

## Anleitung, um Anwendung selbst umzusetzen:

(Falls du die Raketen-Presets auf einem eigenen Server habe möchtest, findest du hier eine Zusammenfassung davon, wie man den Server aufsetzt. Denk daran, dass dann aber der Link in der Typescript Datei angepasst werden muss!)

Voraussetzung hierfür ist der Zugriff auf Hochschulserver:

Logge dich als Erstes mit deinen HFU-Daten per Cisco AnyConnect VPN in das Hochschulnetz ein.

WinSCP: damit die Anwendung richtig funktioniert, musst du einen Zugriff auf Server der Hochschule einrichten, auf die die Anwendung später die gespeicherten Informationen der Rakete/n abspeichern und auch wieder aufrufen kann.

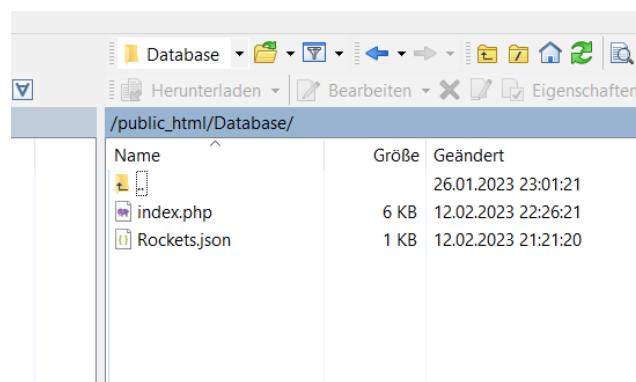
Lade dazu WinSCP herunter und installiere es auf deinem Laptop/Rechner.

1. Für die Verbindung benötigst du folgende Zugangsdaten:
  - Verbindung: SFTP
  - Server: webuser.hs-furtwangen.de
  - Port: 22
  - Benutzer: Ihr HFU-Account Benutzername
  - Passwort: Ihr HFU-Account Passwort

Anschließend musst du die JSON-Datei einfügen, in dieser werden die Daten der Anwendung abgespeichert und verwaltet.

Kopiere dann diesen (<https://github.com/JirkaDellOro/MingiDB/blob/master/index.php>) PHP-Code auf der Public Database deines Servers (auf der rechten Seite) auf derselben Ebene wie die JSON-Datei (sh. Bild)

(Achtung: Der Gebrauch des Data-Servers ist nur für Lehrzwecke!)



Wenn du nun eine Rakete abspeicherst, erscheint sie – anhand des Beispielbild hier – in der Rockets-json-Datei.

## Feuerwerk Anwendung Gebrauchsanleitung:

Schön, dass du dich dafür entschieden hast meine Anwendung auszuprobieren.

Vor dich siehst du auf der linken Seite die „Spielfläche“ (1.), auf der per Mausklick eine Raketenexplosion ausgelöst werden kann. (zur Orientierung sh. Bild unten)

Zu deiner rechten kannst du die verschiedenen Einstellungsparameter erkennen:

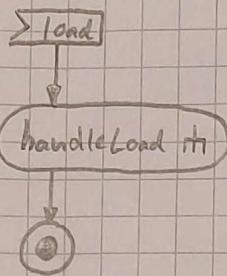
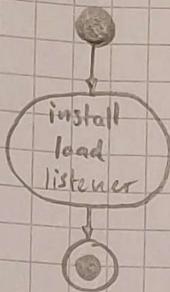
2. *Speed*: Wie schnell sich die Explosion ausbreitet.
3. *Size*: die Größe der entstehenden Partikel.
4. *Amount of particles*: die Anzahl an Partikel, die bei der Explosion „freigesetzt“ bzw. erzeugt werden.
5. *Lifespan*: wie lange die Partikel umherschwirren, bevor sie verschwinden.
6. *Color*: die Farbe der Explosion.
7. *Save current Preset*: die von dir zusammengestellte Rakete kann, nach Eingabe eines Namens im Inputfeld oberhalb des Buttons, gespeichert und zu einem späteren Zeitpunkt wiederaufgerufen werden.
8. *Load*: Gib den Namen der gewünschten, gespeicherten Rakete ein, und lasse sie wieder auf dem Canvas los!
9. *Show All*: Falls du eine Übersicht aller von dir abgespeicherten Raketen einsehen möchtest, oder du dich nicht mehr an die Namen erinnerst, kannst du einfach auf diesen Button klicken und hast sofort einen Überblick.



