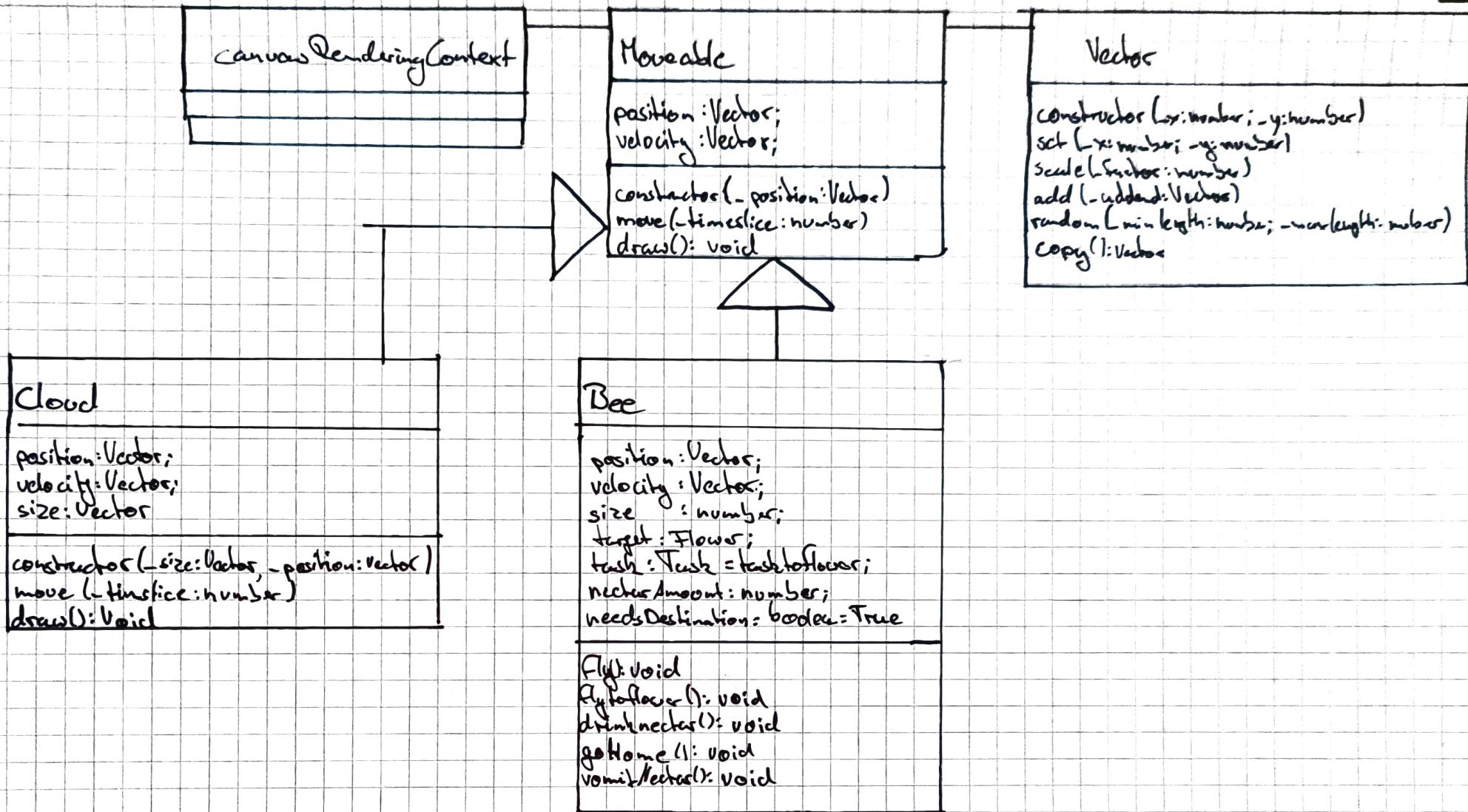
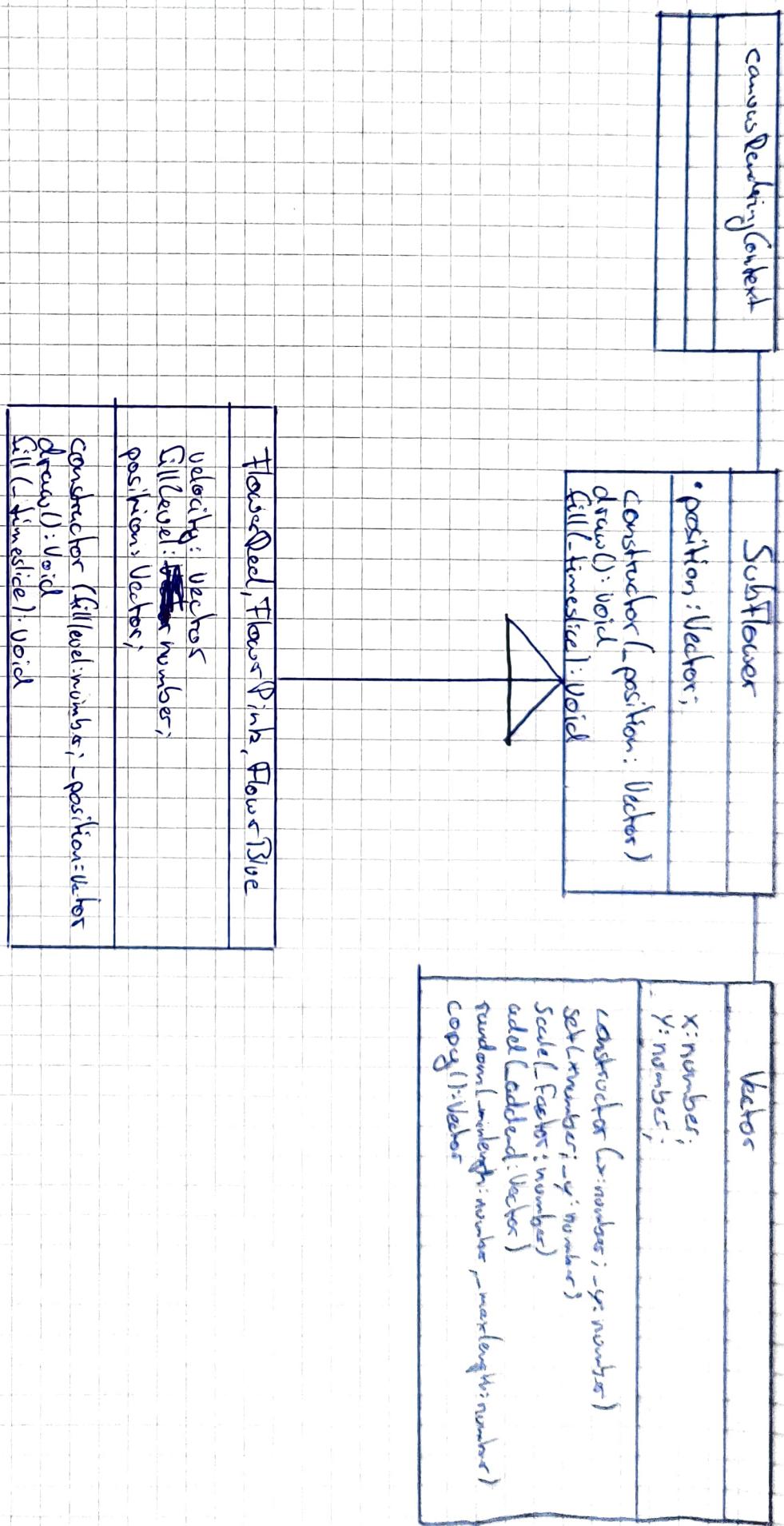


