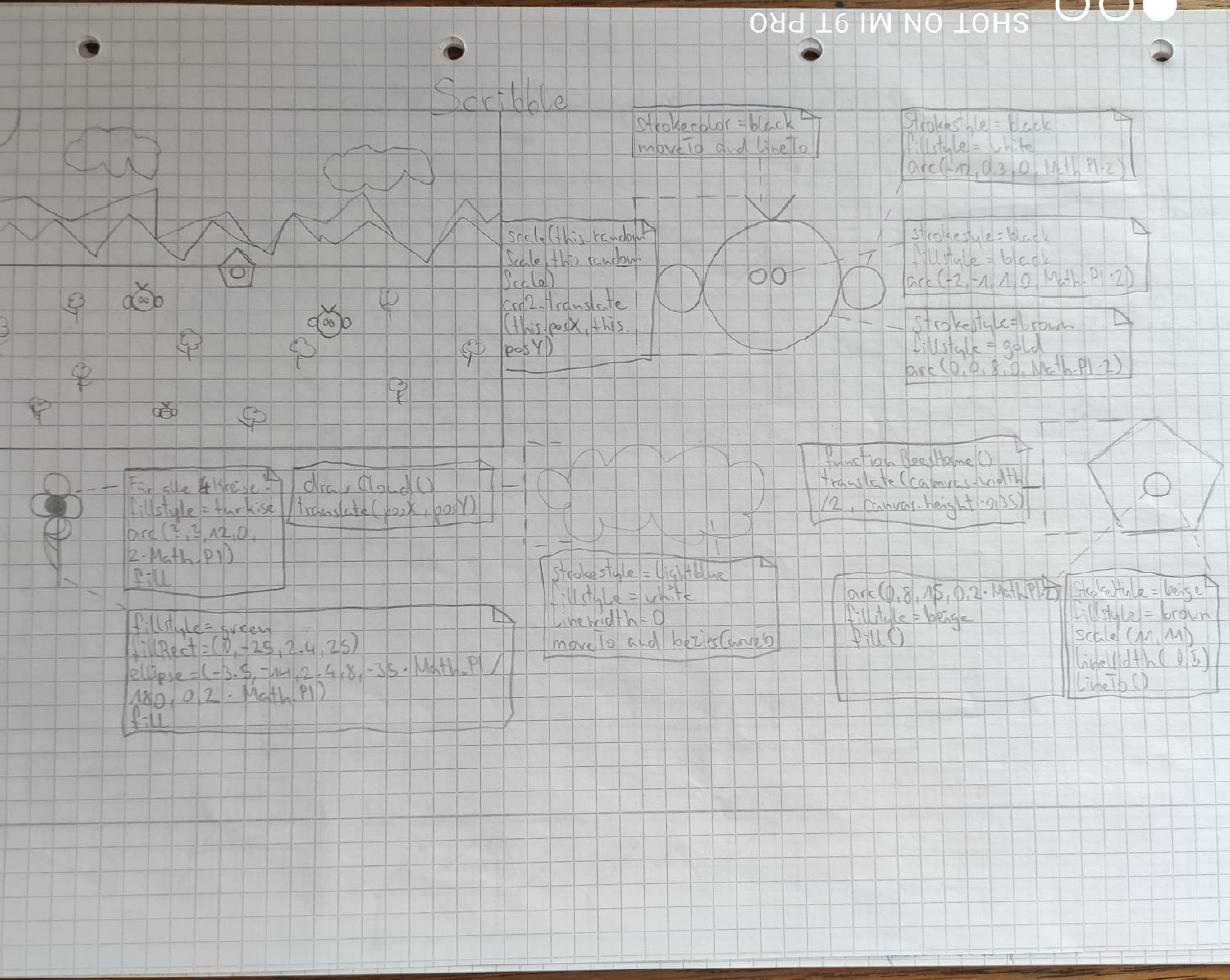
Landschaftsbild: Scribble


Scribble
Strokecolor = black
moveTo and lineTo
Strokestyle = black
fillstyle = white
arc(-12, 0, 3, 0, Math.PI·2)
scale(this.random
Scale, this.random
Scale)
crc2.translate
(this.posX, this.
posY)
strokestyle = black
fillstyle = black
arc(-2, -1, 1, 0, Math.PI·2)
strokestyle = brown
fillstyle = gold
arc(0, 0, 8, 0, Math.PI·2)
Für alle 4 Kreise:
fillstyle = turkise
arc(3, 3, 12, 0,
2·Math.PI)
fill
draw Cloud()
translate(posX, posY)
function BeesHome()
translate(canvas.width
/2, canvas.height·0.35)
fillstyle = green
fillRect = (0, -25, 2.4, 25)
ellipse = (-3.5, -14, 2.4, 8, -35·Math.PI /
180, 0, 2·Math.PI)
fill
strokestyle = lightblue
fillstyle = white
linewidth = 0
moveTo and bezierCurveTo
arc(0, 8, 15, 0, 2·Math.PI)
fillstyle = beige
fill()
Strokestyle = beige
fillstyle = brown
scale(11, 11)
lineWidth(0.5)
lineTo()

