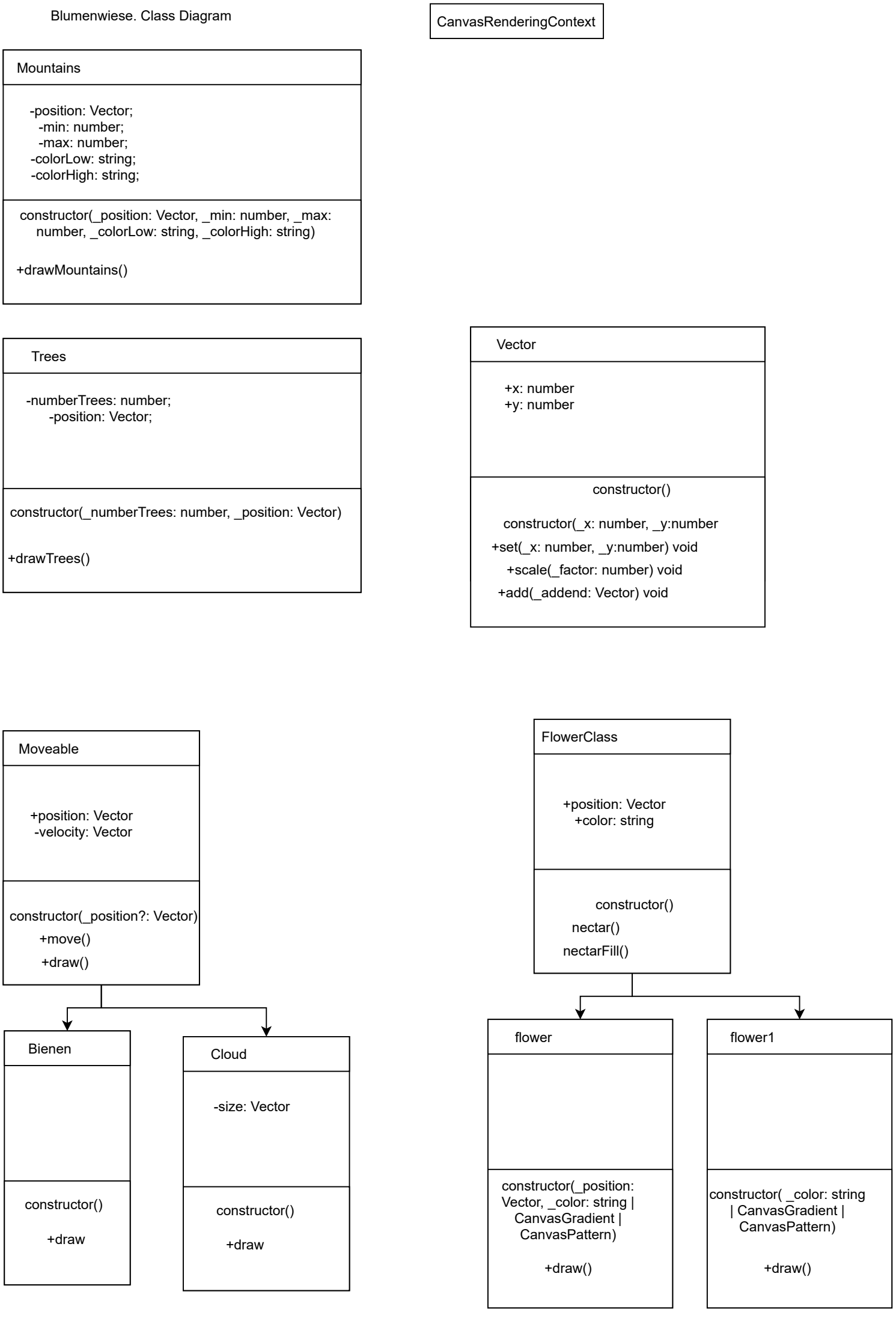
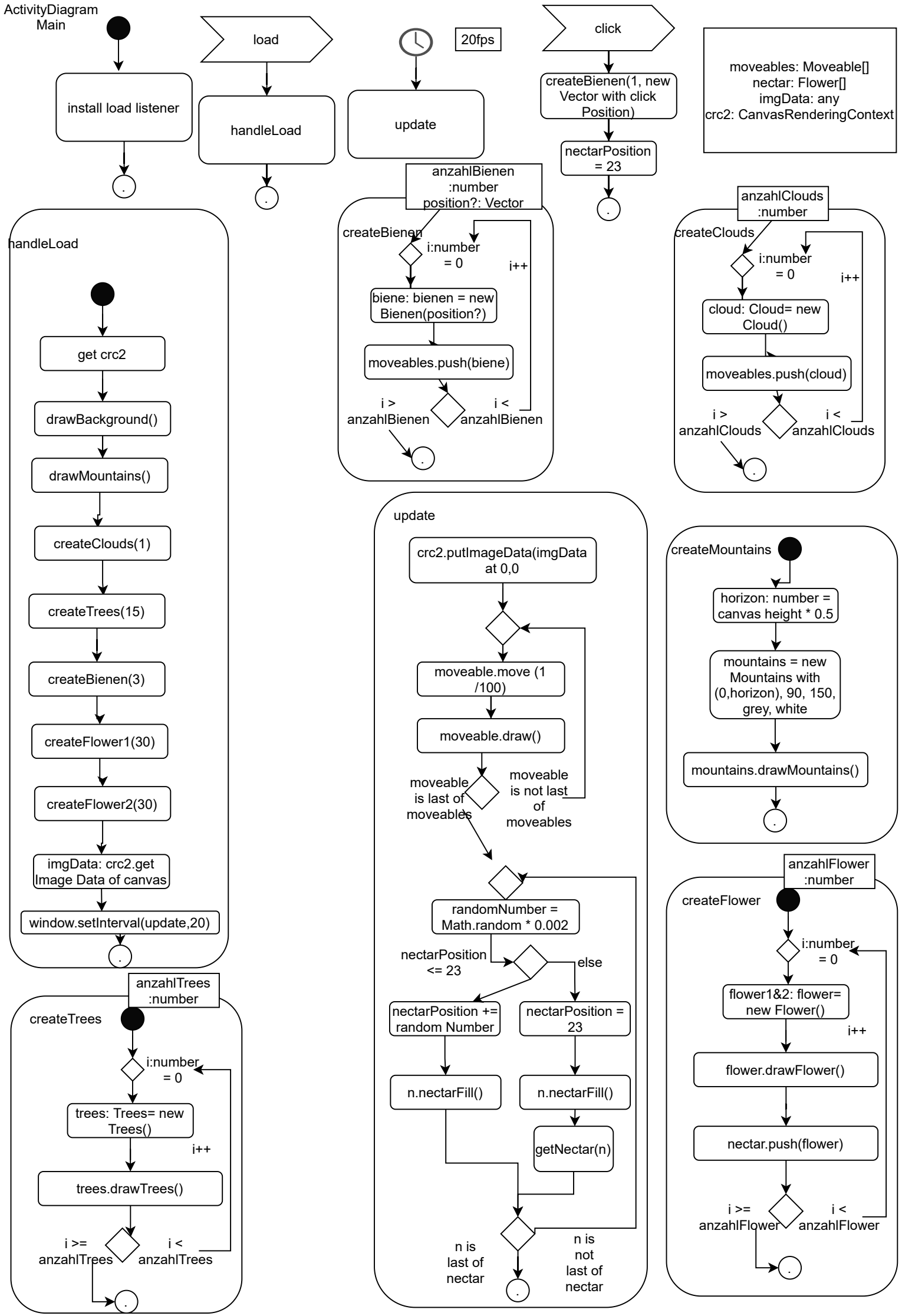
