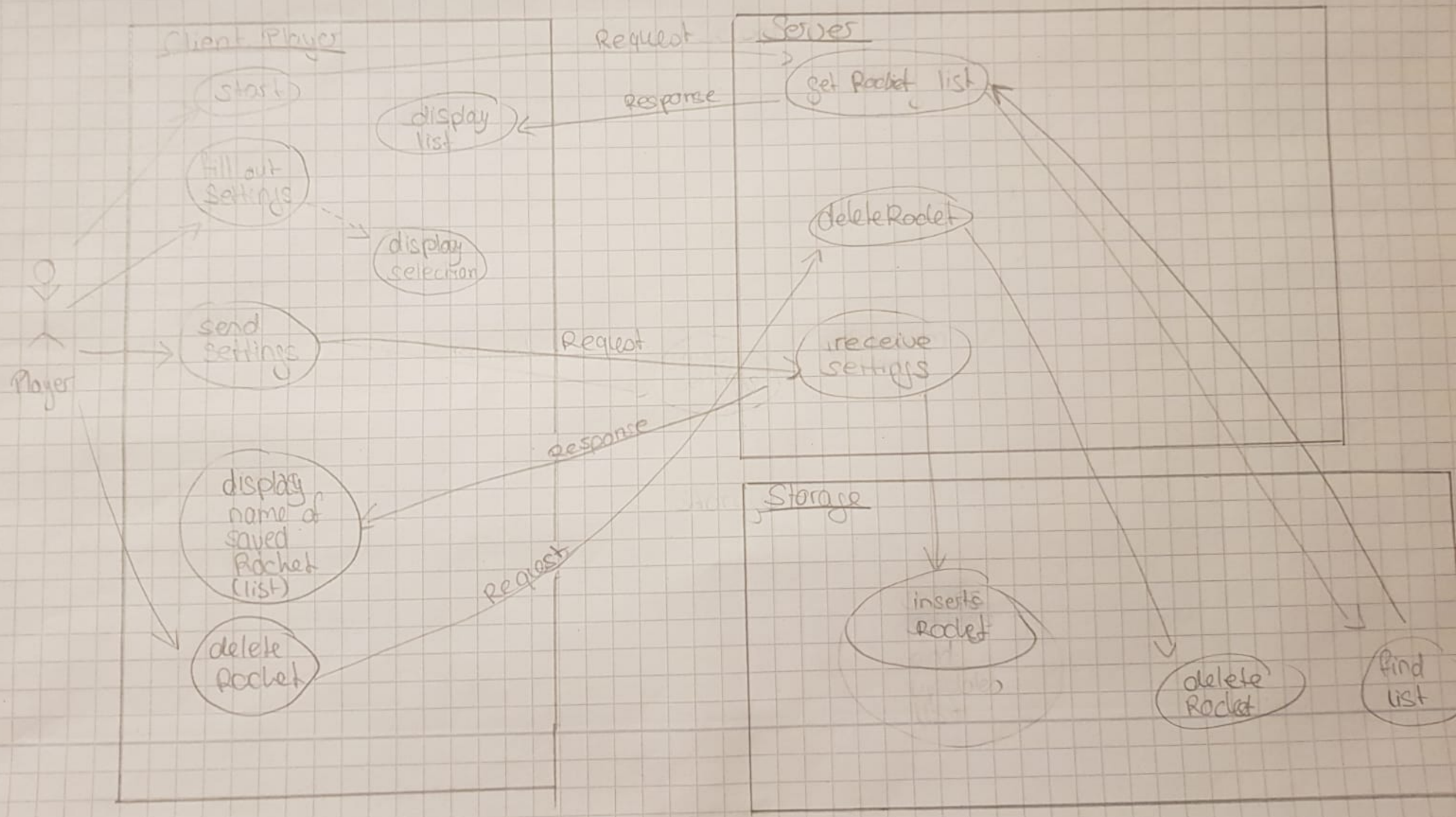
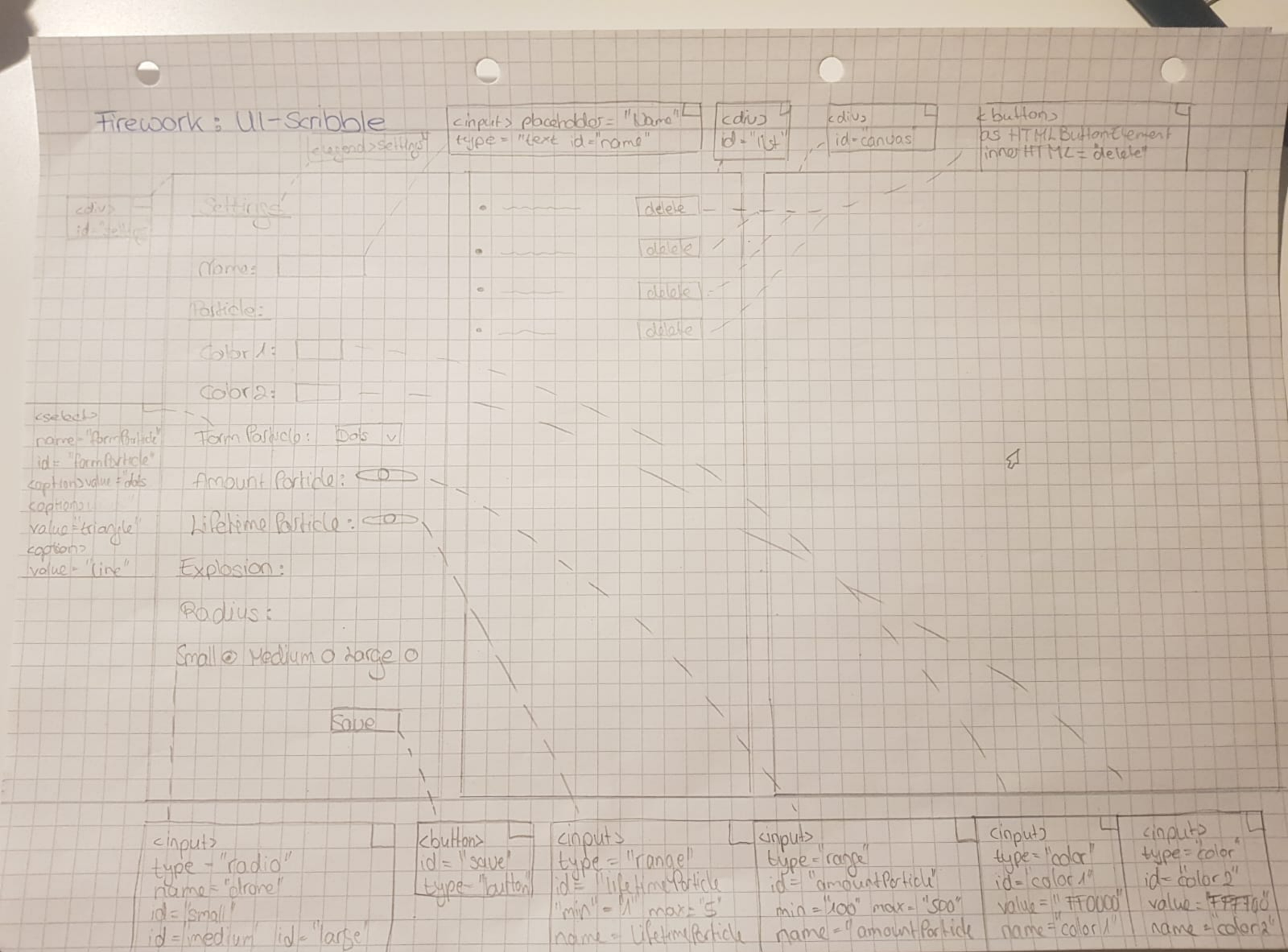
Firework: Use Case Diagramm



