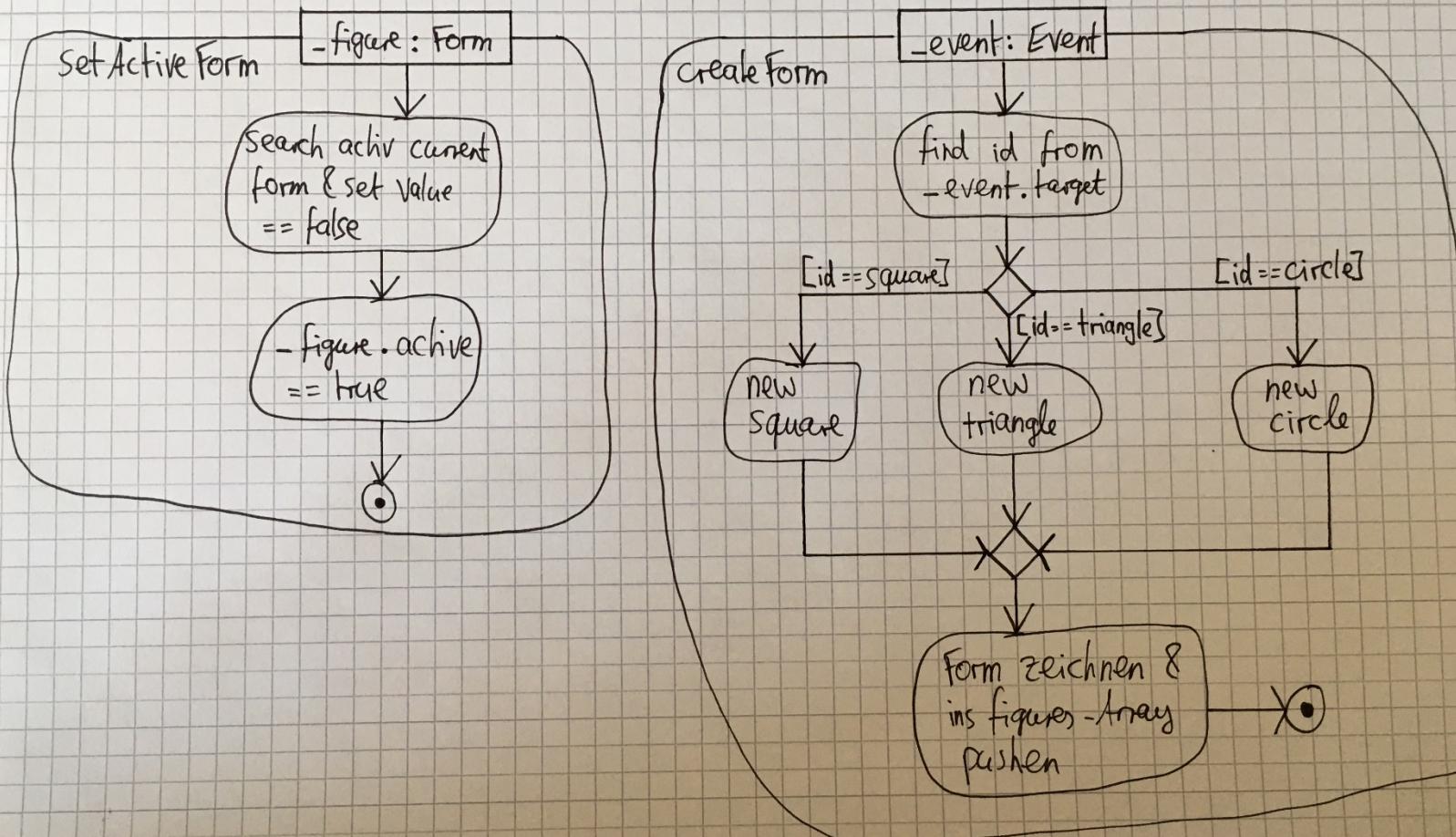
Anhage am Server
verschicken, um
gewünschtes Bild
zu bekommen

