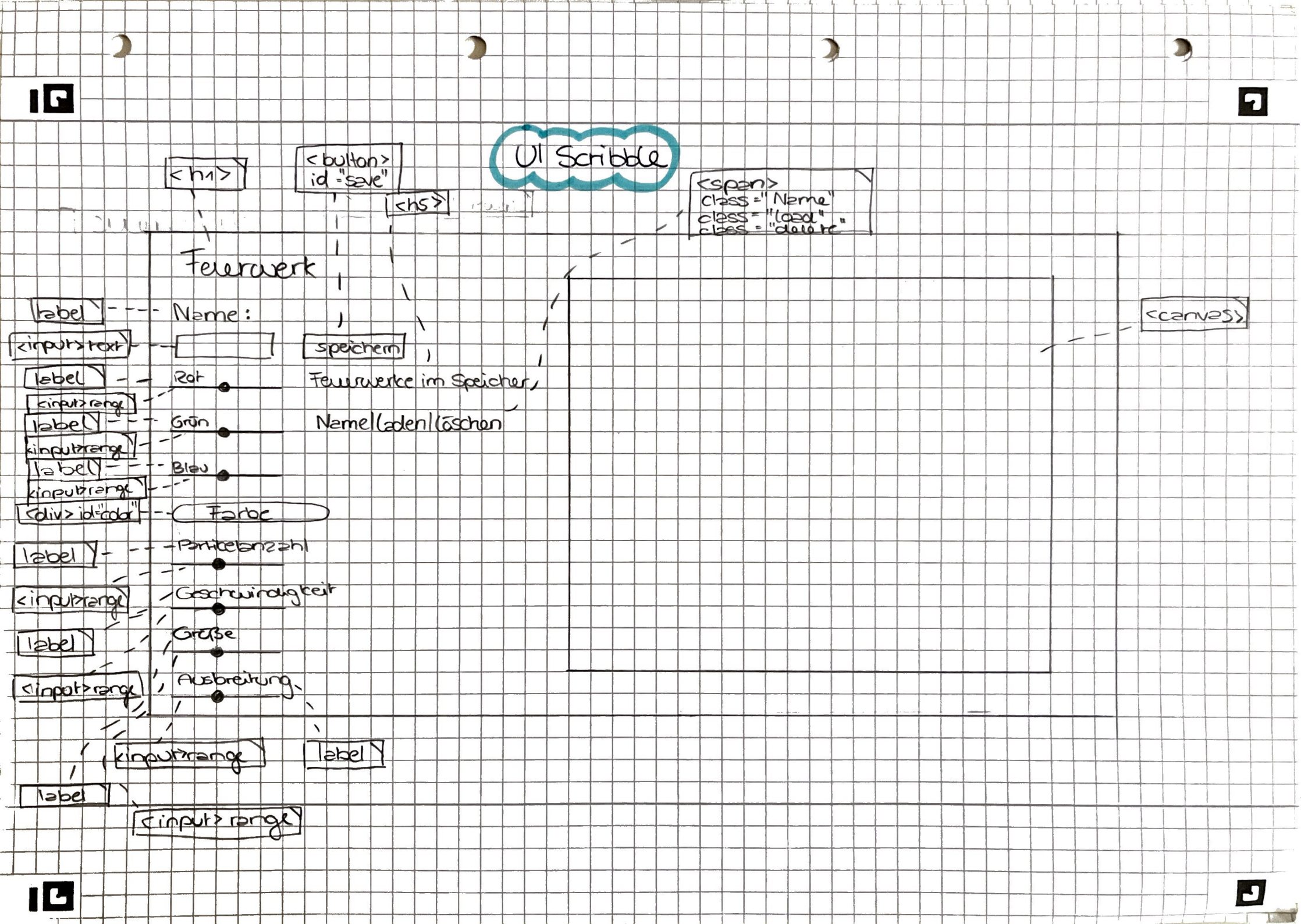
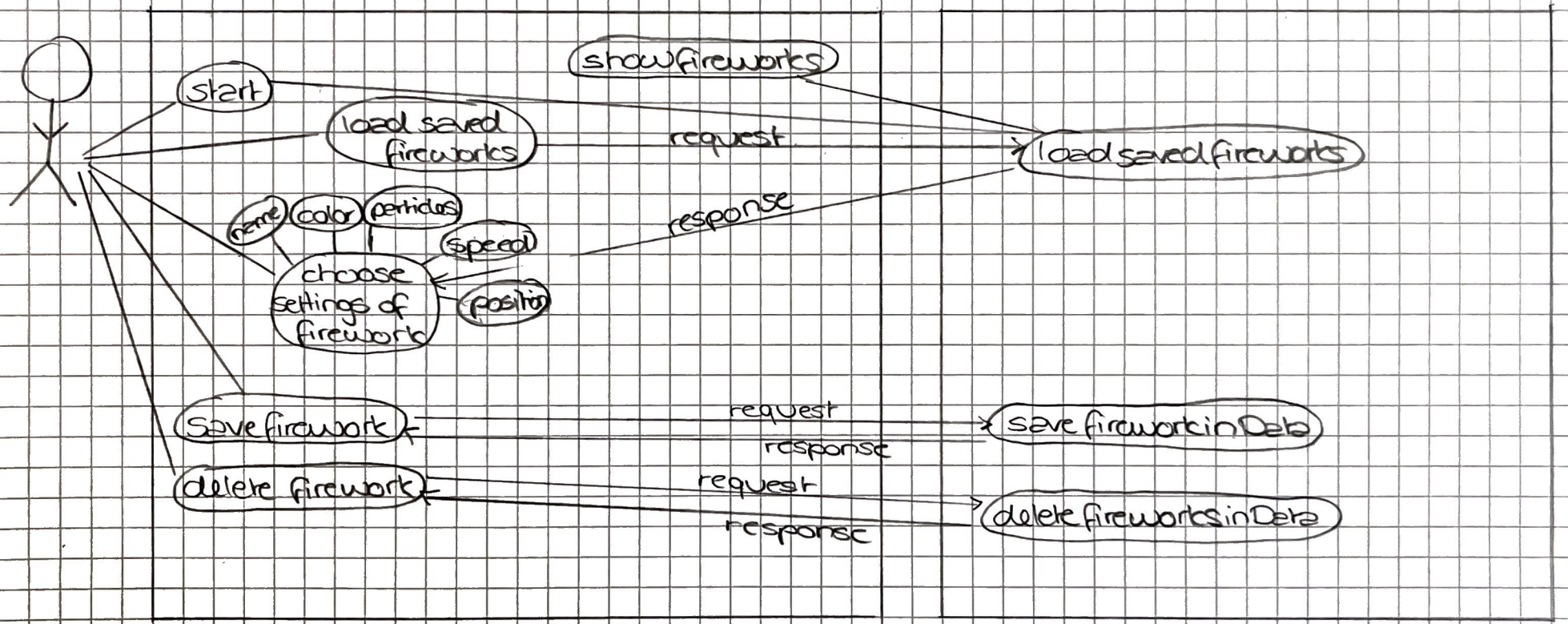
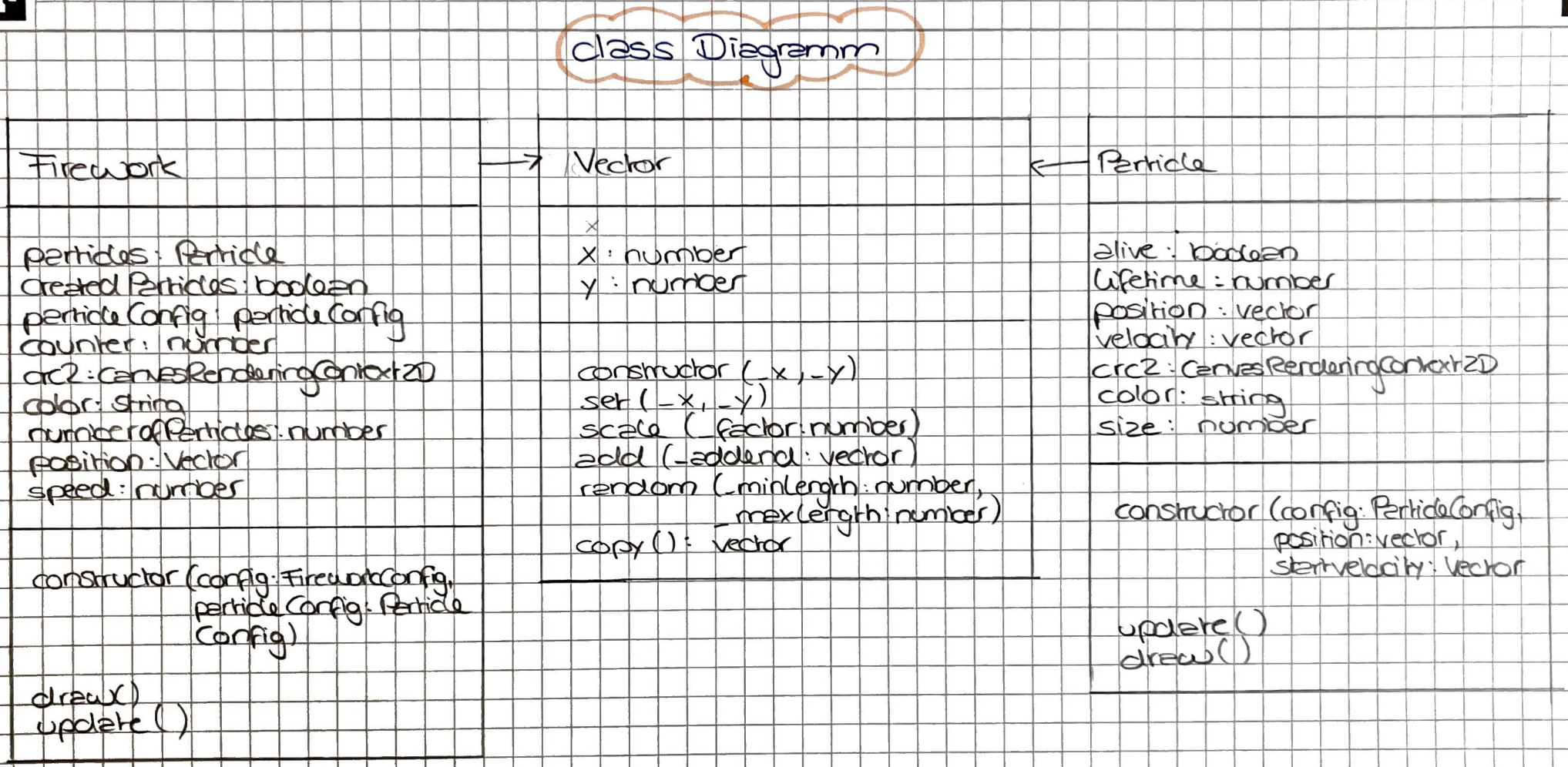