Firework : UI-Scribble



Firework: Datastructur

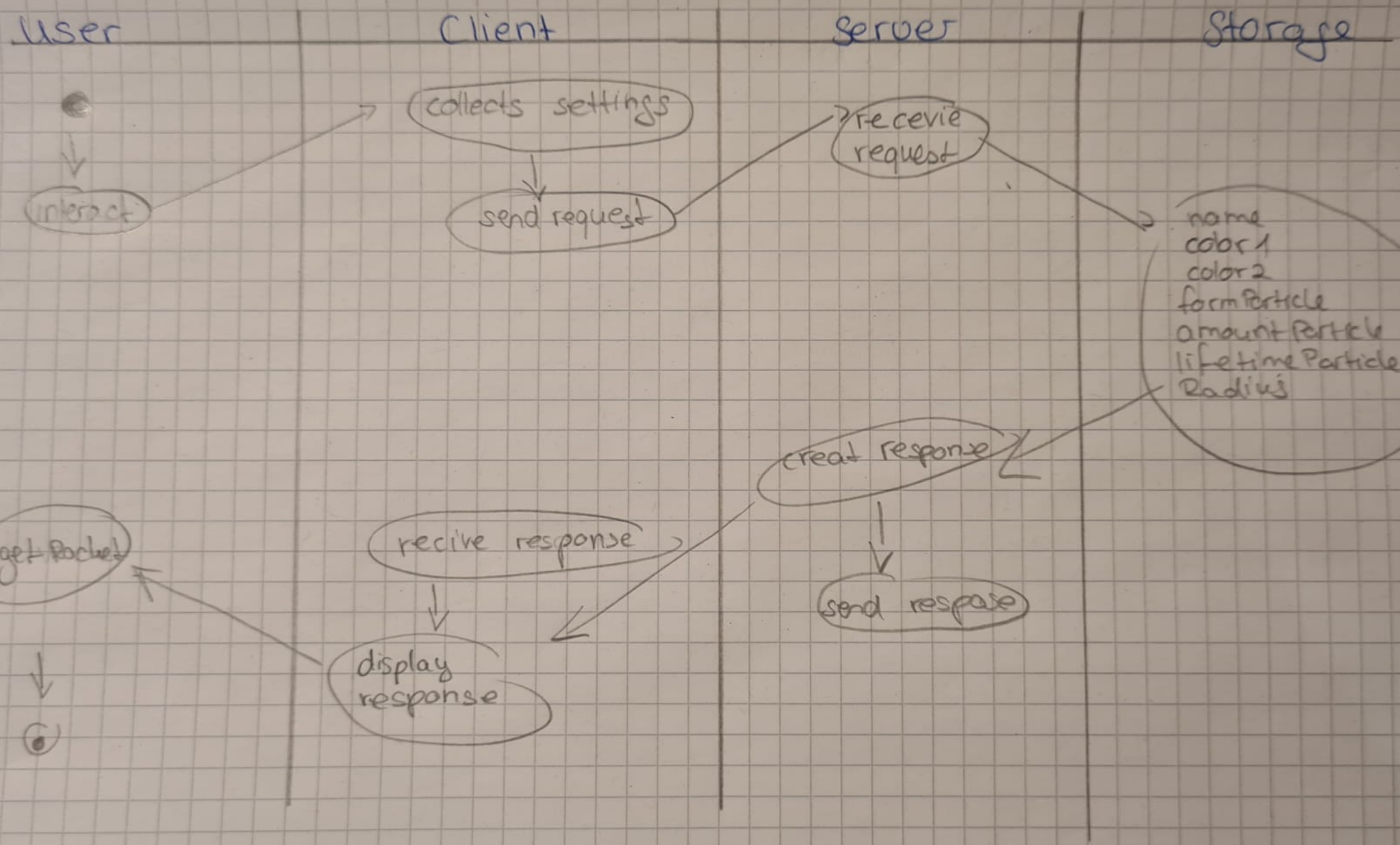
category	name: string/number	value
Settings	name: string	RedYellow
Particle	color1: string	#FF0000
	color2: string	#FFFF00
	formParticle: string	Dots
	amountParticle: number	300
	lifetimeParticle: number	3sec
Explosion	radiusActive	Medium

