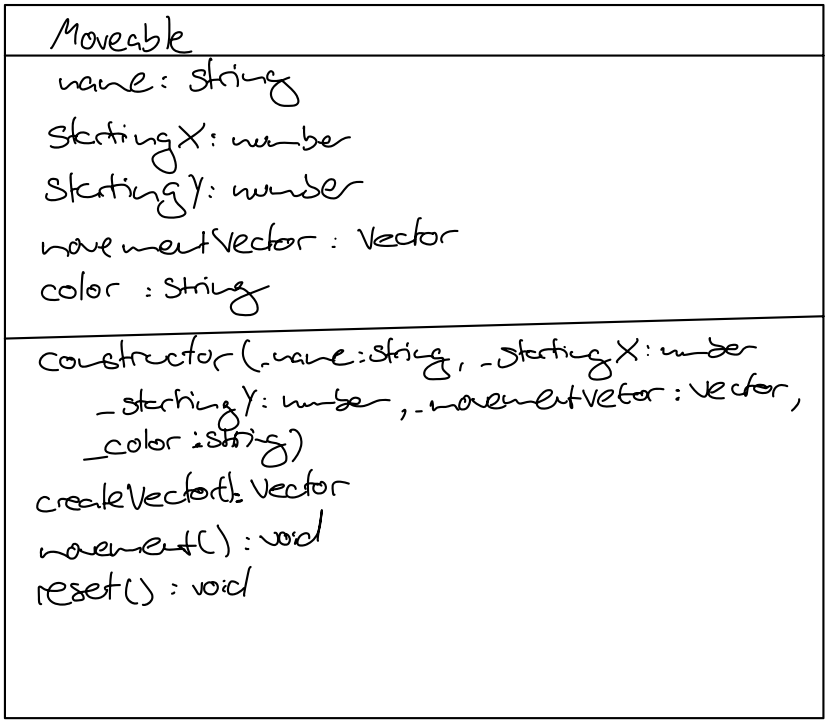
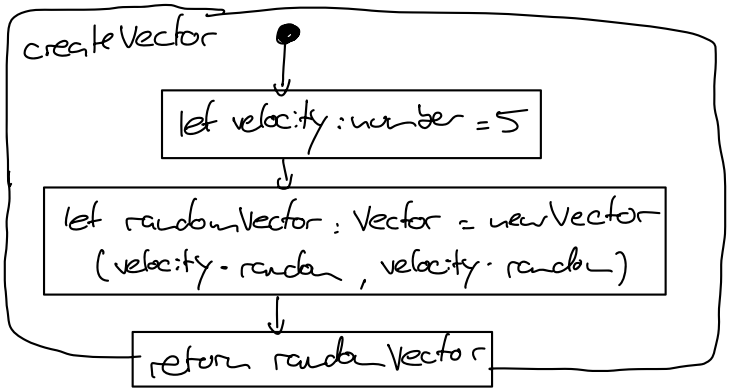
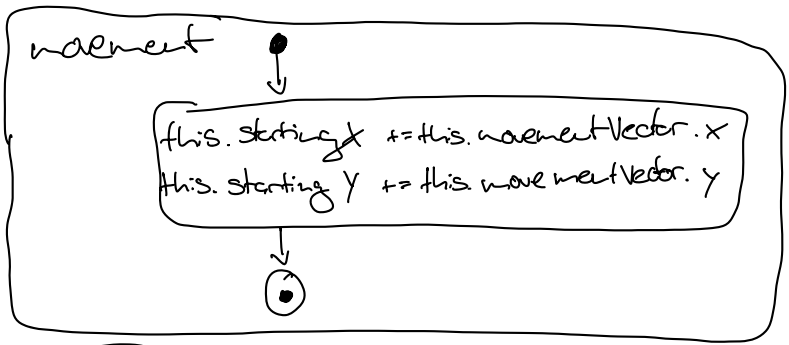
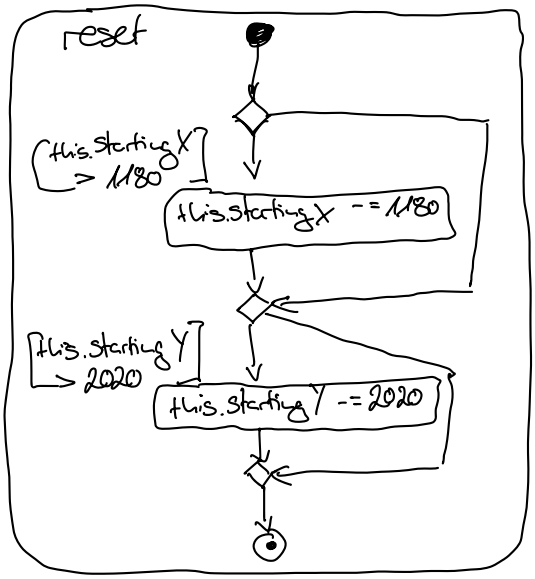