UI Scribble
<h1>
<button>
id="save"
<h5>

class="Name"
class="load"
class="delete"
Feuerwerk
label
Name:
<input> text
speichern
label
Rot
<input> range
Feuerwerke im Speicher:
label
Grün
<input> range
Name | laden | löschen
label
Blau
<input> range
 id="color"
Farbe
label
Partikelanzahl
<input> range
Geschwindigkeit
label
Größe
<input> range
Ausbreitung
<input> range
label
<input> range
<canvas>

Use Case Diagram

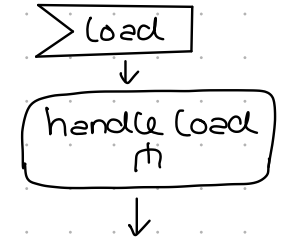


class Diagramm

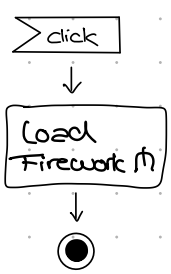


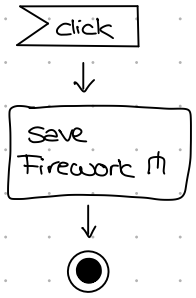
Activity Diagram

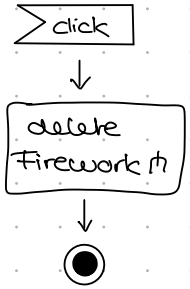
Main

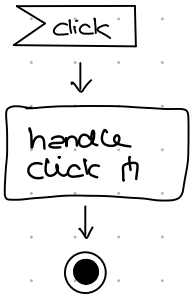


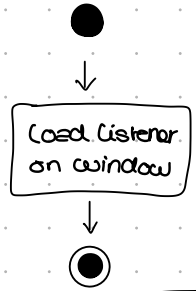

```
let canvas : HTMLCanvasElement;
let crc2 : CanvasRenderingContext;
let fireworks : Firework [] = [];
let rect : DOMRect;
let name : string = "Mein Feuerwerk";
let color : string = "ff0000";
let red : number = 255;
let green : number = 0;
let blue : number = 0;
let numberOfParticles : number = 50;
let speed : number = 5;
let size : number = 1;
let width : number = 0;
```

