

Your fireworks



<div>
id = "cameraContainer"

<canvas>
id = "mainCanvas"
width = 1000
height = 500

<button>
id = "displayButton"

<button>
id = "updateButton"

<div>
id = "buttonContainer"

<div>
id = "rocketArea"

Your rocket from Database

deposit | Show points

<button>
id = "depositButton"

<button>
id = "updateButton"

<div>
id = "displayButton"

```
<div>  
  id="editorContainer"
```

```
<p>
```

```
<div>
```

Create your own treasure...

YOUR

Choose here

```
<fieldset>  
  <input type="text"  
    name="RocketName"  
    id="name"  
    placeholder="Create here"
```

```
<fieldset>  
  <input type="range"  
    name="ExplosionSize"  
    id="explosion"  
    min="1"  
    max="50"  
    step="1"  
    value="1"
```

```
<fieldset>  
  <input type="range"  
    name="Lifetime"  
    id="Lifetime_I"  
    min="1"  
    max="50"  
    step="1"  
    value="10"
```

```
<fieldset>  
  <select type="color"  
    name="Color"  
    value="Color"
```

```
<fieldset>  
  <input type="range"  
    id="avWidth"  
    name="AvWidth"  
    Value="10"  
    min="10"  
    max="1000"  
    step="1"
```

```
<fieldset>  
  <select id="particleType"  
    name="ParticleType">  
    <option value="rectangle">  
      rectangle  
    </option>  
    <option value="line">  
      line  
    </option>
```

```
<fieldset>  
  <input type="range"  
    name="ParticleSize"  
    id="PSize_P"  
    min="10"  
    max="50"  
    step="1"  
    value="10"
```

Name -

Create here

Particle Speed -

Slow

Fast

Lifetime -

1sec

50 sec

Particle Color -



Amount of Particle -

10

1000

Type of Particle -



Size of Particle:

10 Pixel

30 Pixel

display rocket

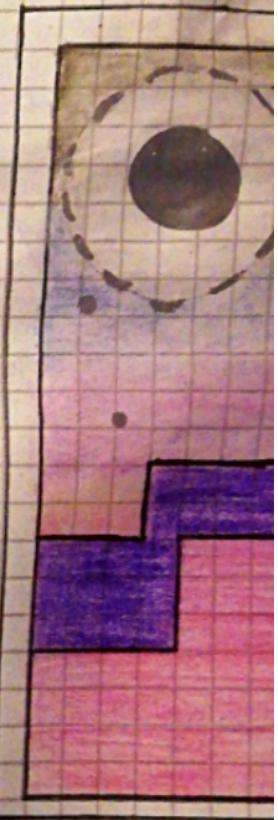
update rocket

```
<button id="displayButton"
```

```
<button id="updateButton"
```

your Rocket

```
<div id="buttonContainer"
```



Name

Create here

Particle Speed

Slow Fast

Lifetime

1 sec 50 sec

Particle Color

▼

Amount of Particle

10 1000

Type of Particle

▼

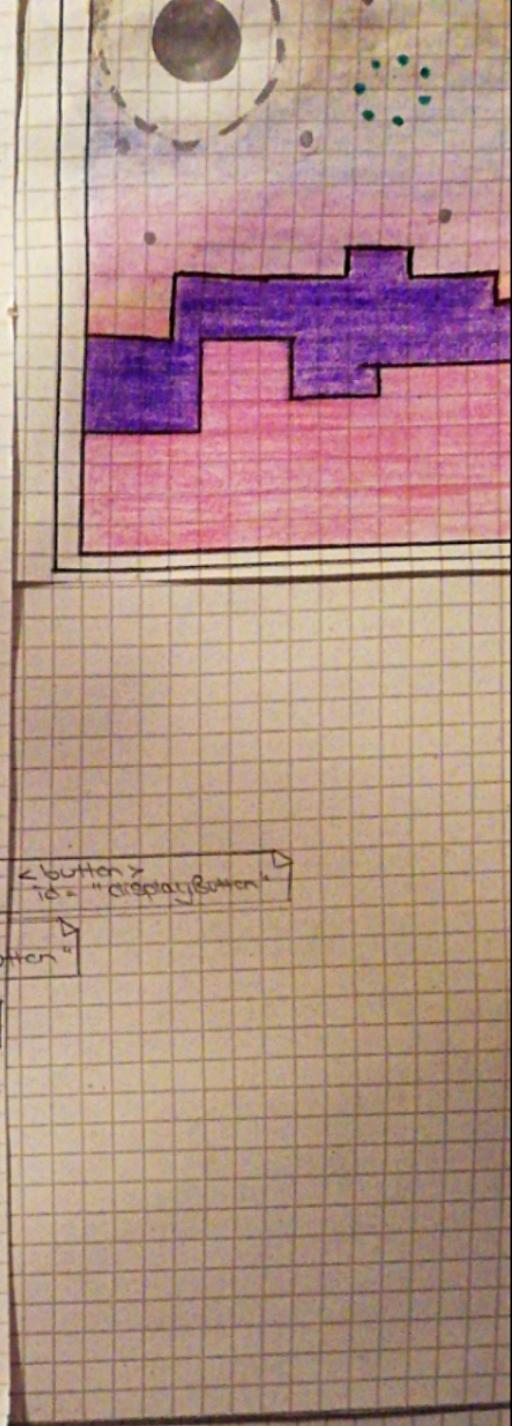
Size of Particle:

100px 30 Pixel

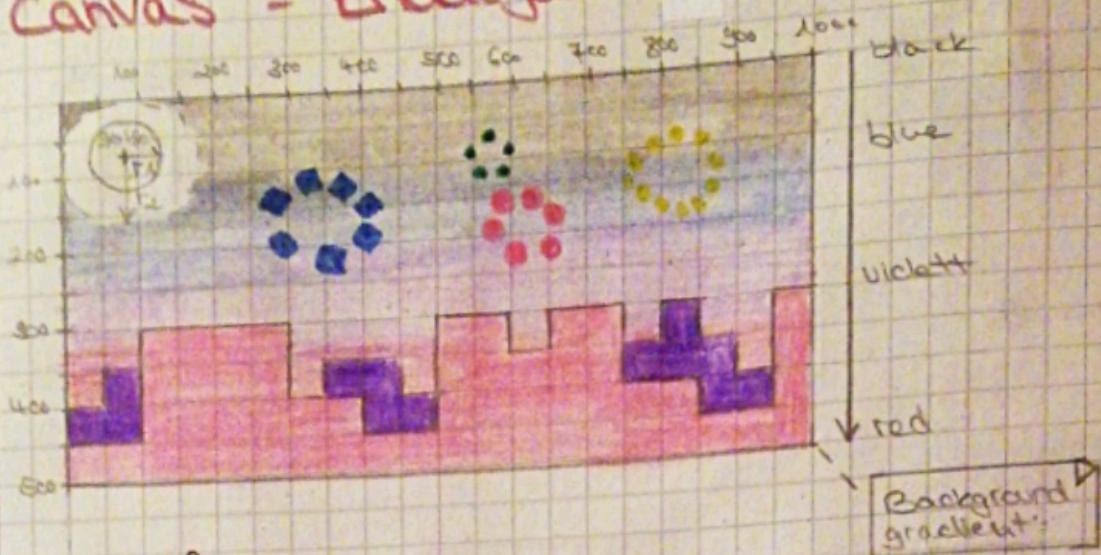
display rocket **update rocket** **button** **id = "Update Button"**

your Rocket **div** **id = "button Container"**

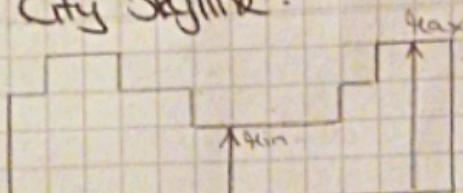
rocket number **Save rocket** **Delete rocket** **Show rocket**