Ich wünsche dir viel Spaß mit der Anwendung!

# EIA2 Endabgabe Wintersemester 2022/2023

## Activity Diagram



export

interface Vector

x: Number  
y: number

const canvas: HTMLCanvasElement  
const ctx: CanvasRenderingContext2D

```

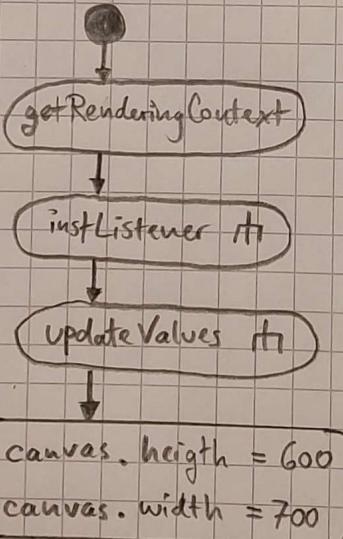
const colorWheel: HTMLInputElement
const speedRange: HTMLInputElement
const sizeRange: HTMLInputElement
const particleRange: HTMLInputElement
const lifespanRange: HTMLInputElement
const presetText: HTMLInputElement
const loadText: HTMLInputElement
const loadButton: HTMLInputElement
const anzeigen: HTMLElement
const showButton: HTMLInputElement
  
```

const serverLink: string

```

let color: string
let size: number
let particleAmount: number
let speed: number
let lifespan: number
let rName: string
  