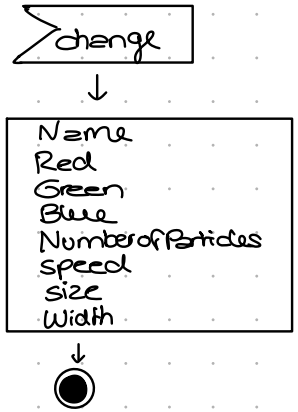


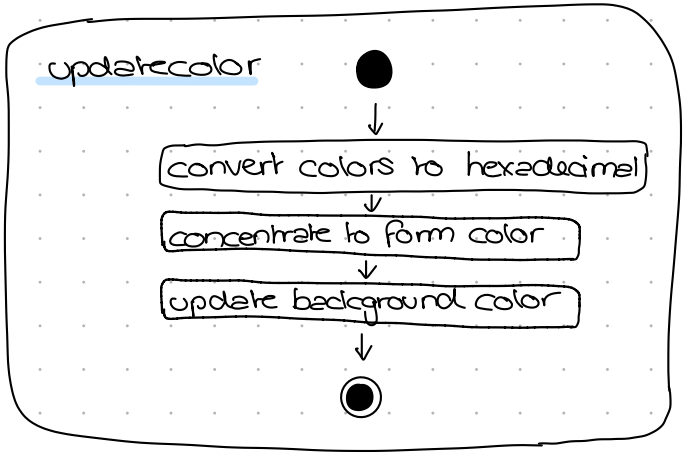


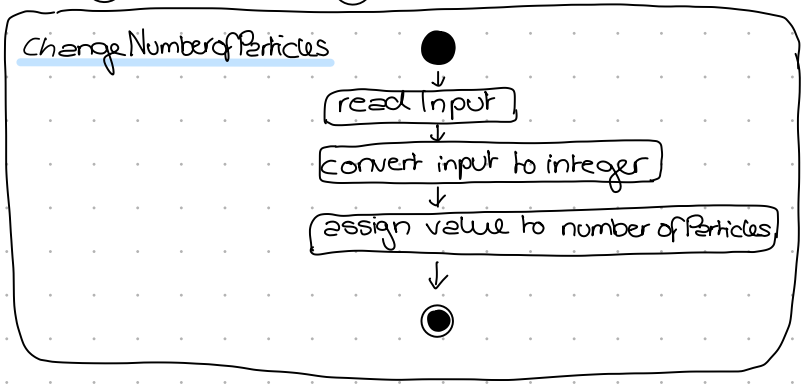


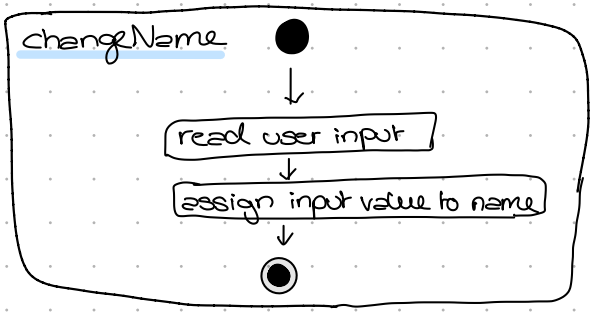


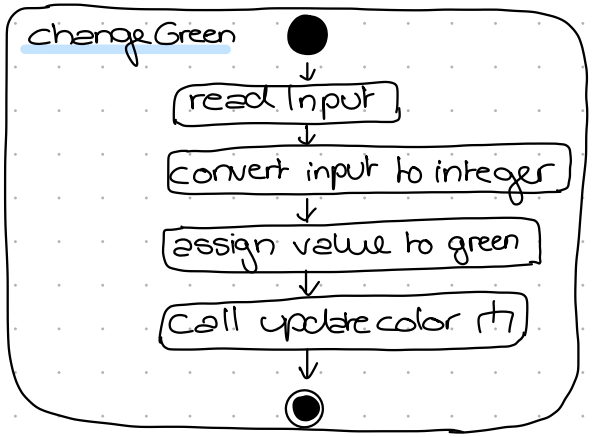


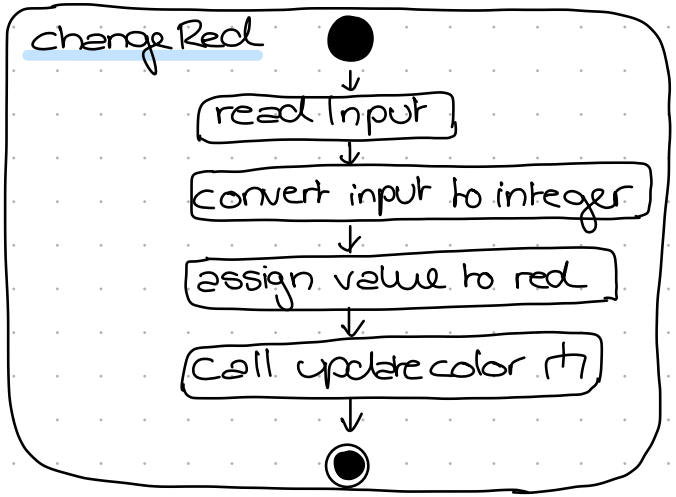


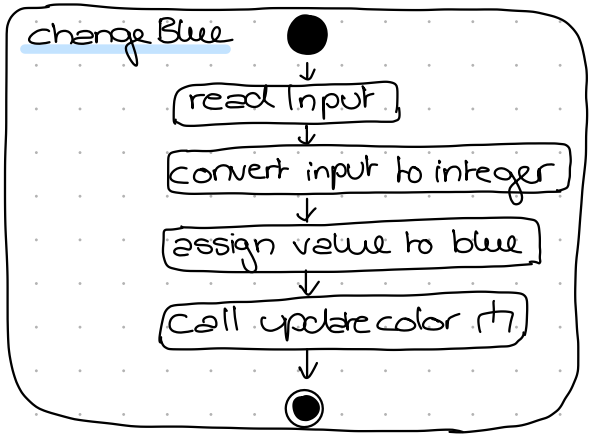


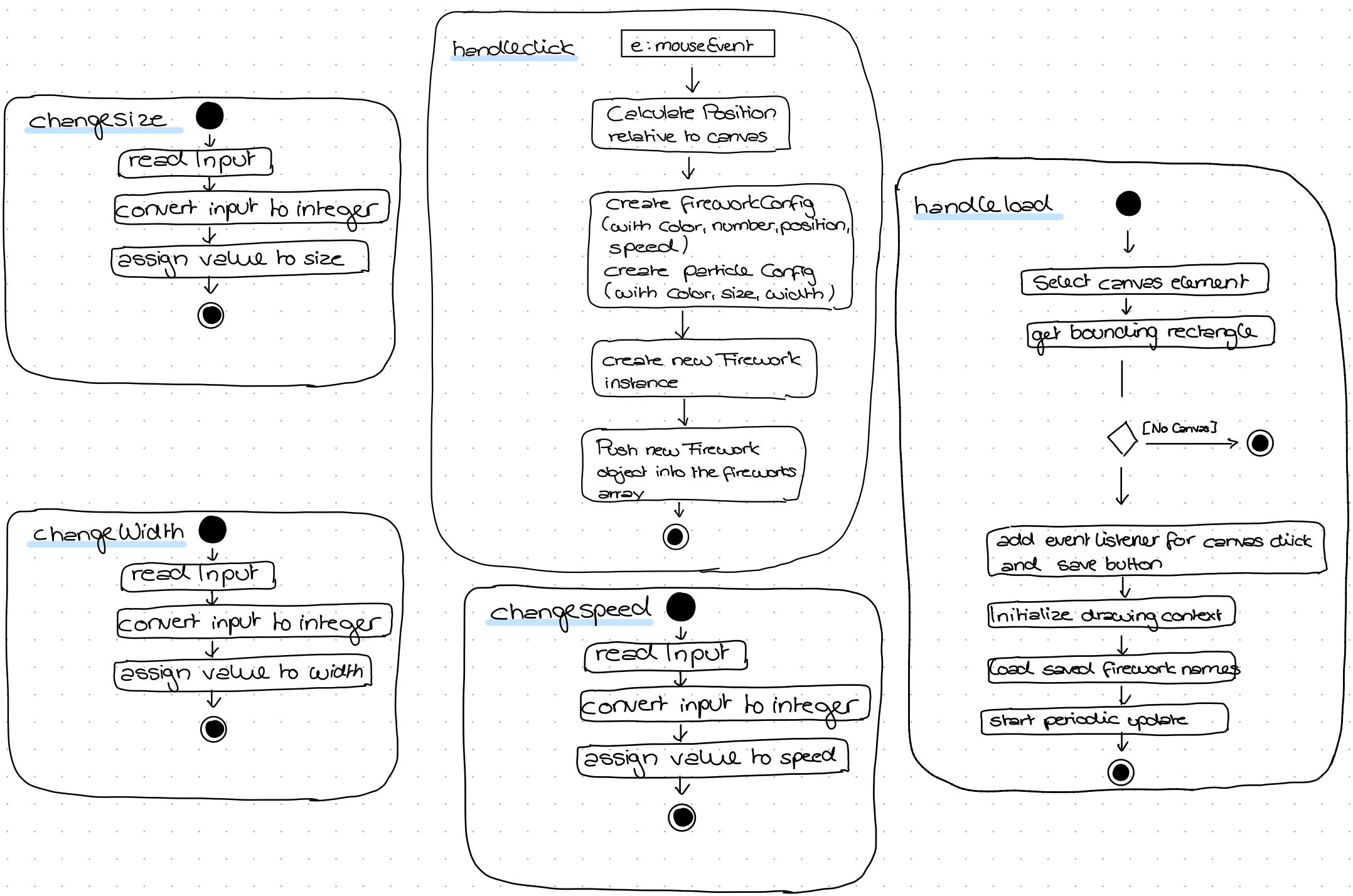


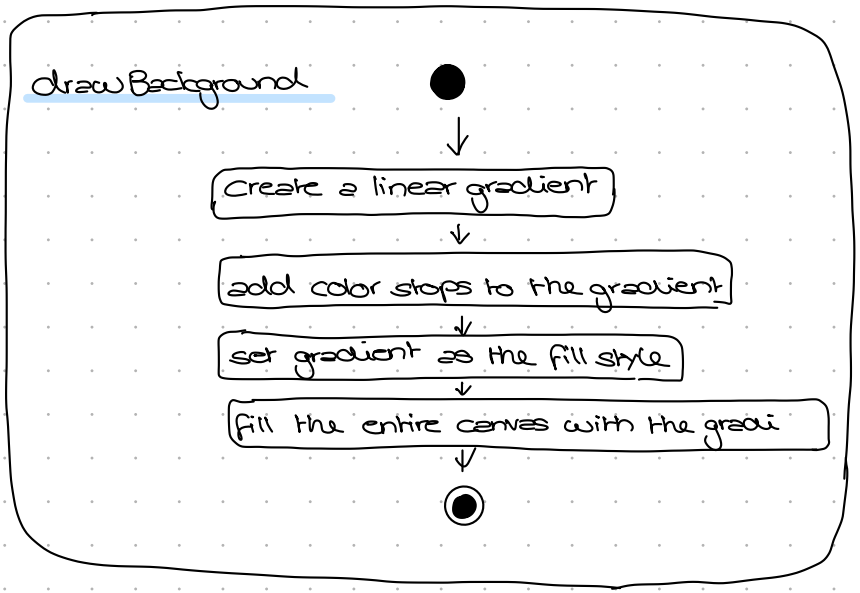


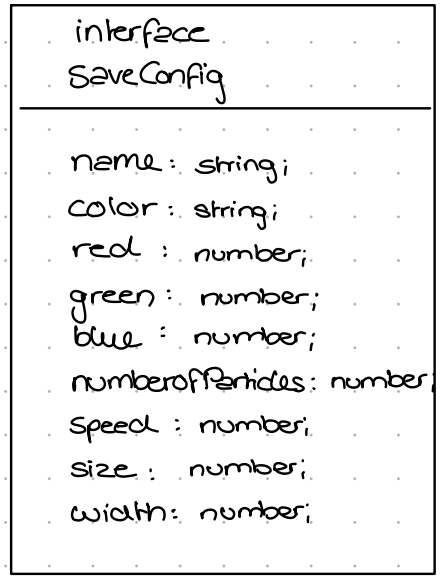


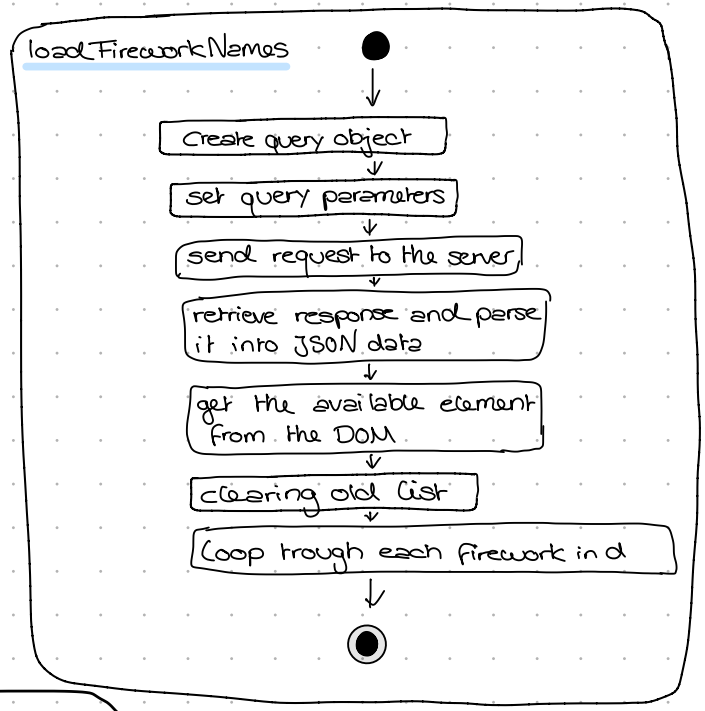