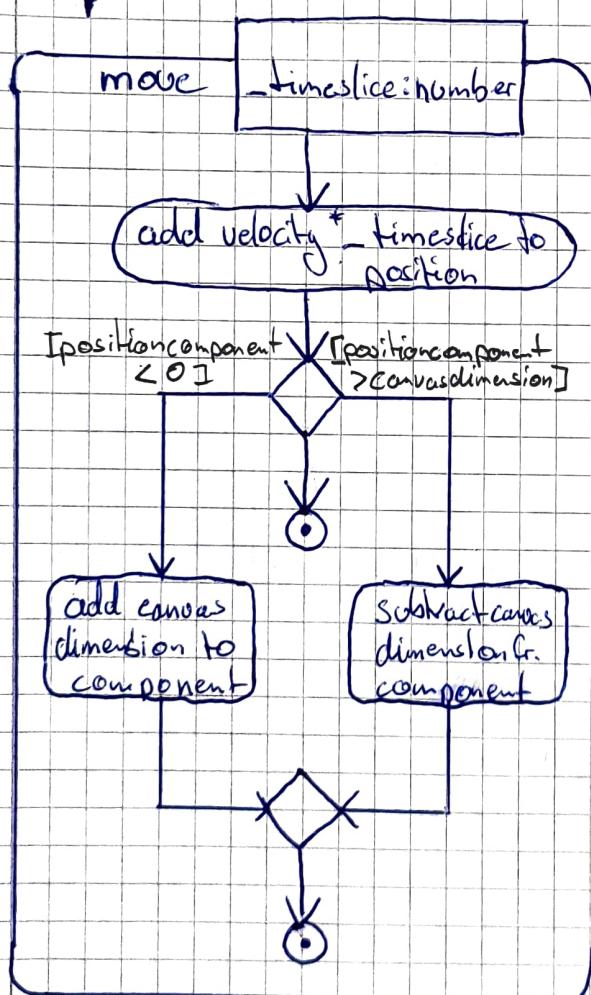
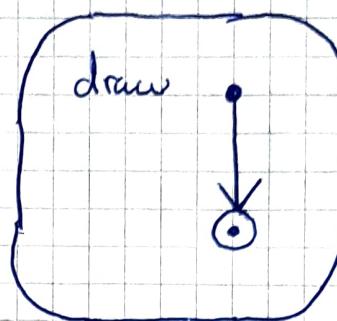
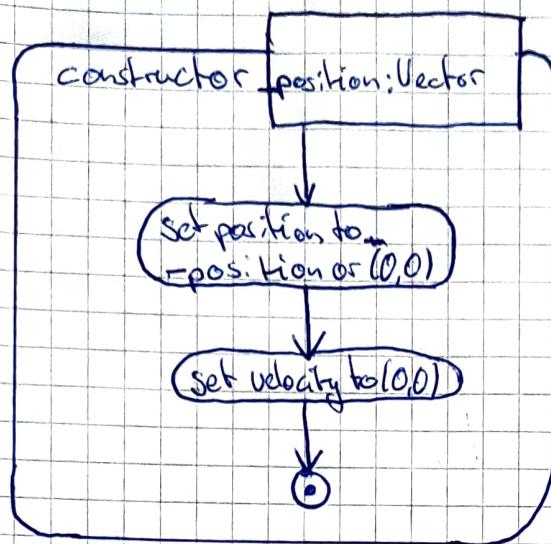
Blumenwiese Class-Diagramm



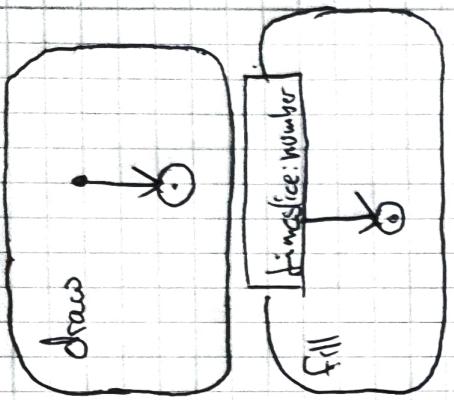
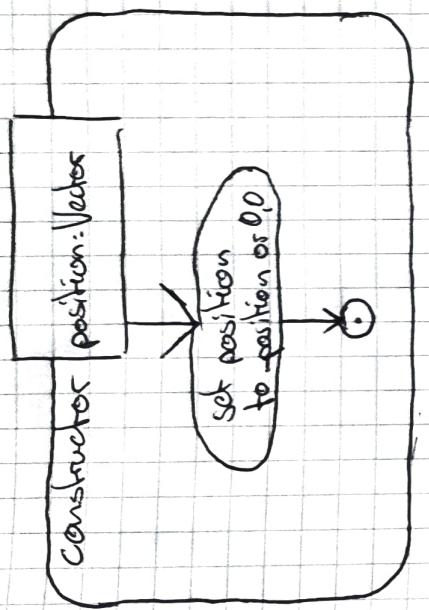
Class Diagram



