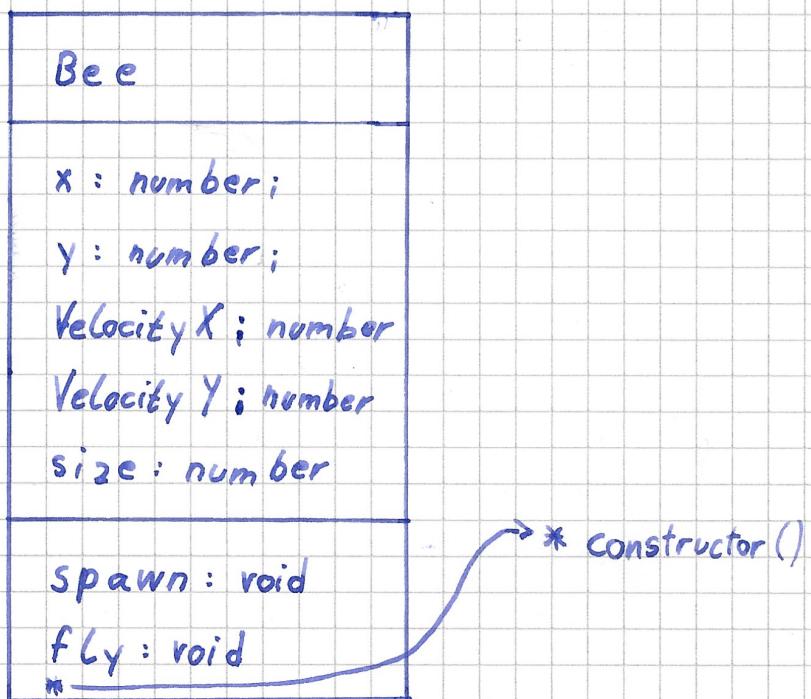
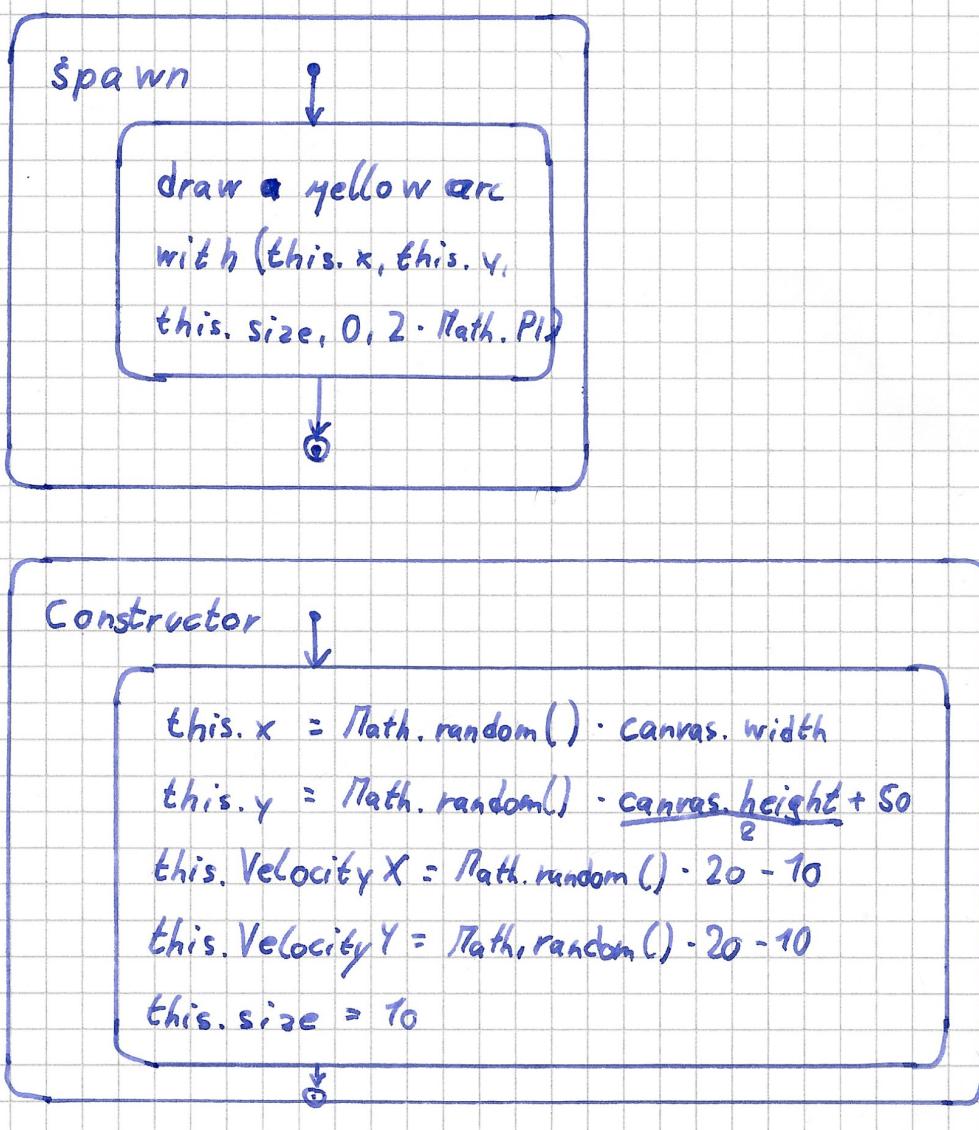