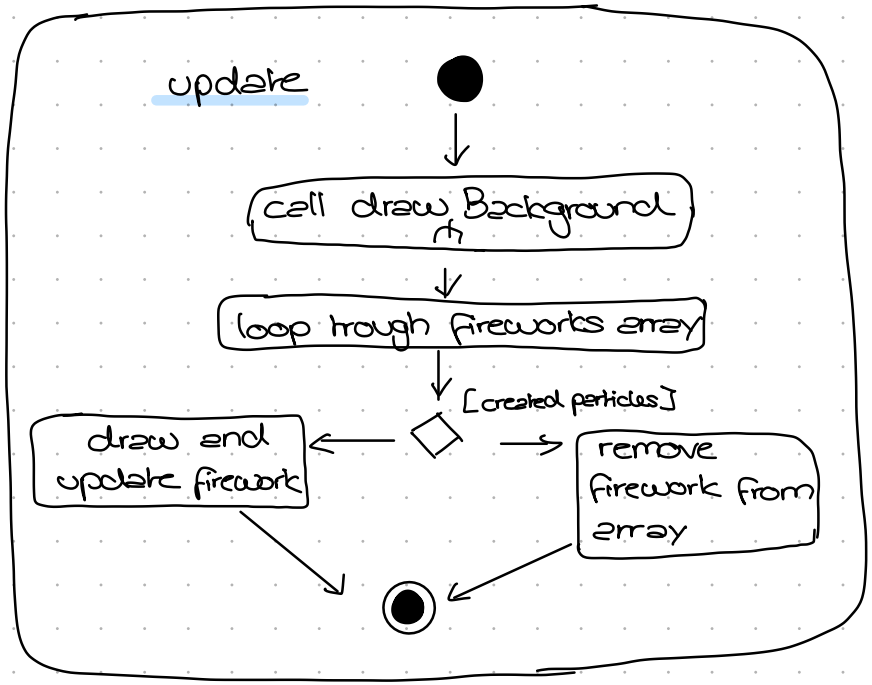


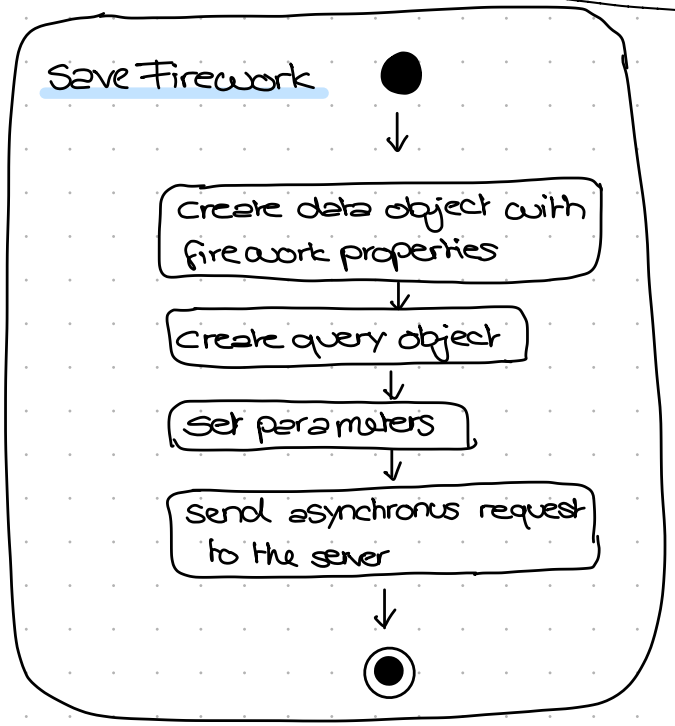

changeSize
read Input
convert input to integer
assign value to size
handleClick
e : mouseEvent
Calculate Position relative to canvas
create fireworkConfig (with color, number, position, speed)
create particle Config (with color, size, width)
create new Firework instance
Push new Firework object into the fireworks array
handleLoad
Select canvas element
get bounding rectangle
[No Canvas]
add event listener for canvas click and save button