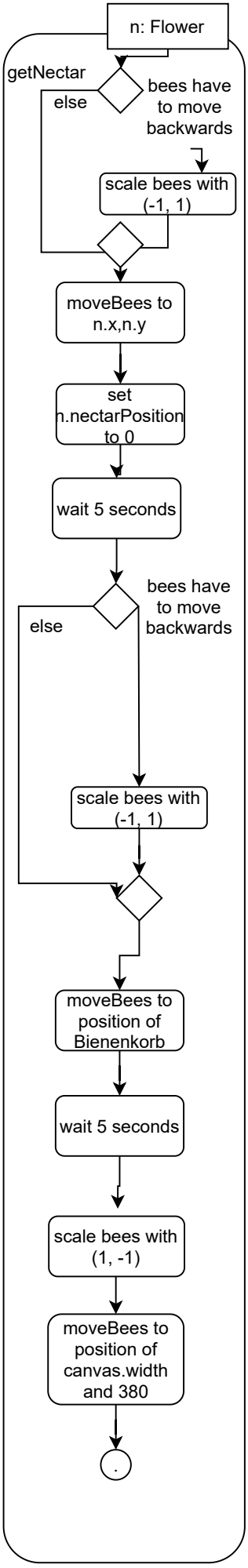
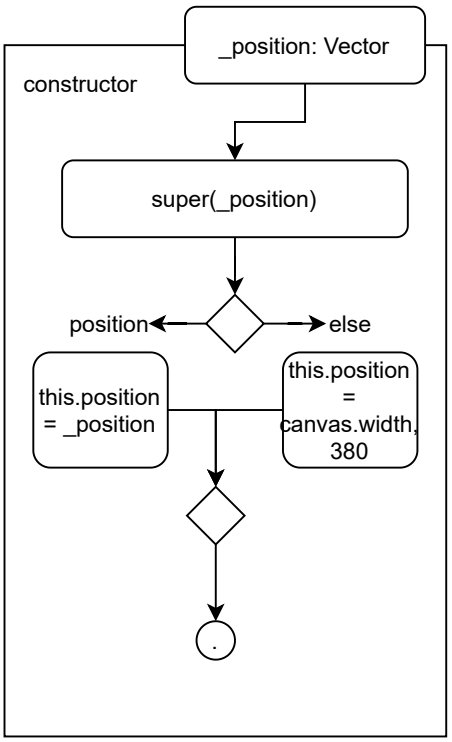
Blumenwiese. Class Diagram