Moveable
name: string
startingX: number
startingY: number
movementVector : Vector
color : string
constructor(_name:string, _startingX: number
_startingY: number, _movementVector: Vector,
_color:string)
createVector(): Vector
movement(): void
reset(): void


createVector
let velocity: number = 5
let randomVector: Vector = new Vector
(velocity * random, velocity * random)
return randomVector

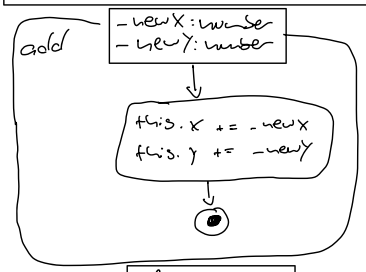

movement
this.startingX += this.movementVector.x
this.startingY += this.movementVector.y

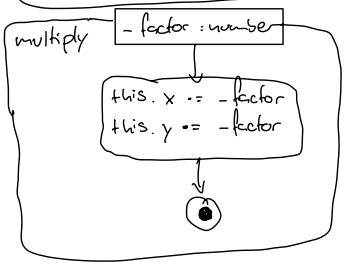

reset
this.startingX
> 1180
this.startingX -= 1180
this.startingY
> 2020
this.startingY -= 2020

Vector

x: number
y: number

constructor (_x: number, _y: number)
add (_newX: number, _newY: number)
multiply (_factor : number)

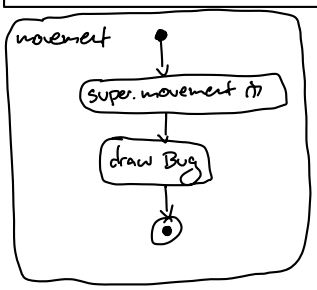




Bug

super(_name, _startingX, _startingY, _color)

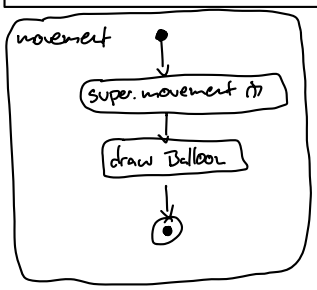
constructor(_name : string, _startingX: number, _startingY: number, _color : string)
movement():void



Balloon

super(_name, _startingX, _startingY, _color)

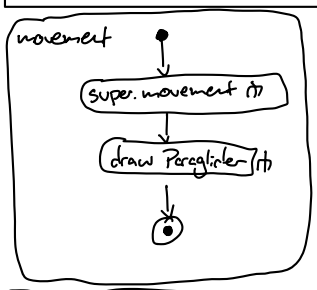
constructor(_name : string, _startingX: number, _startingY: number, _color : string)
movement():void

