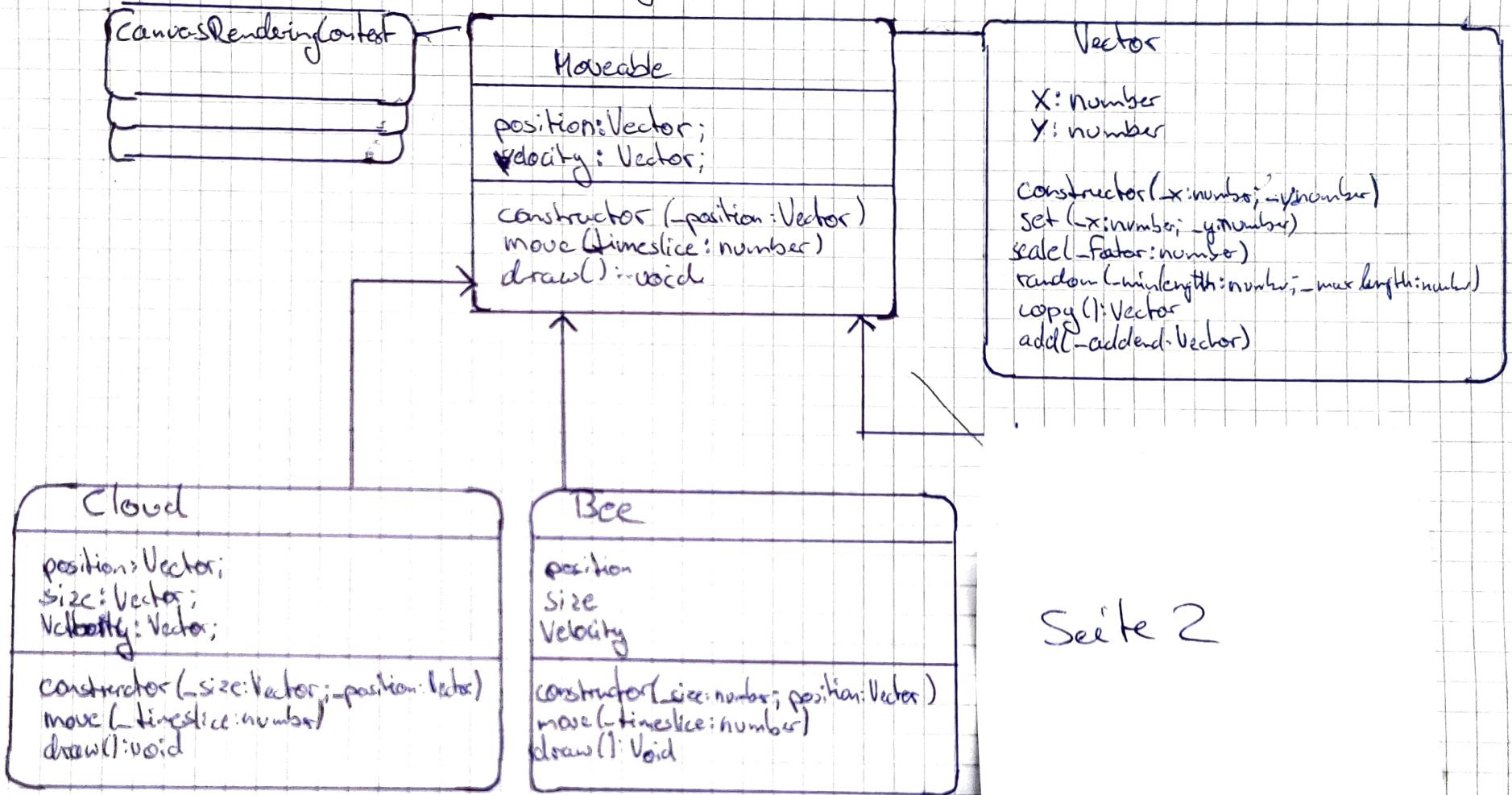
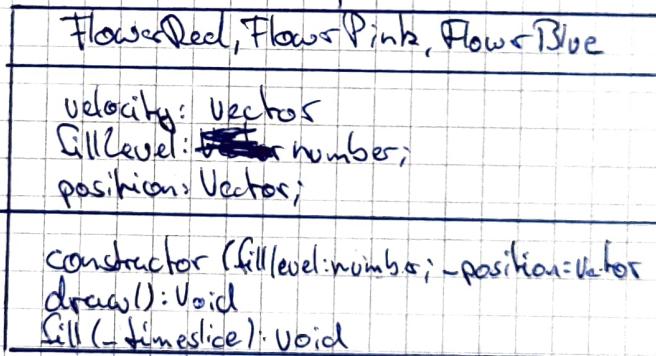
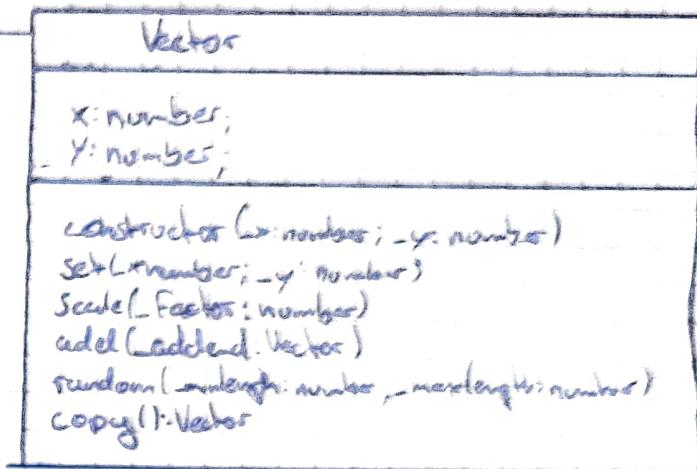
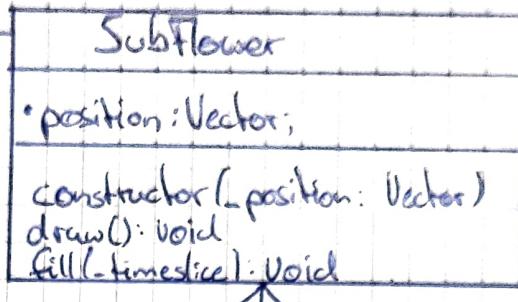
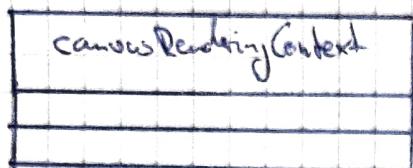