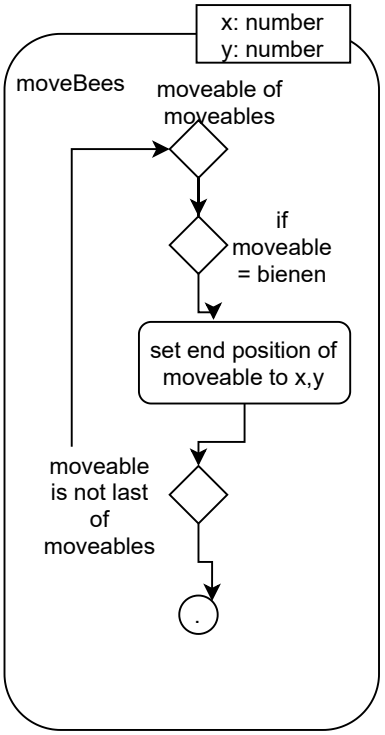


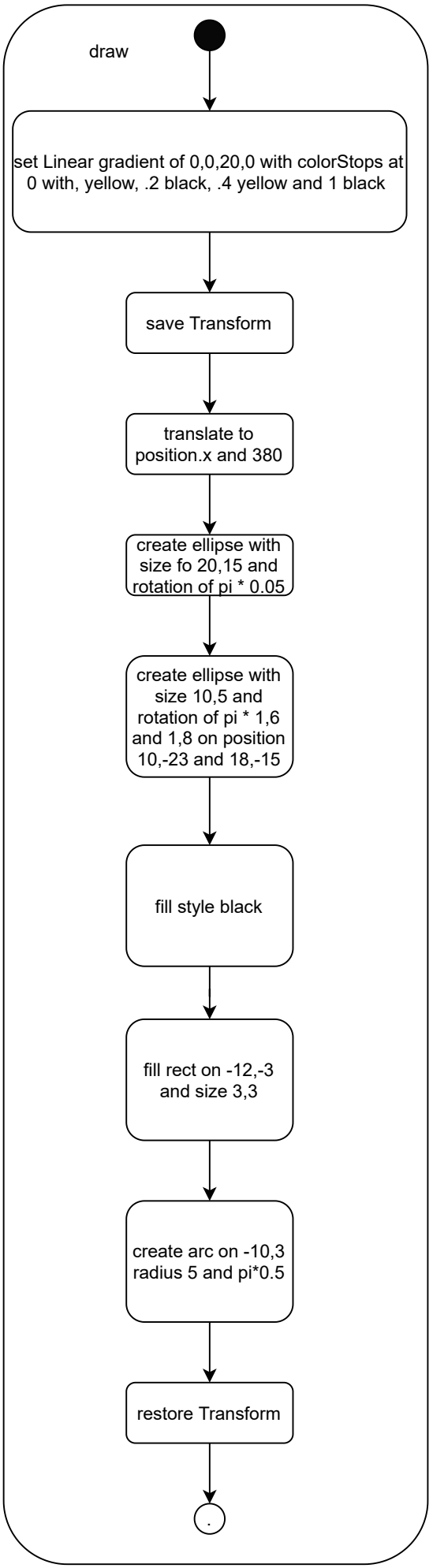

ActivityDiagram
Main
install load listener
load
handleLoad
20fps
update
click
createBienen(1, new Vector with click Position)
nectarPosition = 23
moveables: Moveable[]
nectar: Flower[]
imgData: any
crc2: CanvasRenderingContext
anzahlBienen :number
position?: Vector
createBienen
i:number = 0
i++
biene: bienen = new Bienen(position?)
moveables.push(biene)
i > anzahlBienen
i < anzahlBienen
anzahlClouds :number
createClouds
i:number = 0
i++
cloud: Cloud= new Cloud()
moveables.push(cloud)
i > anzahlClouds
i < anzahlClouds
handleLoad
get crc2
drawBackground()
drawMountains()
createClouds(1)
createTrees(15)
createBienen(3)
createFlower1(30)
createFlower2(30)
imgData: crc2.get Image Data of canvas
window.setInterval(update,20)
update
crc2.putImageData(imgData at 0,0
moveable.move (1 /100)
moveable.draw()
moveable is last of moveables
moveable is not last of moveables
randomNumber = Math.random * 0.002
nectarPosition <= 23
else
nectarPosition += random Number
nectarPosition = 23
n.nectarFill()
n.nectarFill()
getNectar(n)
n is last of nectar
n is not last of nectar
createMountains
horizon: number = canvas height * 0.5
mountains = new Mountains with (0,horizon), 90, 150, grey, white
mountains.drawMountains()
anzahlTrees :number
createTrees
i:number = 0
trees: Trees= new Trees()
i++
trees.drawTrees()
i >= anzahlTrees
i < anzahlTrees
anzahlFlower :number
createFlower
i:number = 0
flower1&2: flower= new Flower()
i++
flower.drawFlower()
nectar.push(flower)
i >= anzahlFlower
i < anzahlFlower