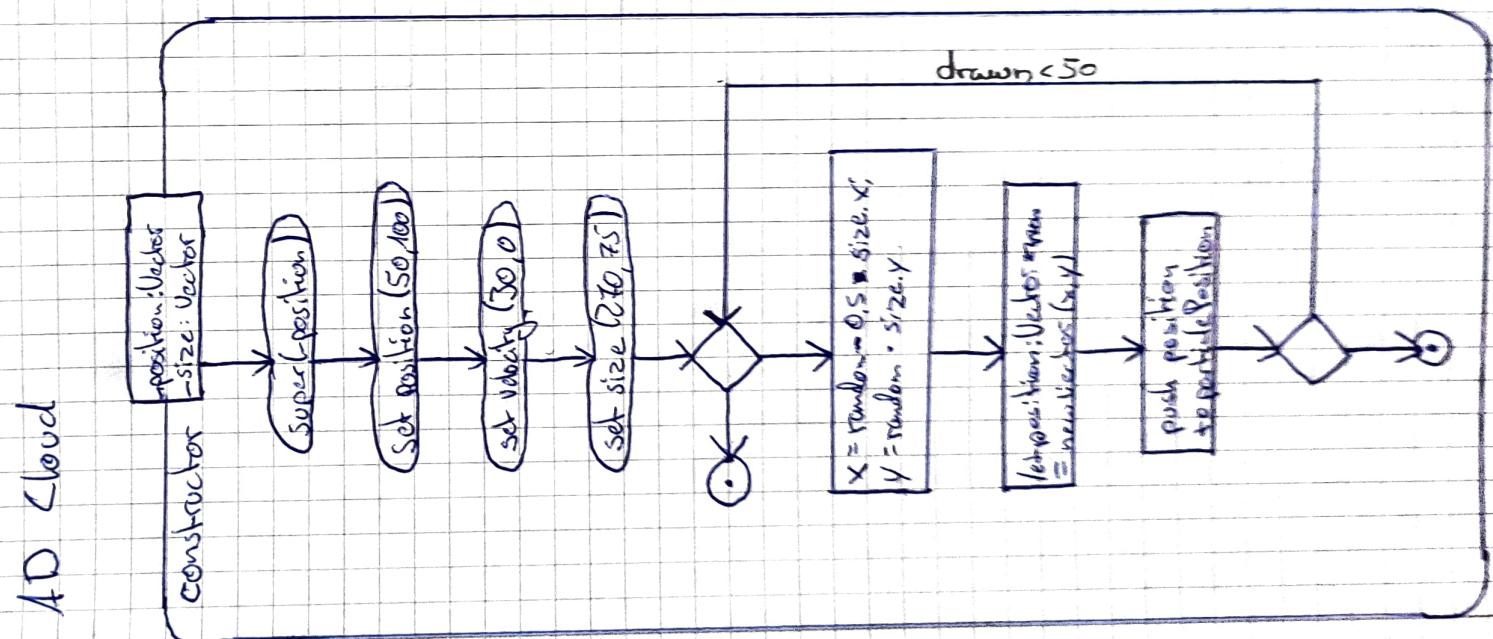
Blumenwiese; AD - Moveable

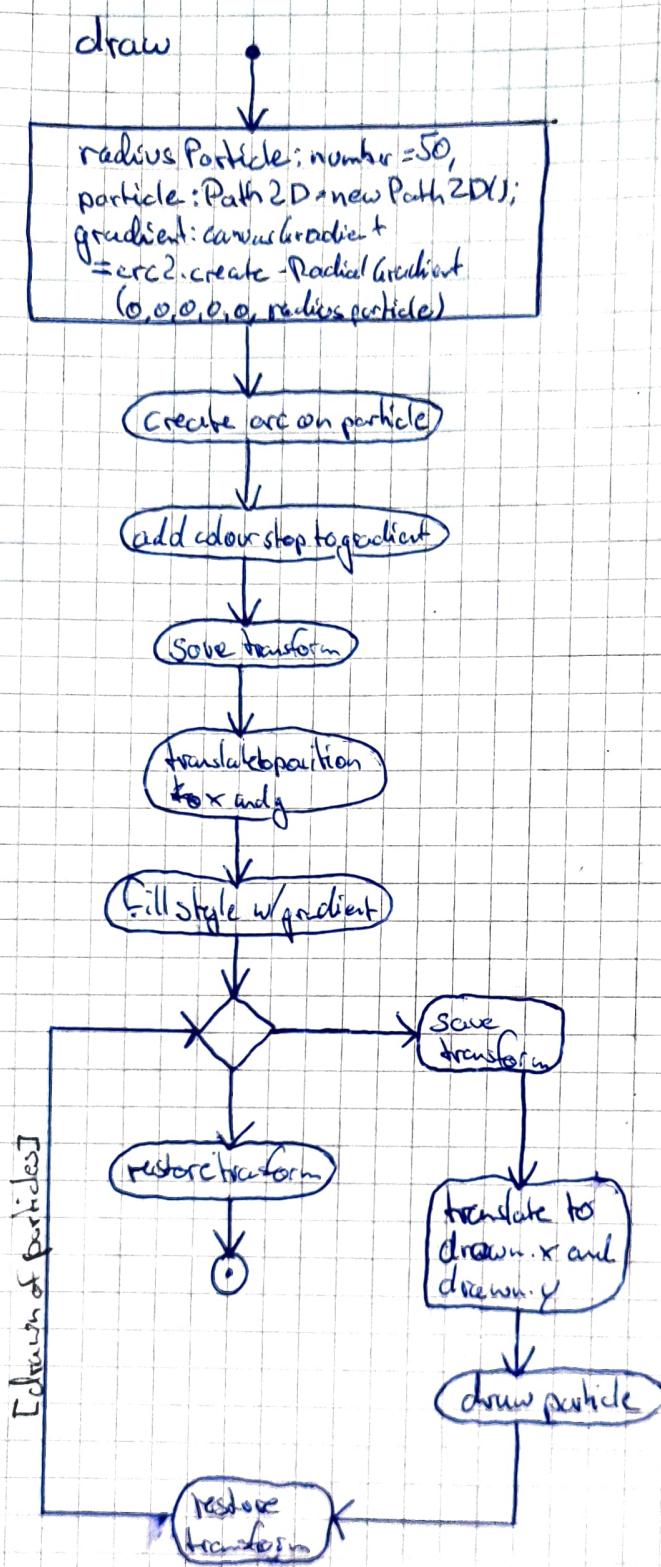


SubFlower

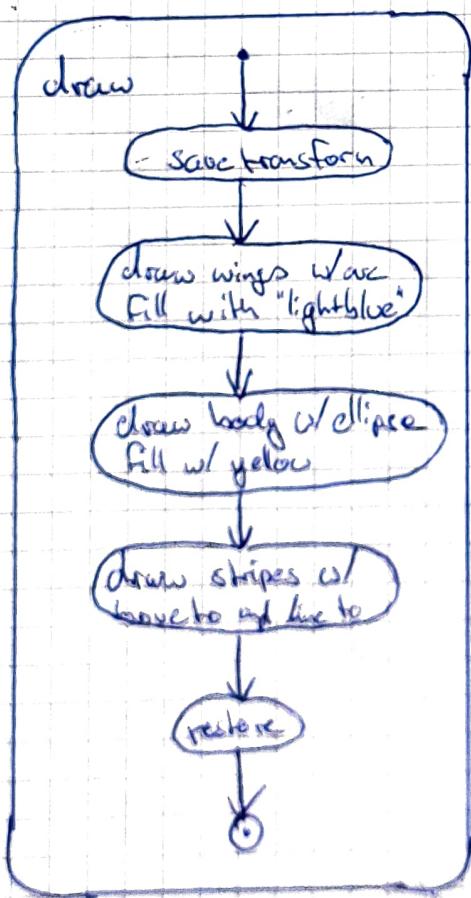
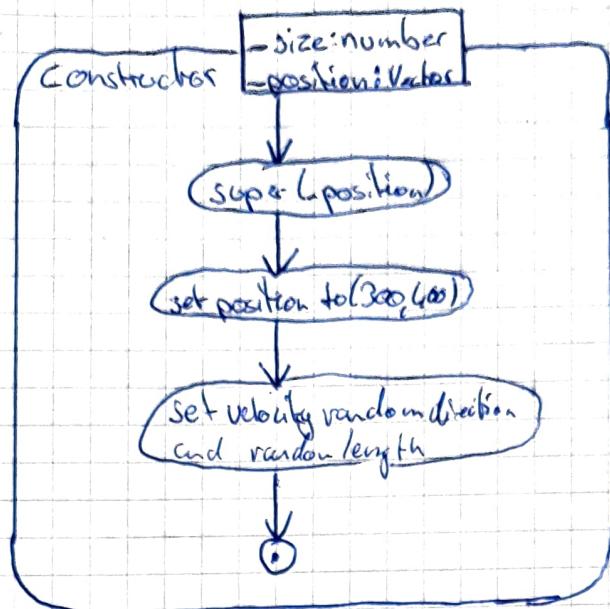


1D Cloud

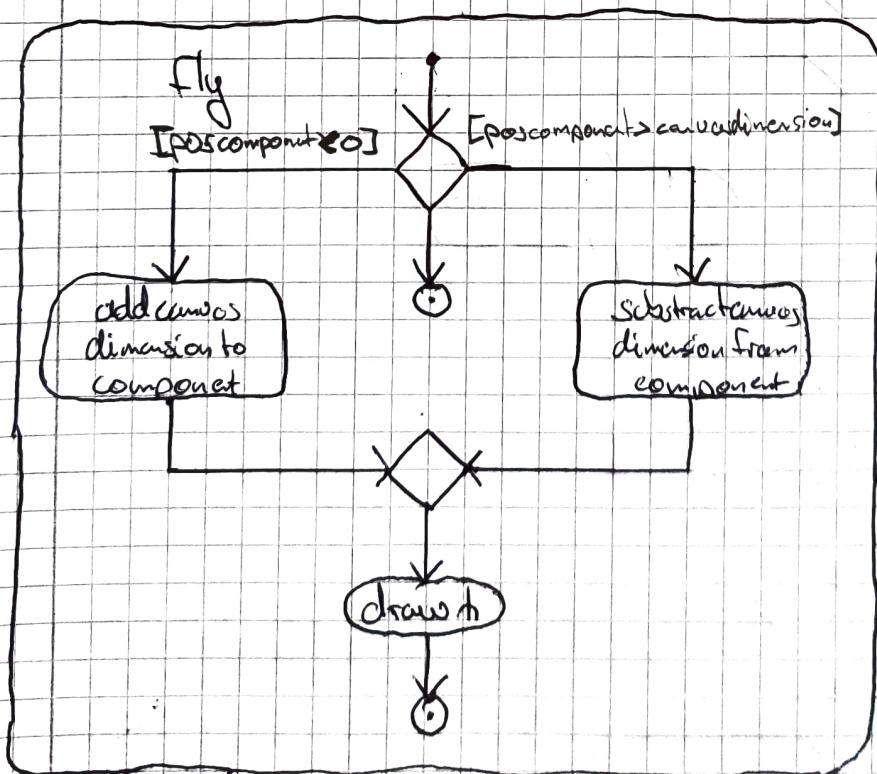
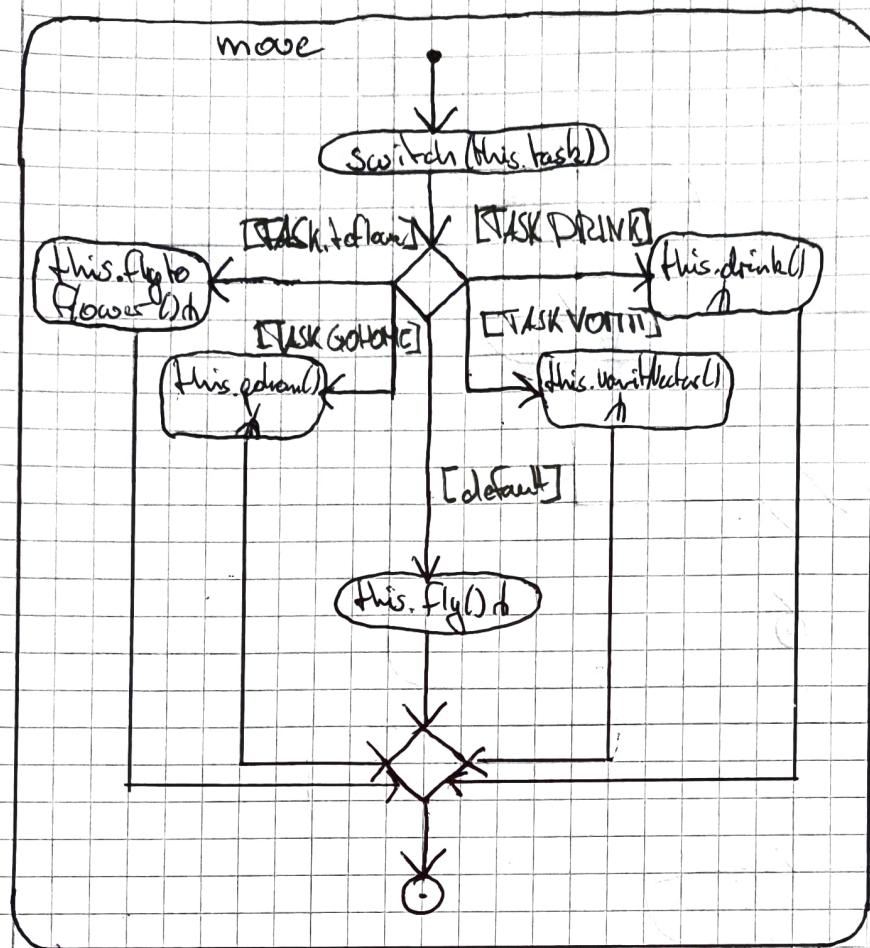