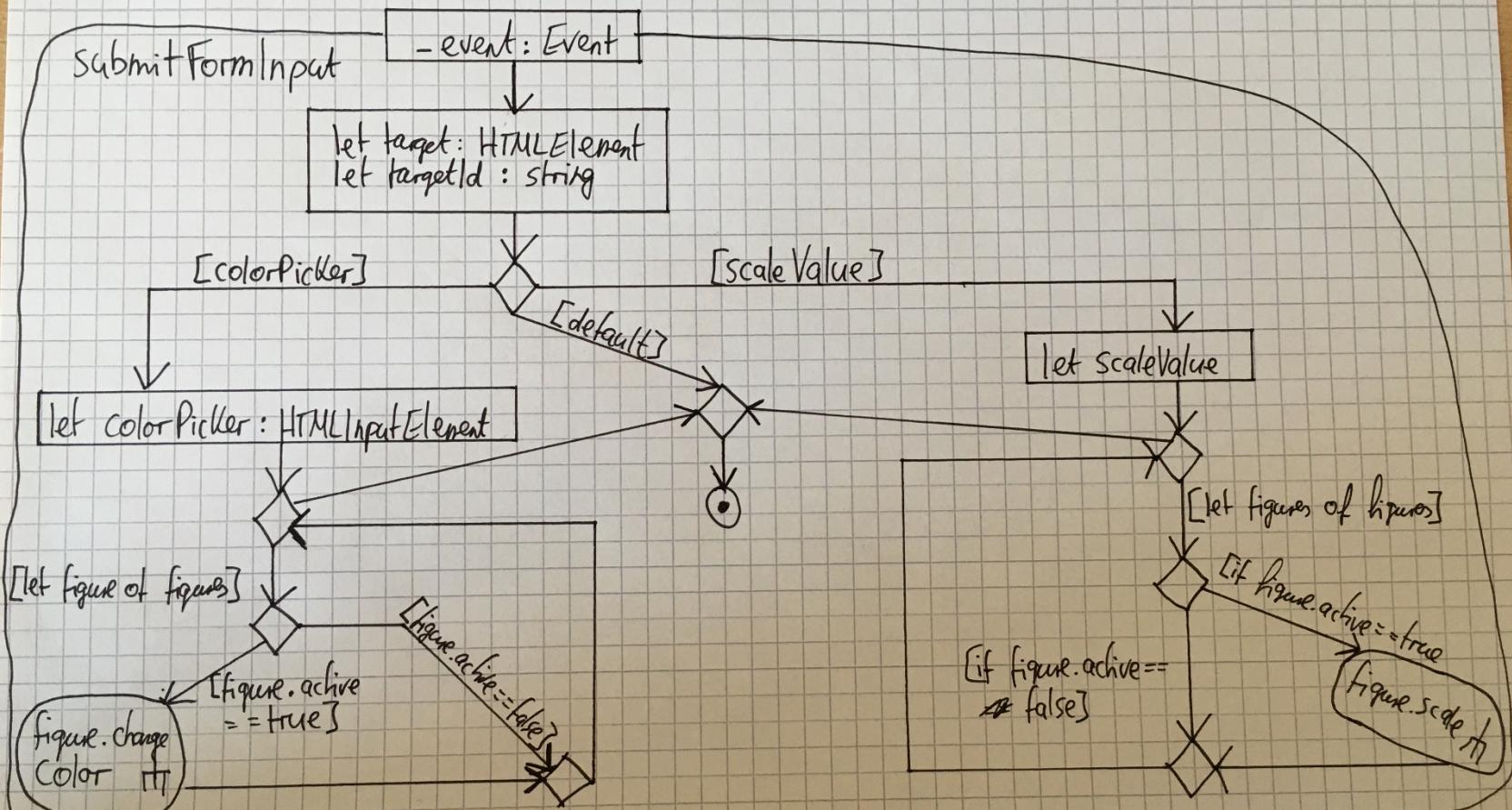
Bilder in der
datalist mit
id = library
anzeigen











constructor

```
this.active = true  
this.rotation = 0  
this.position = new Vector  
width/2 & height/2  
this.size = new Vector (50,50)  
this.moveType = "move"  
this.color = #fff
```