Canvas - Endabgabe



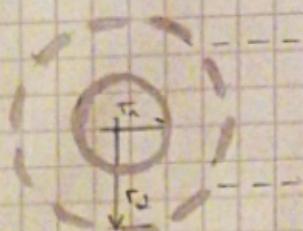
- ## • City Skyline:



width of skyline
= canvas - width

colors:
front row: #1B0A24
back: #14C718

- Moon:



$$r_1 = 35$$

- gradient (f_2)
- bright white
- $a = 0$

- ## • Stars :



starCart - 100
randomly placed
on canvas
with Math.random

- ## • Particles:

> Rectangle

> Dot

> Line

• Particle "Rectangle"



fillStyle = ▶
- color

Size = ParticleSize
Min = 1 Pixel
Max = 10 Pixel

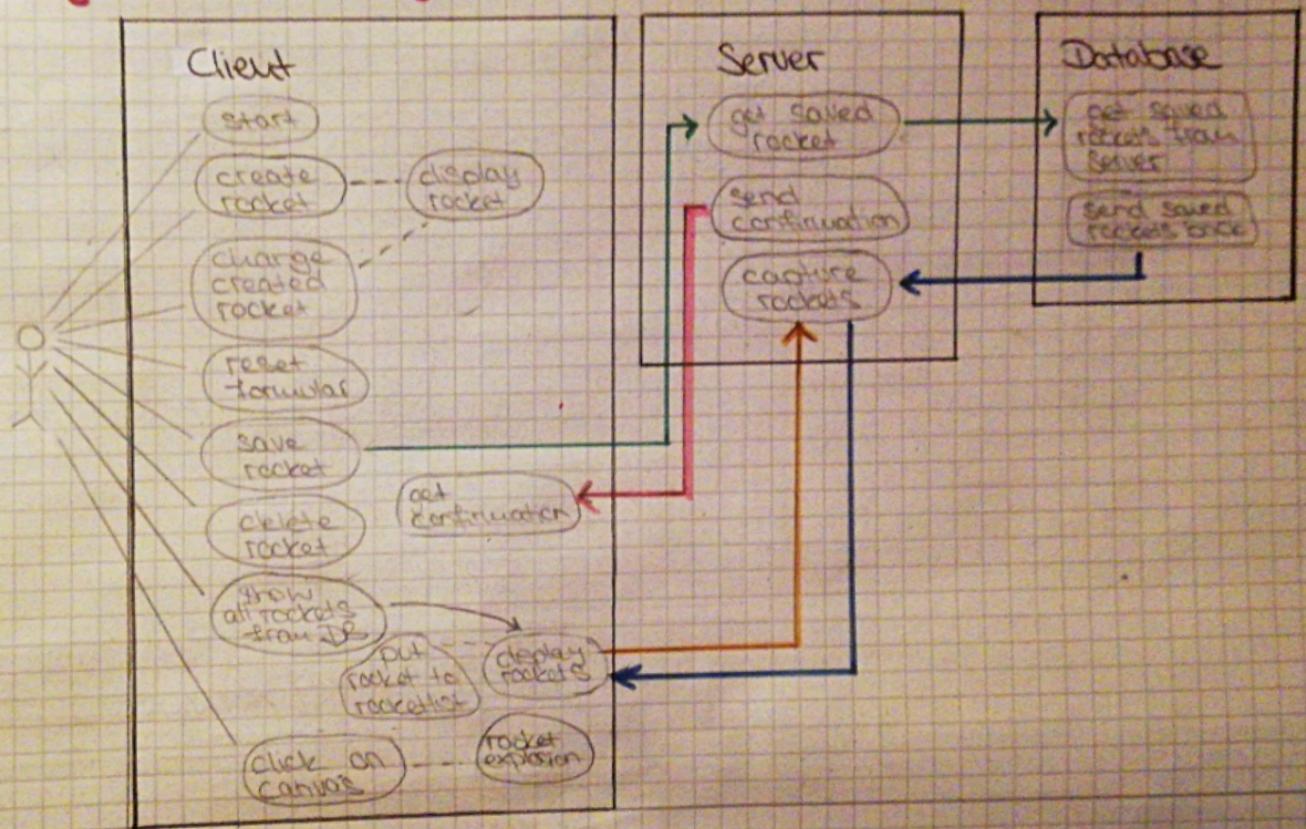
• Particle "Dot"



r = - particleRadius

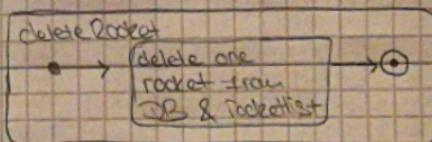
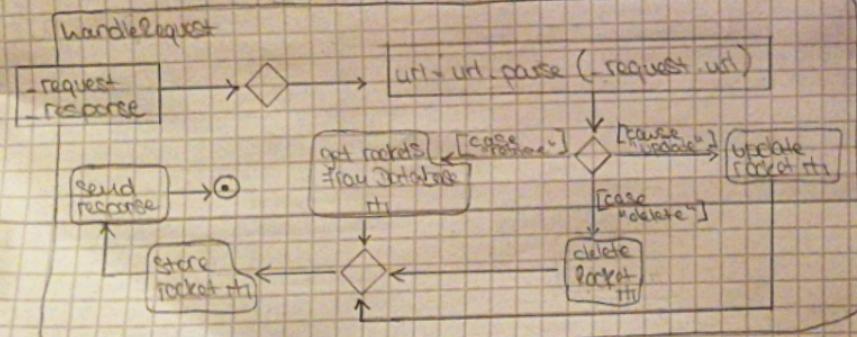
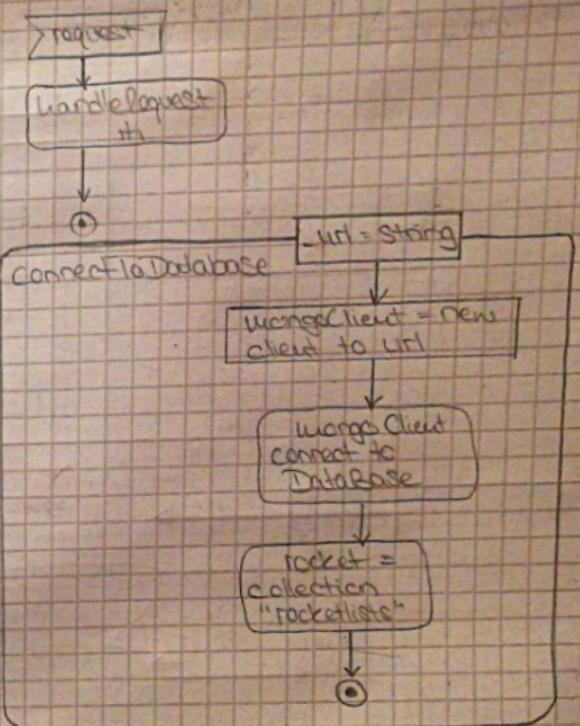
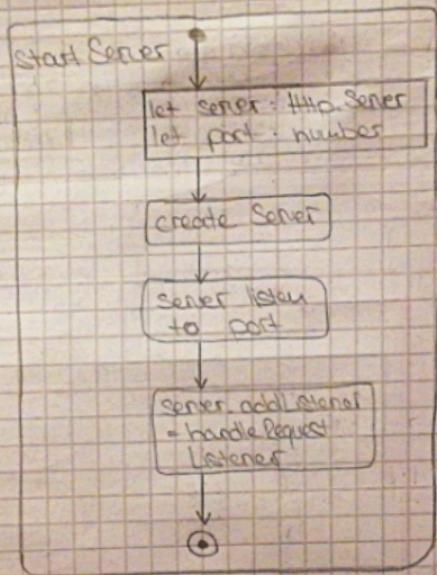
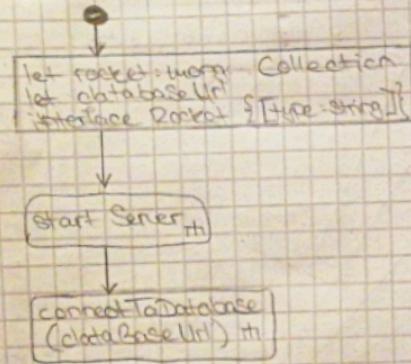
fillStyle = - color