Blumenwiese: Class Diagramm

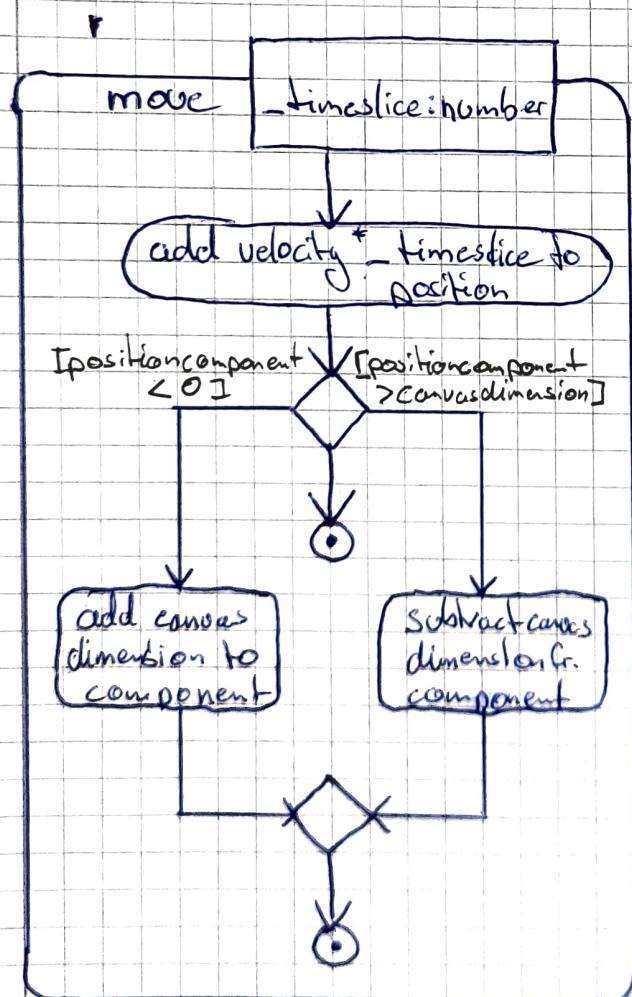
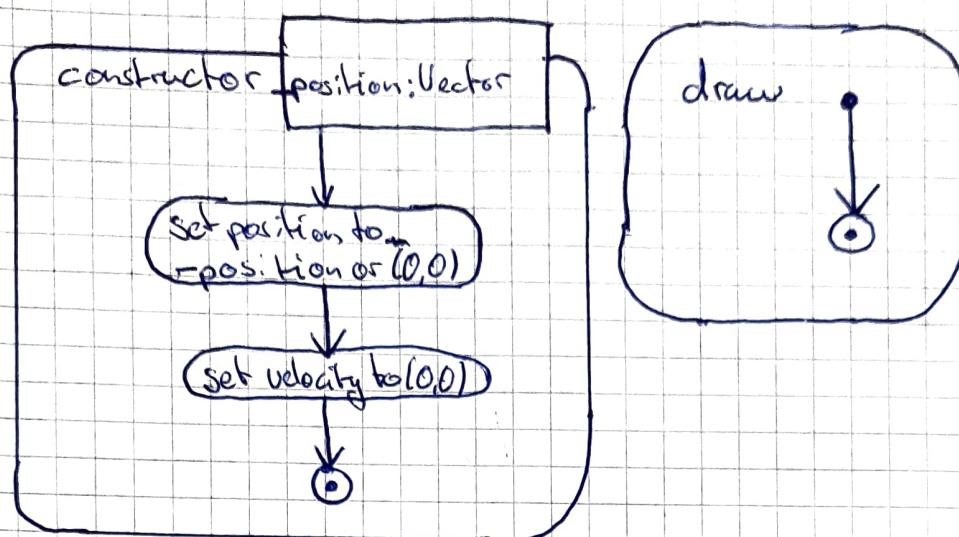