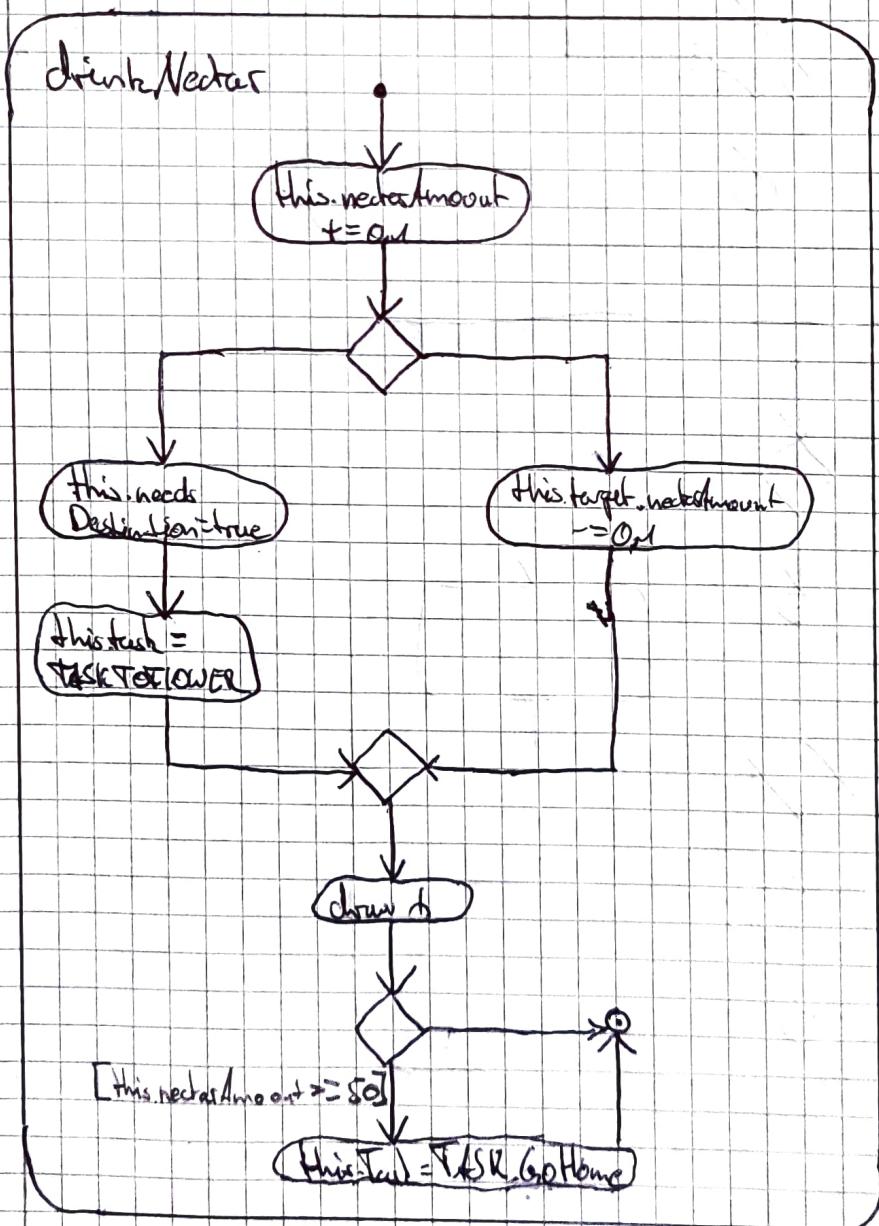
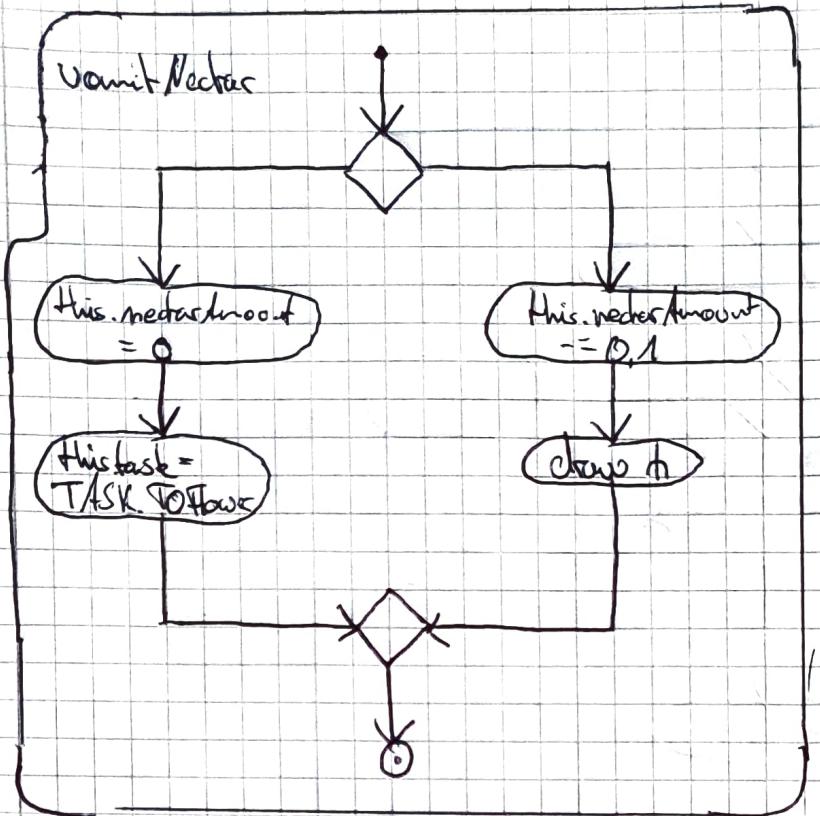