Use Case Diagramm - Endabgabe



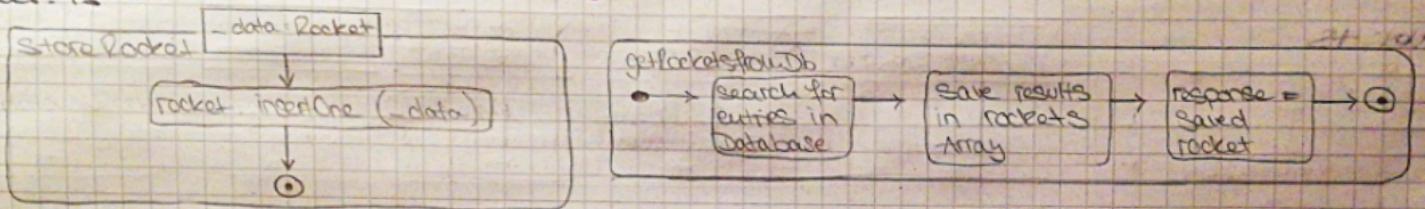
Aktivitätsdiagramm 1 - Endanfrage

Server.ts



Aktivitätsdiagramm 2 Endabgabe

Server.ts



UpdateRocket

```

let url: URL parse
let abName: String | String []
let rocketName: String | String []
let rocketExplosion: String | String []
let rocketLifetime: String | String []
let rocketColor: String | String []
let rocketAircraft: String | String []
let rocketParticleType: String | String []
let rocketParticleSite: String | String []
  
```

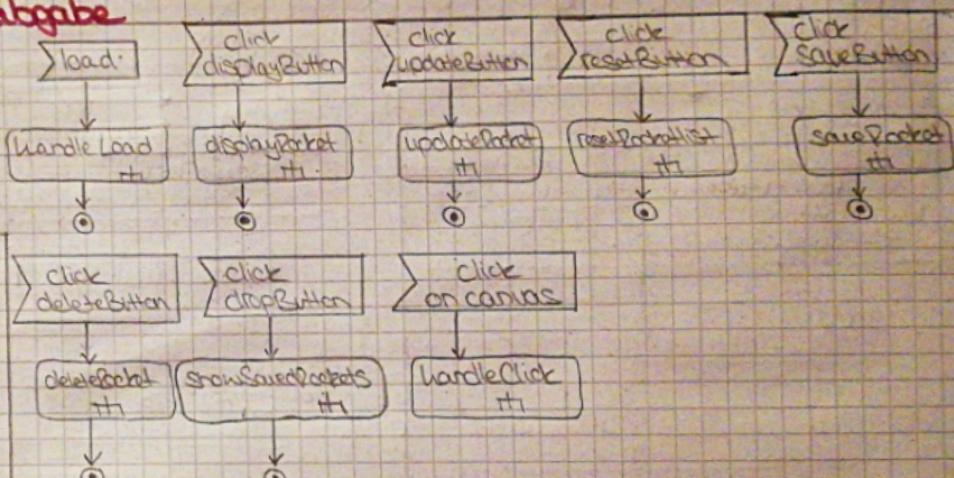
rocket.updateOne
update all rocket components

response =
"rocket updated!"

Aktivitätsdiagramm 3 Endabgabe

main.ts

```
install load
Listener
let url: String (HTTPS-Link zu Heroku)
let buttonClicked: number = 0
let rockets: any
let currentRocket: string
export let imageData: ImageData
export let crc2: CanvasRenderingContext2D
let form: HTMLFormElement
let firework: Firework [] = []
let fps: number = 10
let canvas: HTMLCanvasElement | null
```



HandleLoad

event: Event

form: HTMLFormElement
(> query "form")
canvas: HTMLCanvasElement
(> query "canvas")

declare crc2: CanvasRenderingContext2D
Implementiere Background
with drawCanvas ()

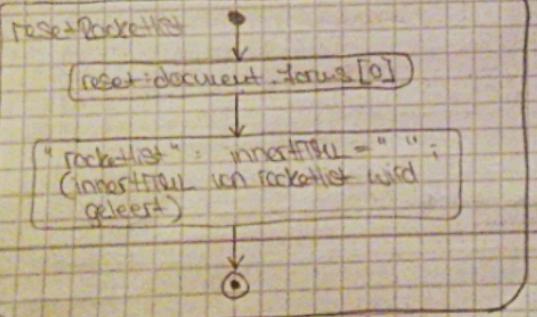
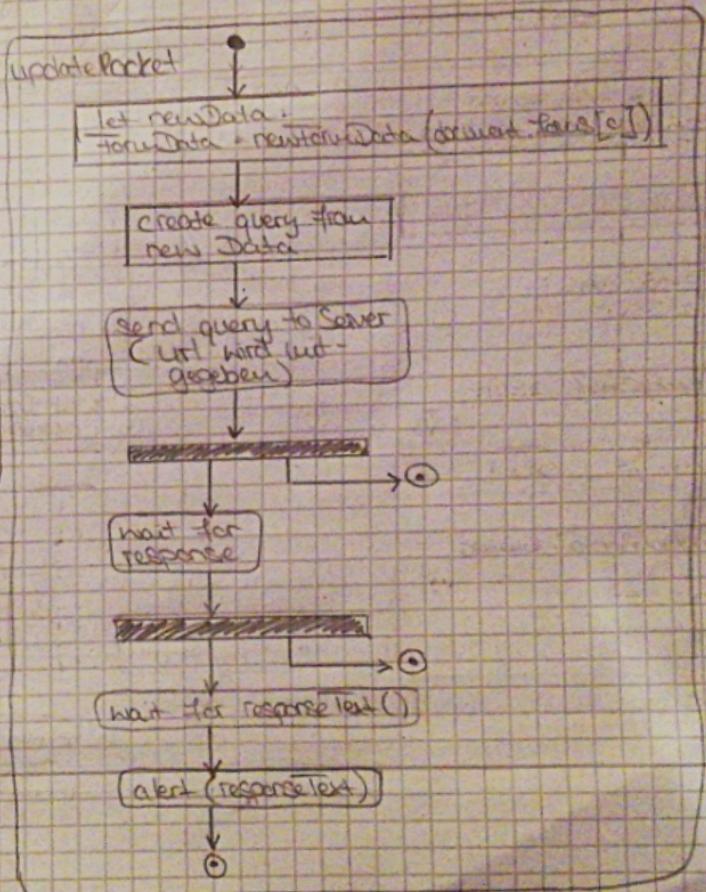
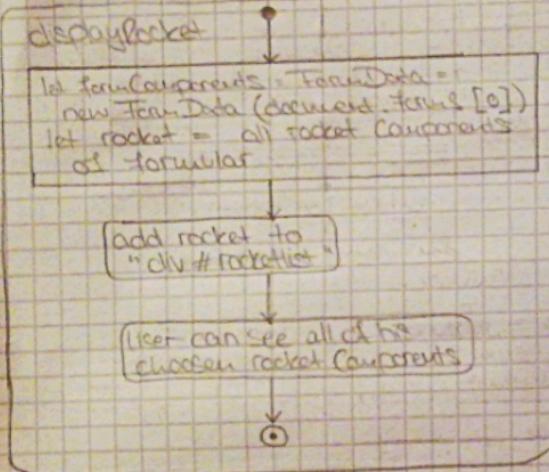
Find Buttons
2 canvas &
install click
Listener