Seite 2

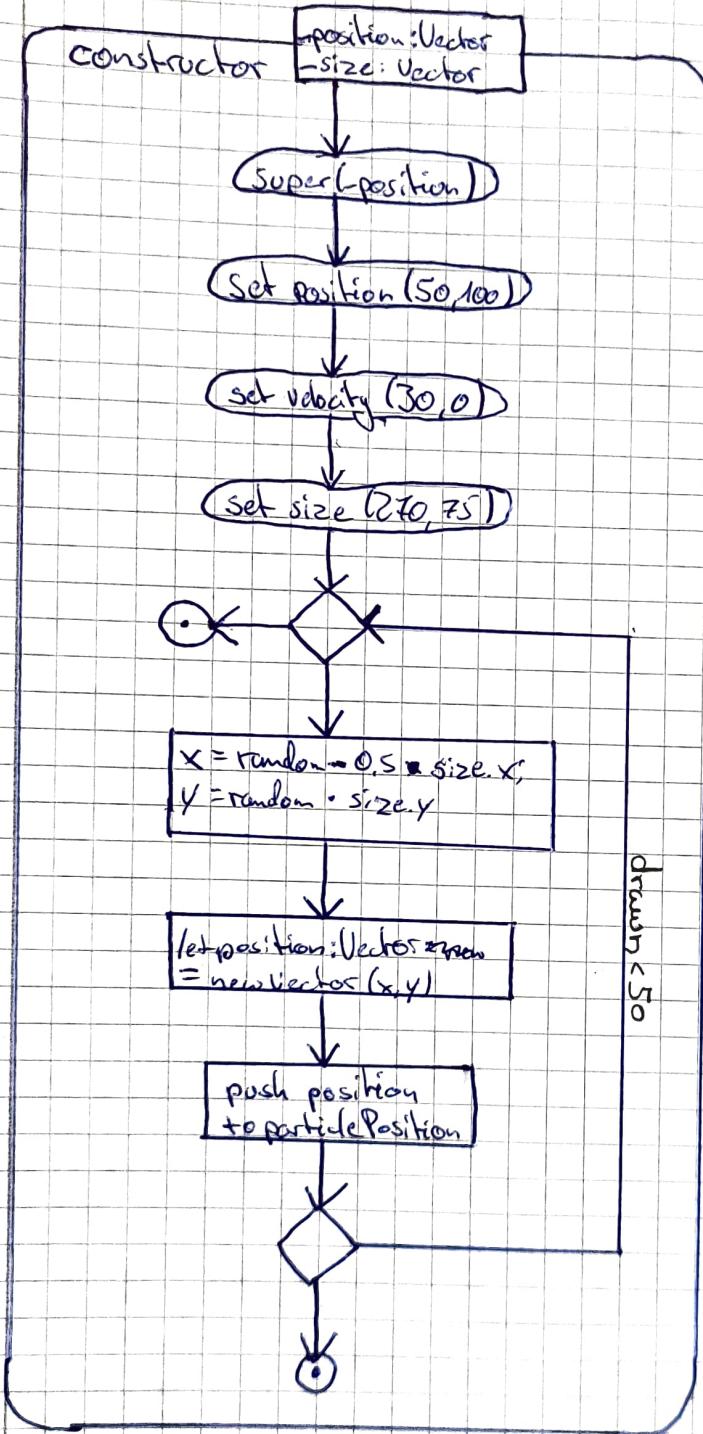
Class Diagram



Blumenwiese: AD - Movable



2D Cloud



draw

```
radiusParticle: number = 50,  
particle: Path2D = new Path2D();  
gradient: canvasGradient  
= ccc2.createRadialGradient  
(0,0,0,0,0, radiusParticle)
```

Create arc on particle

Add colour stop to gradient

Save transform

translate position
to x and y

Fill style w/gradient

Save transform

Restore transform

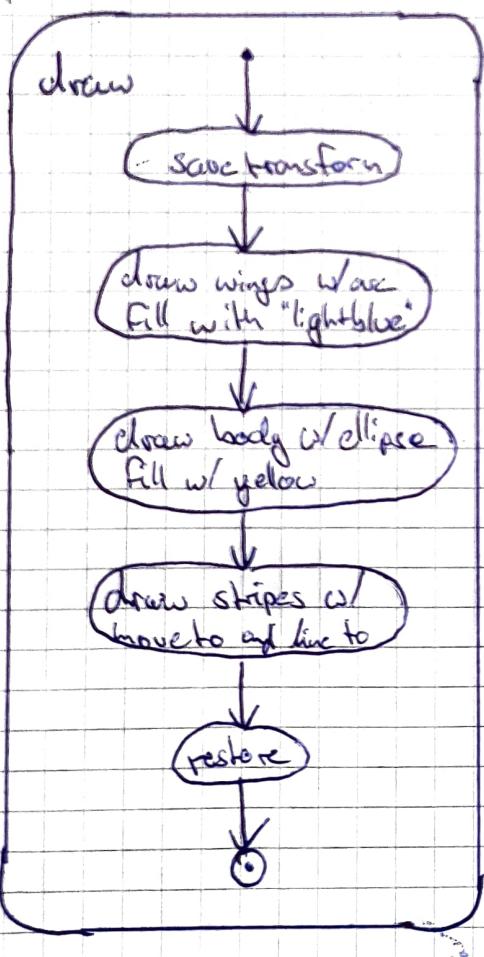
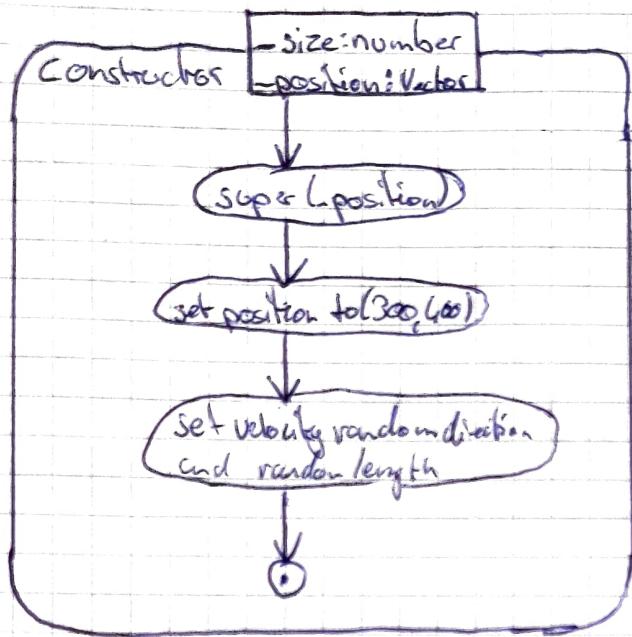
translate to
drawn.x and
drawn.y