[data]

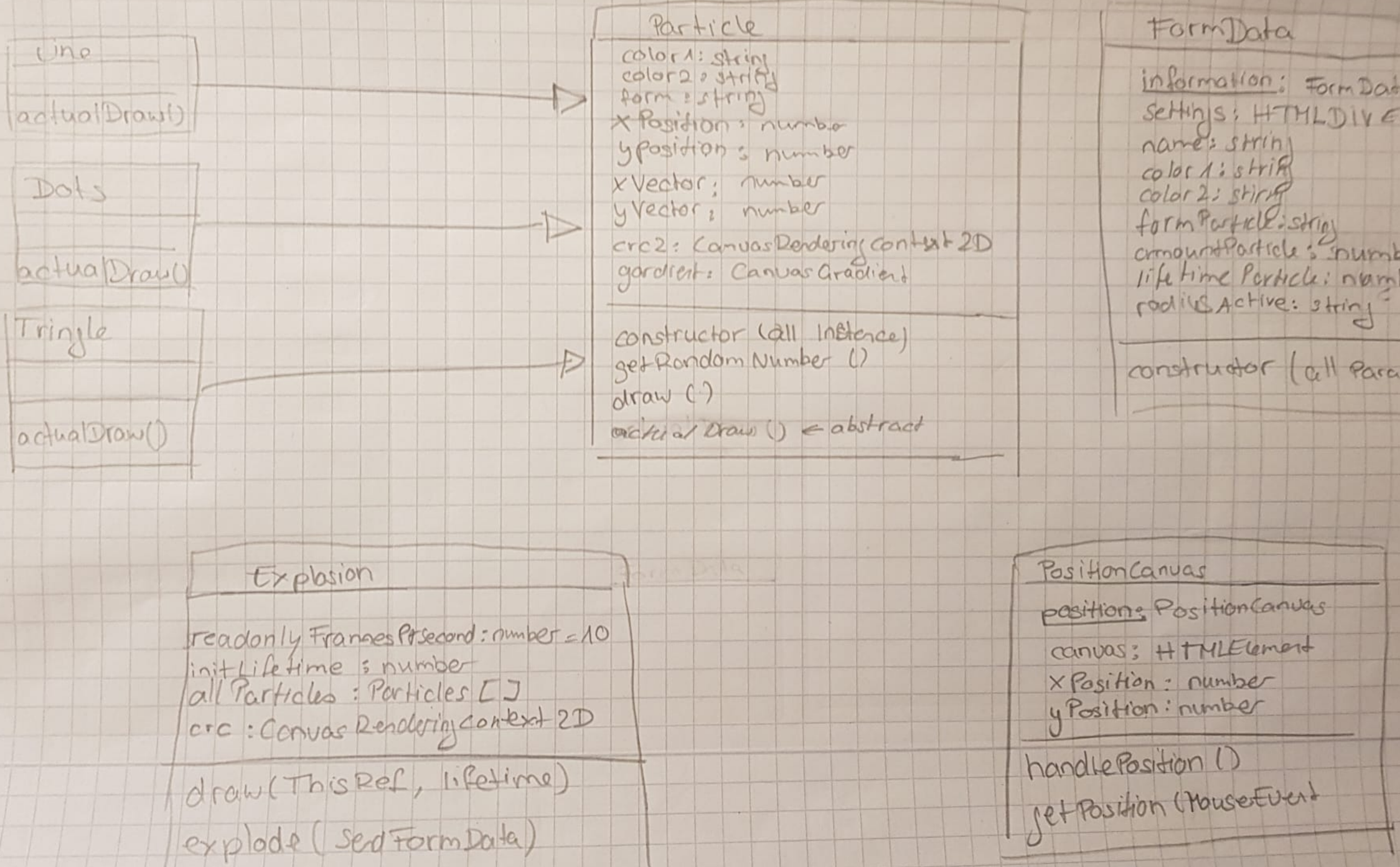
[category]

[value]