AD Bee

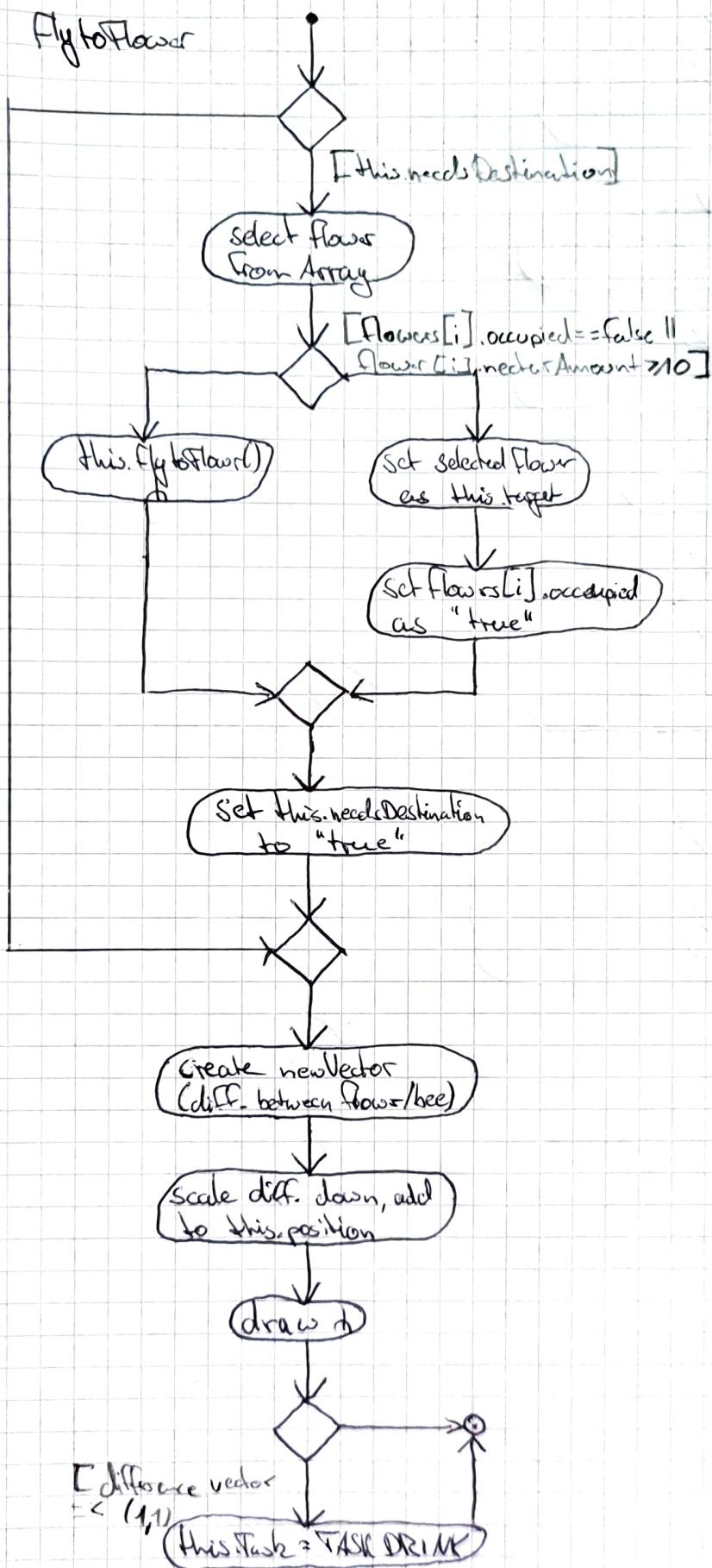


Biene



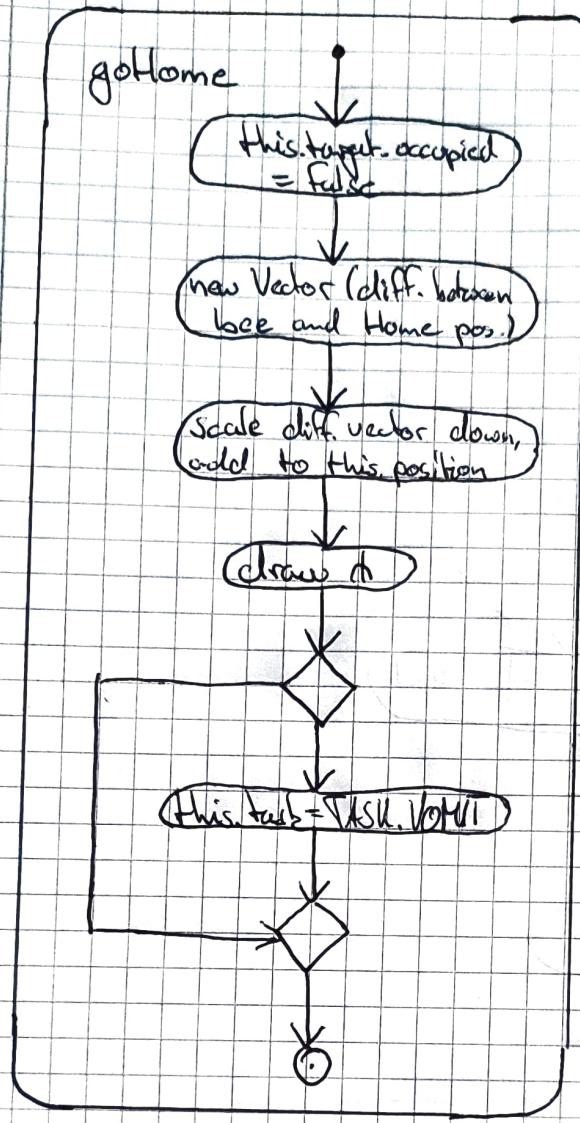


Fly to flower

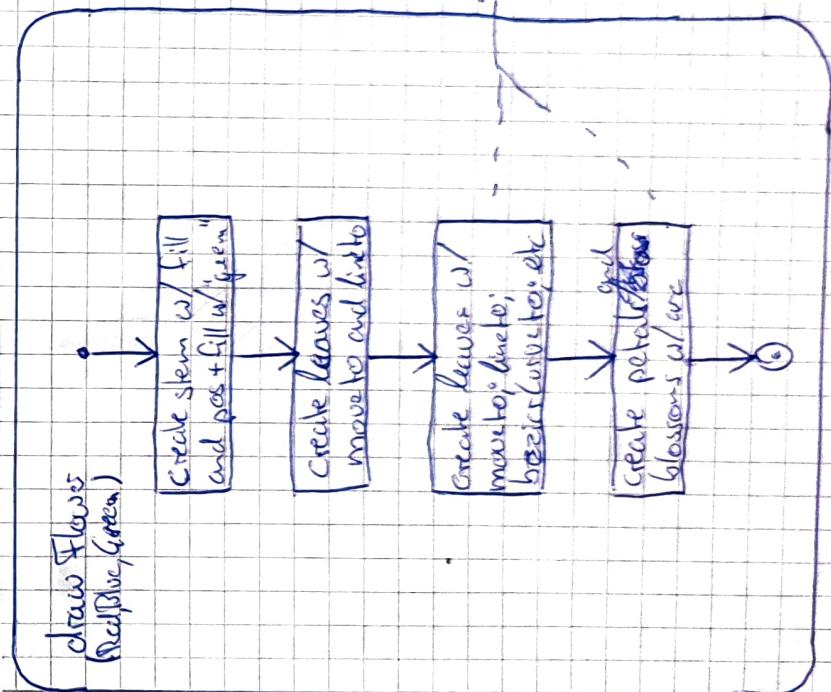
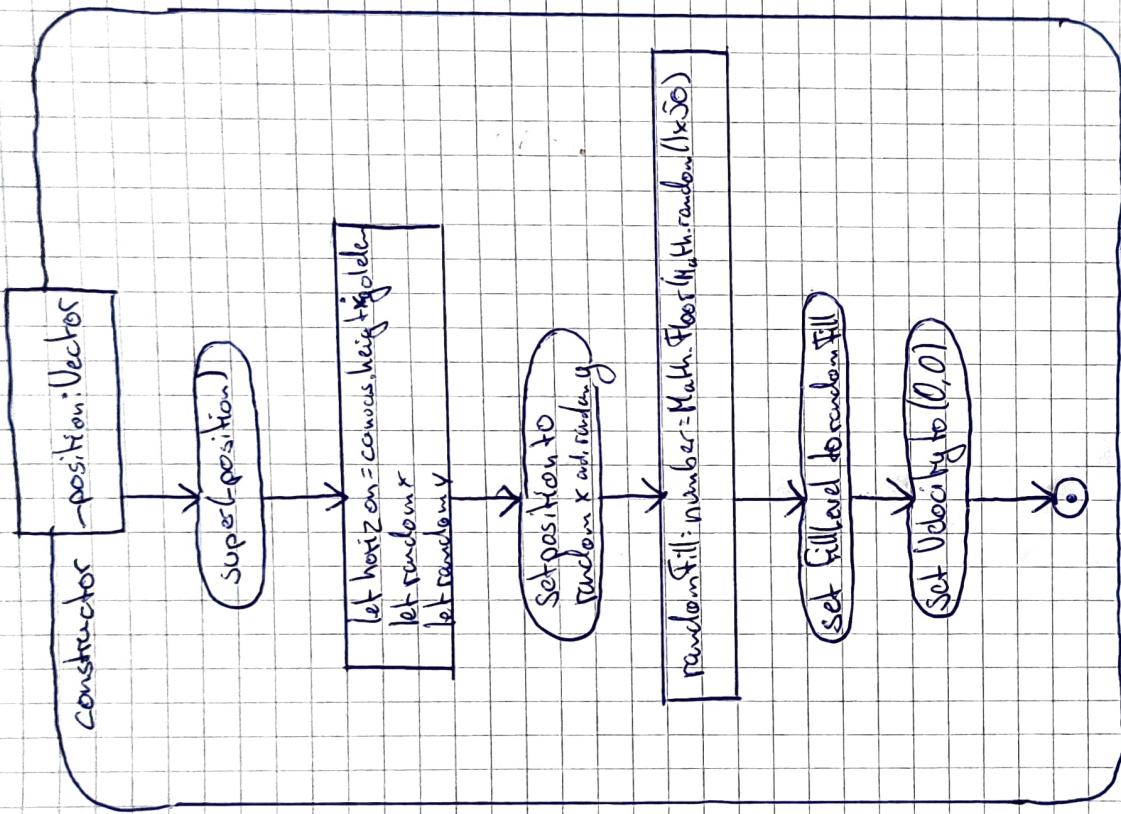