```

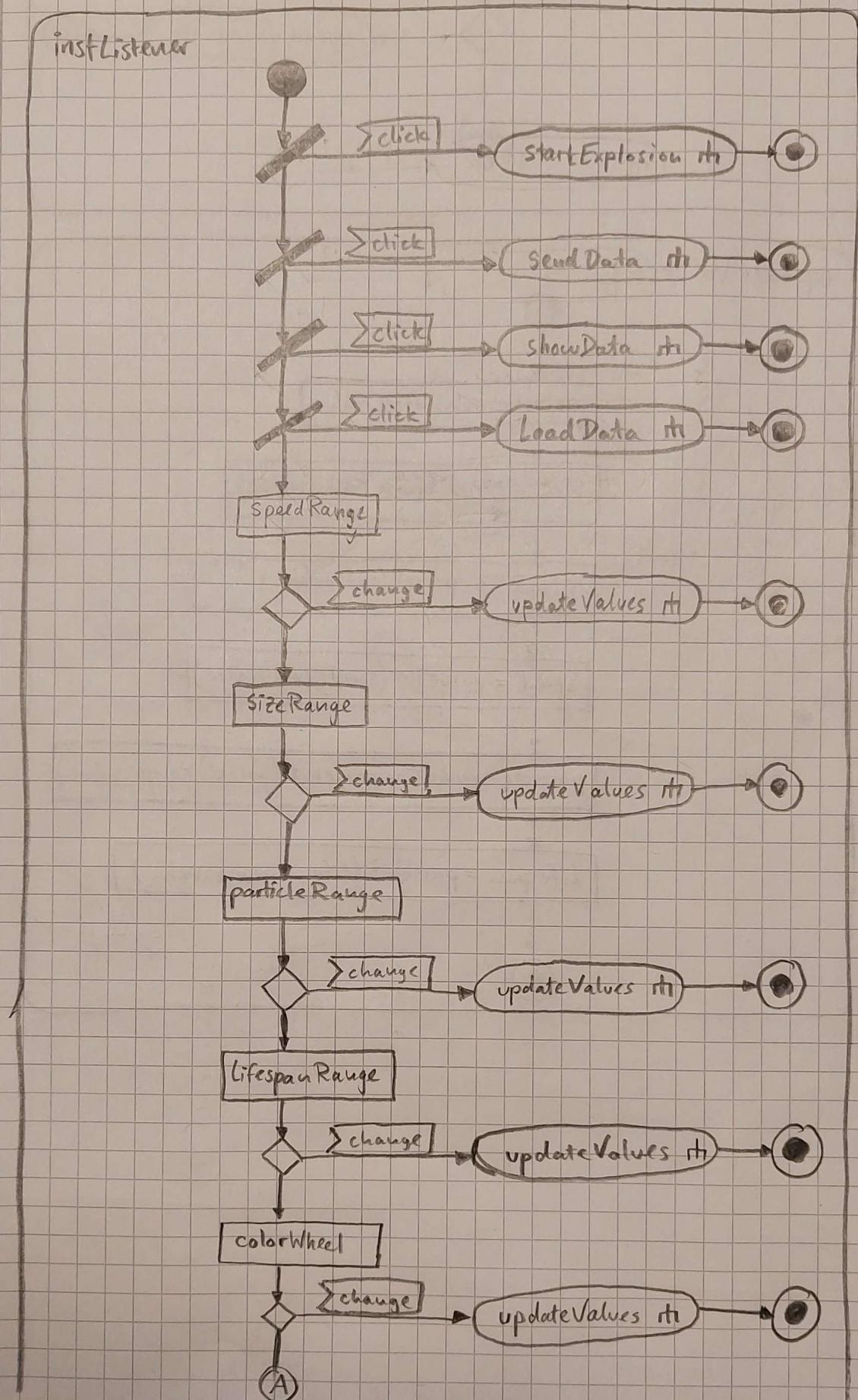
handleLoad



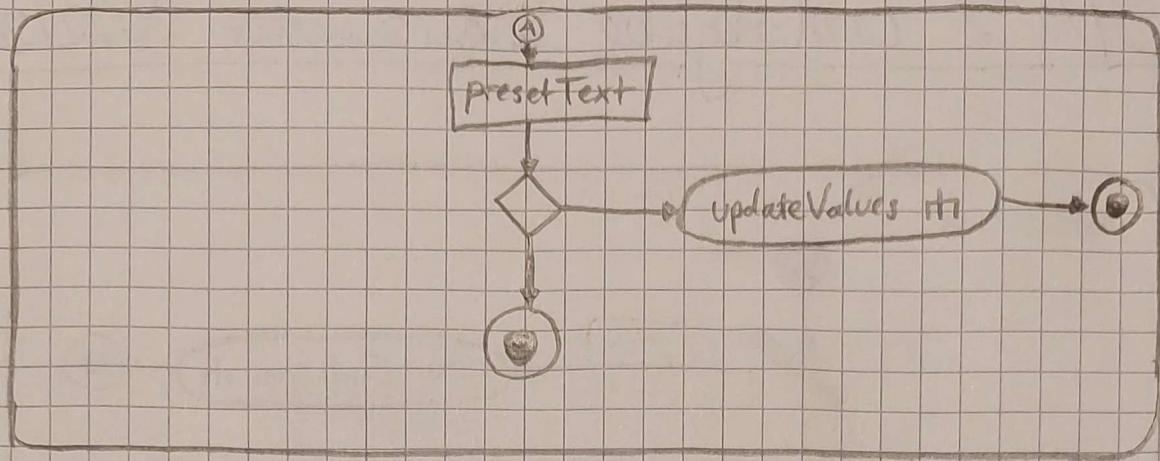
click on canvas