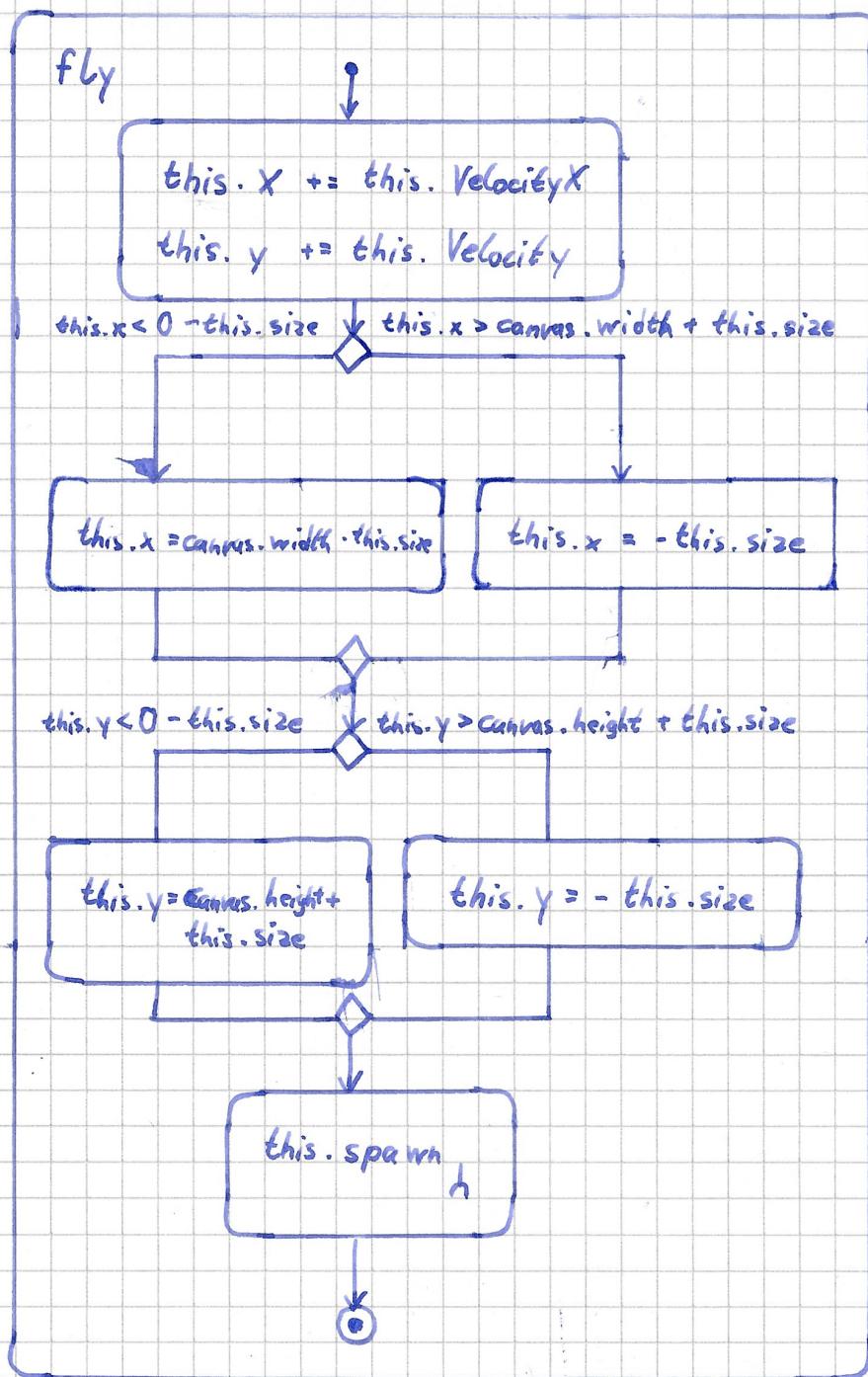
Bee : Class Diagram

29.05.24

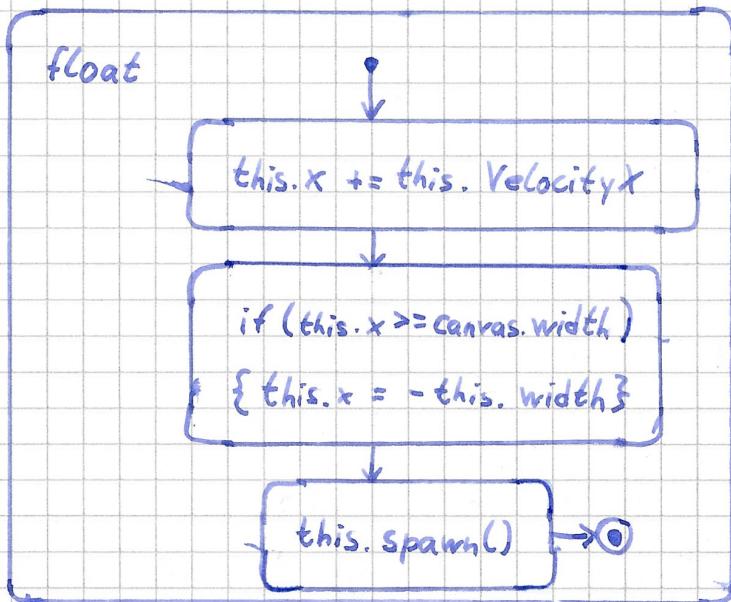