[Difference vector
less than (1,1)]

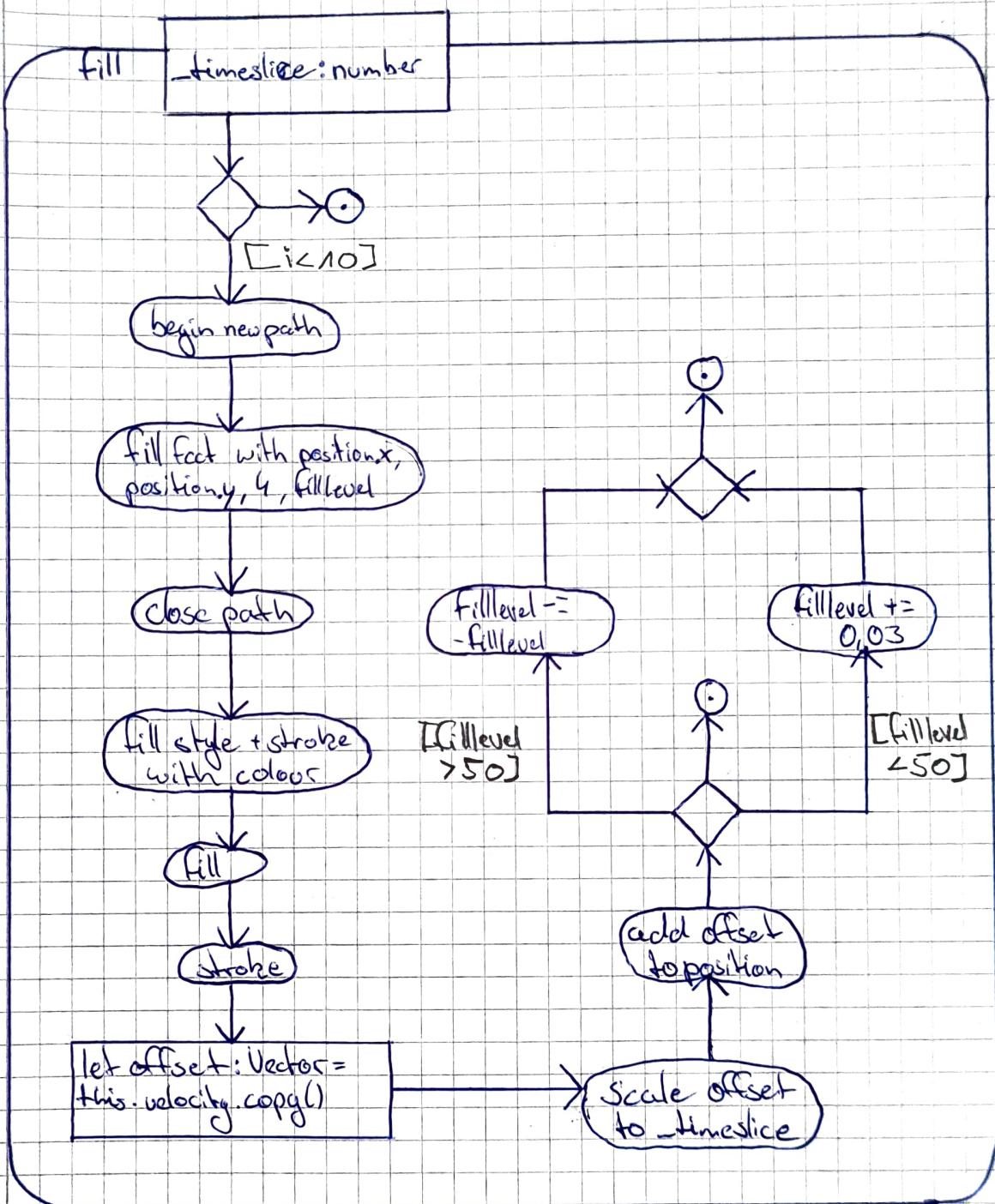
(this.task = TASK DRINK)



4D Flowers



repetitiv der
einfachen Blume



Wiese

IG

```
let moreBees = Morebees();
export let imageData: ImageData;
export let crc2: canvasRenderingContext2D;
export let goldenNumber = 0.5;
```

install local listener

load

handle load

-event: event

handleLoad

get canvas from html

get rendering context

```
let horizon: number = crc2.canvas.height * golden
let posMountains: Vector = {x: 0, y: horizon};
let posTreeStart: Vector = {x: 28, y: horizon + 5};
let posTreeEnd: Vector = {x: crc2.canvas.width / 2, y: horizon + 5};
let posBush: Vector = {x: 400, y: horizon + 100};
```

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

create flower()

create clouds()

create tree()



add click listener
on canvas for
moreBees()