draw particle

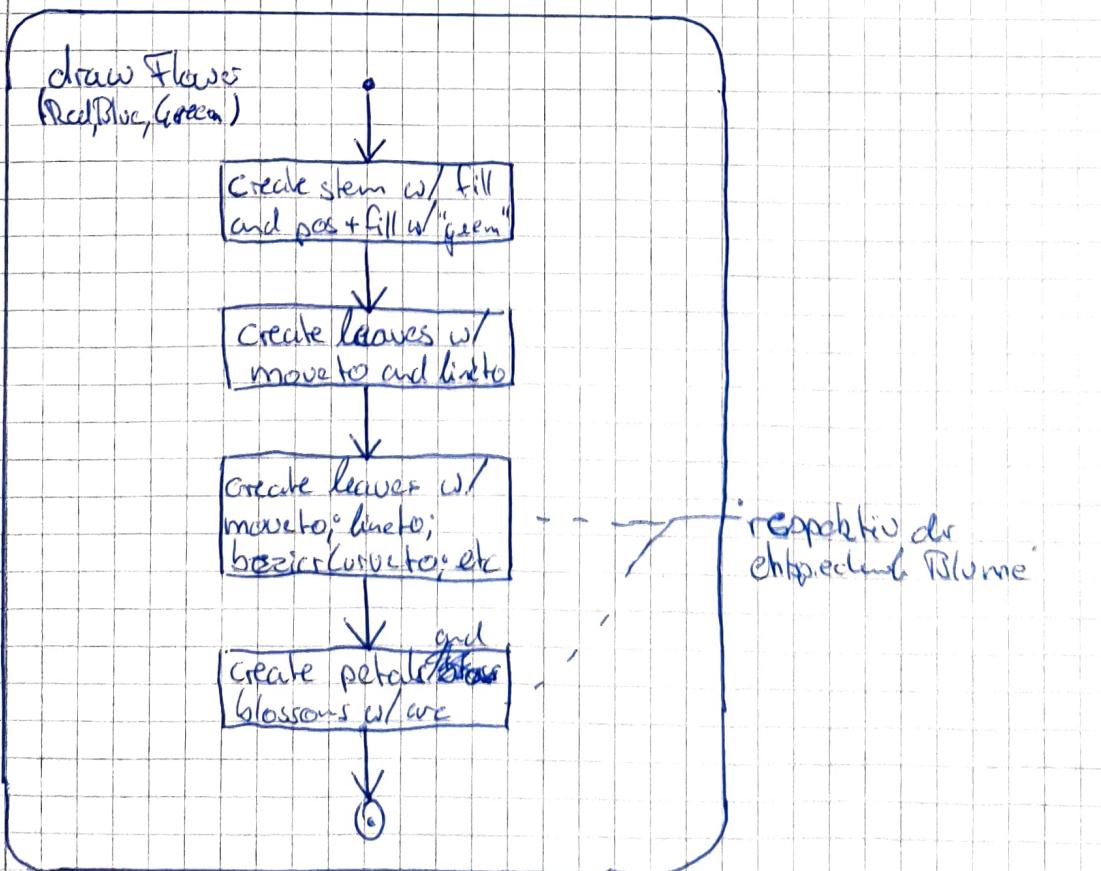
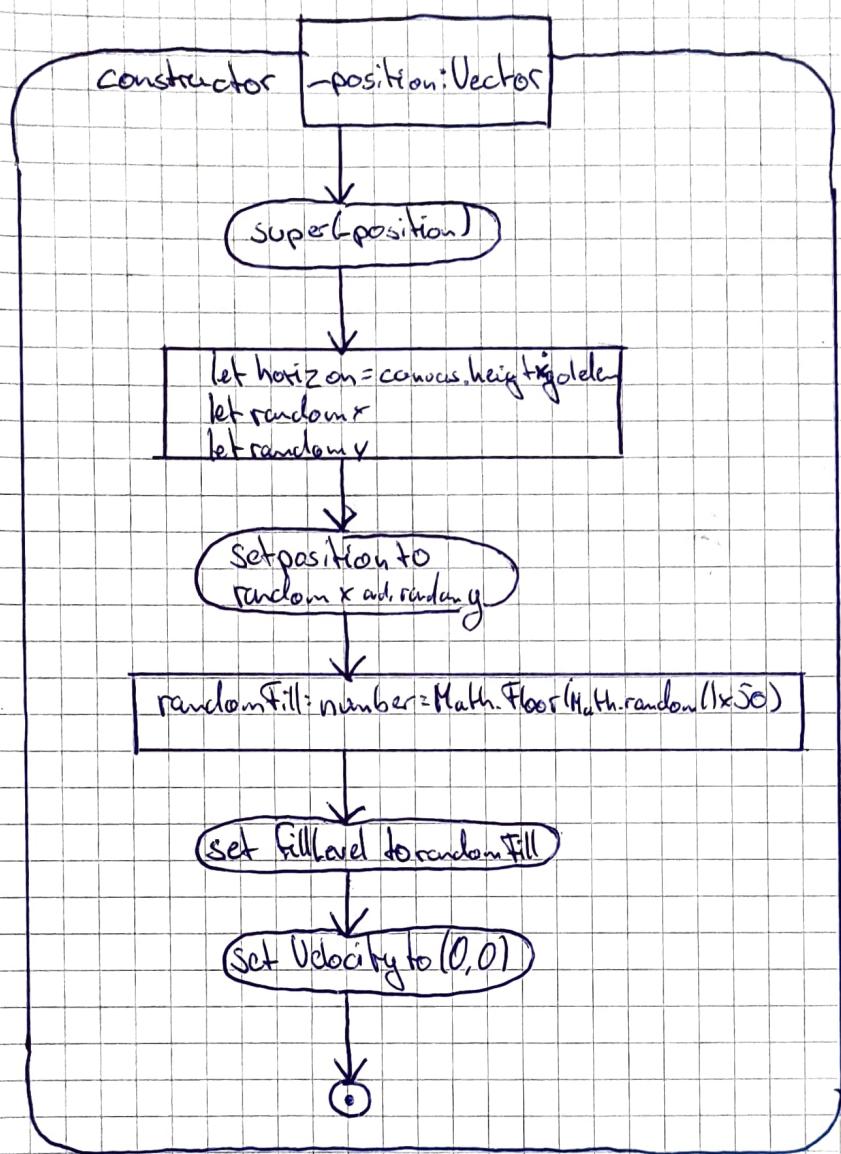
Restore transform