0 ← window.setInterval (update, 10 / fps) ←

Fertig
update in
10 / 8

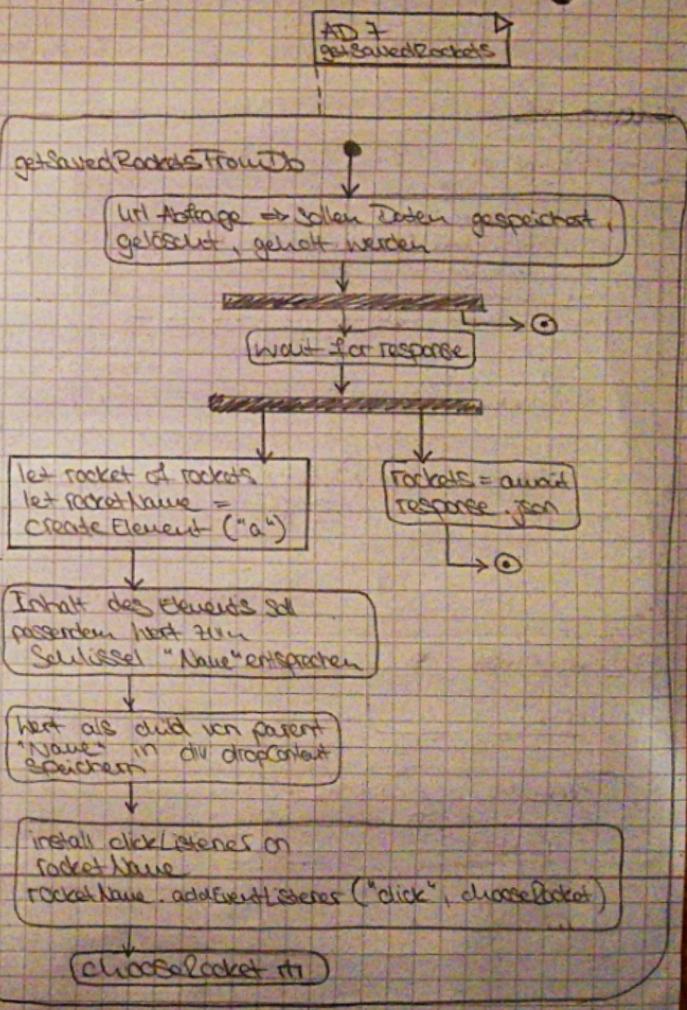
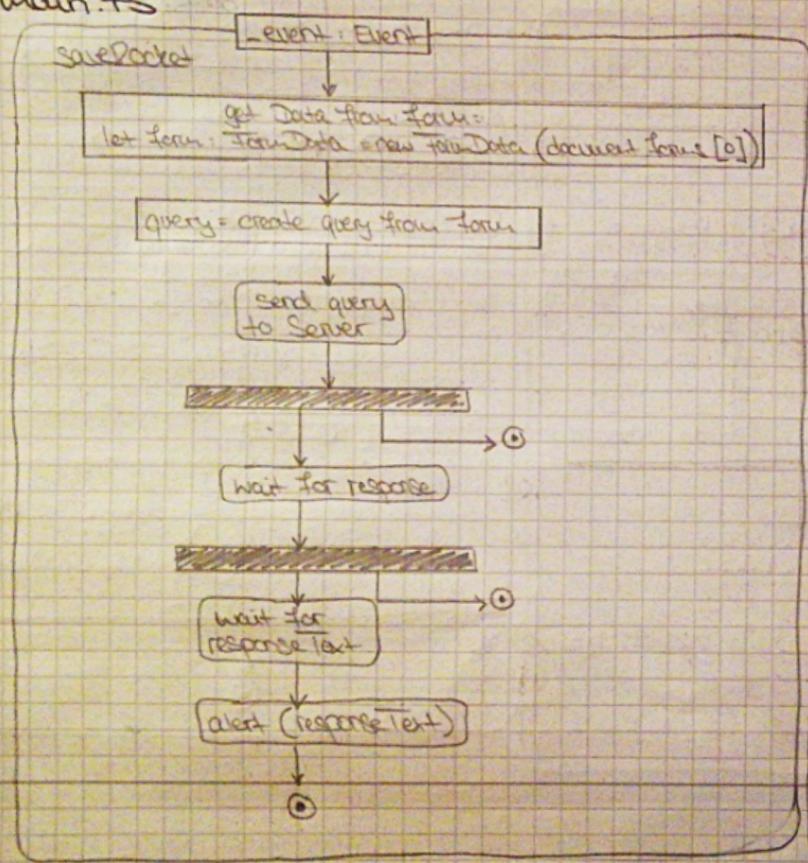
Aktivitätsdiagramm 4 Endabgabe