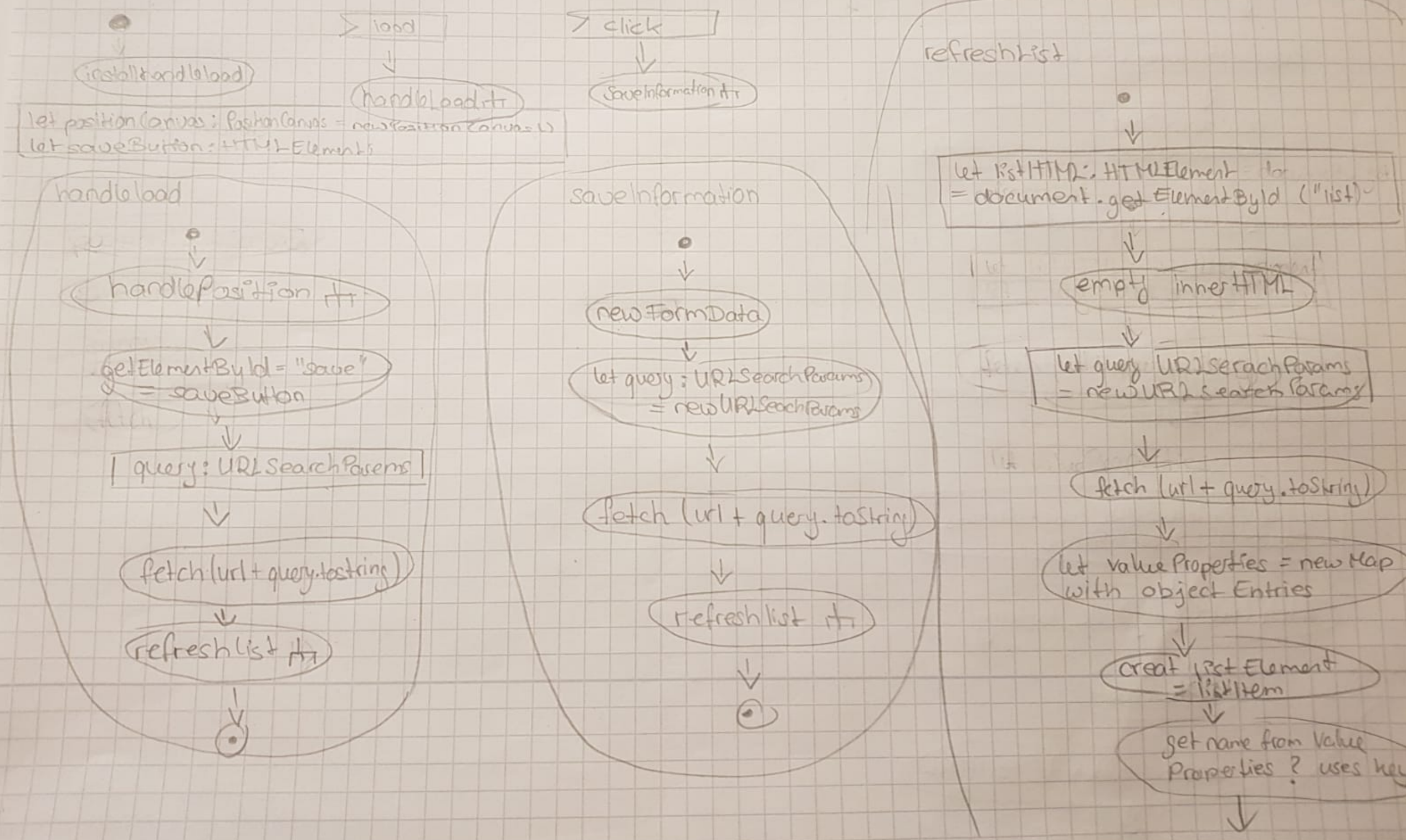
Firework : Swimlane



Firework: Class Diagramm



Firework: Activity Diagramm : Main



refreshList

↓

addEventListener("click") on listItem

↓

new FormData
with all entries from FormData

↓

LIST.HTML.appendchild
(listItem)

↓

create deleteButton with delete
and innerHTML and addEventListener
click on Button

↓

let queryDelete: URLSearchParams
= new URLSearchParams

↓

fetch(url + queryDelete.toString())