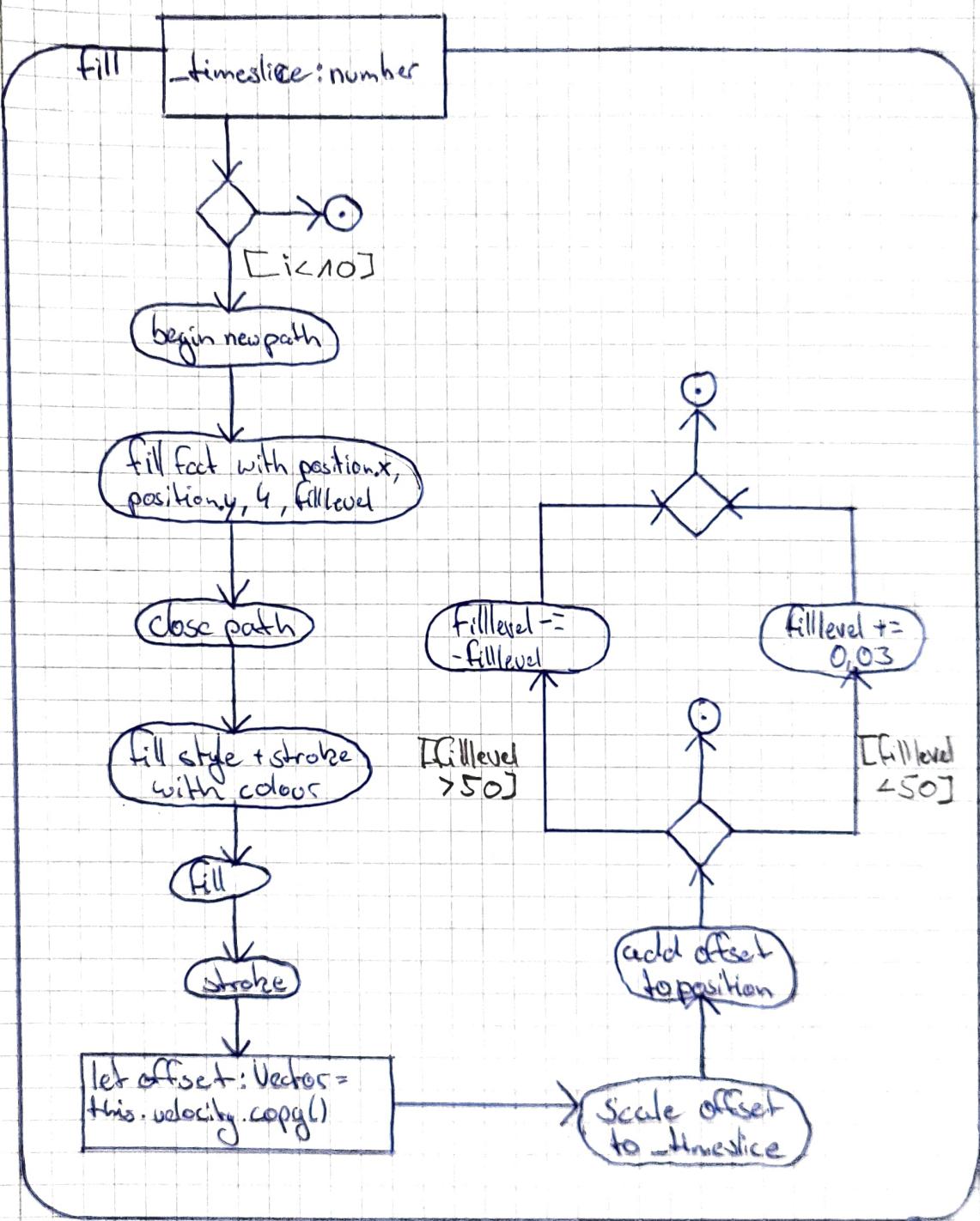
[drawn.x
and
drawn.y]

AD Bee



4D Flowers



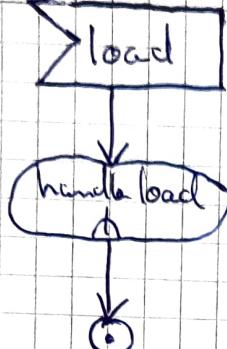


IG

```

let moveable = Moveables[] = [];
export let imageData: ImageData;
export let crc2: canvasRenderingContext2D;
export let goldenNumber = 0.5;
  
```

install load listener



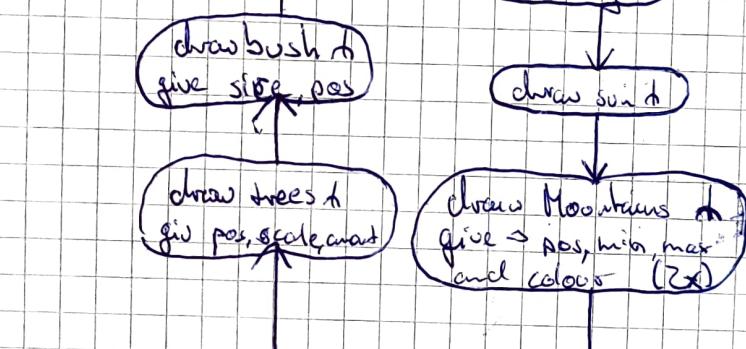
```

imageData = crc2.getImageData(0, 0, crc2.canvas.width, crc2.canvas.height)
  
```

create flower()

create clouds()

create bed()



Wiese

event: Event

handle load

get canvas from html

get rendering context

```

let horizon: number = crc2.canvas.height * goldenNumber;
let posMountains: Vector = {x: 0, y: horizon};
let posTreeStart: Vector = {x: 28, y: horizon + 5};
let posTreeEnd: Vector = {x: crc2.canvas.width, y: horizon + 5};
let posBush: Vector = {x: 400, y: horizon + 100}
  