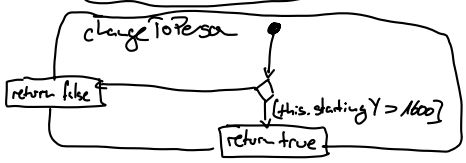


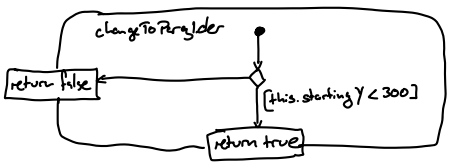
Paraglider

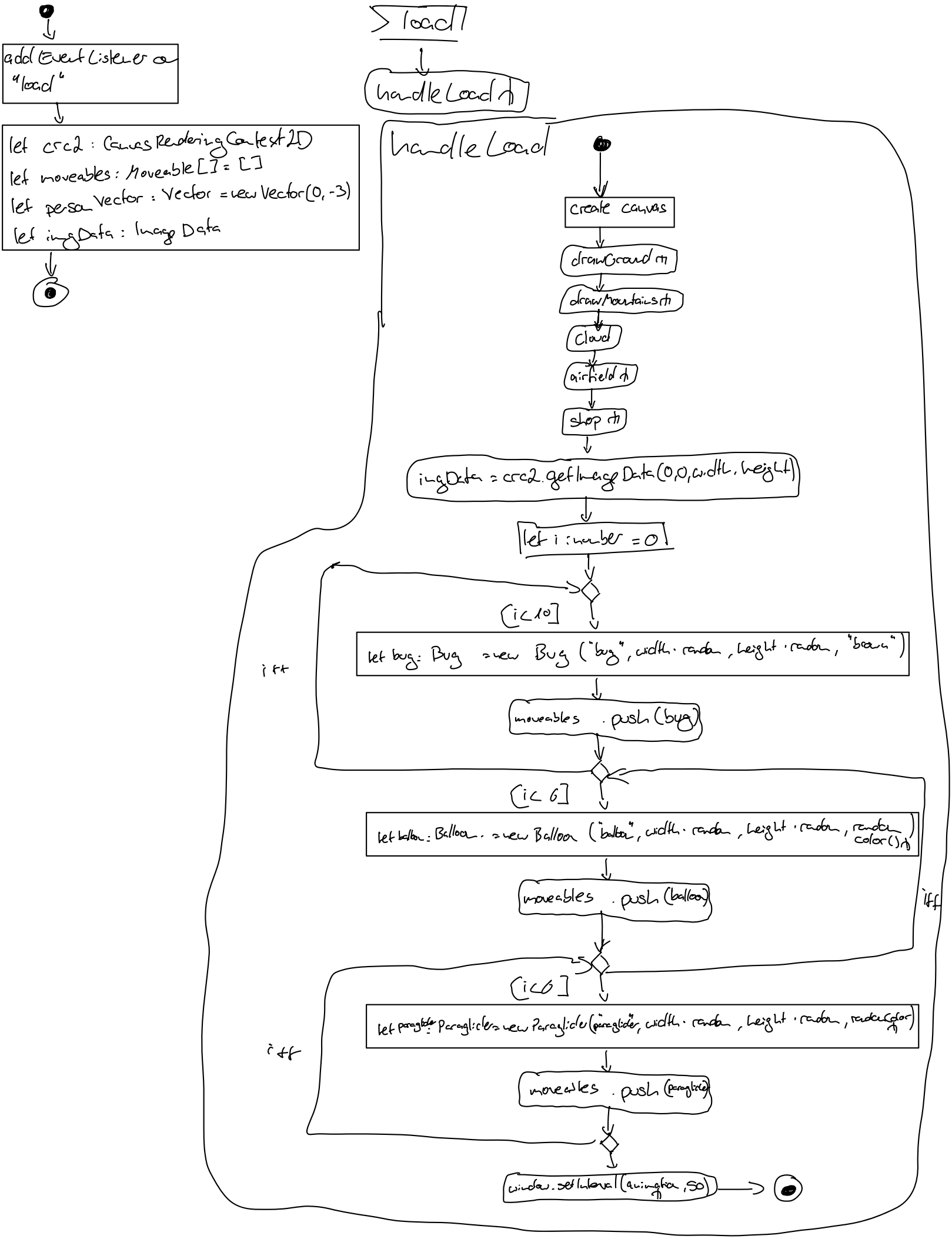
super(_name, _startingX, _startingY, _color)