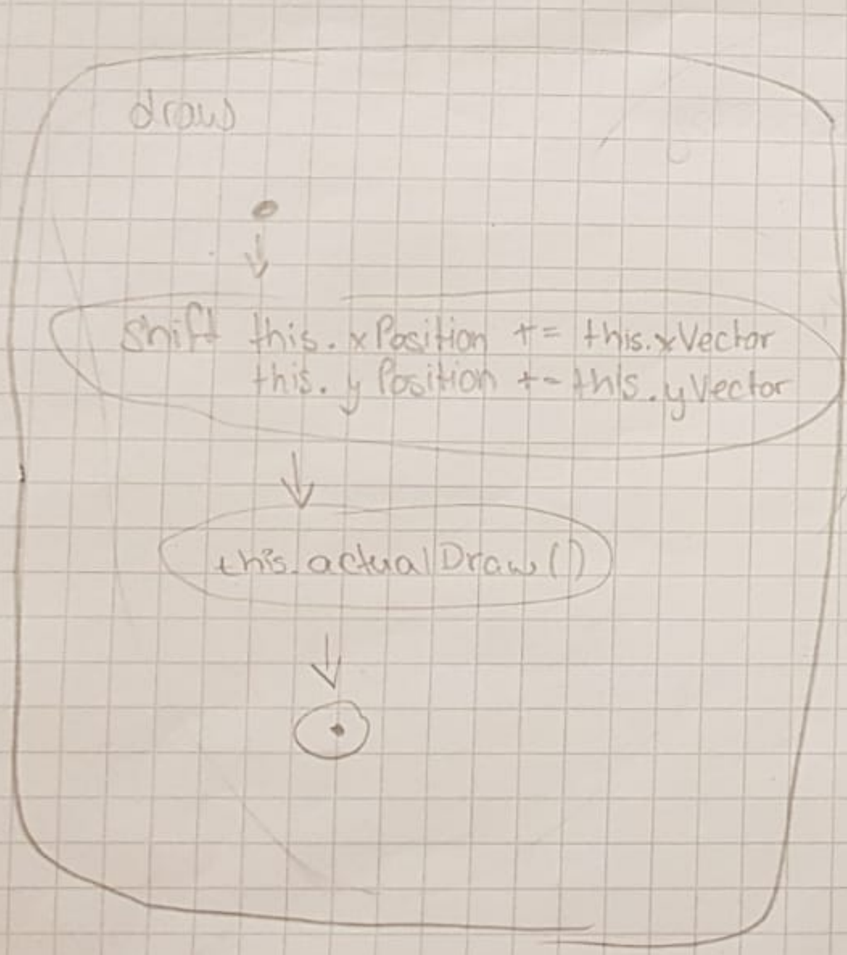
↓

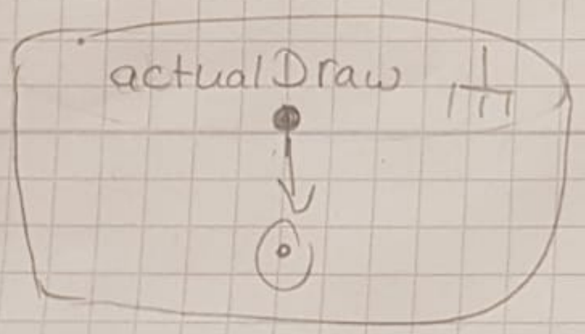
refreshList

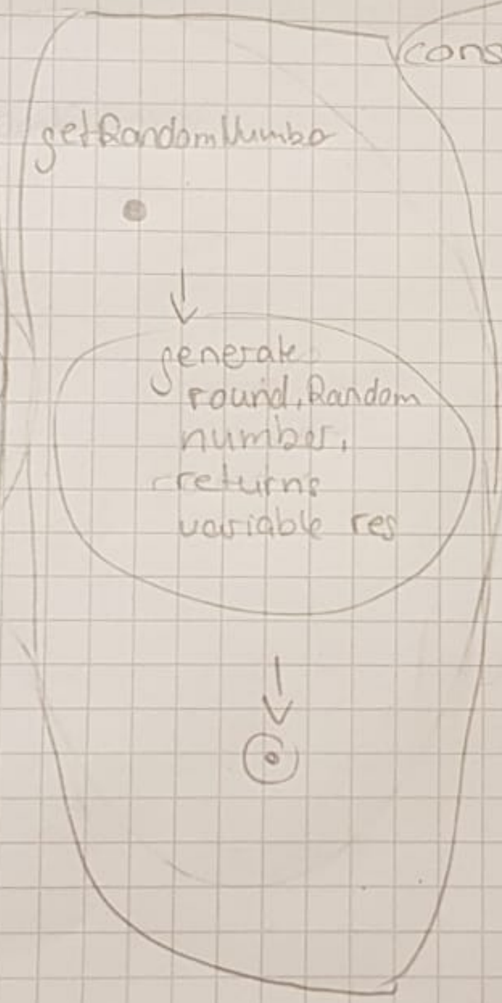
↓

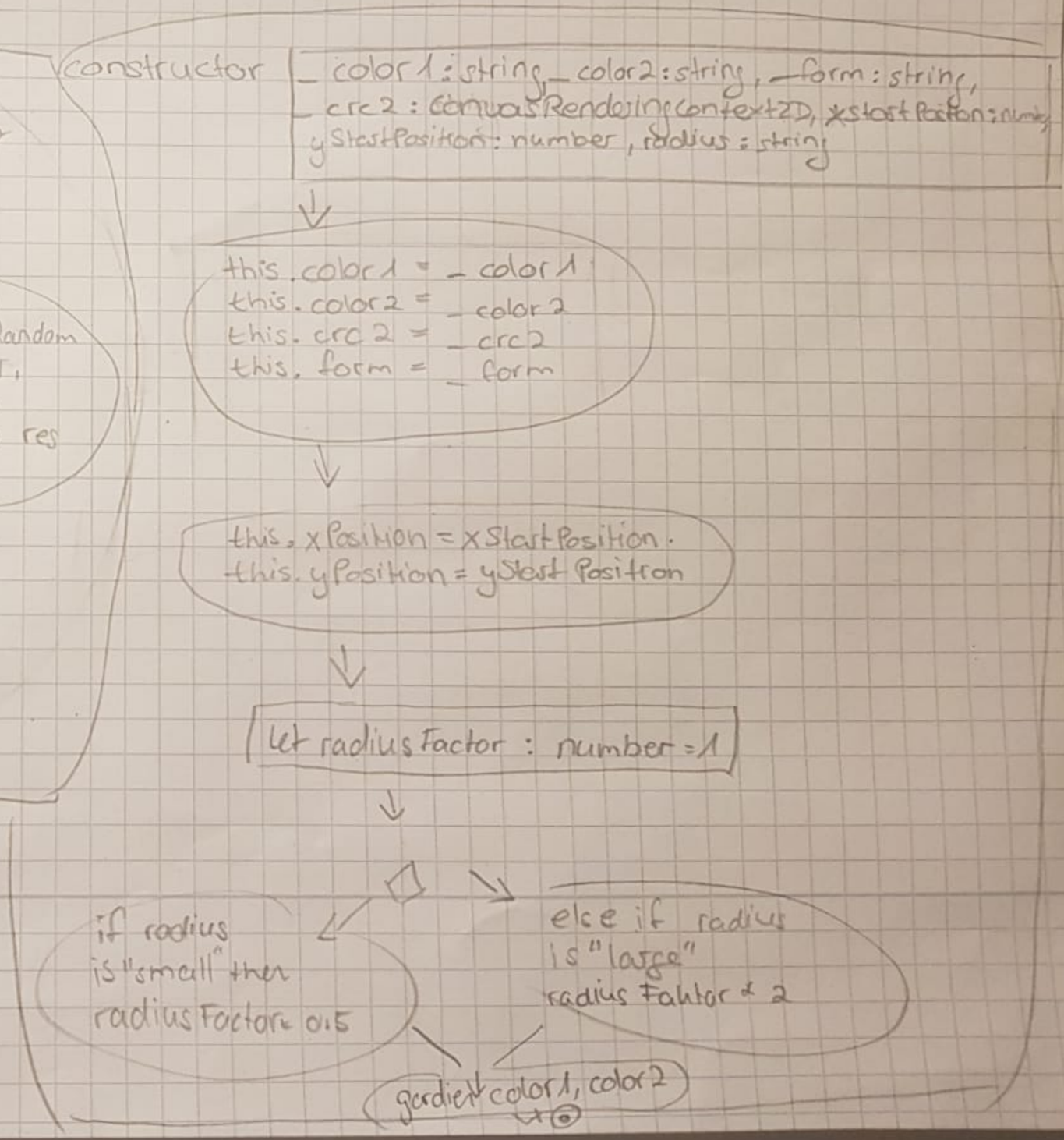
appendButton to listHTML

Firework: Activity Diagramm : Particle

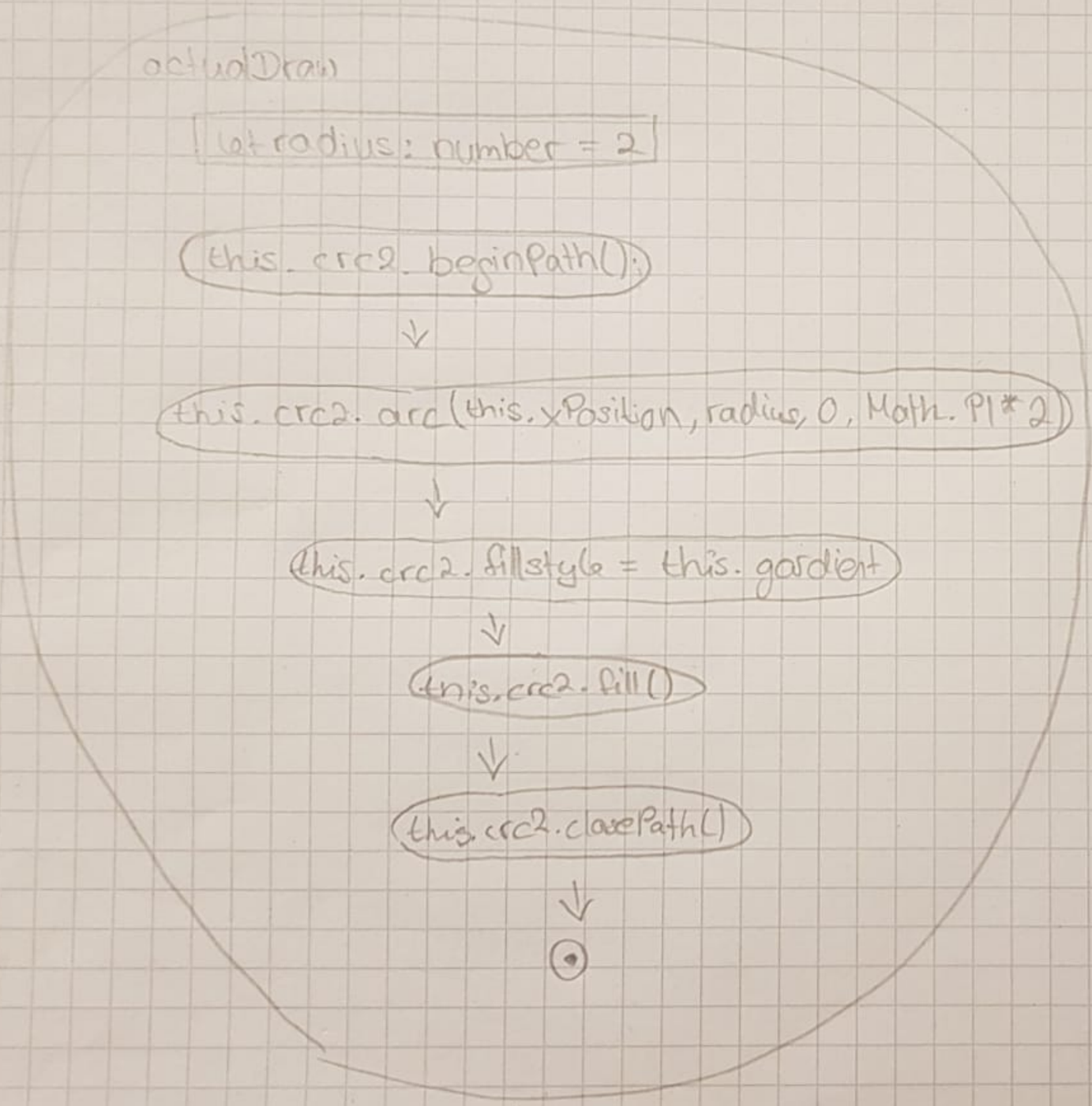