main.ts

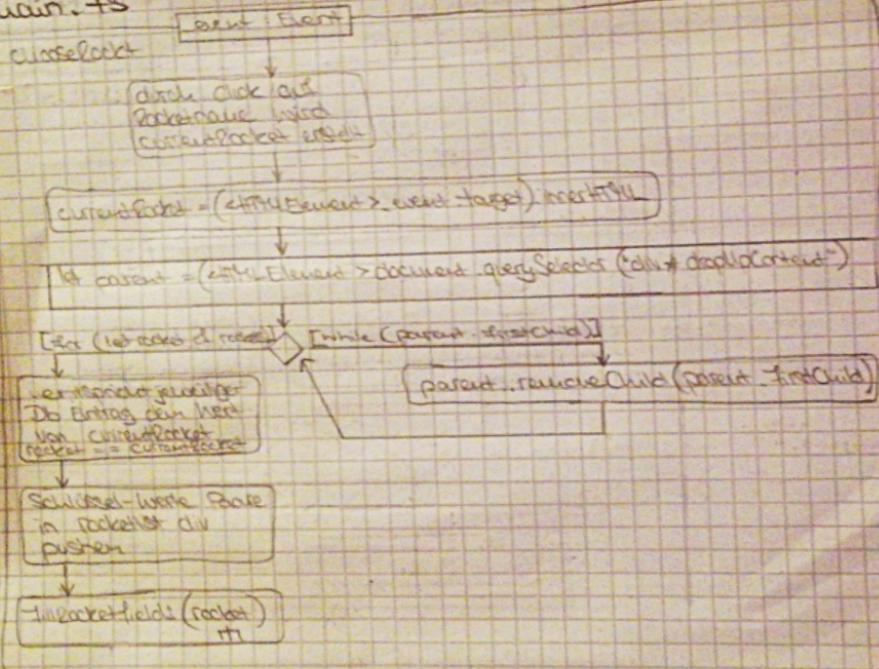


Aktivitätsdiagramm 5 Endabgabe

main.ts



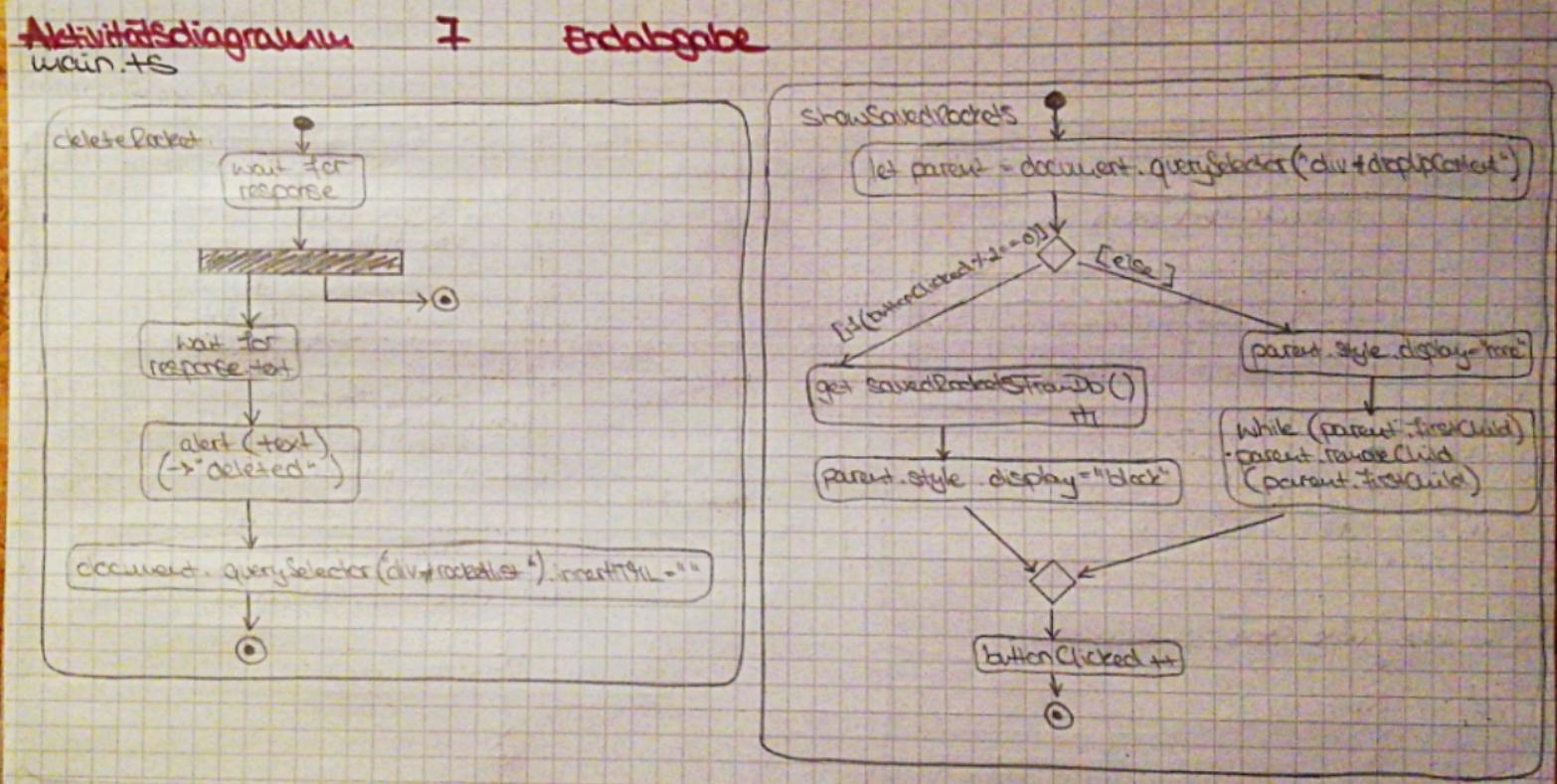
Aktivitätsdiagramm 6 Endabgabe



2.1 Packed Fluids

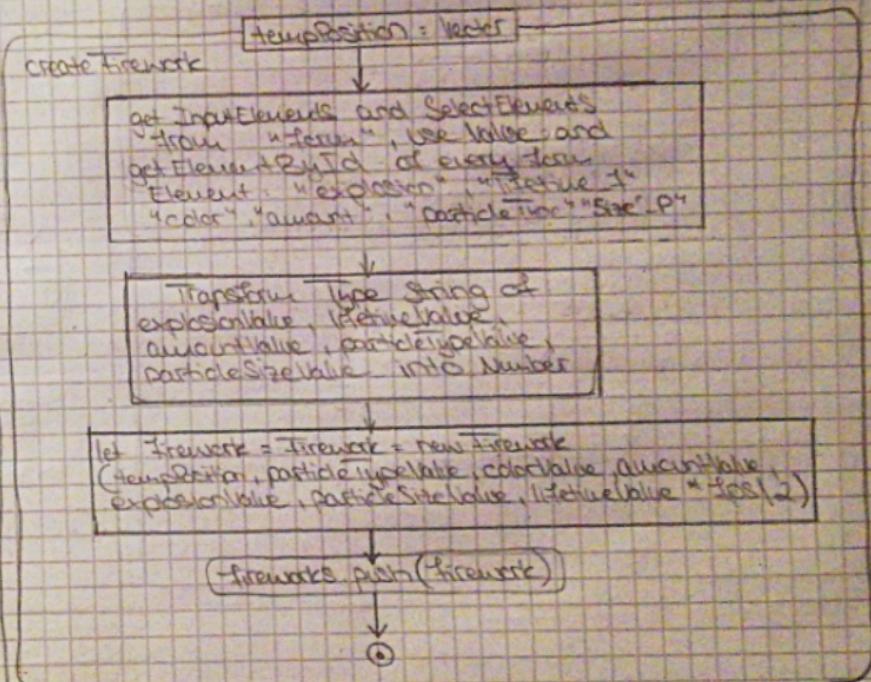
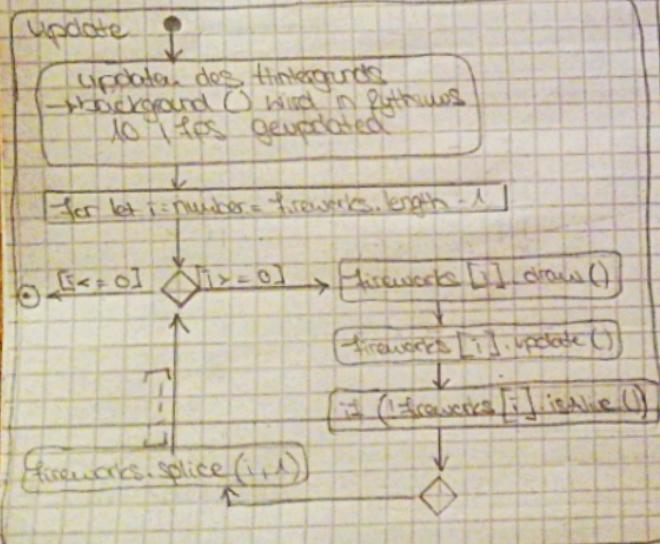
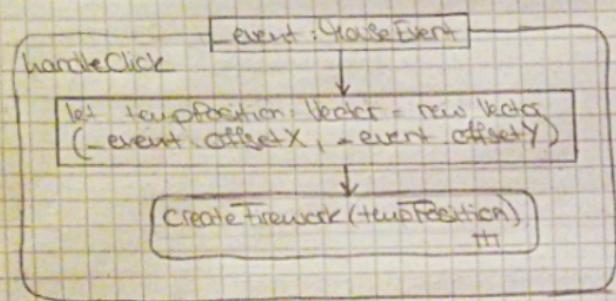
• To work with Data from Form:
→ Use querySelector to get
connected with Elements

```
document.querySelector("mat + rawe") value  
= rocket ["rocke have"]  
document.querySelector("mat + explosion") value  
= rocket ["Explodeze"]  
document.querySelector("mat + lifetime") value  
= rocket ["Lifetime"]  
document.querySelector("select + color") value  
= rocket ["Color"]  
document.querySelector("input + forward") value  
= rocket ["forward"]  
document.querySelector("select + particleType") value  
= rocket ["ParticleType"]  
document.querySelector("input + Site_P") value  
= rocket ["ParticleSite"]
```