```

draw bush

draw sun

draw moon

4D Mountains

drew Mountains

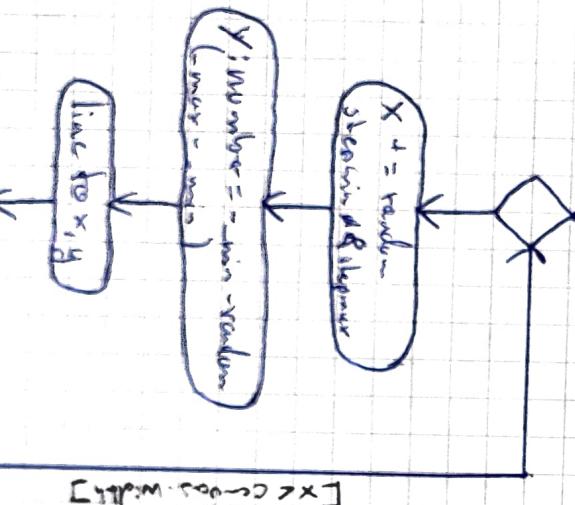
- positionVector, minNumber, maxNumber,
- stepmax & number = 50
- colorLow: string!
- colorHigh: string!

```
stepmin: number = 10
stepmax & number = 50
x: number = 0
```

Save mountain

Random to position

Move to 0

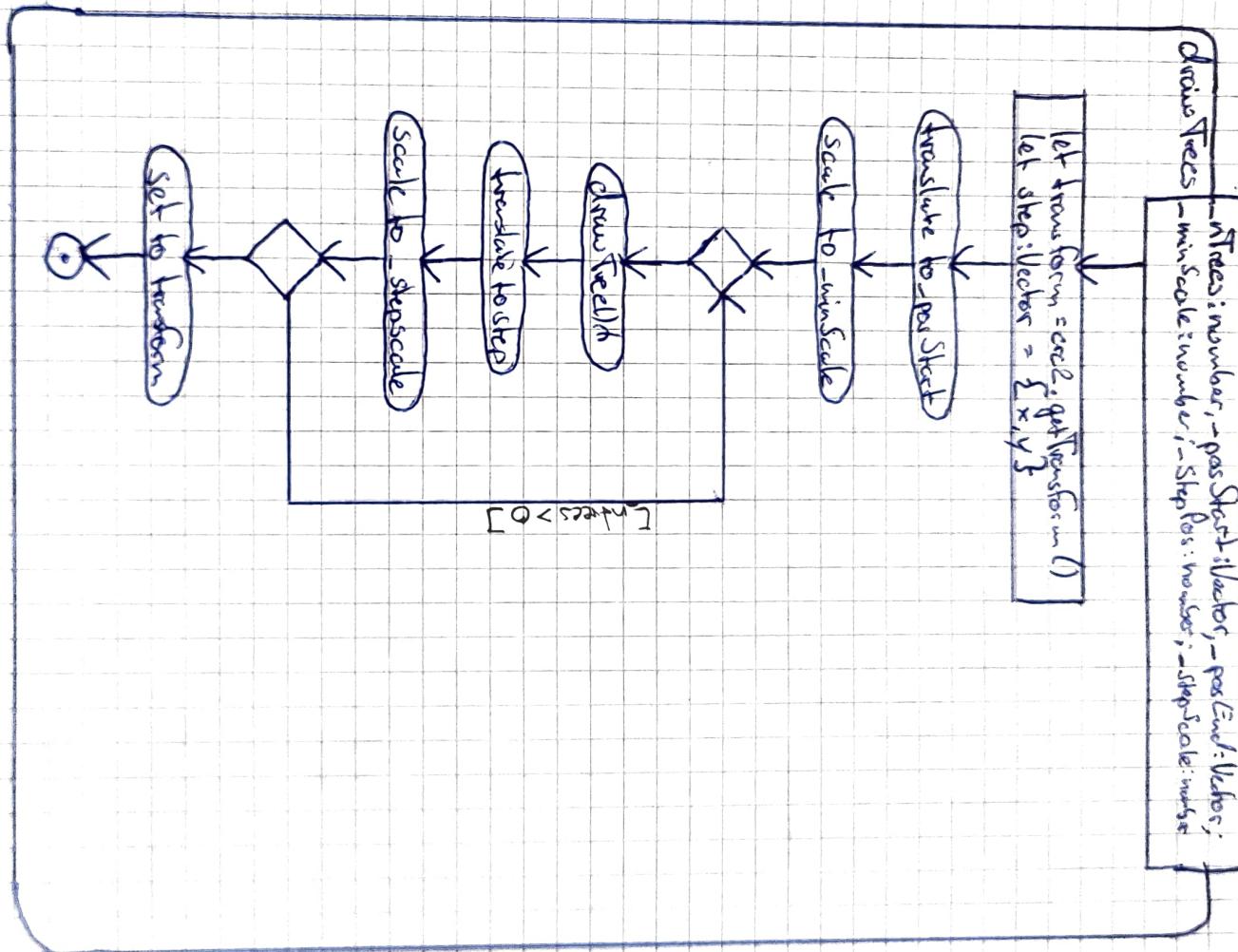


1D Trees

```
drawTrees : number, -> posStart:Vector, -> posEnd:Vector:  
-> minScale:number, -> stepPos:numbers, -> stepScale:number
```

```
let transform = eccl: getTransform()
```

```
let stepVector = {x,y,z}
```



drawtree

```
nBranches: number = 50  
maxRadius: number = 60  
branchData2D: Data2D = new Data2D()
```

Create arc w/ maxRadius

Fill with "brown"

Save transform

Translate to (0, -100)

[nBranches > 0]

```
y: number = random * 350  
size: number = 1 - y / 200  
x: number = (random * 0.5) * 2 * maxRadius
```

Save transform

translate to (0, -y)