Initialize drawing context
load saved firework names
start periodic update
changeWidth
read Input
convert input to integer
assign value to width
changeSpeed
read Input
convert input to integer
assign value to speed

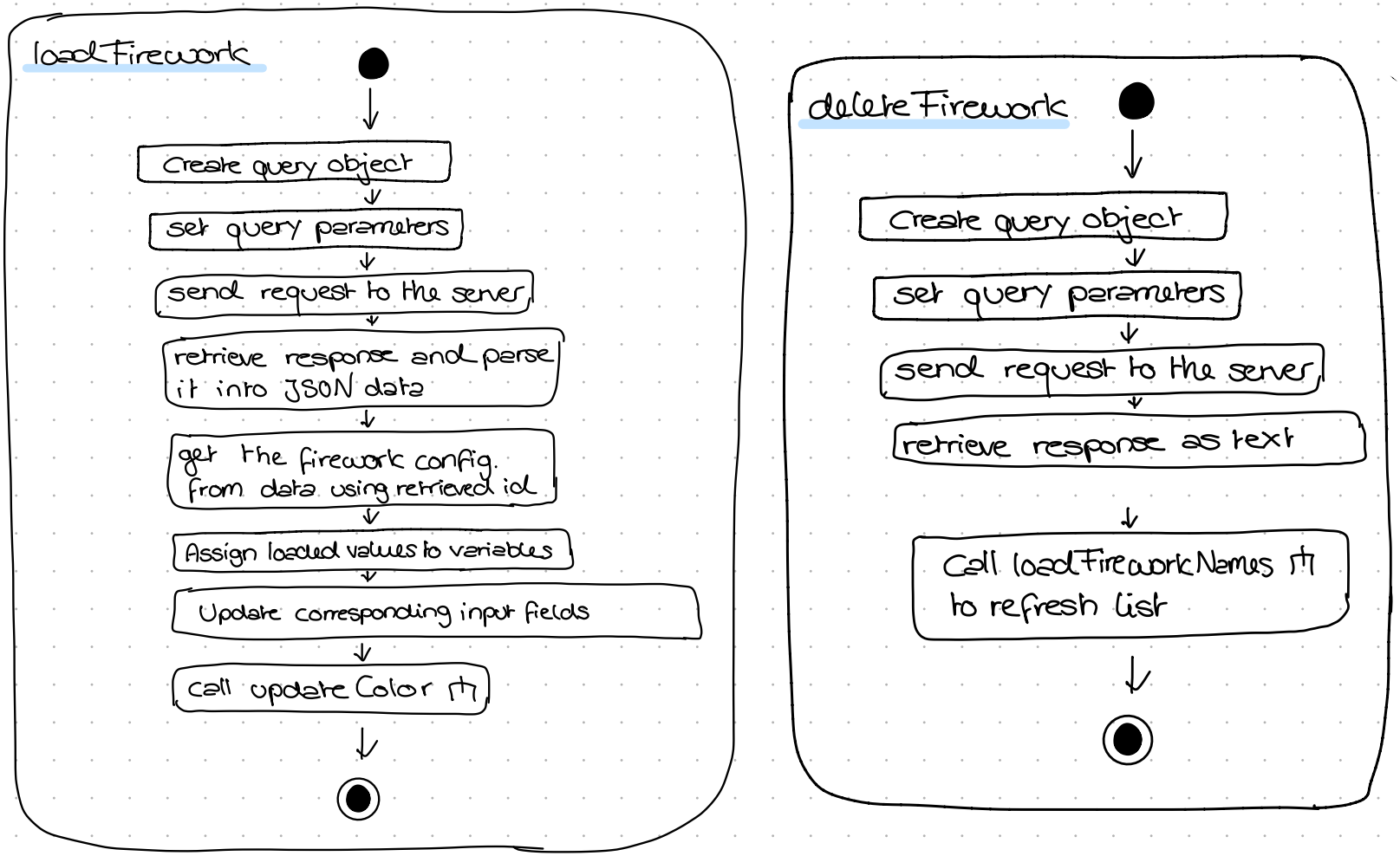

drawBackground
Create a linear gradient
add color stops to the gradient
set gradient as the fill style
fill the entire canvas with the gradi


interface
SaveConfig
name : string;
color : string;
red : number;
green : number;
blue : number;
numberofParticles : number;
speed : number;
size : number;
width: number;


loadFireworkNames
Create query object
set query parameters
send request to the server
retrieve response and parse it into JSON data
get the available element from the DOM
clearing old list
Loop trough each firework in d

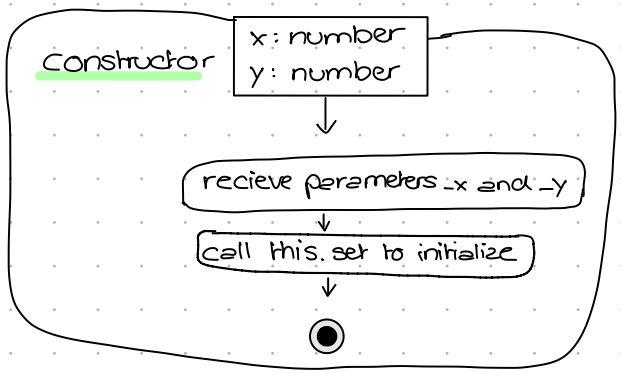

update
call drawBackground
loop trough fireworks array
[created particles]
draw and update firework
remove firework from array