ActivityDiagram:
Bienen

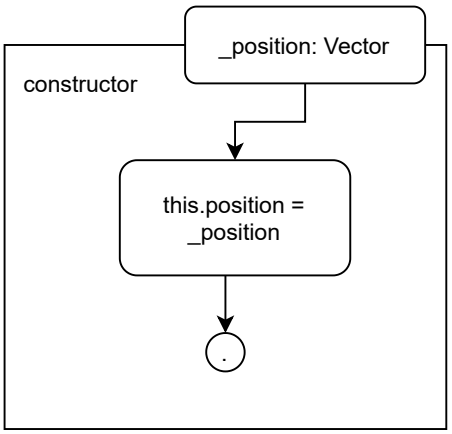


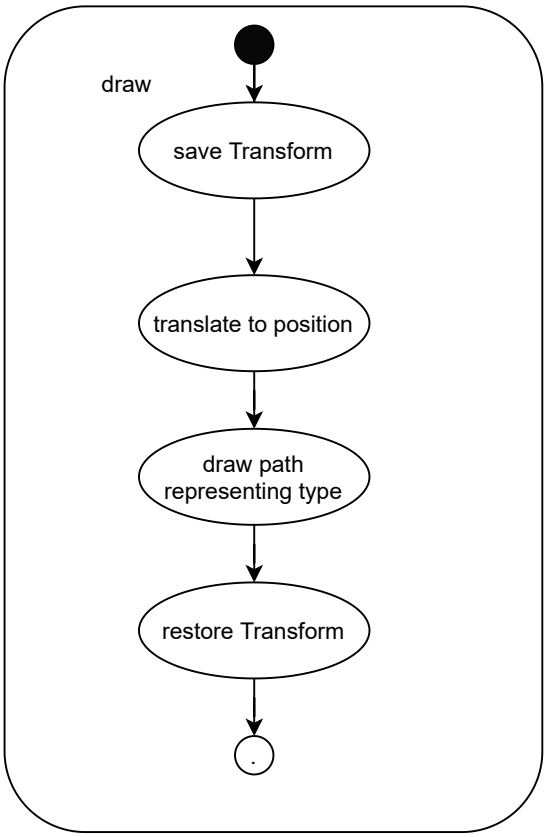




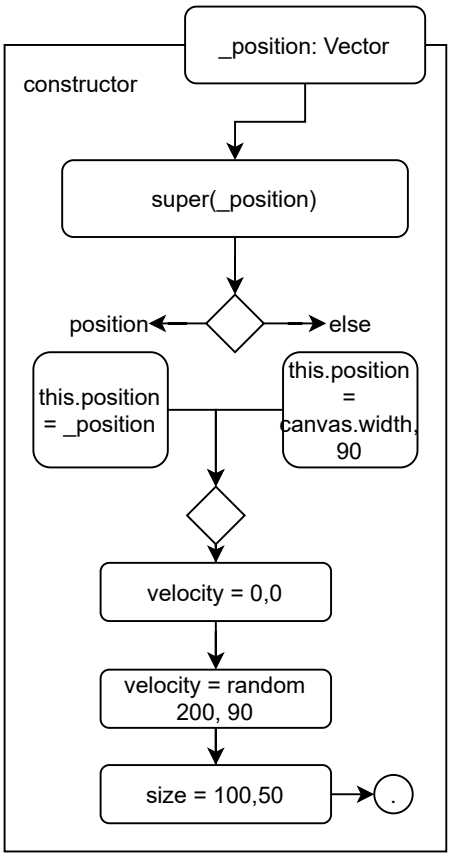


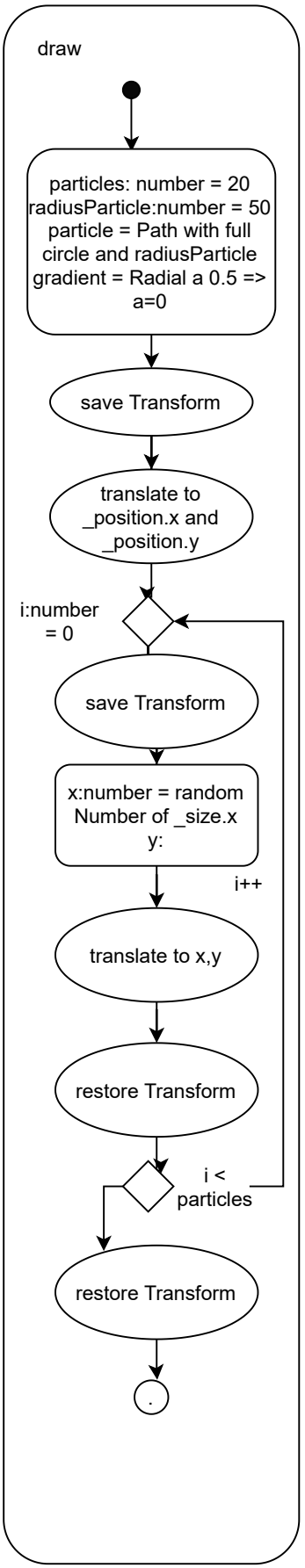
ActivityDiagram:
Bienenkorb





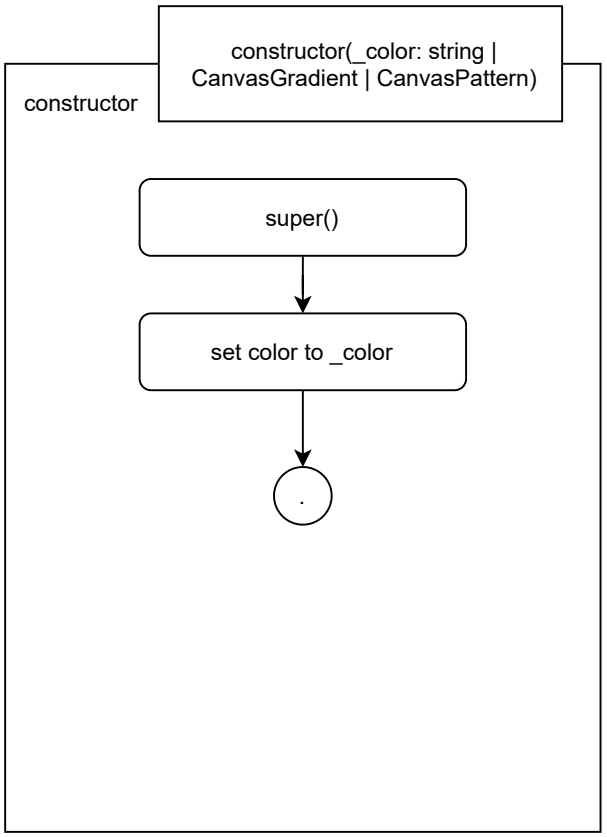
ActivityDiagram