start Explosion

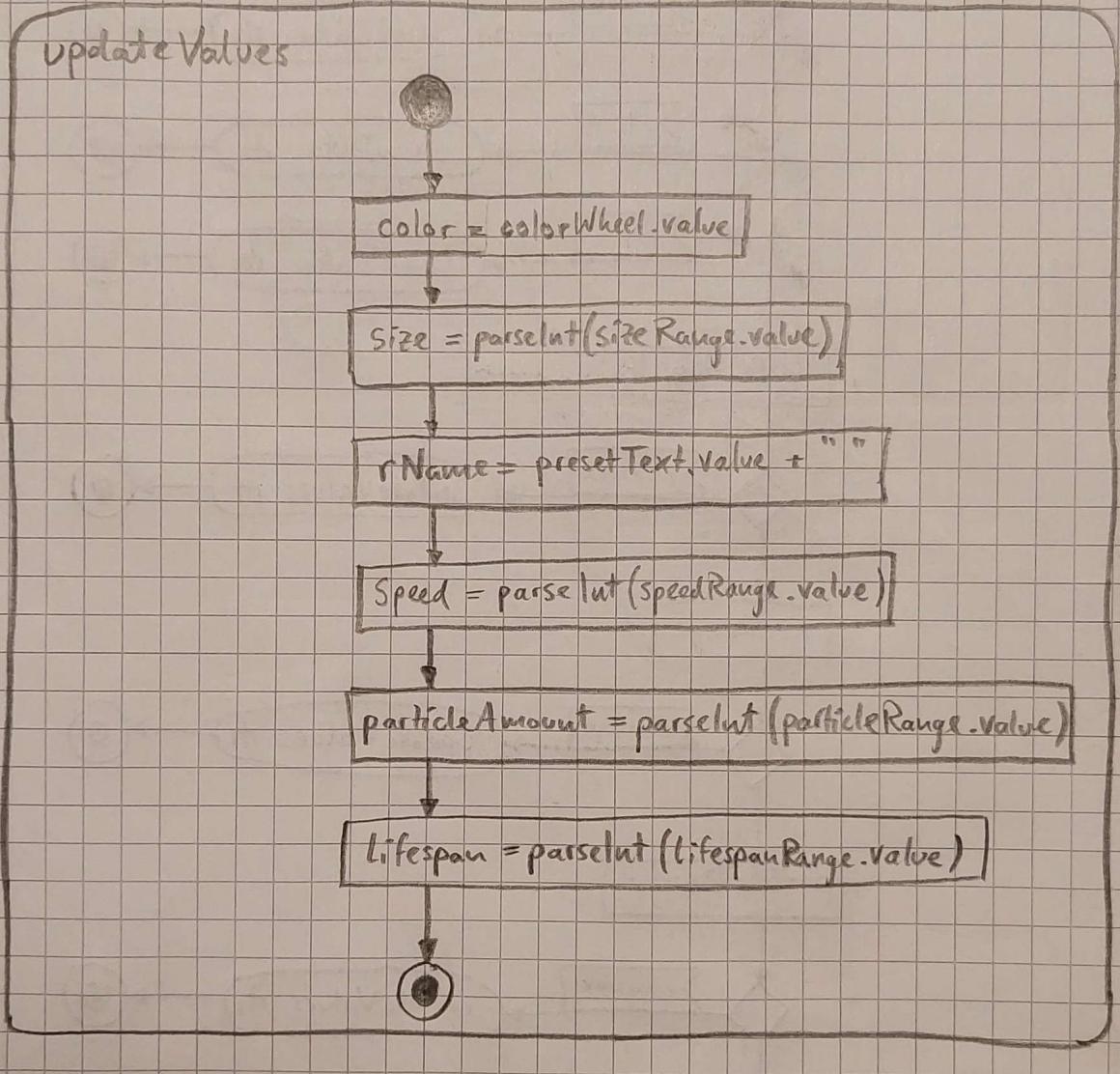
# EIA 2 Endabgabe Wintersemester 2022/2023