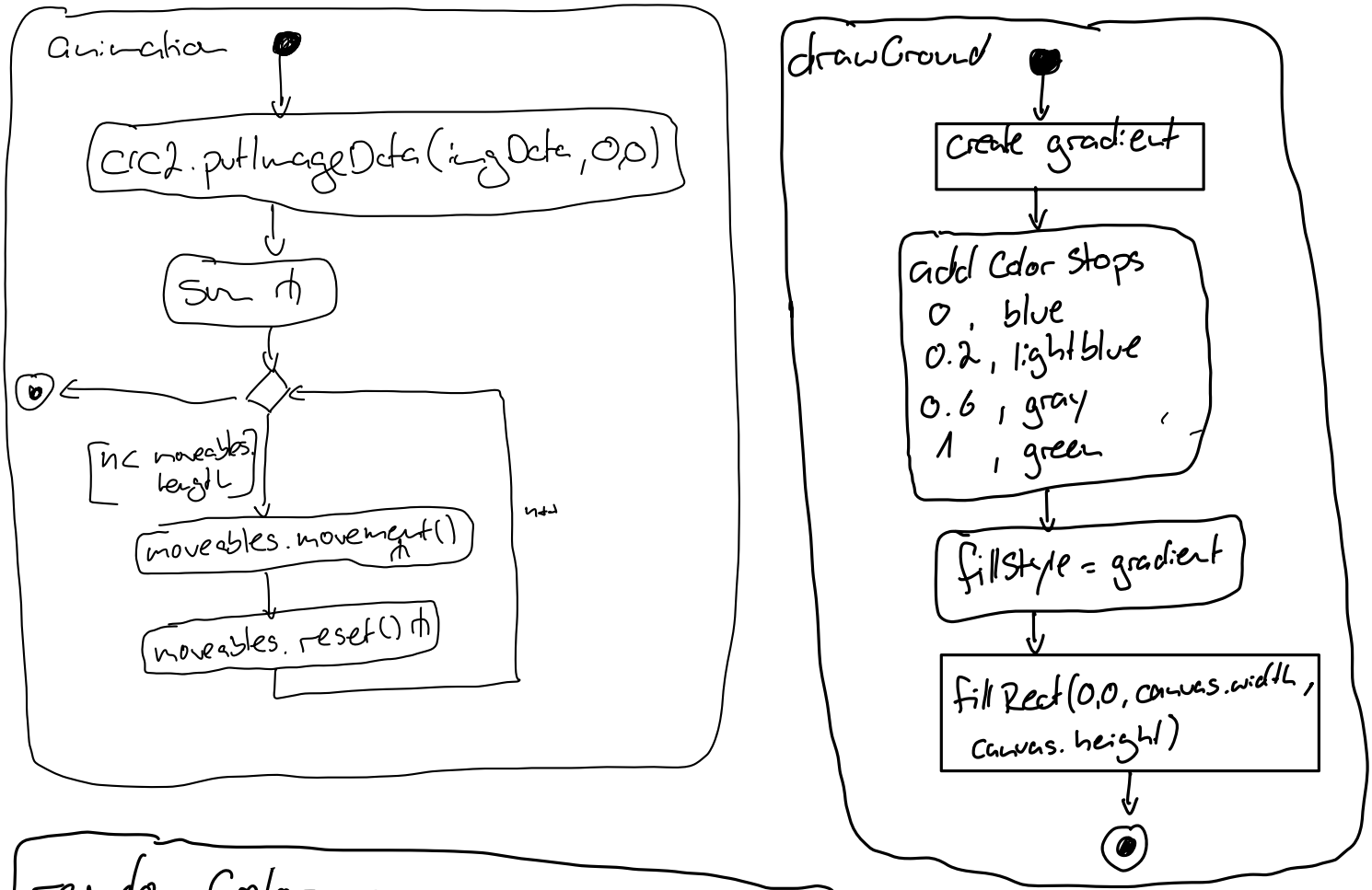
constructor(_name : string, _startingX: number, _startingY: number, _color : string)
movement():void