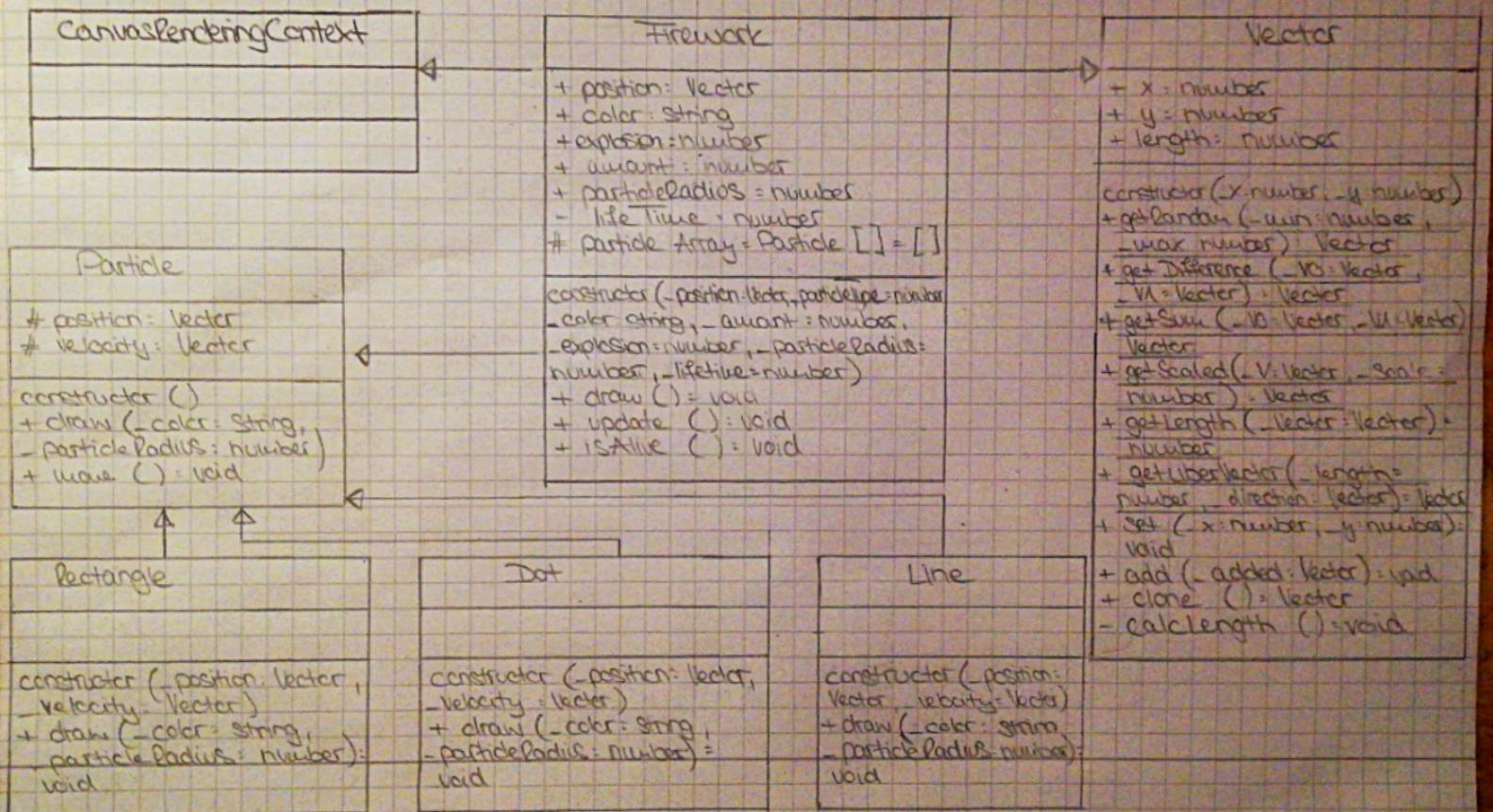


Aktivitätsdiagramm 8 Endgabe

main.ts

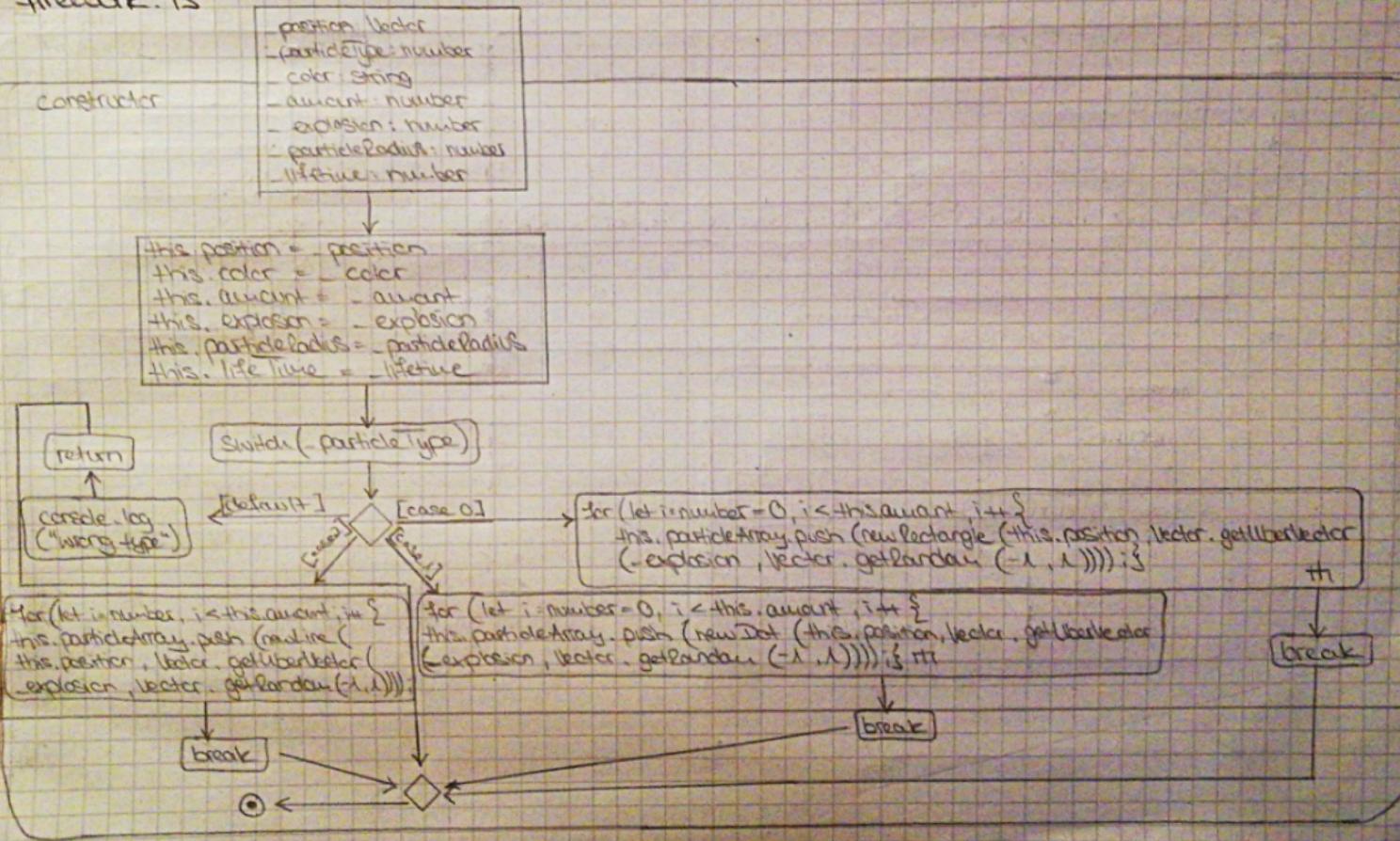


Klassendiagramm - Endabgabe 9.10



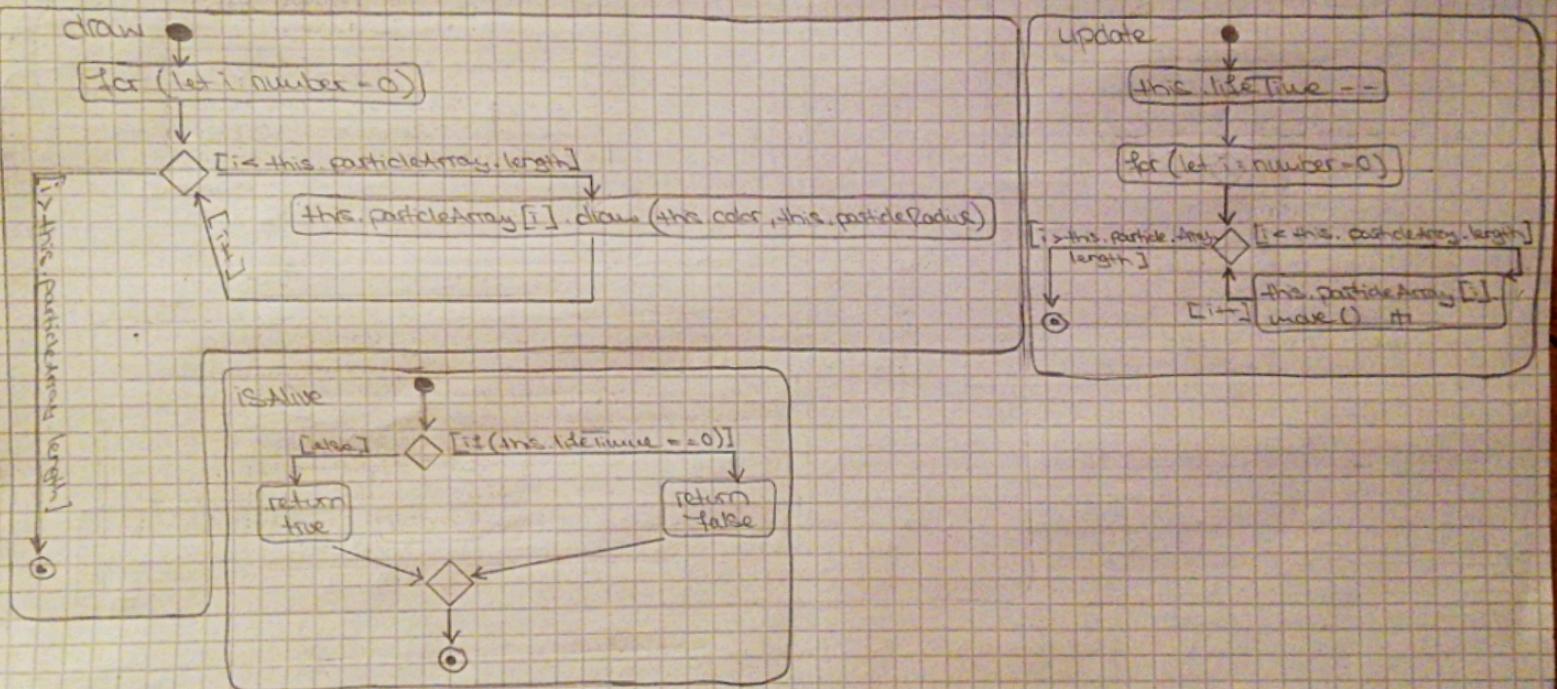
Aktivitätsdiagramm 9.1 Endabgabe

firework.ts



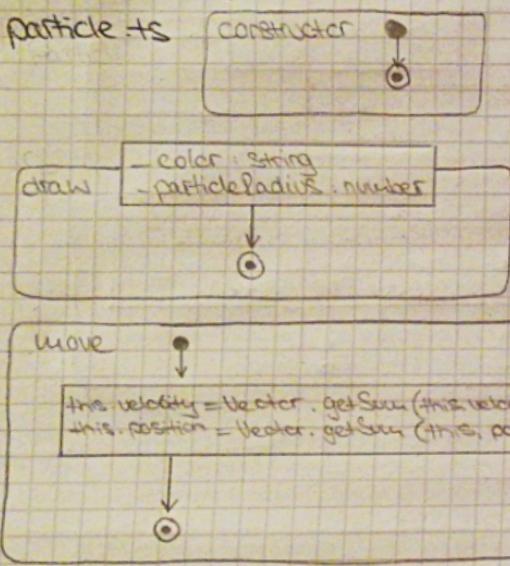
Aktivitätsdiagramm 9.2 Endabgabe

firework.ts

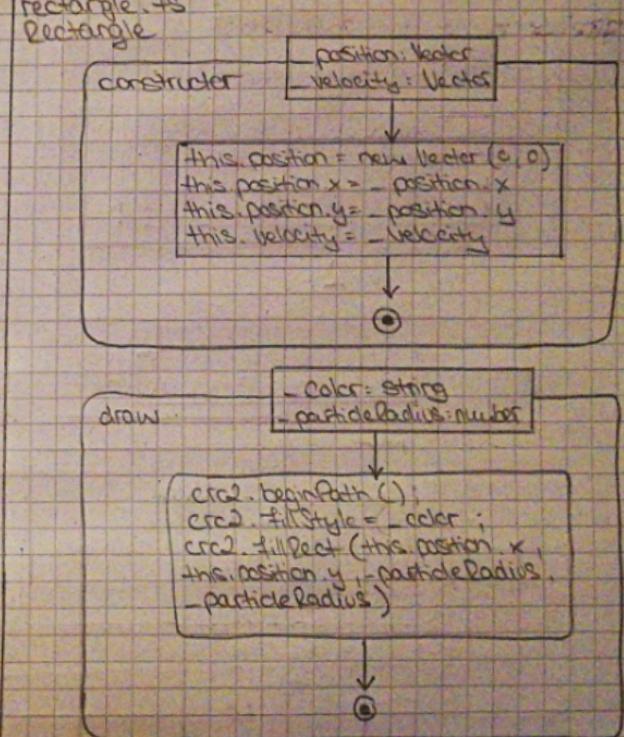


Aktivitätsdiagramm 10 Endabgabe

Particle ts



rectangle, +S
rectangle



Activitätsdiagramm

M

Endabgabe

dot.ts

constructor