Cloud

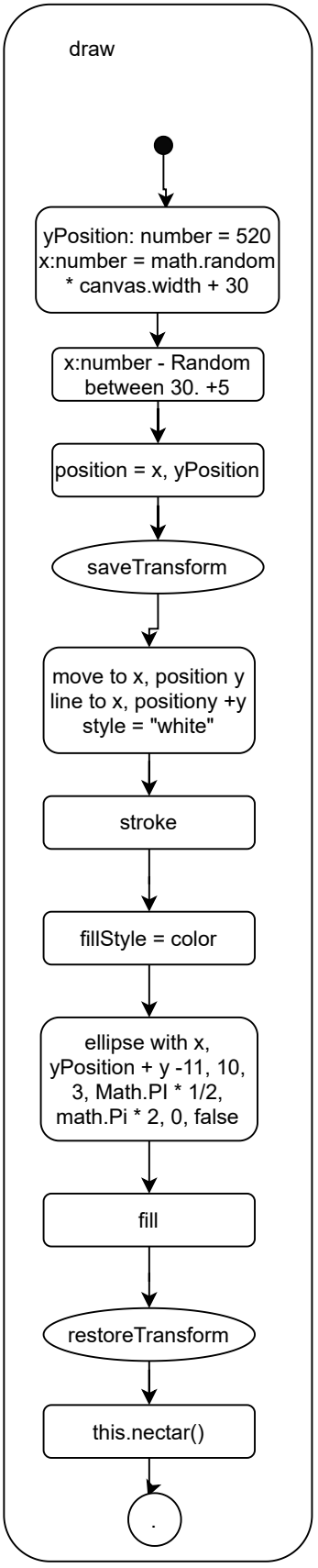


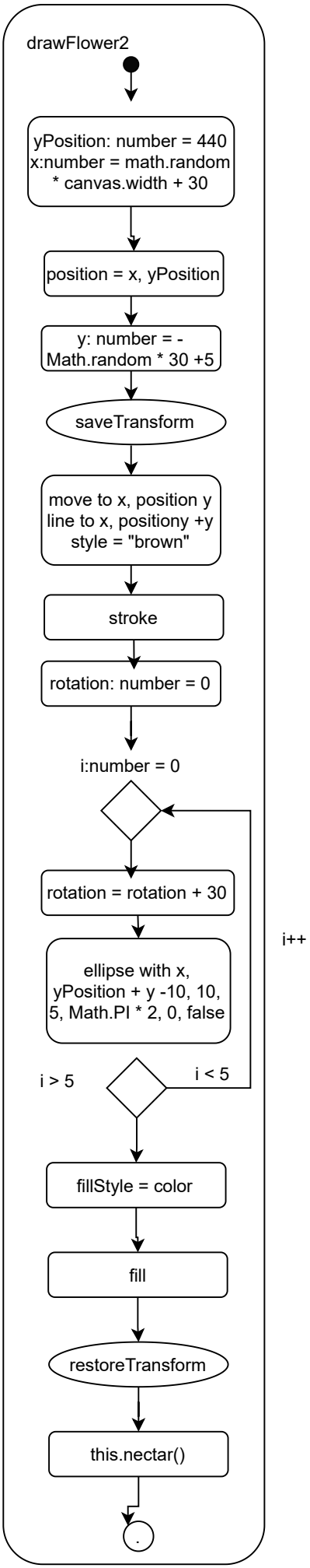


ActivityDiagram for
flower and flower1

nectarPosition:
number = 0



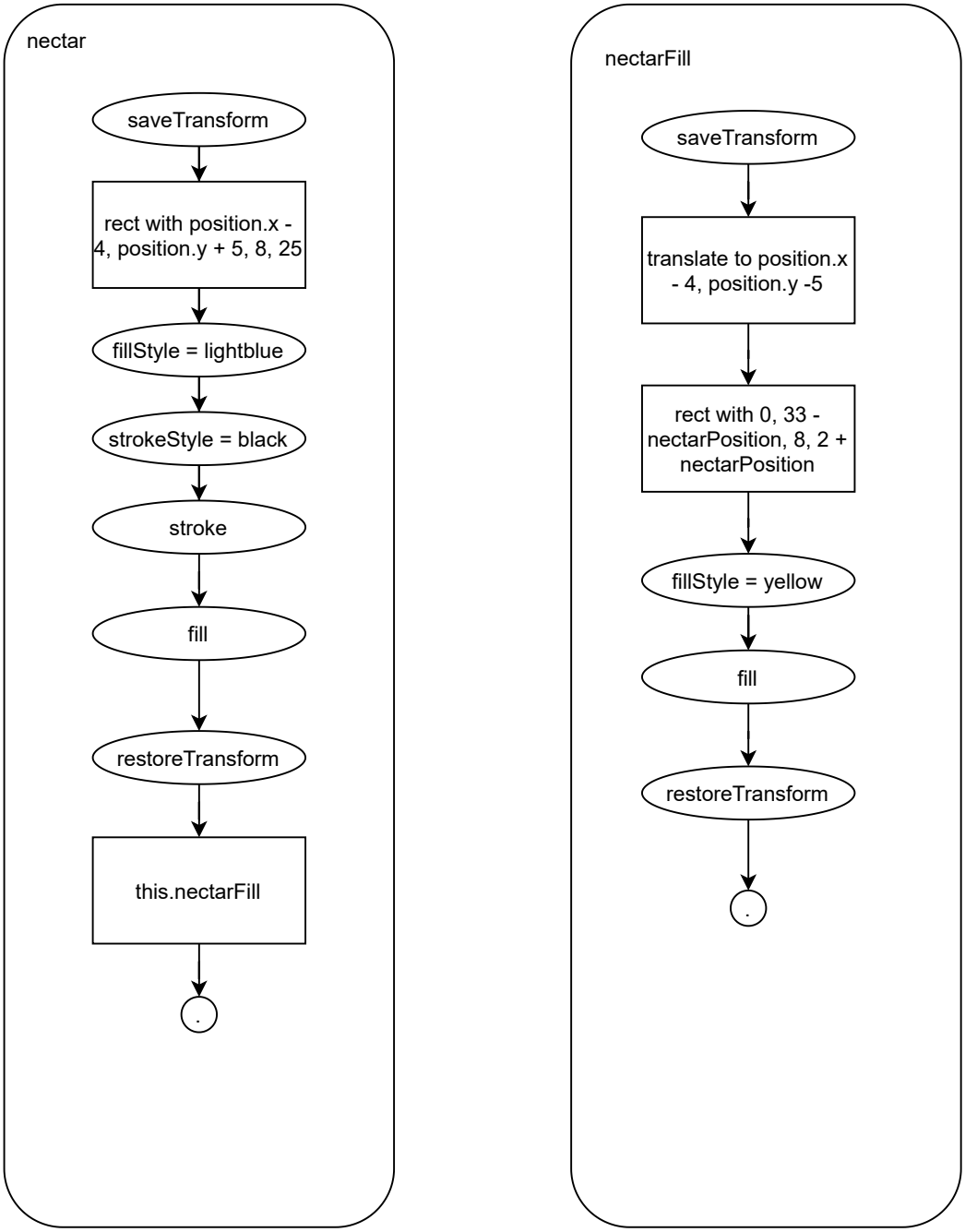




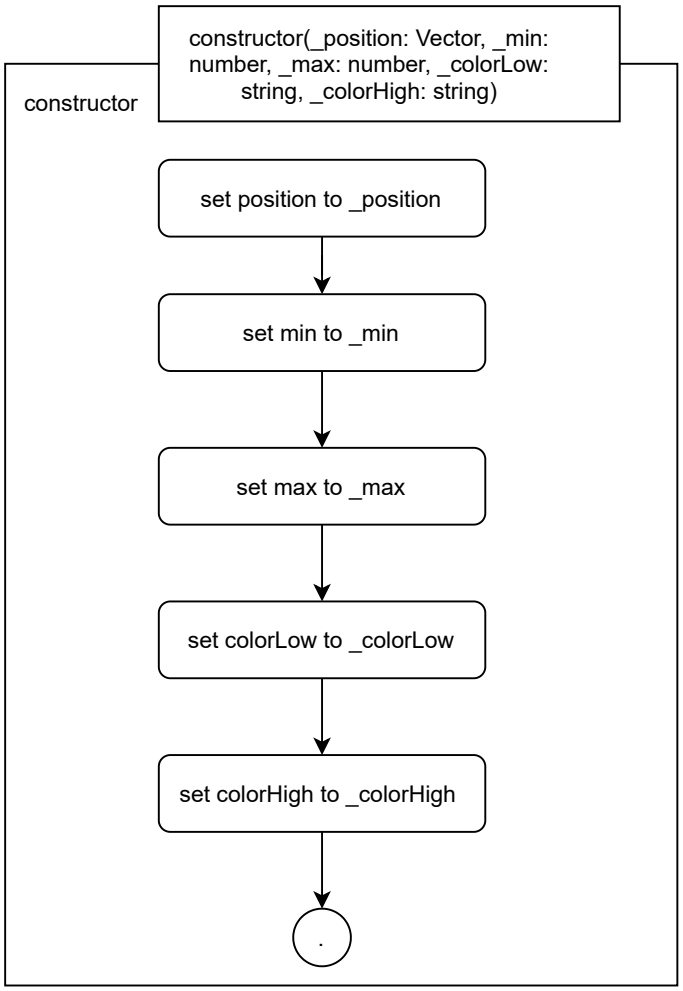
FlowerClass:
Activity
Diagram