animate

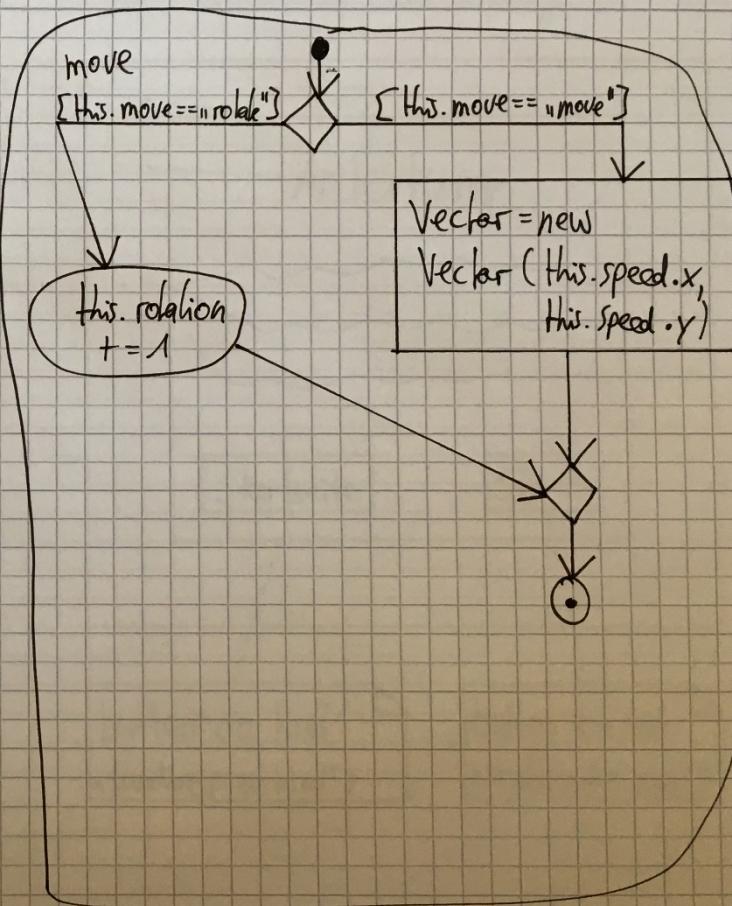
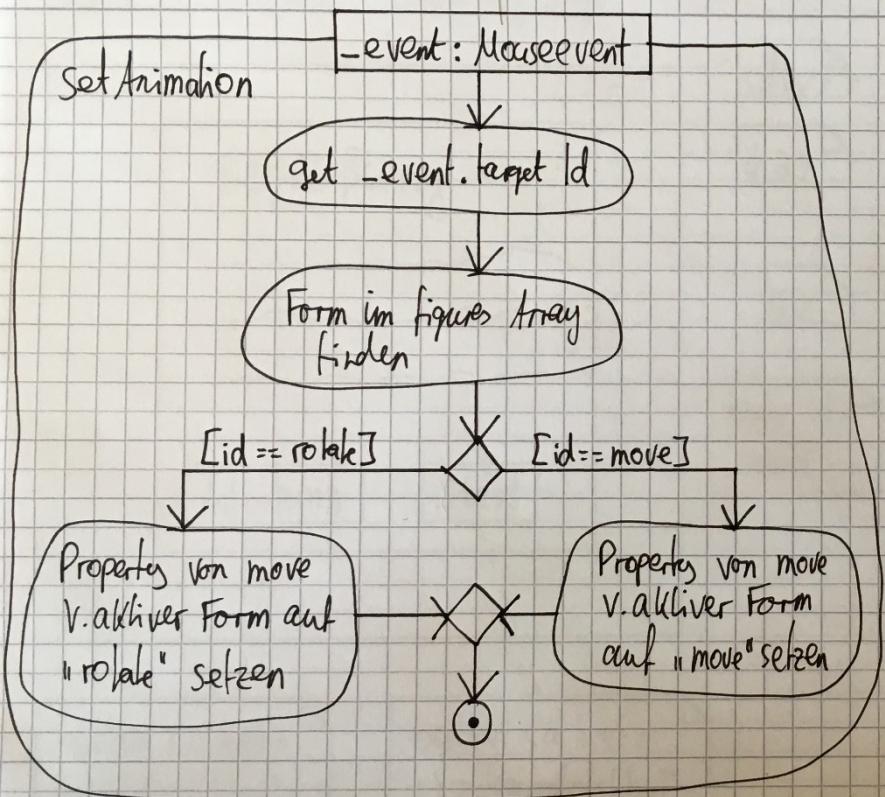
```
crc2.putImageData  
(backgroundImage,0,0)
```

let figure at figures

[figure verfügbar]

figure.move(1/2)

figure.draw()



update FormList

[if child available]

first child aus
AllForms löschen

let title: HTML
SpanElement

neuen Span
erstellen

neuen Span in
Übersicht-div
einfügen

let list: HTMLSpanElement

Typ anzeigen mit Farbvalue
aus Farbfeld

.addEventlistener auf span
mit setvalue funktion

an Übersicht - div anfügen