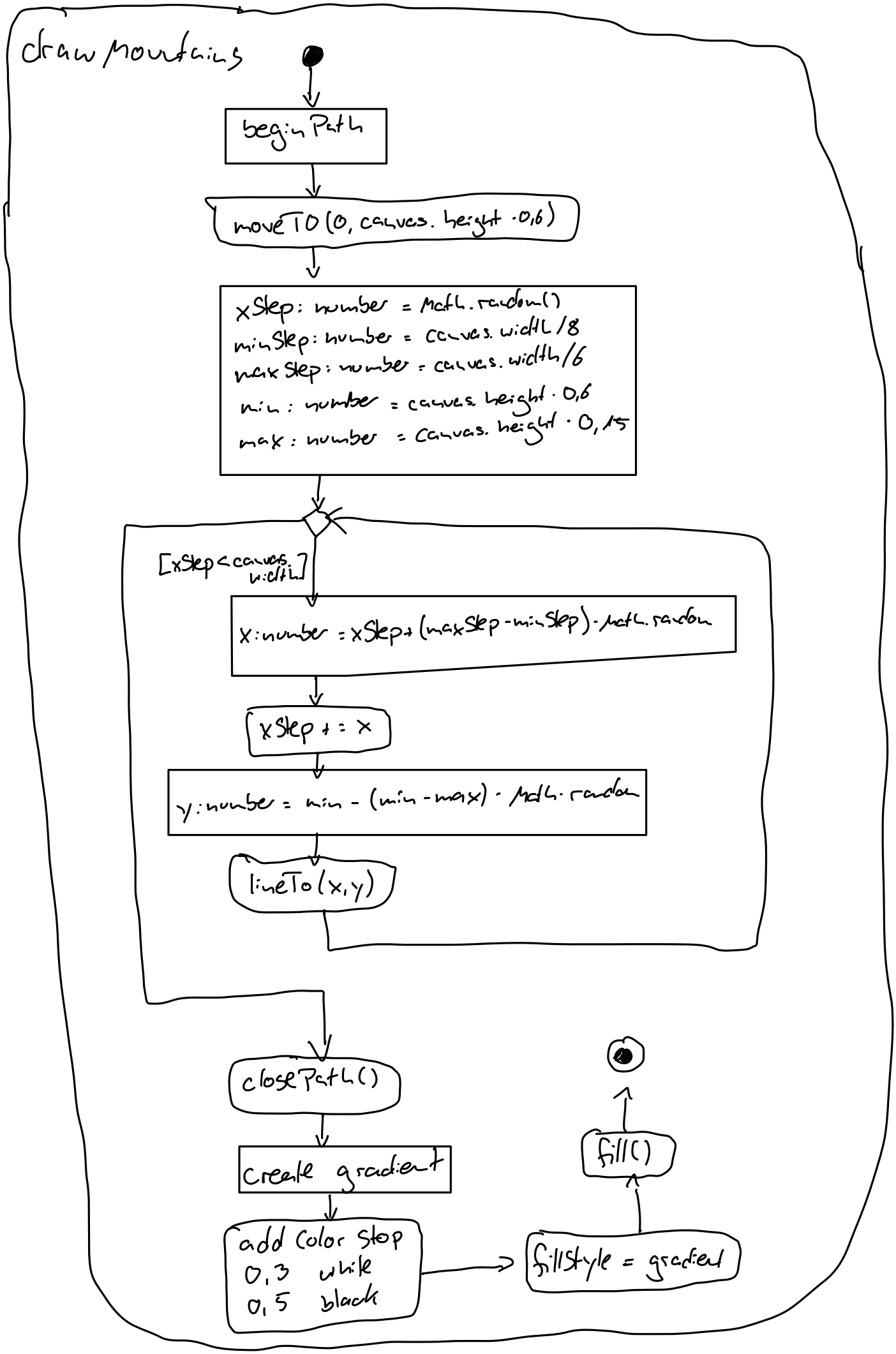


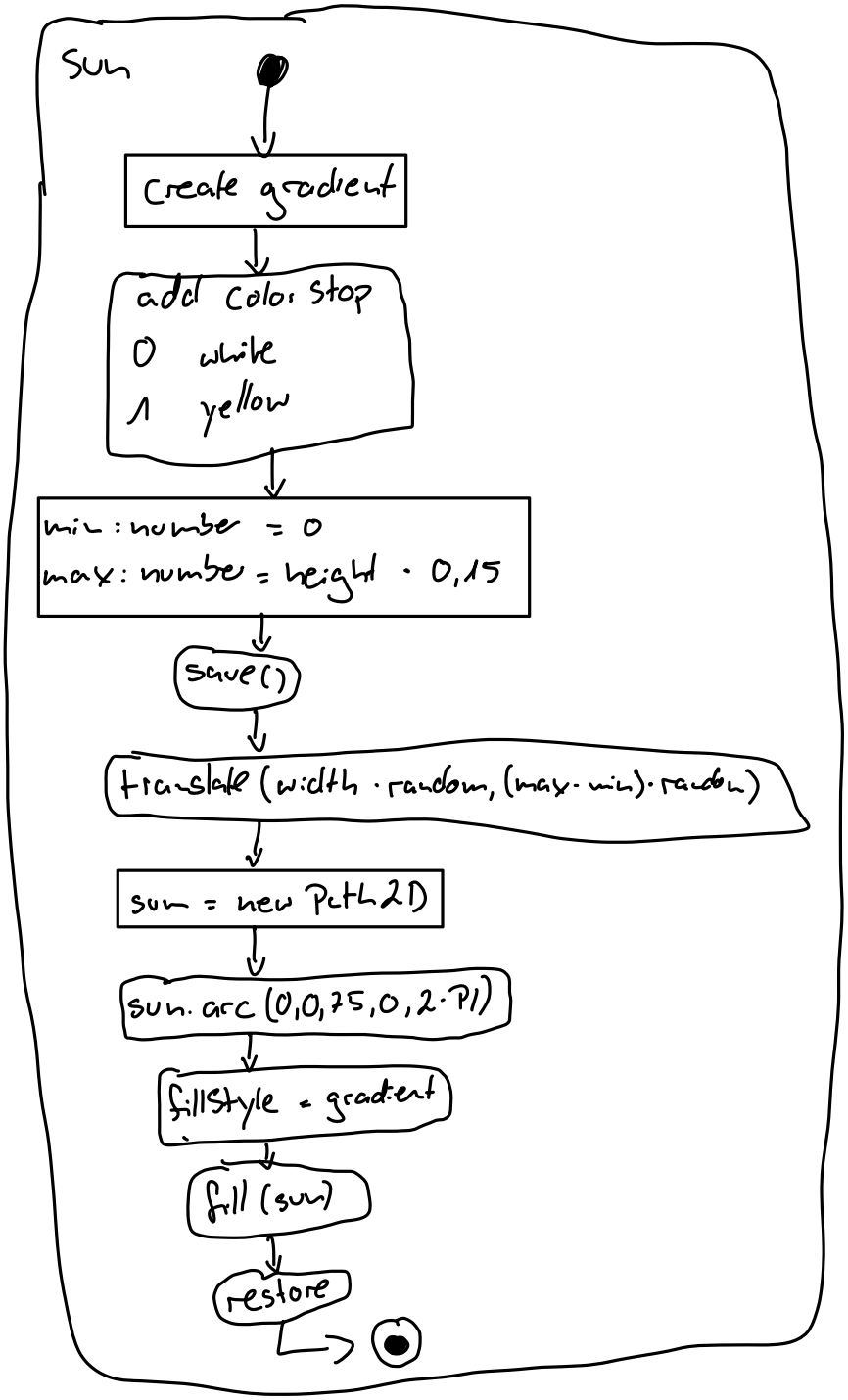


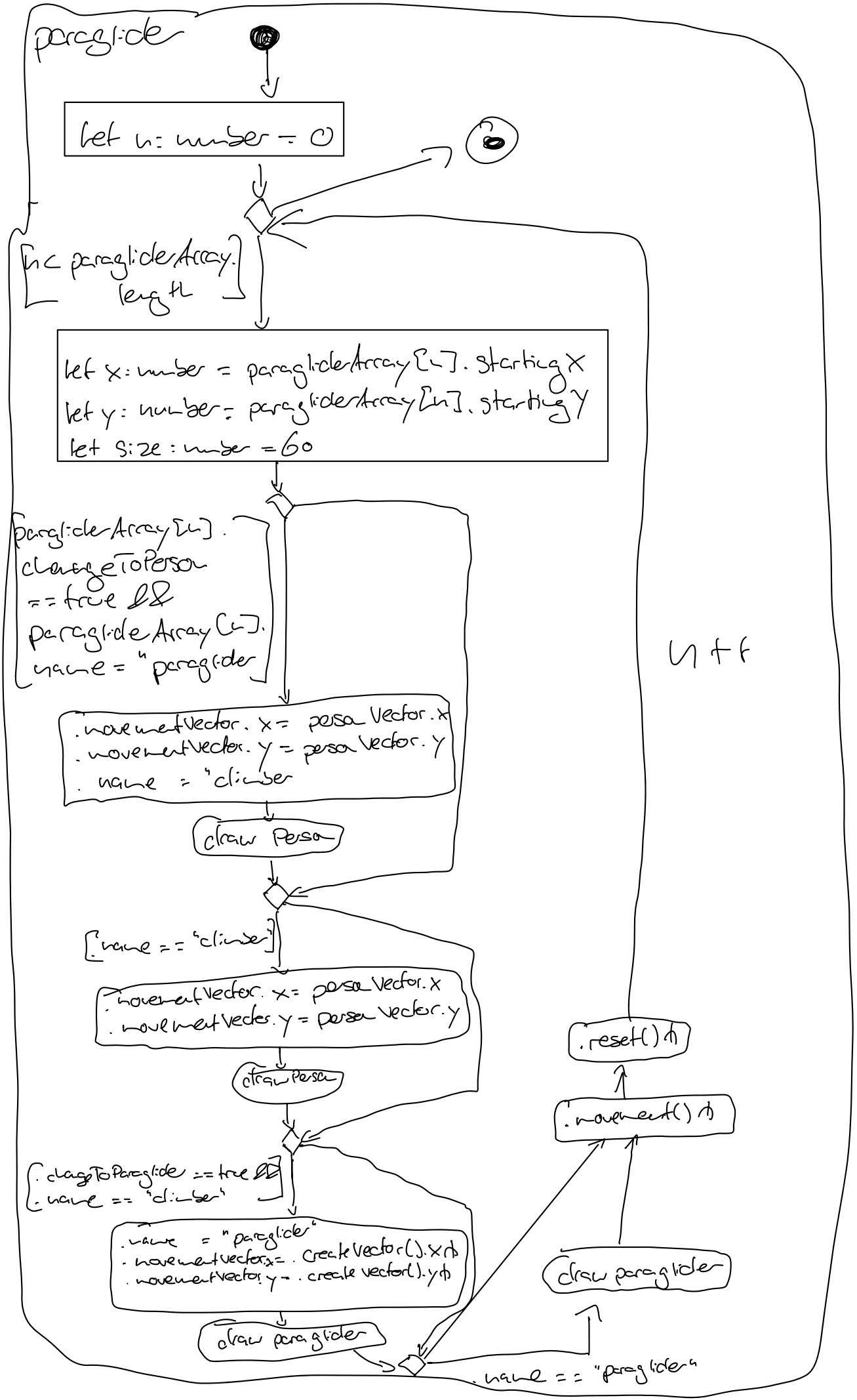


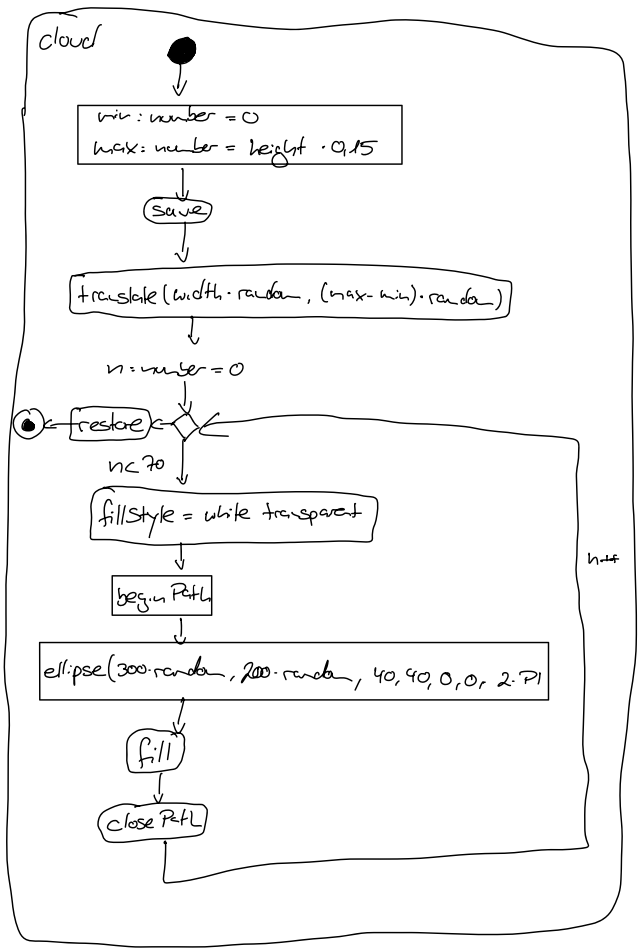

addEventListener on "load"
let crc2 : CanvasRenderingContext2D
let moveables : Moveable[] = []
let personVector : Vector = new Vector(0, -3)
let imgData : ImageData
load()
handleLoad()
handleLoad
create canvas
drawGround()
drawMountains()
cloud()
airfield()
shop()