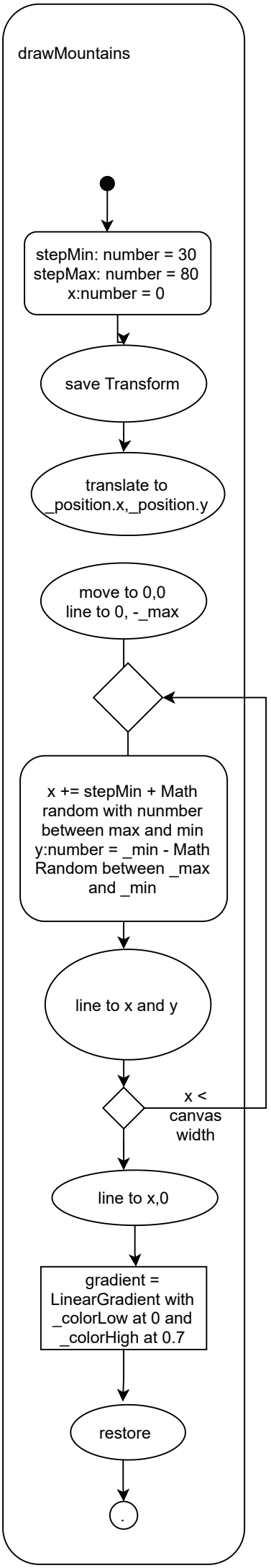
position: Vector
color: string

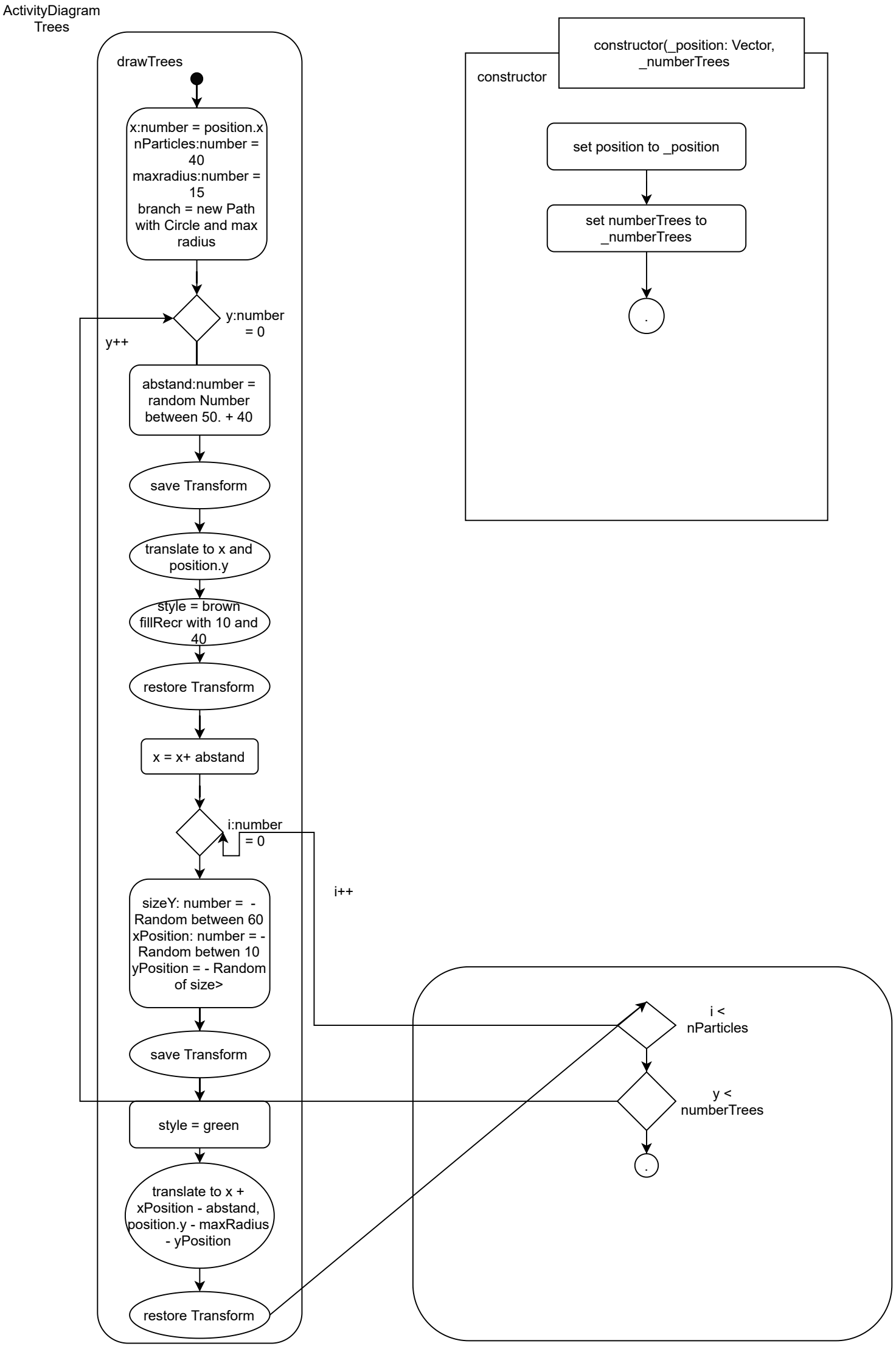
public abstact draw



ActivityDiagram
Mountains






ActivityDiagram
Trees
drawTrees
x:number = position.x
nParticles:number = 40
maxradius:number = 15
branch = new Path with Circle and max radius
y:number = 0
y++
abstand:number = random Number between 50. + 40
save Transform
translate to x and position.y
style = brown fillRecr with 10 and 40
restore Transform
x = x+ abstand
i:number = 0
i++
sizeY: number = - Random between 60
xPosition: number = - Random betwen 10
yPosition = - Random of size>
save Transform
style = green
translate to x + xPosition - abstand, position.y - maxRadius - yPosition
restore Transform
i < nParticles
y < numberTrees
.
constructor
constructor(_position: Vector, _numberTrees
set position to _position
set numberTrees to _numberTrees
.