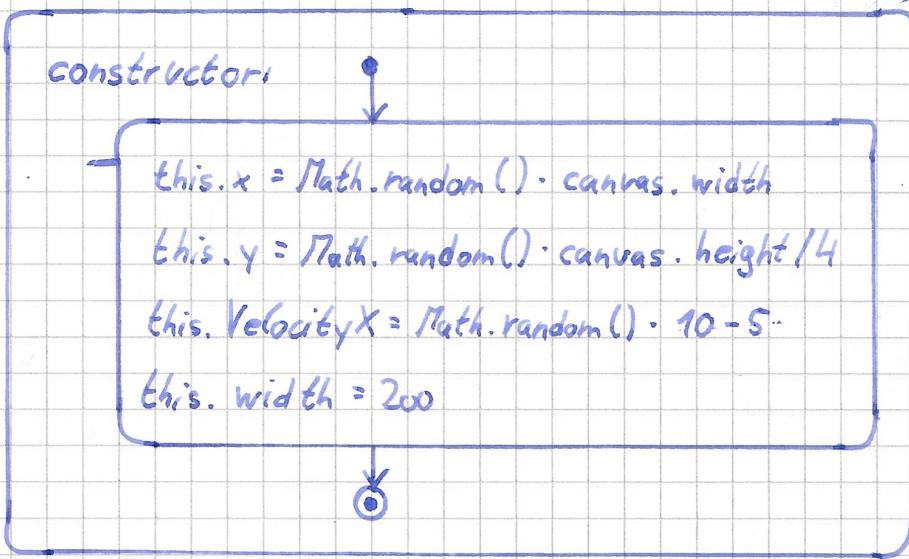
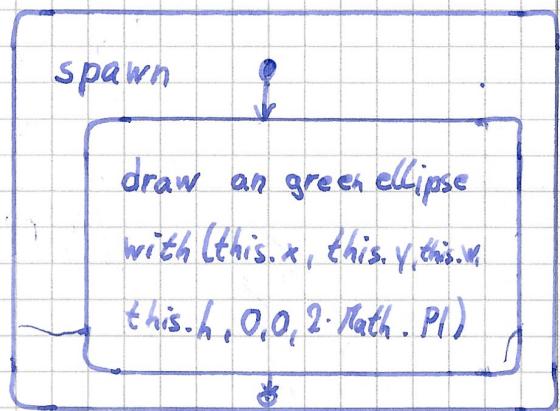
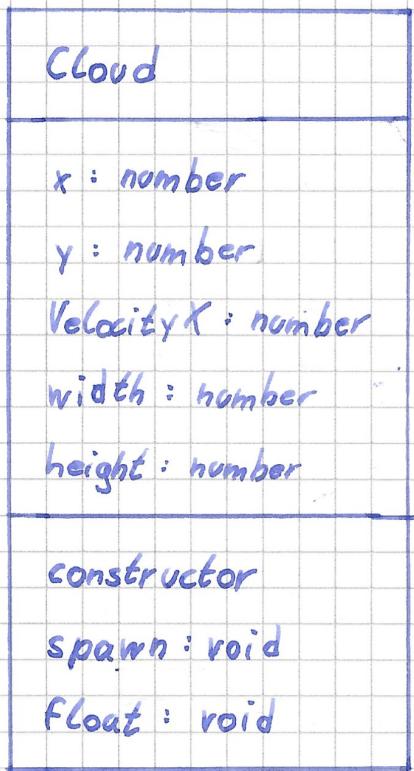