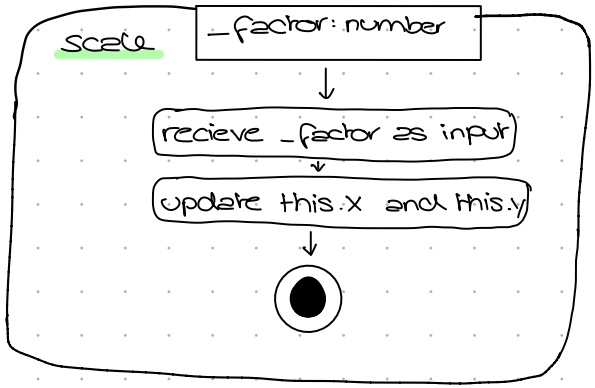

SaveFirework
Create data object with firework properties
Create query object
Set parameters
Send asynchronus request to the server

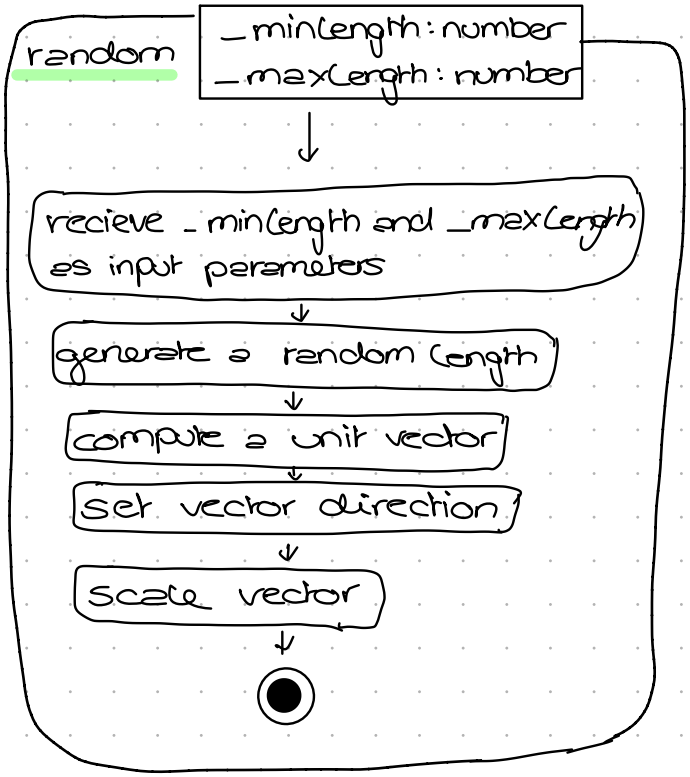

loadFirework
Create query object
set query parameters
send request to the server
retrieve response and perse it into JSON data
get the firework config. from data using retrieved id
Assign loaded values to variables
Update corresponding input fields
call updateColor
deleteFirework
Create query object
set query parameters
send request to the server
retrieve response as text
Call loadFireworkNames to refresh list