- position : Vector

- velocity : Vector

Super()

this.position = new Vector(0,0)
this.position.x = -position.x
this.position.y = -position.y
this.velocity = -velocity



draw

- color : String

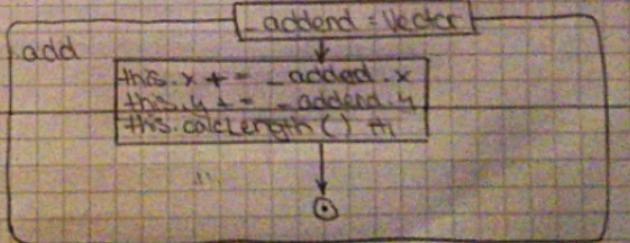
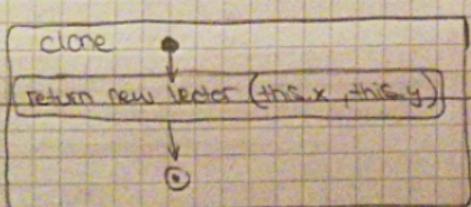
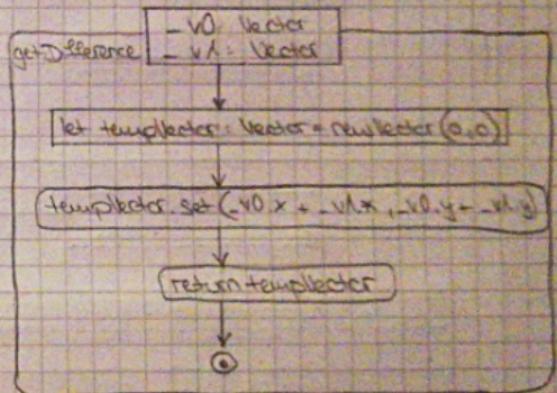
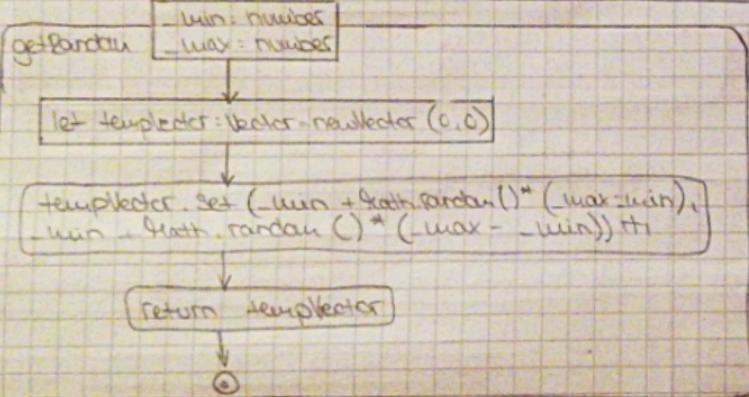
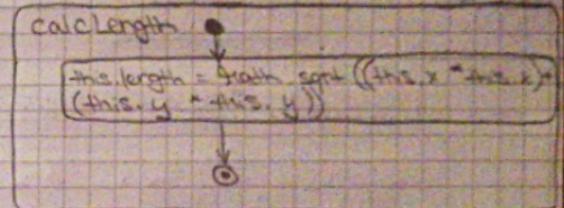
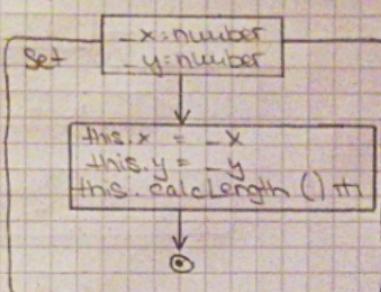
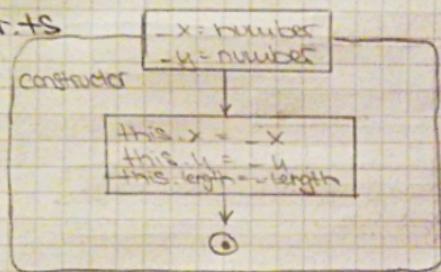
- particleRadius : number