Bee : Activity Diagram

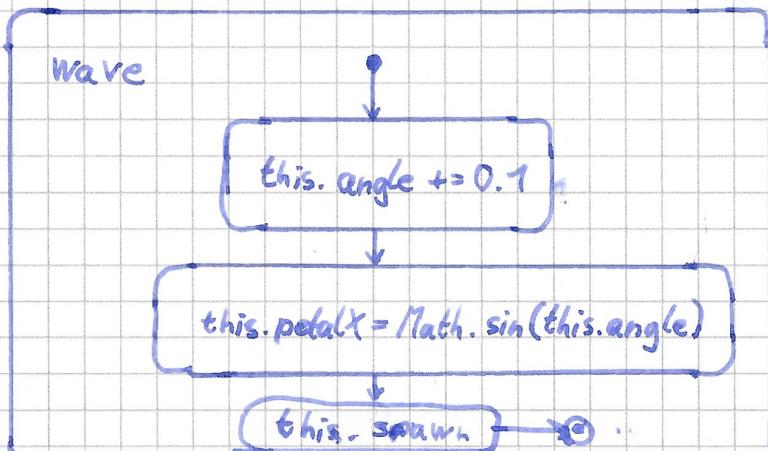
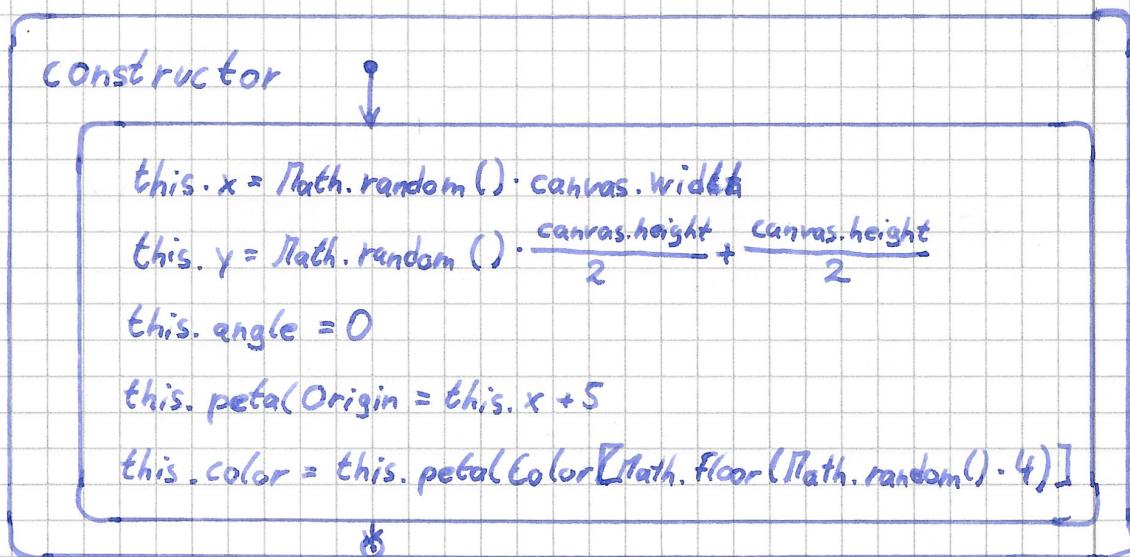
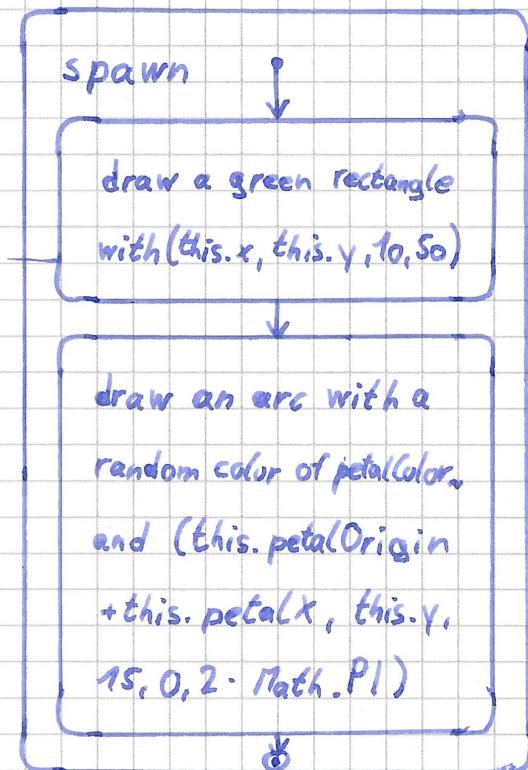
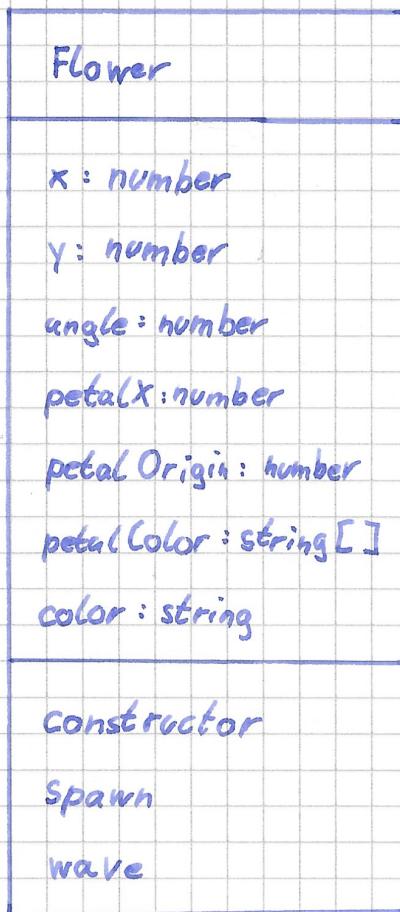