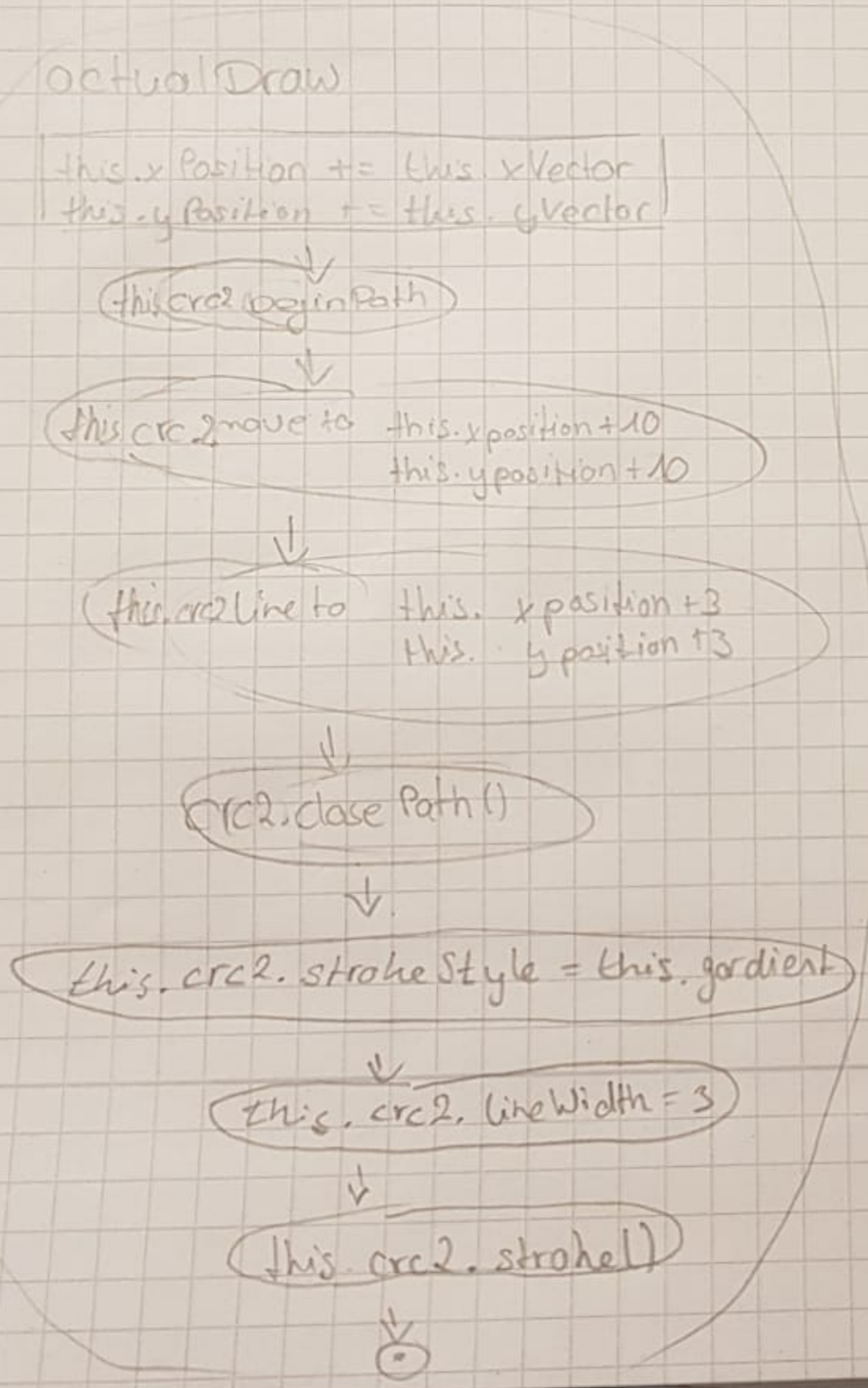




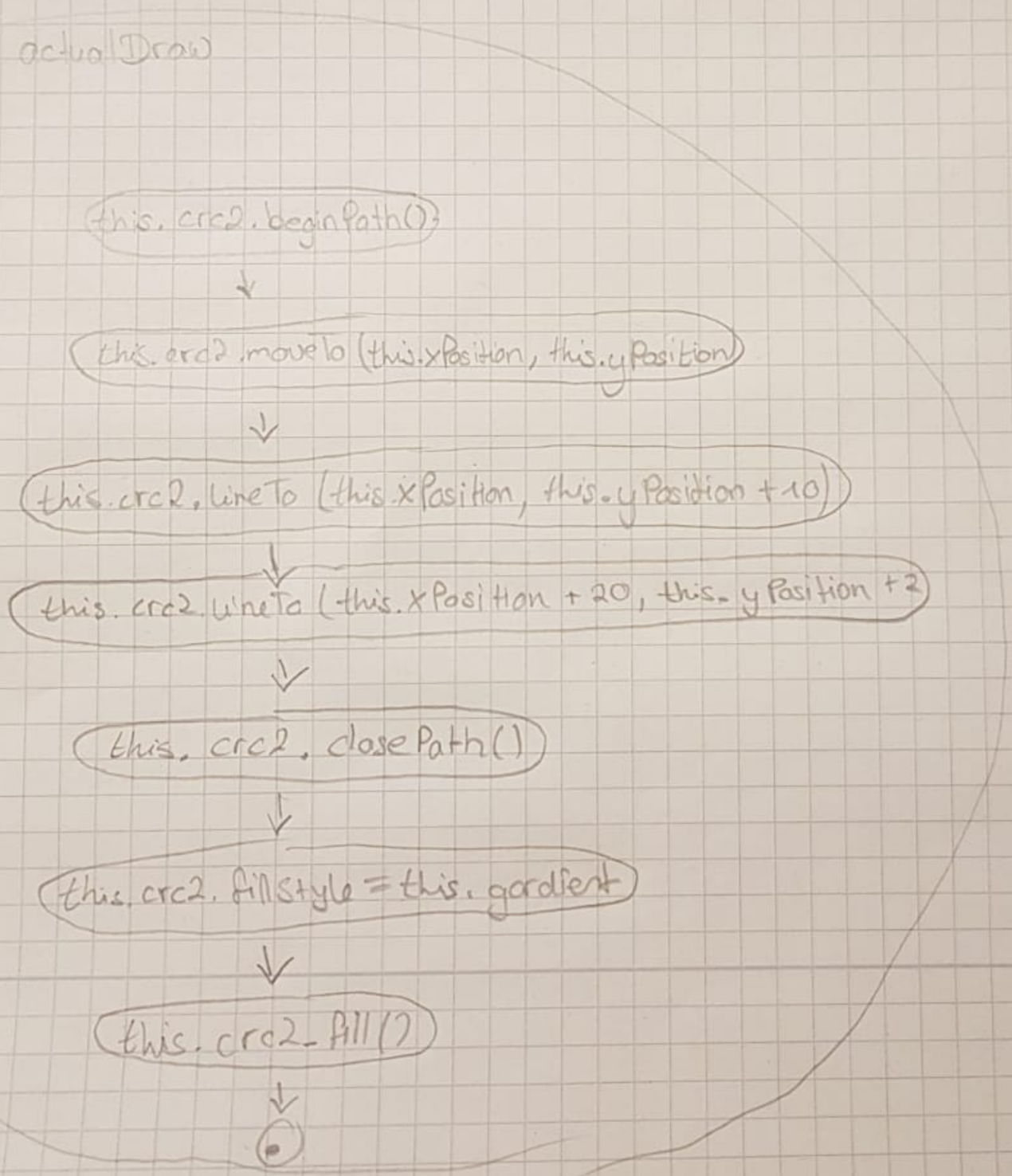
Firework: Activity Diagram: Dots



Firework: Activity Diagramm: Line



Firework: Activity Diagramm : Triangle



Firework: Activity Diagramm : Explosion

readonly FramesPerSecond : number = 10

lifetime : number

allParticles : Particle [ ];

crc : CanvasRenderingContext2D

explode | SendFormData : FormData, x : number, y : number

if we we don't have FormData

return

let newCanvas = creatCanvas

style newCanvas with css elements

appeard newCanvas to 'id = canvas'