cont. instListener



update Values



Start Explosion

- event : MouseEvent

```
let rect = canvas.getBoundingClientRect  
let x: number = e.clientX - rect.left  
let y: number = e.clientY - rect.top  
let spawnpoint: Vector = x, y  
let rocket: Rocket = new Rocket
```

rocket.launch() +

launch

```
let interval: number  
interval = setInterval
```

ctx.resetTransform

draw

```
this.time  
ctx.fillStyle = "rgba"  
ctx.fillRect  
ctx.fillStyle = this.color
```

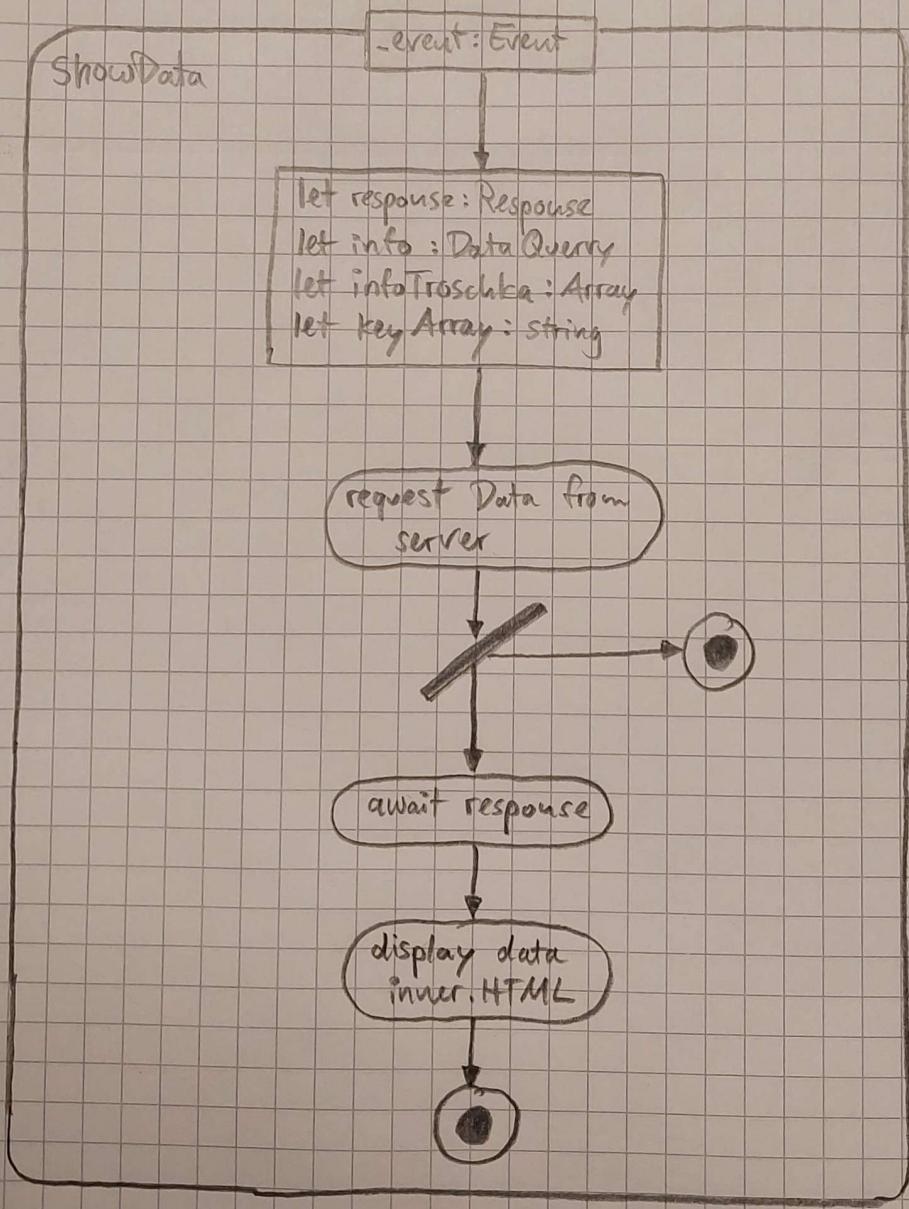
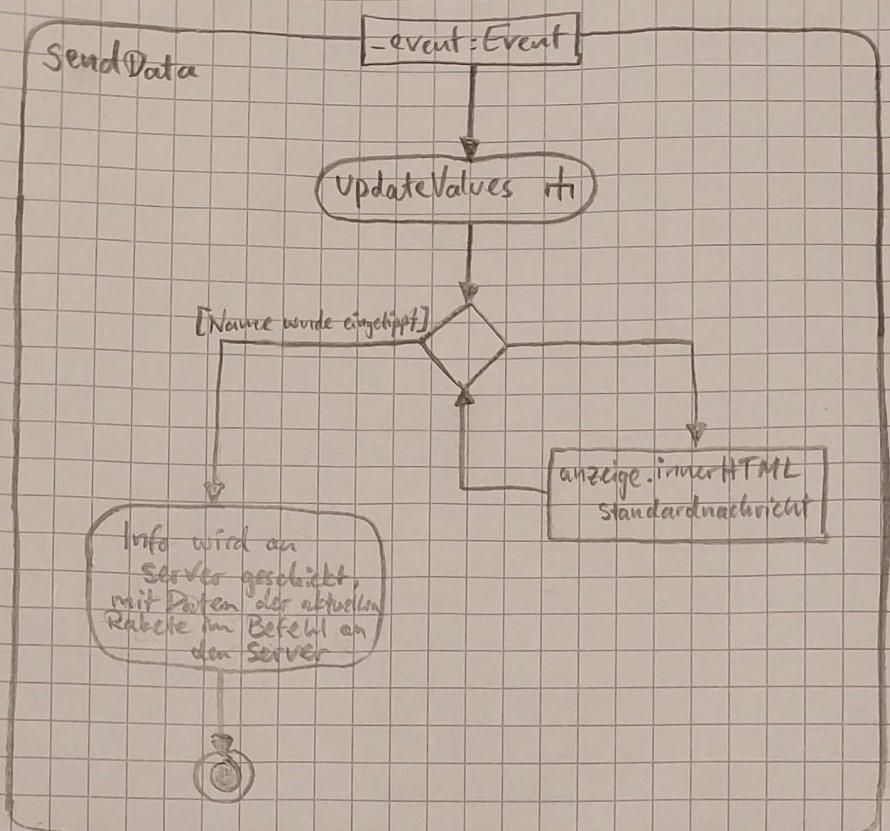
let x: number = 0