Cloud : Class Diagram + Activity Diagram

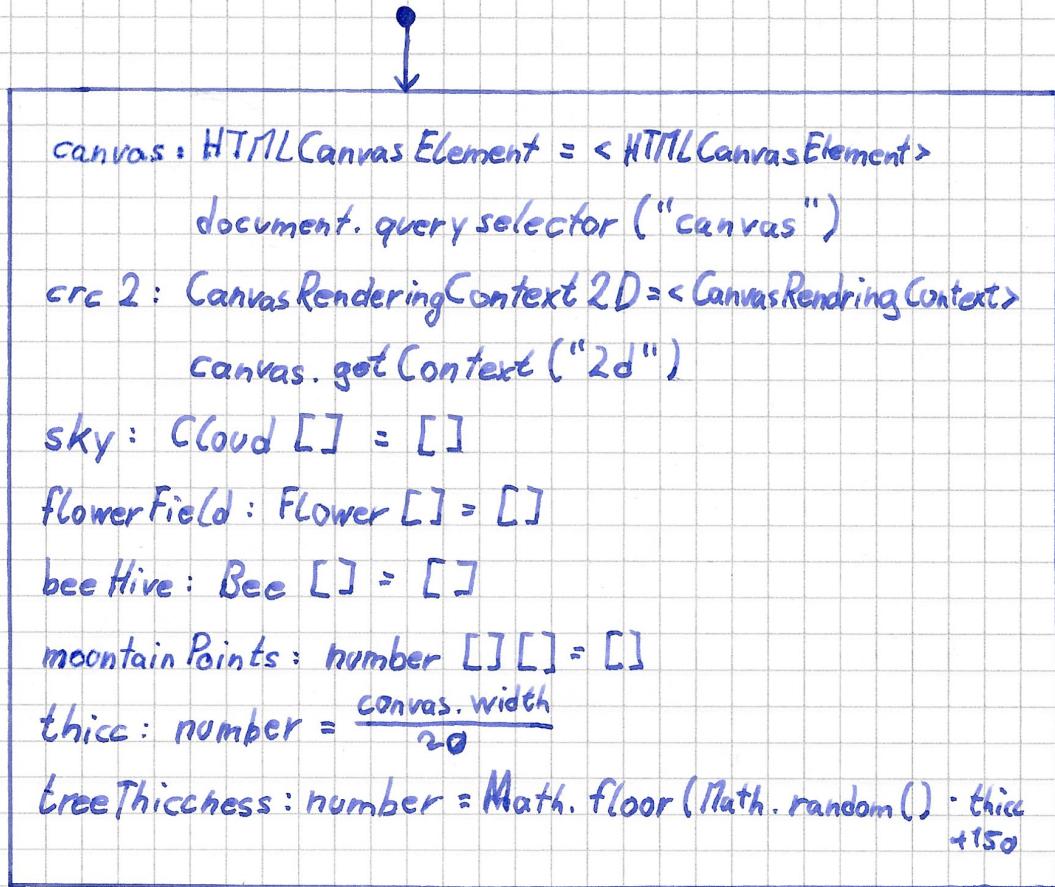


Flower : Class Diagram + Activity Diagram

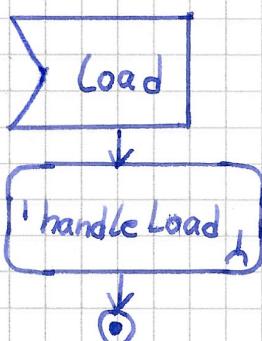


Animierte Wiese : Activity Diagram

28.05.21



install Load Listener



handleLoad

```
fillSky (Math.random() * 70 + 1)
```

```
fillField (Math.random() * 50 + 1)
```

```
fillHive (5)
```

```
generate Mountains (Math.random() * 2 + 3)
```

```
setInterval (updateIMG, 16)
```

fillSky

-cnt: number

i: number = 0

if i < cnt
 then

```
Sky.push(new Cloud)
```

i++

-cnt: number

i: number

if i <= cnt
 then

```
beeHive.push(new Bee)
```

int

fill Field

-cnt: number

i: number = 0

if i < cnt
 then

```
flowerField.push(new Flower)
```

i++

generateMountains

- mntCnt: number

mntPoints.push([0, canvas.height / 2])

i: number = 0

mntPoints.push([0, canvas.height / 2])

X
i < mntCnt - 1

trash: number[] = []

trash.push(i * canvas.width / mntCnt * 2)

trash.push(Math.random() * canvas.height / 2)

mntPoints.push(trash)

L i++

' draw Mountains

move canvas to

(mntPoints[0][0], mntPoint[0][1])