draw bush in
give slope, pos

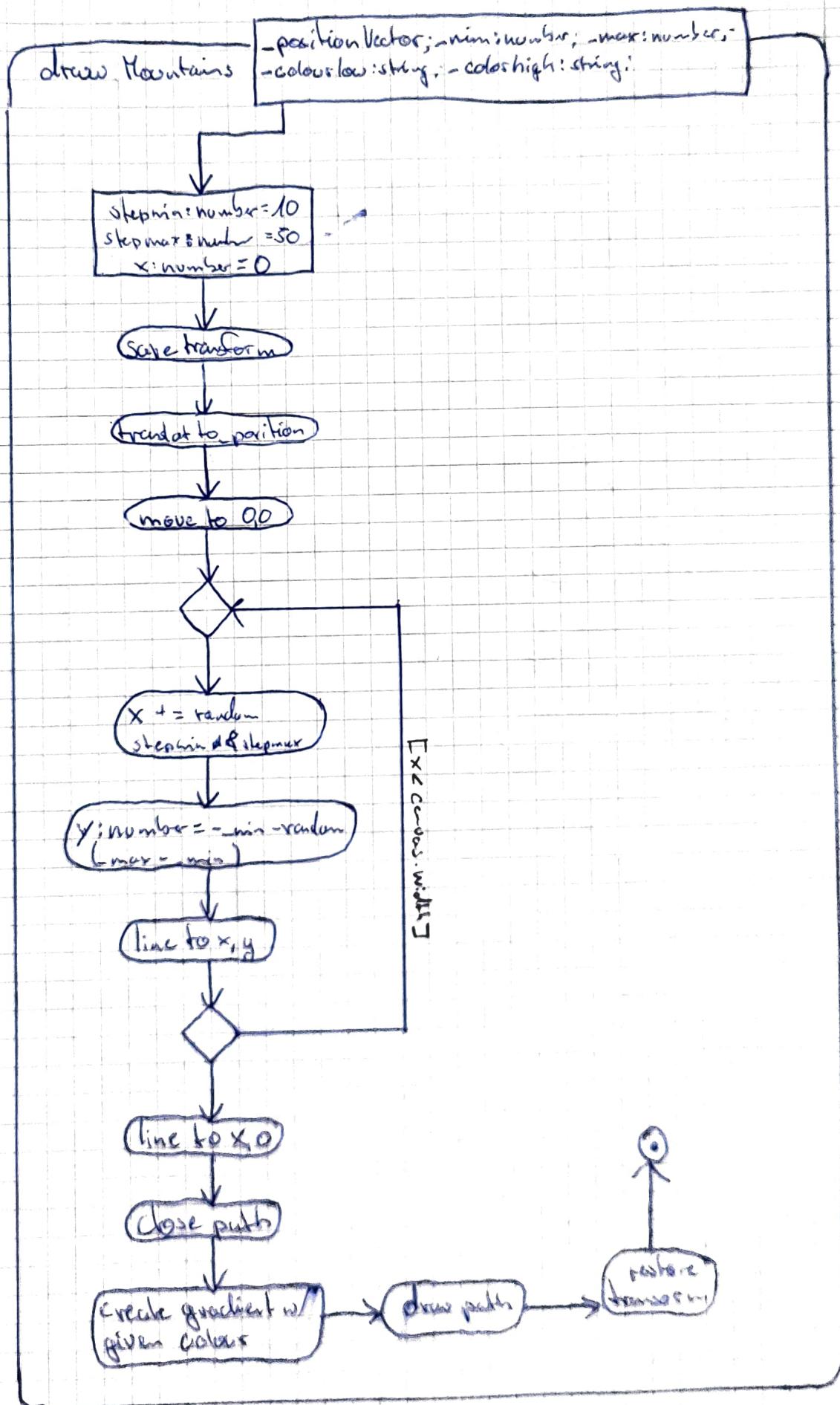
draw trees in
give pos, scale, count

draw log in

draw sun in

draw Mountains in
give → pos, min, max
and colors (2x)

2D Mountains



1D Trees

```
drawTrees(nTrees: number, -posStart: Vector, -posEnd: Vector;  
-minScale: number; -stepPos: number; -stepScale: number)
```

```
let transform = crc2.getTransform()
```

```
let step: Vector = {x, y}
```

```
translate to -posStart
```

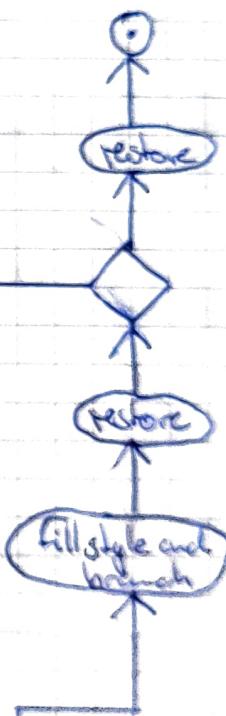
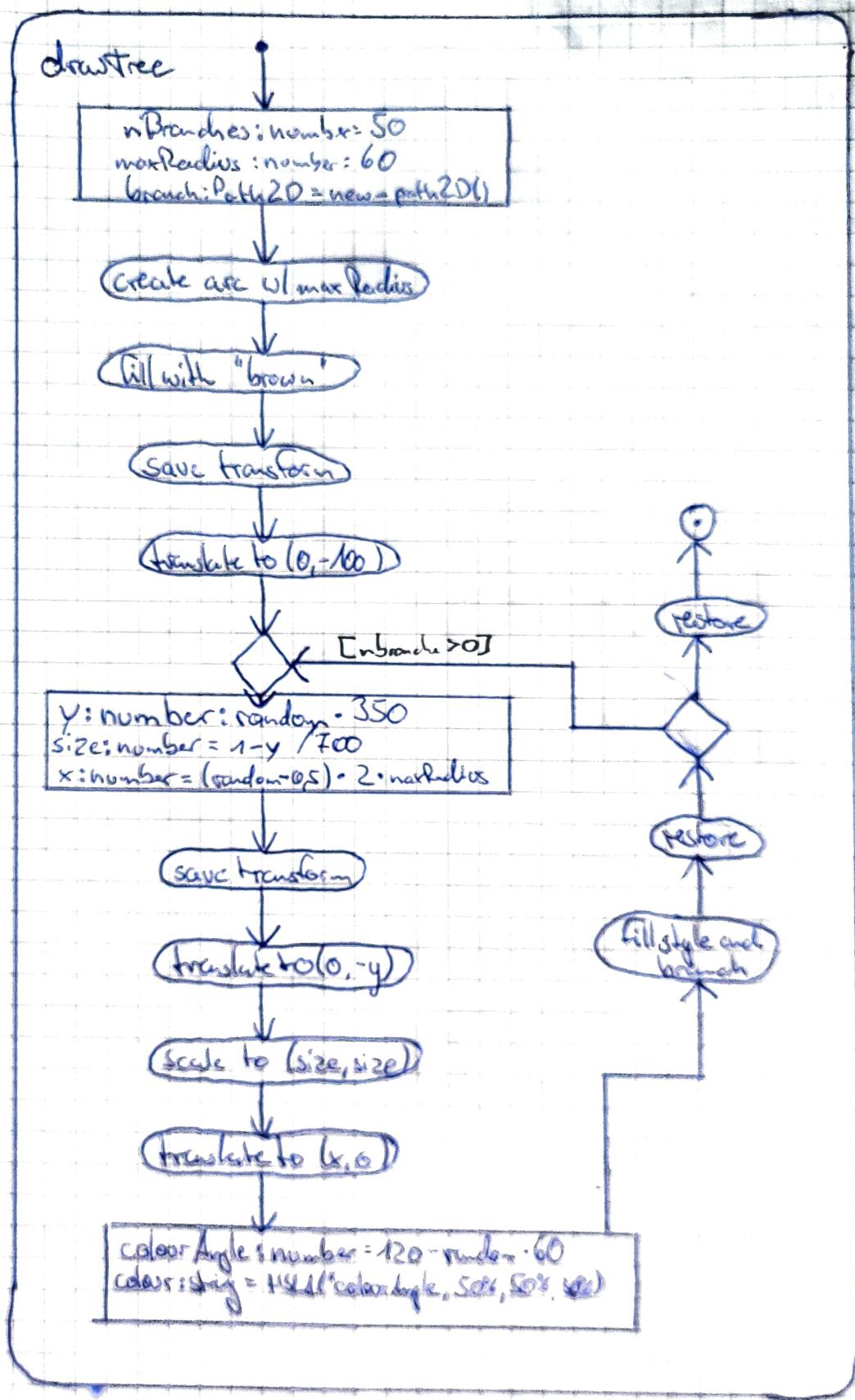
```
scale to -minScale
```

```
drawTree1th
```

```
translate to step
```

```
scale to stepScale
```

```
set to transform
```



drawBush
positionVector;
size: Vector;

nParticles: number = 20;
radiusParticle: number = 15;
particlePath2D = new Path2D();
gradient = createRadialGradient

create arc w/ radiusParticle

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)