Scale to (size, size)

translate to (x, 0)

```
colorAngle: number = 120 - random * 60  
colorString = `HSL("colorAngle, ${colorAngle}, 50%, 40%)`
```

Fill style and branch

Report

Restore

drawBush

```
positionVector  
size:Vector;
```

```
nParticles: number = 20;  
radiusParticle: number = 15;  
particle: Particle = new Particle();  
gradient = createRadialGradient
```

Create arc w/ gradient

Add colour stop x2

Save transform

translate to
-position

fill w/ gradient

Save transform

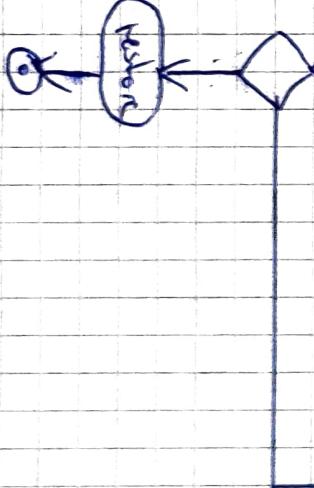
```
X: number = (random - 0.5) * size.x;  
Y: number = (random + size.y);
```

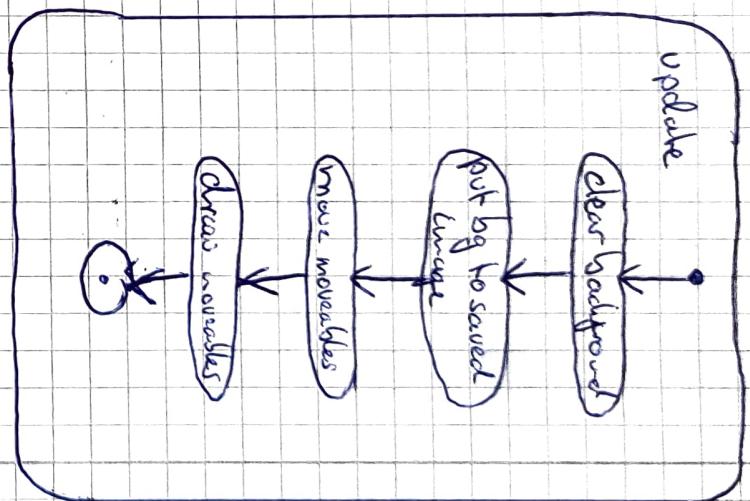
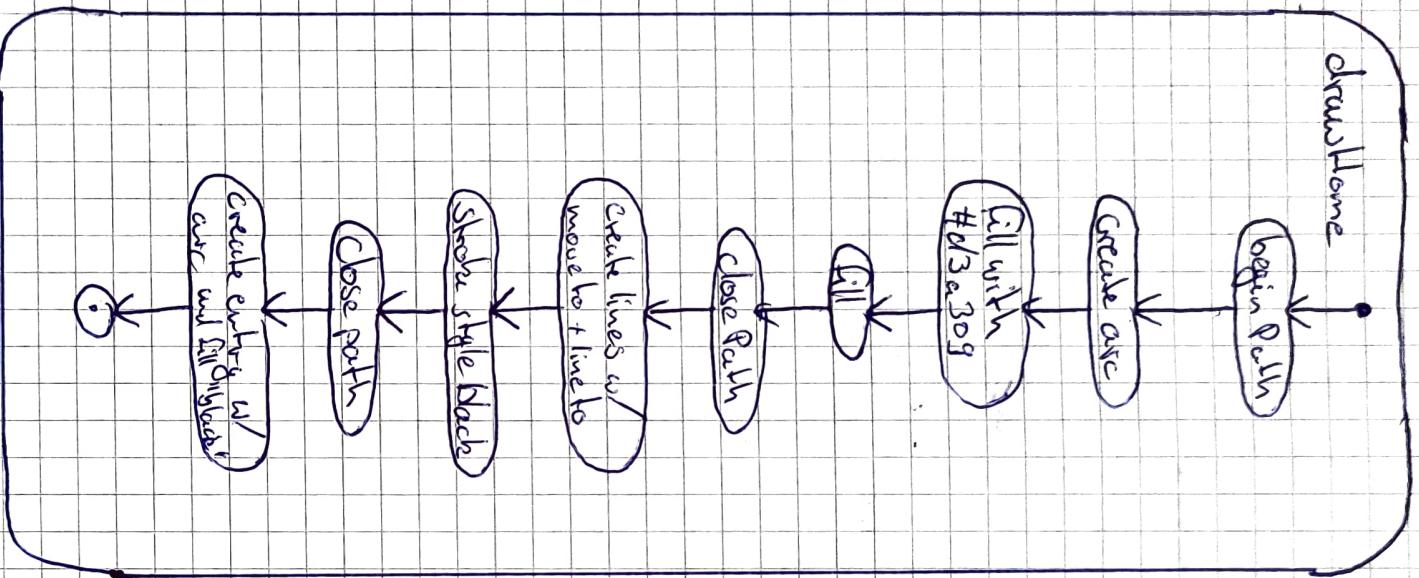
translate x, y