imgData = crc2.getImageData(0,0,width, height)
let i : number = 0
[i<10]
let bug: Bug = new Bug("bug", width * random, height * random, "brown")
moveables.push(bug)
i++
[i<6]
let balloon: Balloon = new Balloon("balloon", width * random, height * random, randomColor())
moveables.push(balloon)
i++
[i<6]
let paraglider: Paraglider = new Paraglider(paraglider, width * random, height * random, randomColor())
moveables.push(paraglider)
i++
window.setInterval(animation, 50)

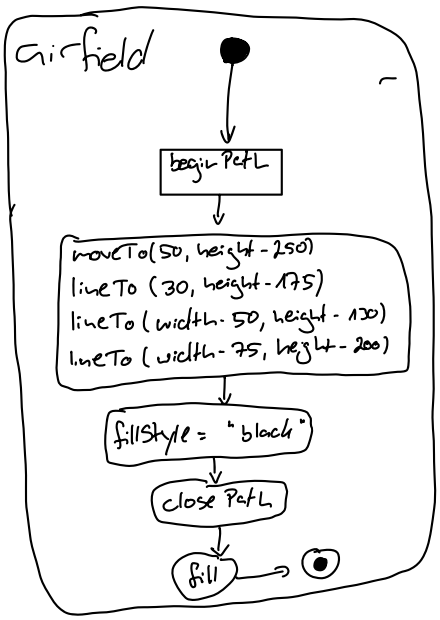

Animation
crc2.putImageData(imgData,0,0)
Sum
[n< moveables.length]
moveables.movement()
moveables.reset()
drawCrowd
create gradient
add Color Stops
0, blue
0.2, lightblue
0.6, gray
1, green
fillStyle = gradient
fillRect(0,0, canvas.width, canvas.height)
randomColor
let r: number = Math.random() * 256
let g: number = Math.random() * 256
let b: number = Math.random() * 256
"rgb(" + r + "," + g + "," + b + ")"