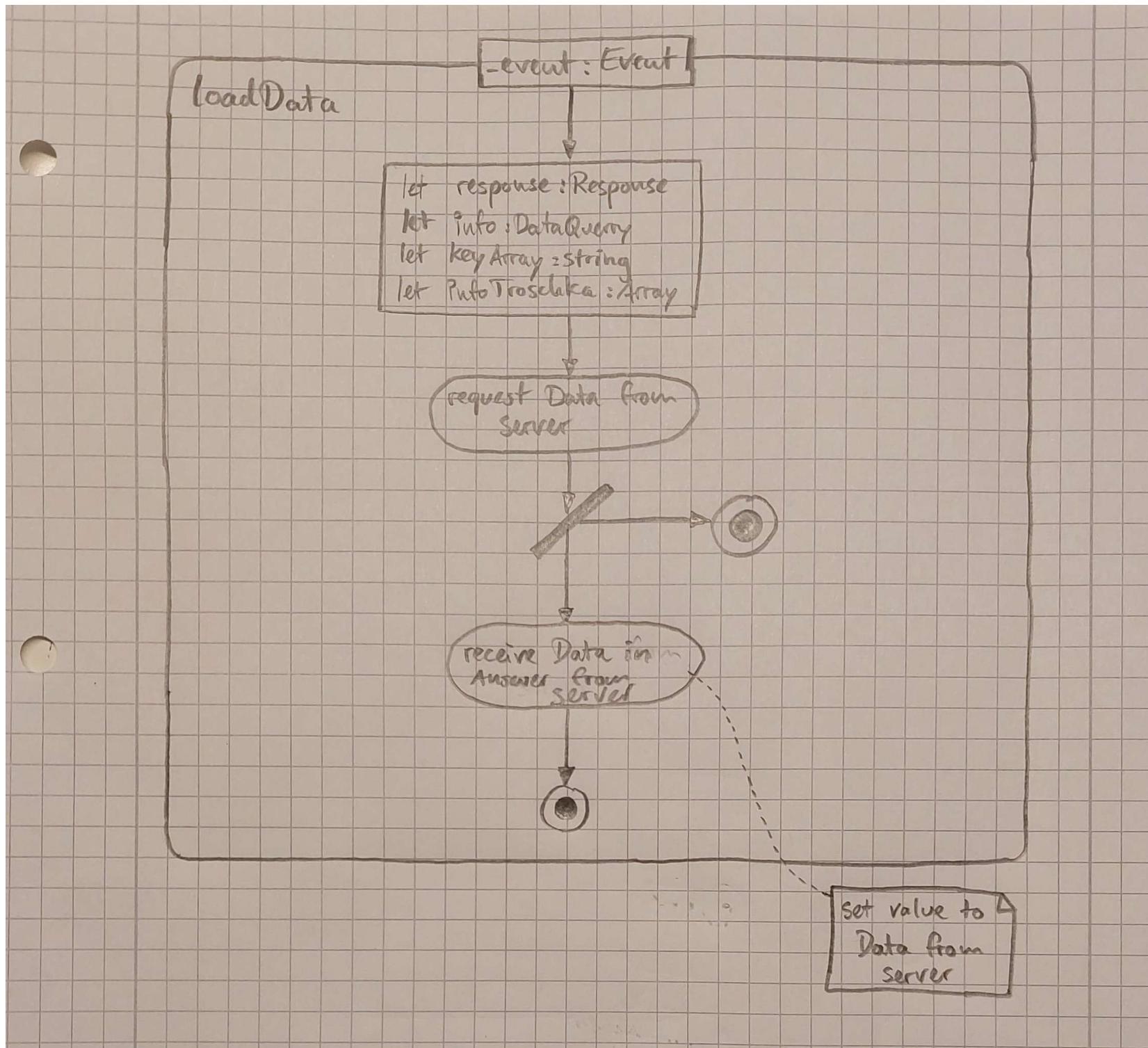
```
ctx.fillRect(this... etc.)  
ctx.fillRect(this... etc.)  
ctx.fillRect(this... etc.)  
ctx.fillRect(this... etc.)
```