this. allParticles = [ ]

this. crc = new Canvas. get context ("2d")

index < SendFormData

let newElement : Particle

draw | ThisRef: Explosion, lifetimer: number

if fistElementChild from canvas == ThisRef.crc.canvas

else

clear canvas

drawParticle black

count lifetime down

if lifetime bigger then 0

else

clear canvas

draw all Podicles
draw ( ) ++
settimeout (1/ExplorionFramesPerSecond)

explode

sendFormData . fromParticle

case "dots"
→ newElement = newDots with DataValus

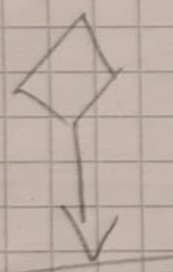
case "triangle"
→ newElement = newTriangle with DataValus

case "like" → newElement = newLike with DataValus

default → console.log

back to for-Loope

for-loop



allParticles [index] = newElement

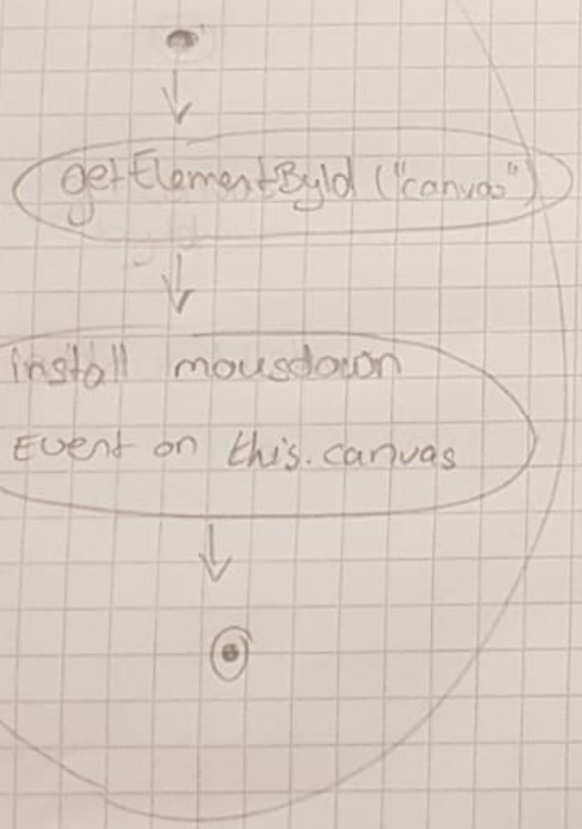
let lifetime : number = SendFormData.lifetimeParticle * Explosion.FramesPerSecond
this.initlifetime = lifetime

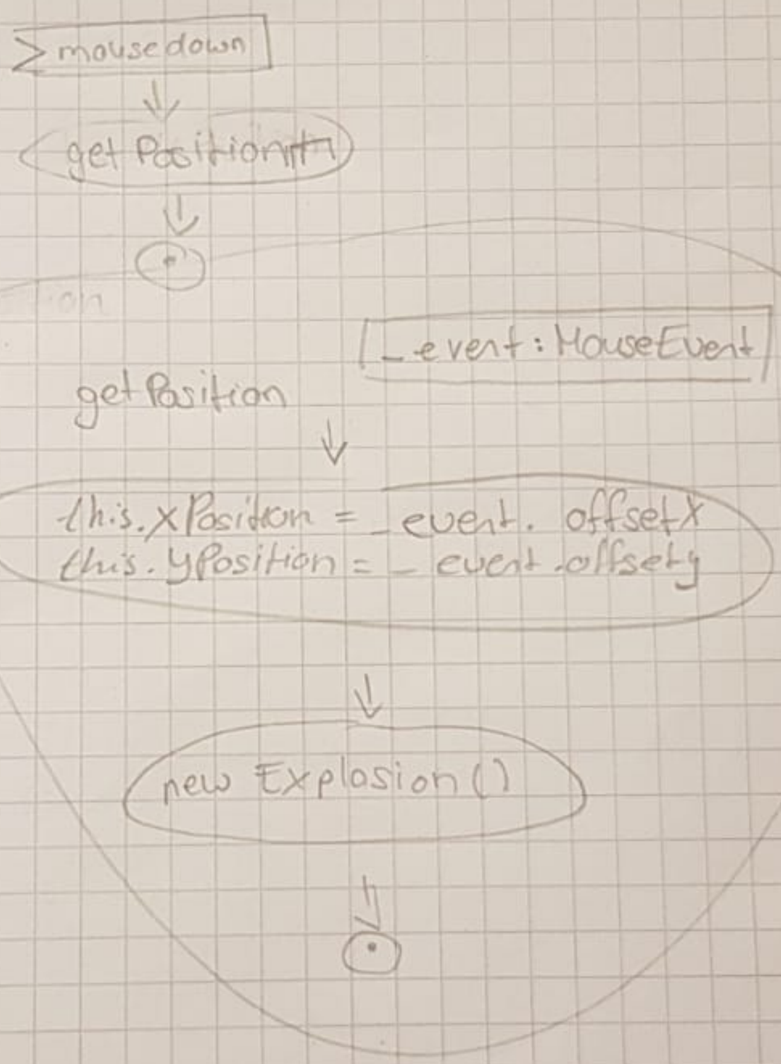
Explosion.draw (this, lifetime)

Firework: Activity Diagramm: PositionCanvas

position : PositionCanvas
canvas : HTMLElement
xposition : number
yposition : number

handlePosition





Firecoork : Activity Diagramm : FormData

information : FormData

setting: HTMLDivElement = document.querySelector('#settings')

name: string

color1: string

color2: string

formParticle: number

amountParticle: number

lifeTimeParticle: number

radiusActive: string

Constructor

- name?: string, - color1?: string, - color2?: string,
- formParticle?: string, - amountParticle?: number,
- lifeTimeParticle?: number, - radiusActive?: string