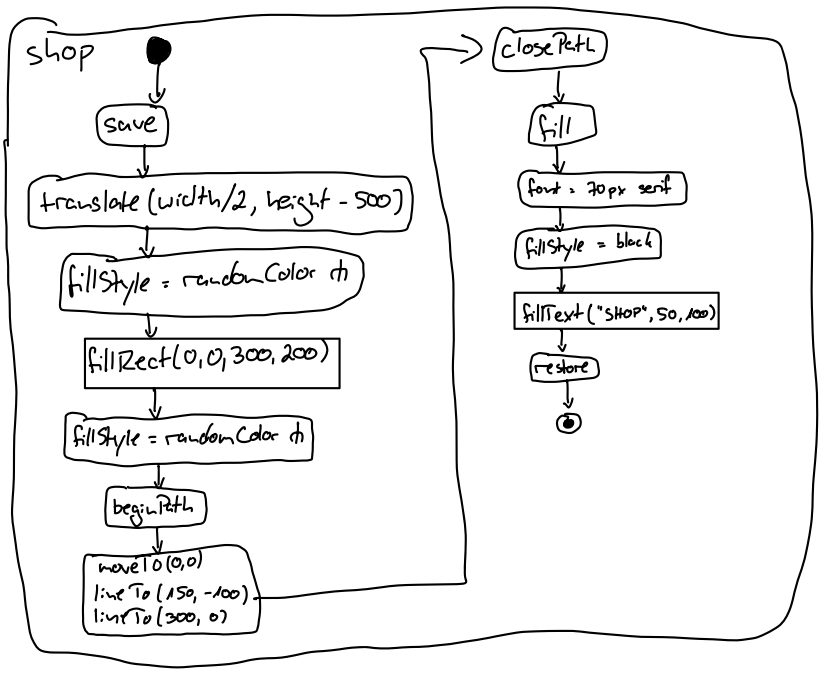

drawMountains
beginPath
moveTO(0, canvas.height ·0,6)
xStep: number = Math.random()
minStep: number = canvas.width/8
maxStep: number = canvas.width/6
min: number = canvas.height · 0,6
max: number = canvas.height · 0,15
[xStep < canvas.width]
x:number = xStep + (maxStep - minStep) · Math.random
xStep += x
y:number = min - (min - max) · Math.random
lineTo(x,y)
closePath()
create gradient
add Color Stop
0,3 white
0,5 black
fillStyle = gradient
fill()


Sun
Create gradient
add Color Stop
0 white
1 yellow
min : number = 0
max : numbe = height · 0,15
save()
translate (width · random, (max - min) · random)
sun = new Path2D
sun.arc (0,0,75,0,2·PI)
fillStyle = gradient
fill (sun)
restore


paraglider
let n: number = 0
n < paragliderArray.length
let x: number = paragliderArray[n].startingX
let y: number = paragliderArray[n].startingY
let size: number = 60
paragliderArray[n].changeToPerson == true && paragliderArray[n].name = "paraglider
.movementVector.x = personVector.x
.movementVector.y = personVector.y
.name = "climber
draw Person
.name == "climber"
.movementVector.x = personVector.x
.movementVector.y = personVector.y
draw Person
.changeToParaglide == true && .name == "climber"
.name = "paraglider"
.movementVector.x = .createVector().x/5
.movementVector.y = .createVector().y/5
draw paraglider
.name == "paraglider"
draw paraglider
.movement()
.reset()
n++


cloud
min : number = 0
max: number = height · 0.15
save
translate (width · random, (max - min) · random)
n : number = 0
restore
n < 70
fillStyle = white transparent
beginPath
ellipse (300 · random, 200 · random, 40, 40, 0, 0, 2 · PI
fill
closePath
n++


airfield
beginPath
moveTo (50, height - 250)
lineTo (30, height - 175)
lineTo (width - 50, height - 130)
lineTo (width - 75, height - 200)
fillStyle = "black"
closePath
fill


shop
save
translate (width/2, height - 500)
fillStyle = randomColor rh
fillRect (0, 0, 300, 200)
fillStyle = randomColor rh
beginPath
moveTo (0, 0)
lineTo (150, -100)
lineTo (300, 0)
closePath
fill
font = 70px serif
fillStyle = black
fillText ("SHOP", 50, 100)
restore