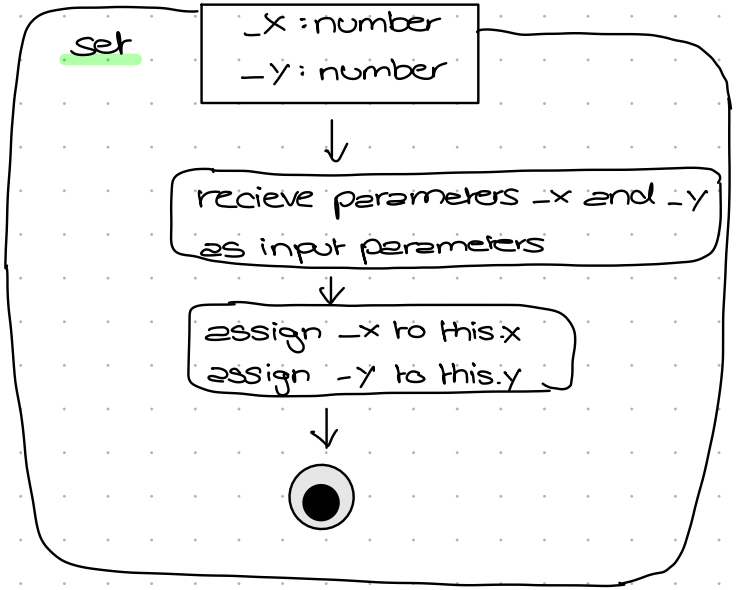
Activity Diagram

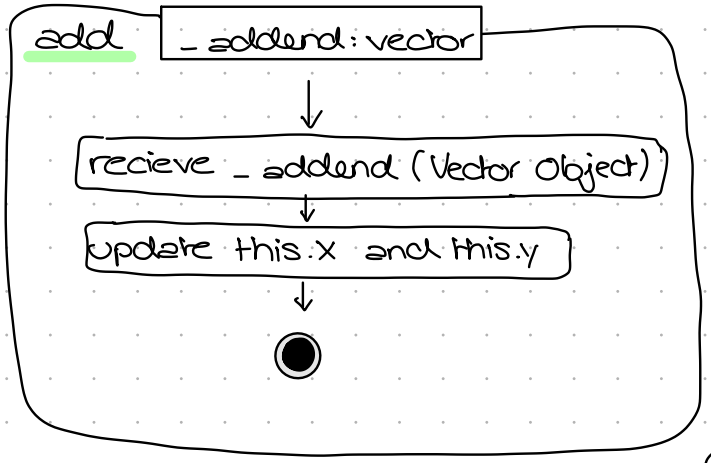
Vector

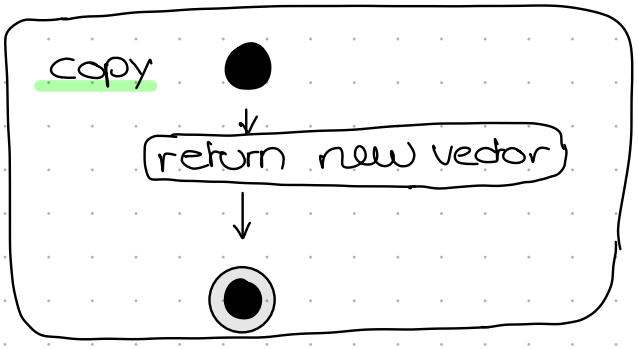






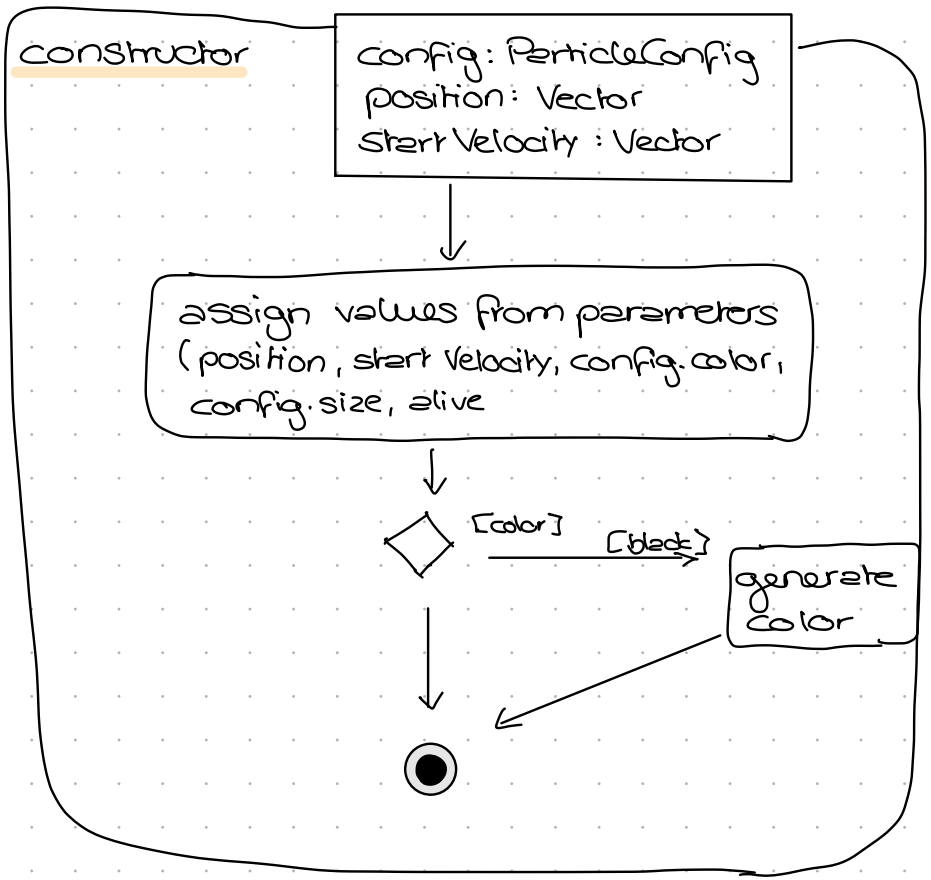


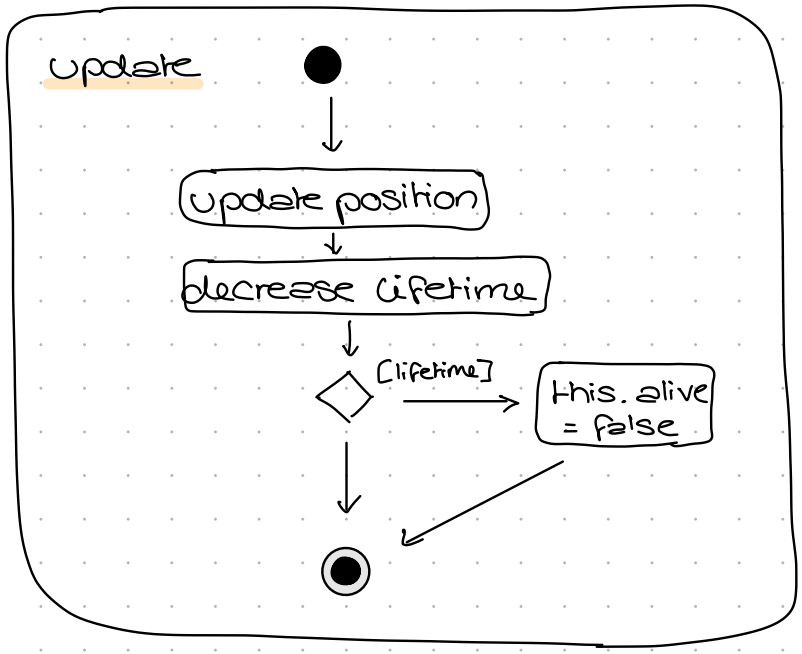


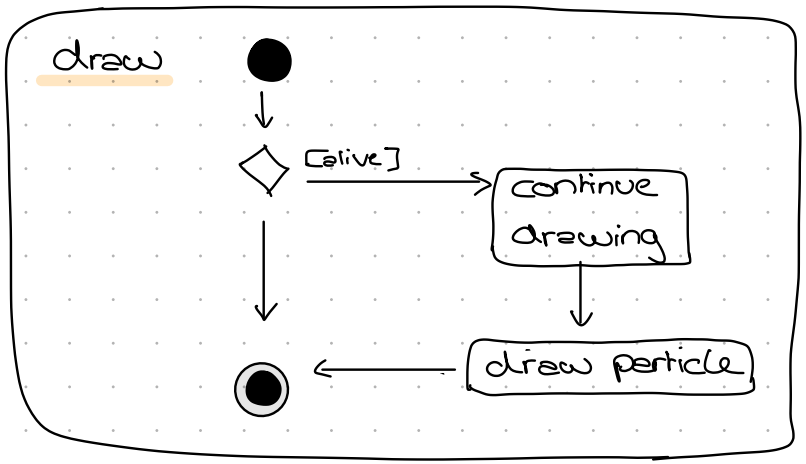


Activity Diagram

Particle







Activity Diagram

Firework