[this.time < this.lifespan]

[this.time > this.lifespan]

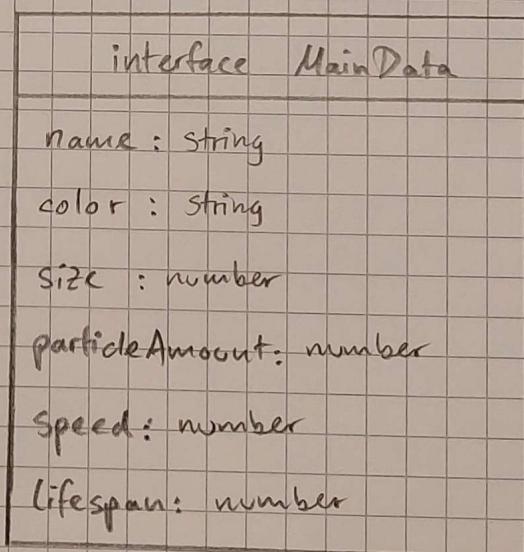
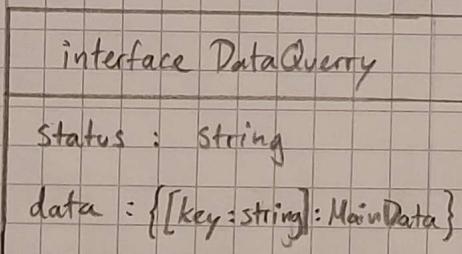
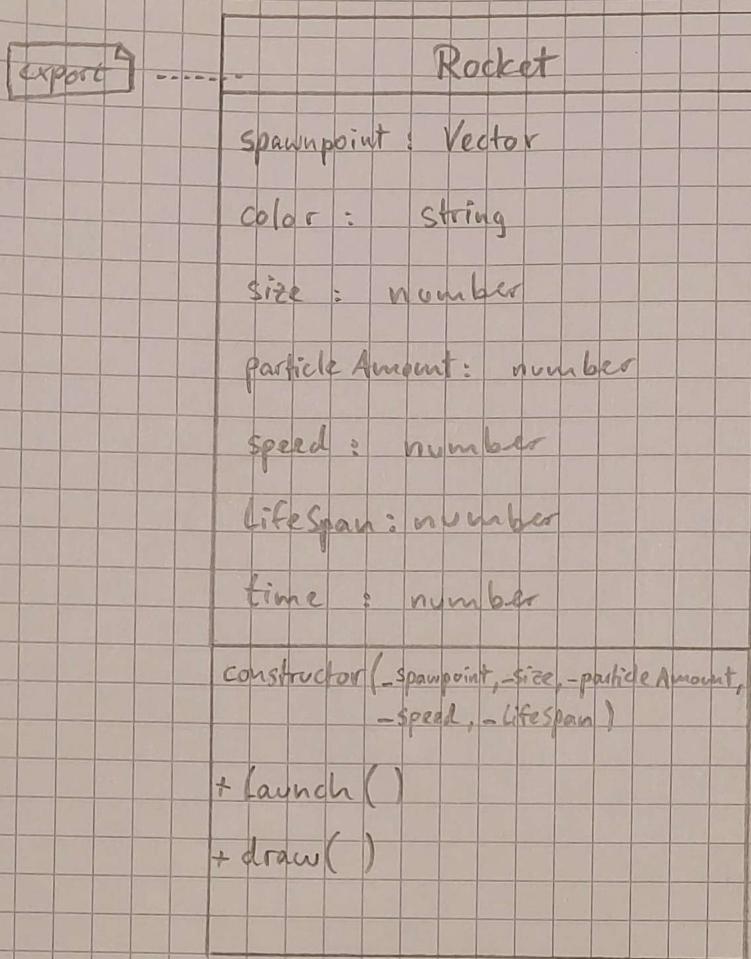
clearInterval  
ctx.clearRect





# EIA2 Endabgabe Wintersemester 2022/2023

## Class Diagram



# EIA 2 Endabgabe Wintersemester 2022/2023 Feuerwerk

UI-Scribble

PC/Komputer-Bildschirm

Text / Namen  
der Parameter  
`<sp>`

`<input> id=speed`  
`<input> id=size`

`<h2>`

Titel bzw. Überschrift

2000

99

00

00

00

00

00

<div data-bbox="635 2690 675 2695</div><div data-bbox="635 2695

# EIA2 Endabgabe Wintersemester 2022/2023 Feuerwerk

Use-Case-Diagramm V3