fill canvas gray

i: number = 0

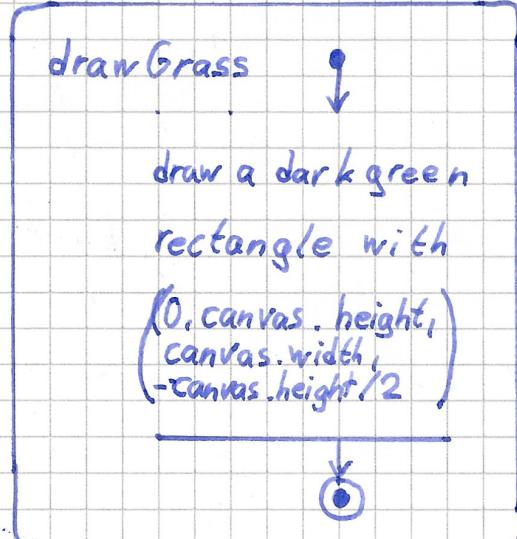
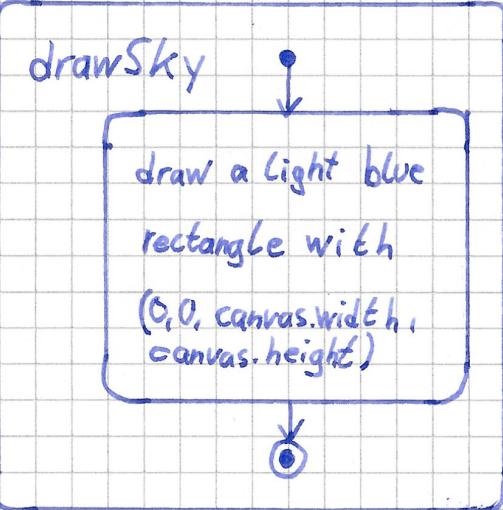
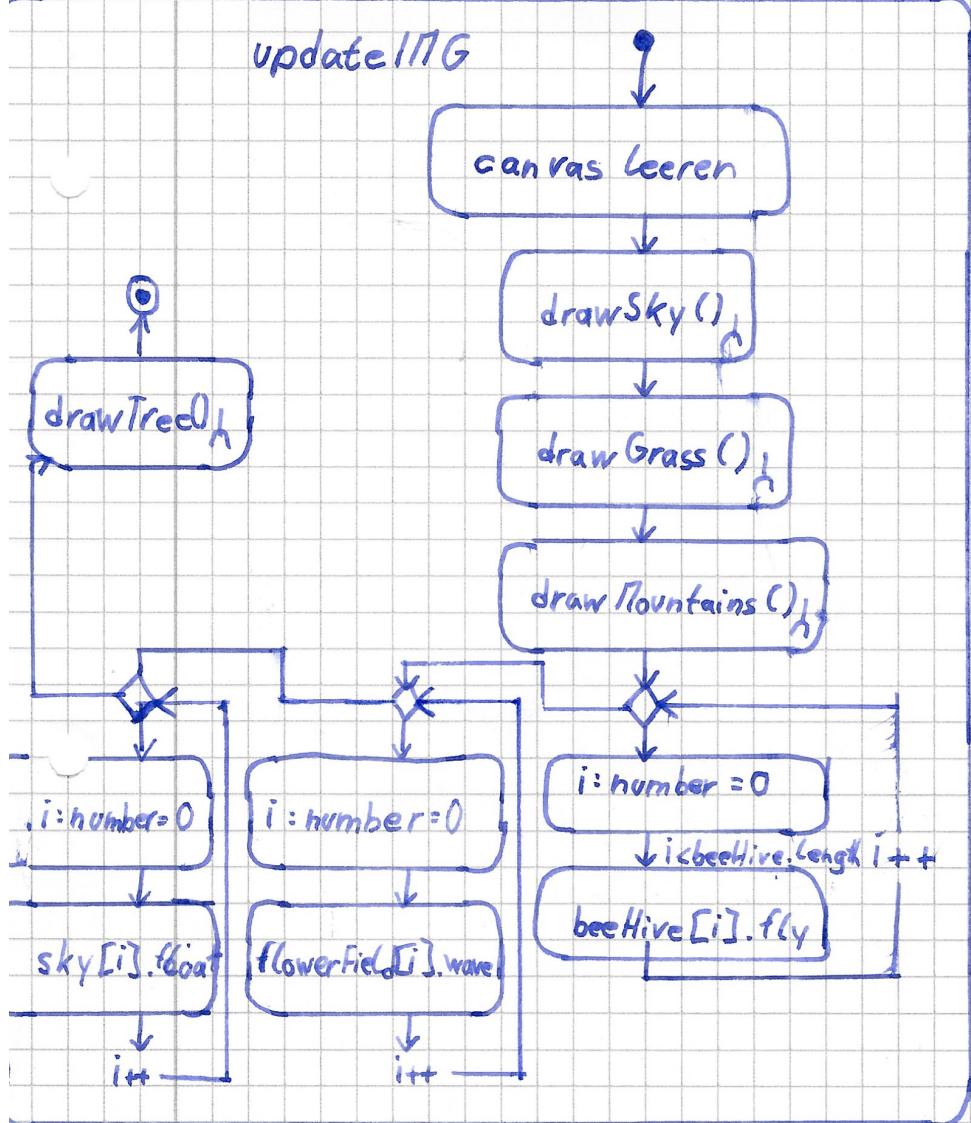
X
i < mntPoints.length

move line to

(mntPoints[i][0], mntPoints[i][1])

i++

update / T/G



drawTree



$$\text{height: number} = \frac{3 \cdot \text{canvas.height}}{4}$$

pos: number [] = [0, canvas.height]



draw a brown rectangle with
(pos[0], pos[1], thick, height)



draw a dark green arc with
(pos[0]+thick/2, pos[1]-height, treeThickness, 0, 2 * Math.PI)