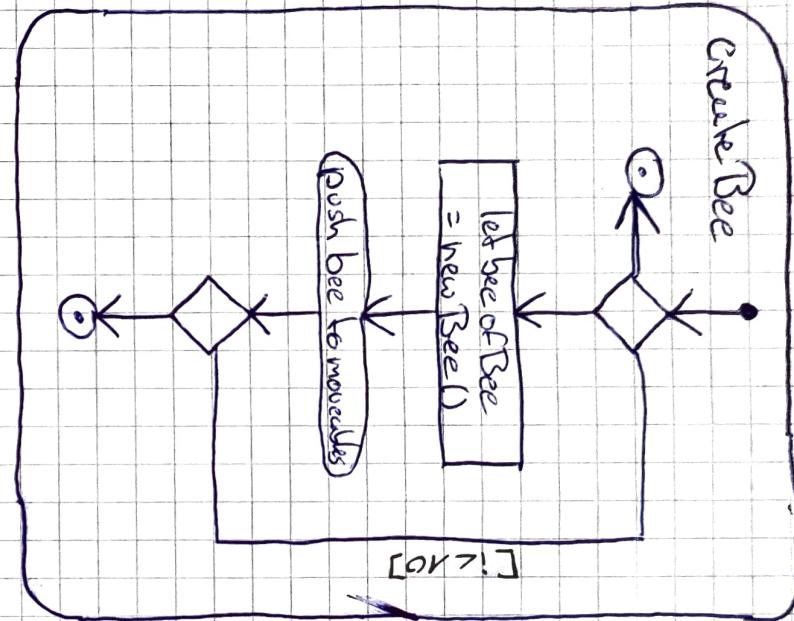
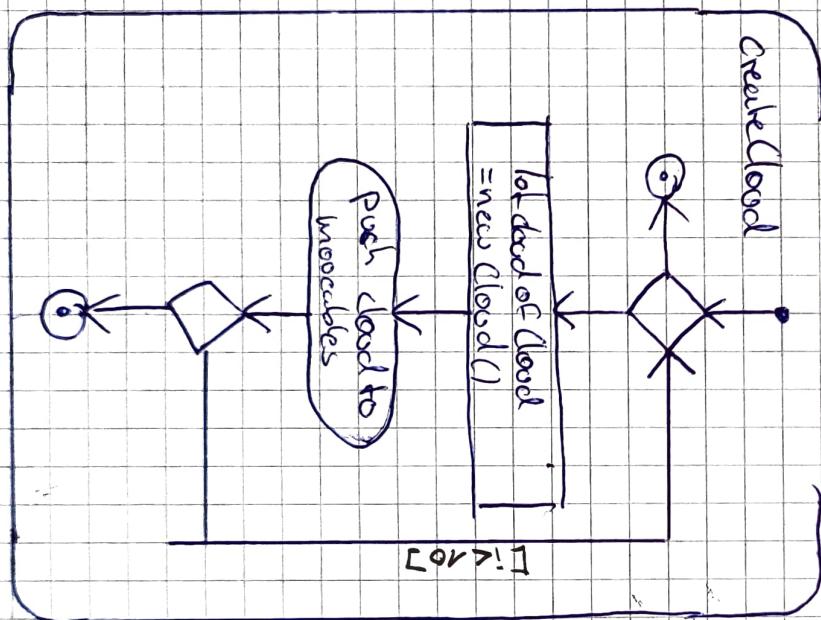
fill w/ particle

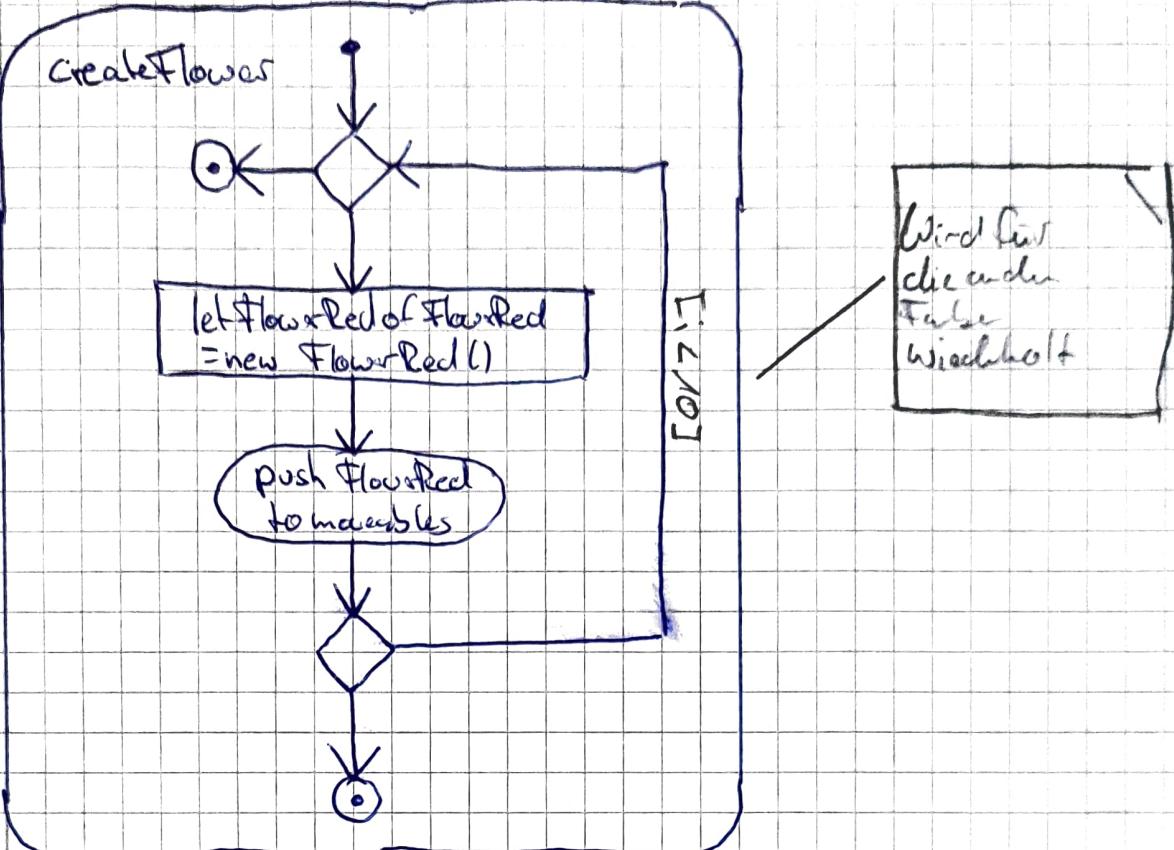
Nothing

[drawBush]









Windkun
die anden
Tabe
Wischholt