crcl.beginPath();
crcl.fillStyle = -color;
crcl.arc(this.position.x,
this.position.y, -particleRadius,
0, 2 * Math.PI);
crcl.fill



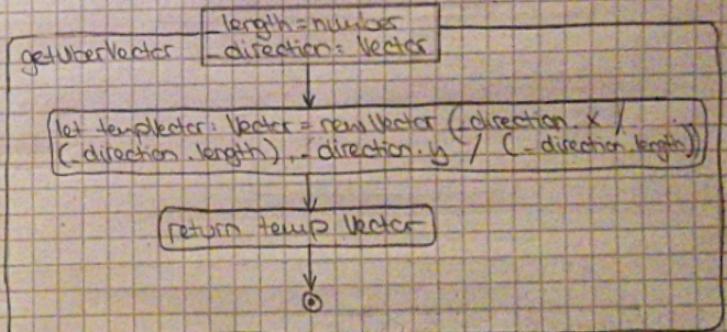
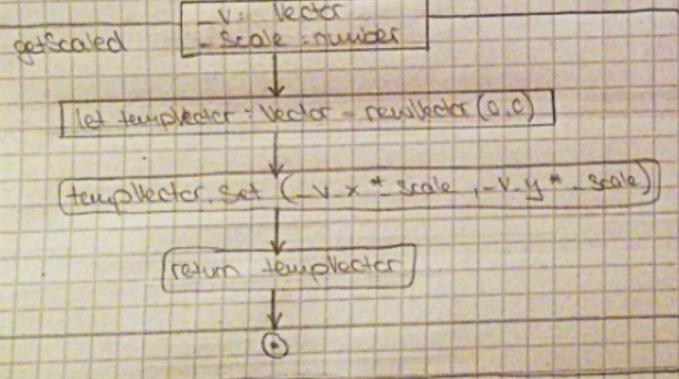
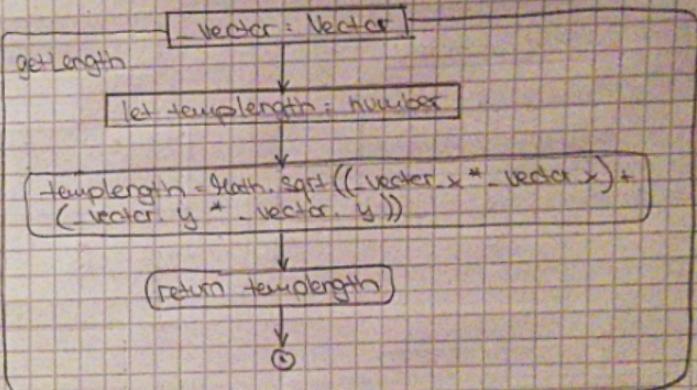
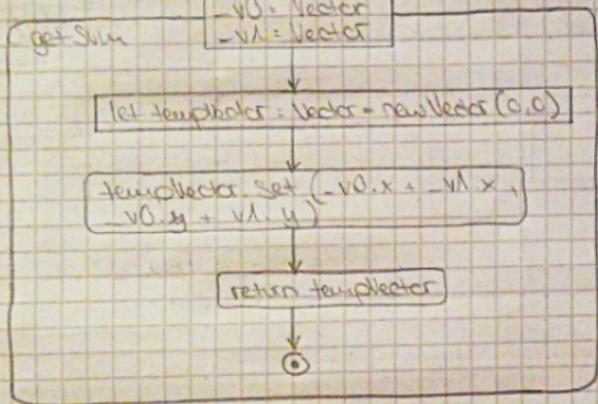
Aktivitätsdiagramm 12 Endabgabe

Vector.ts



Aktivitätsdiagrammum 13 Endabgabe

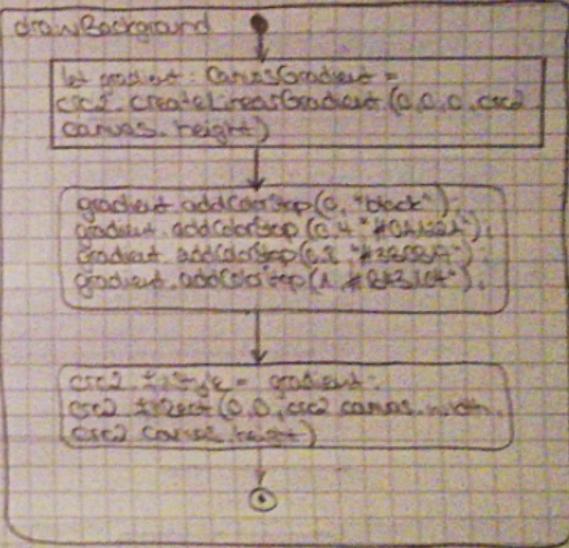
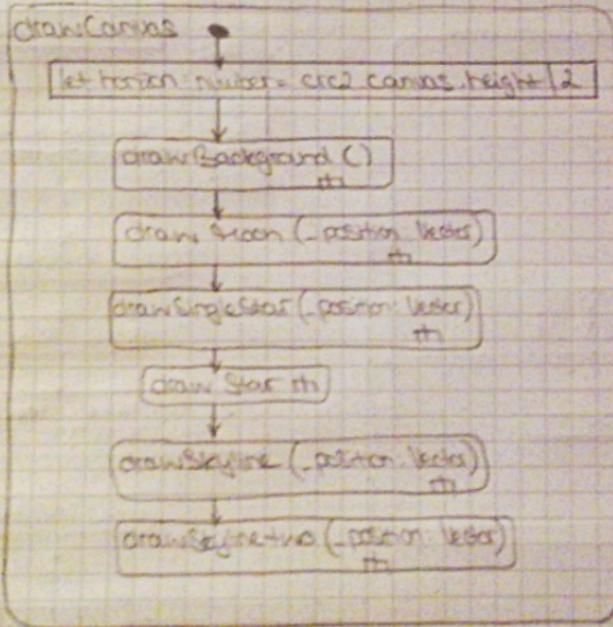
Vector.t3



Aktivitätsdiagramm 14 Endabgabe

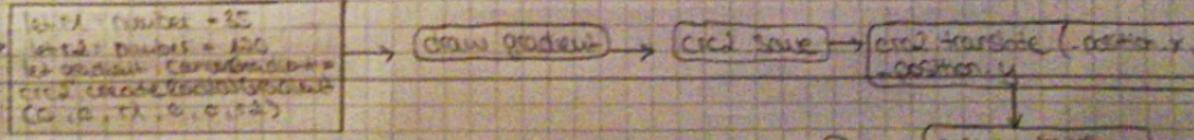
background.ts

Vector
x : number
y : number



drawMoon

position: Vector



Aktivitätsdiagramm 15 Erdgabe

background.ts