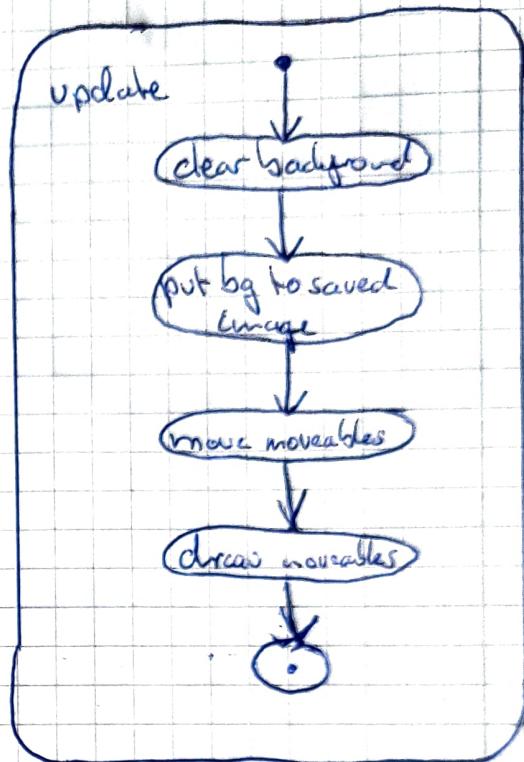
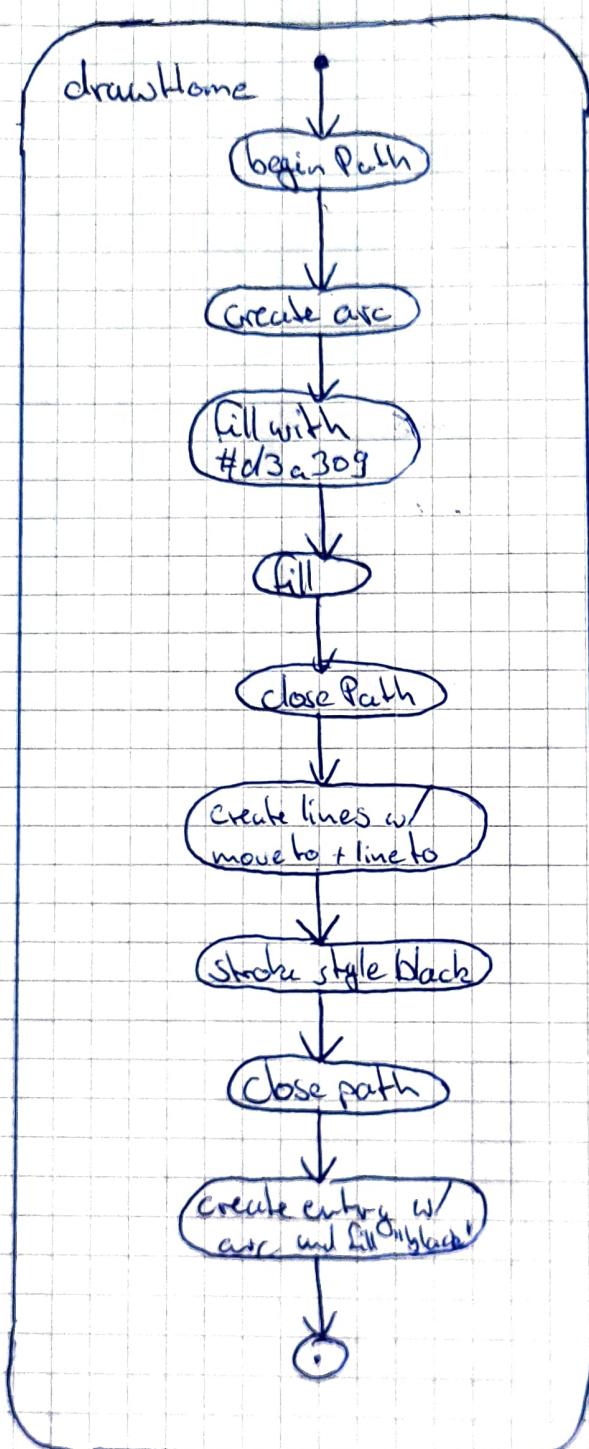
translate x, y

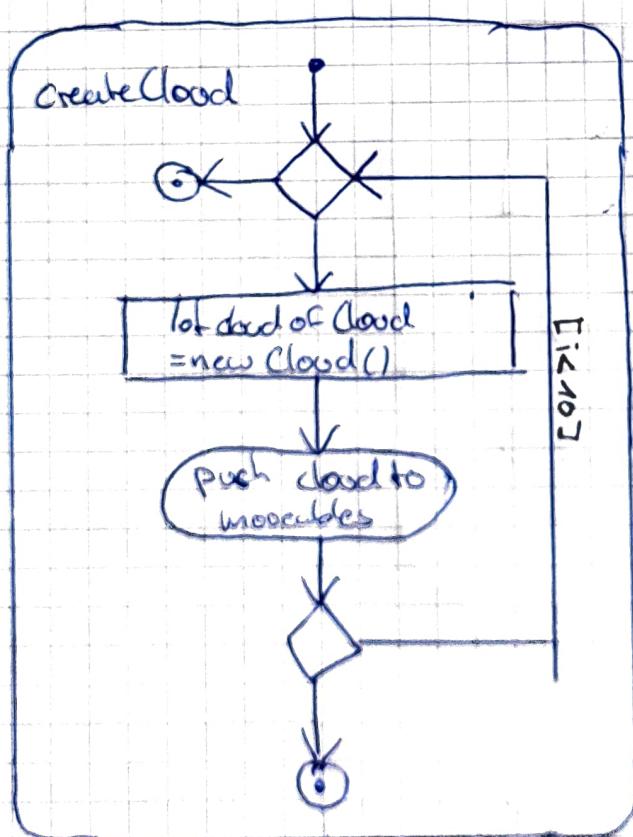
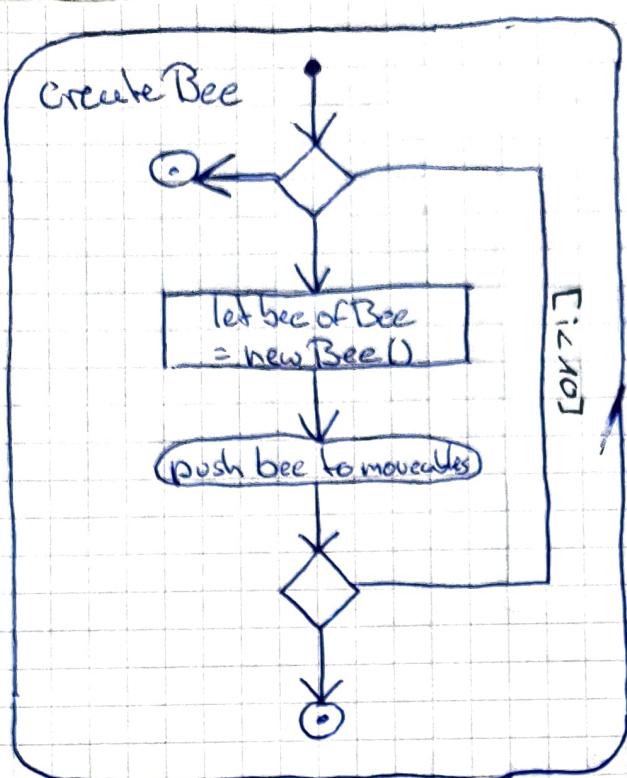
Fill w/ particle

Restore

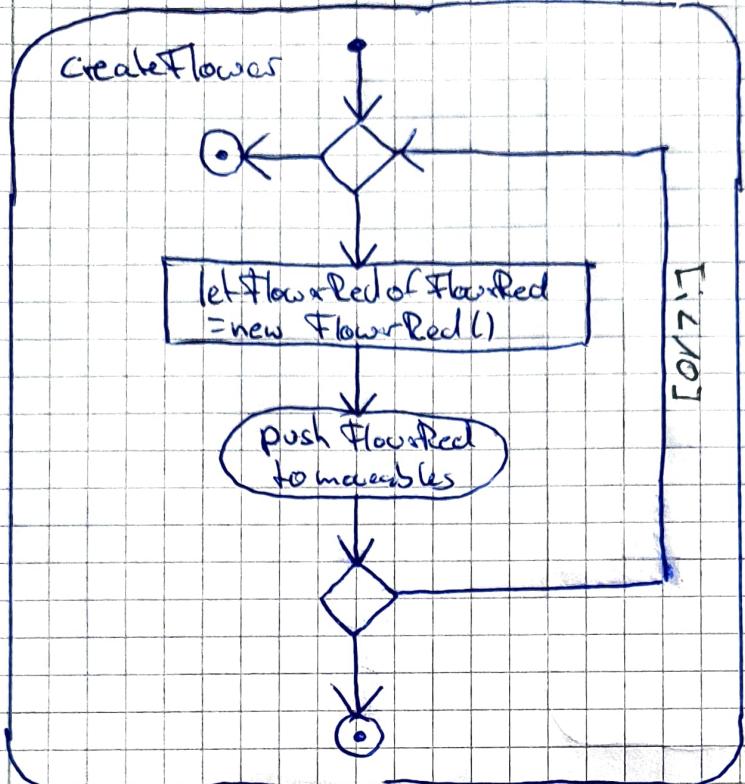


Restore





CreateFlower



WindFan
die and die
false
Wiedeholt

moreBees