Class Diagramms

Flower

xPos: number

yRandomMin: number

yRandomMax: number

nectarValue: number = Math.floor...

yPos: number

randomScale: number = 0.5 + Math.random...

nectarLength: number = 5

nectarCounter = number = 0

---

constructor(_xPos: number, _yRandomMin: number, _yRandomMax: number)

public draw(): void

public updateNectar(): void

Flower 1

constructor(_xPos: number, _yRandomMin: number, _yRandomMax: number)

public draw(): void

public updateNectar(): void

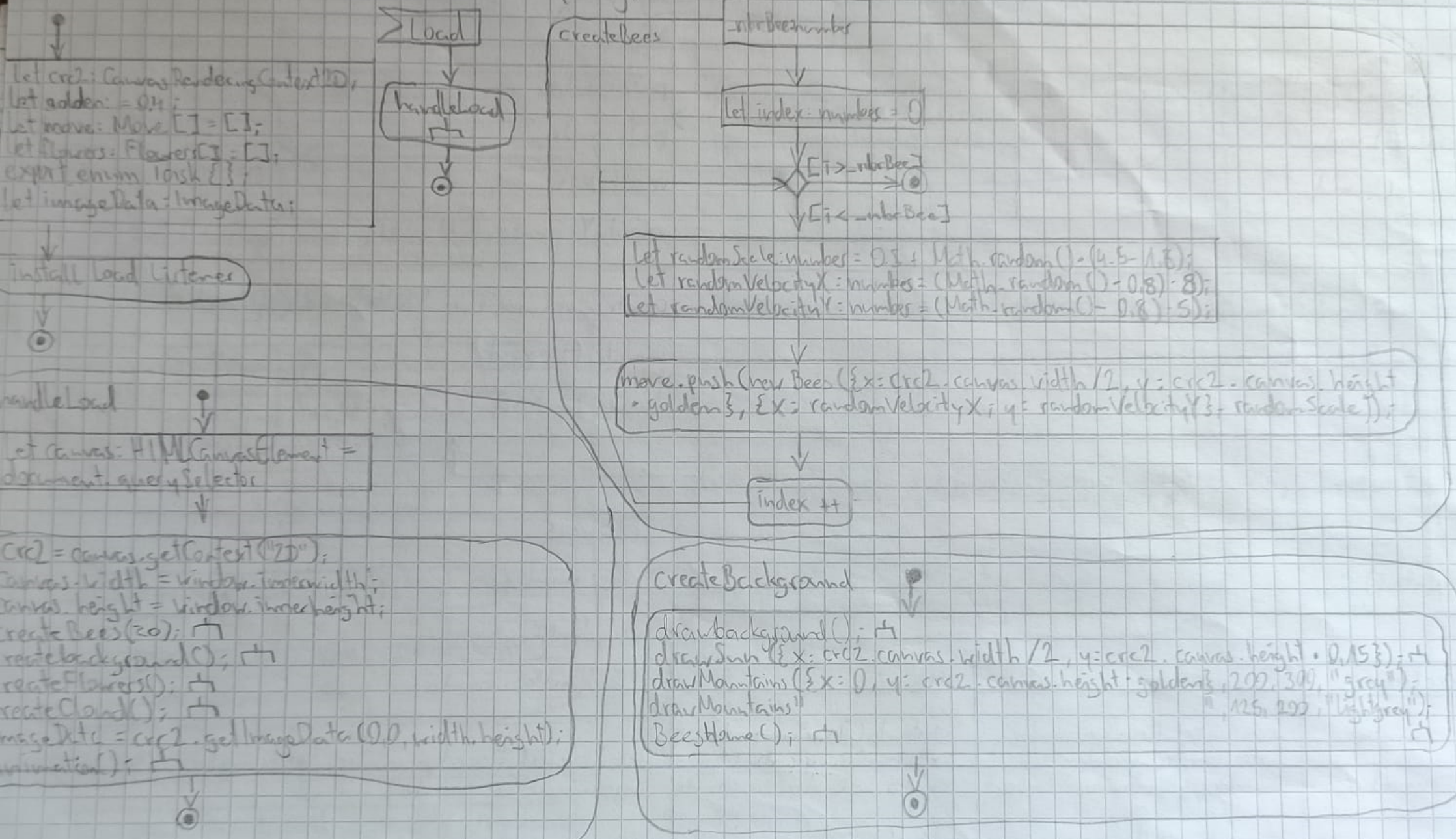
Flower 2

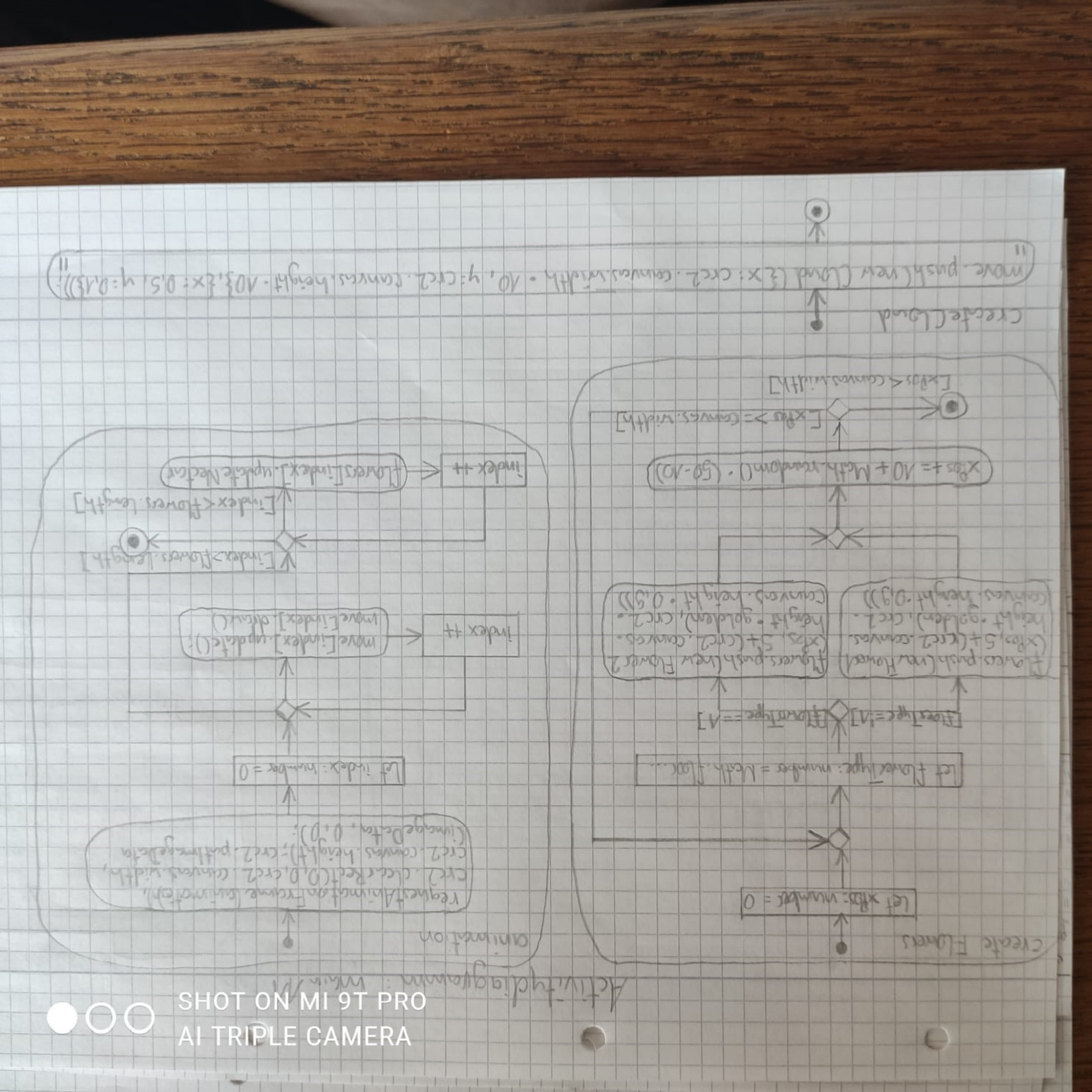
constructor(_xPos: number, _yRandomMin: number, _yRandomMax: number)

public draw(): void

public updateNectar(): void

Activitydiagram: main.ts





Activitydiagram: background 1/1

interface Vector

drawBackground

Let gradient: CanvasGradient =
crc2.createLinearGradient

gradient Eigenschaften

crc2.fillStyle = gradient
crc2.fillRect()

drawMountains

- position: Vector;
- stepMin: number
- stepMax: number
- color: string

Let stepMin: number = 50;
let stepMax: number = 110;
let x: number = 0;

crc2.save();
crc2.translate(-position.x, -position.y);
crc2.beginPath();
crc2.moveTo();
crc2.lineTo();

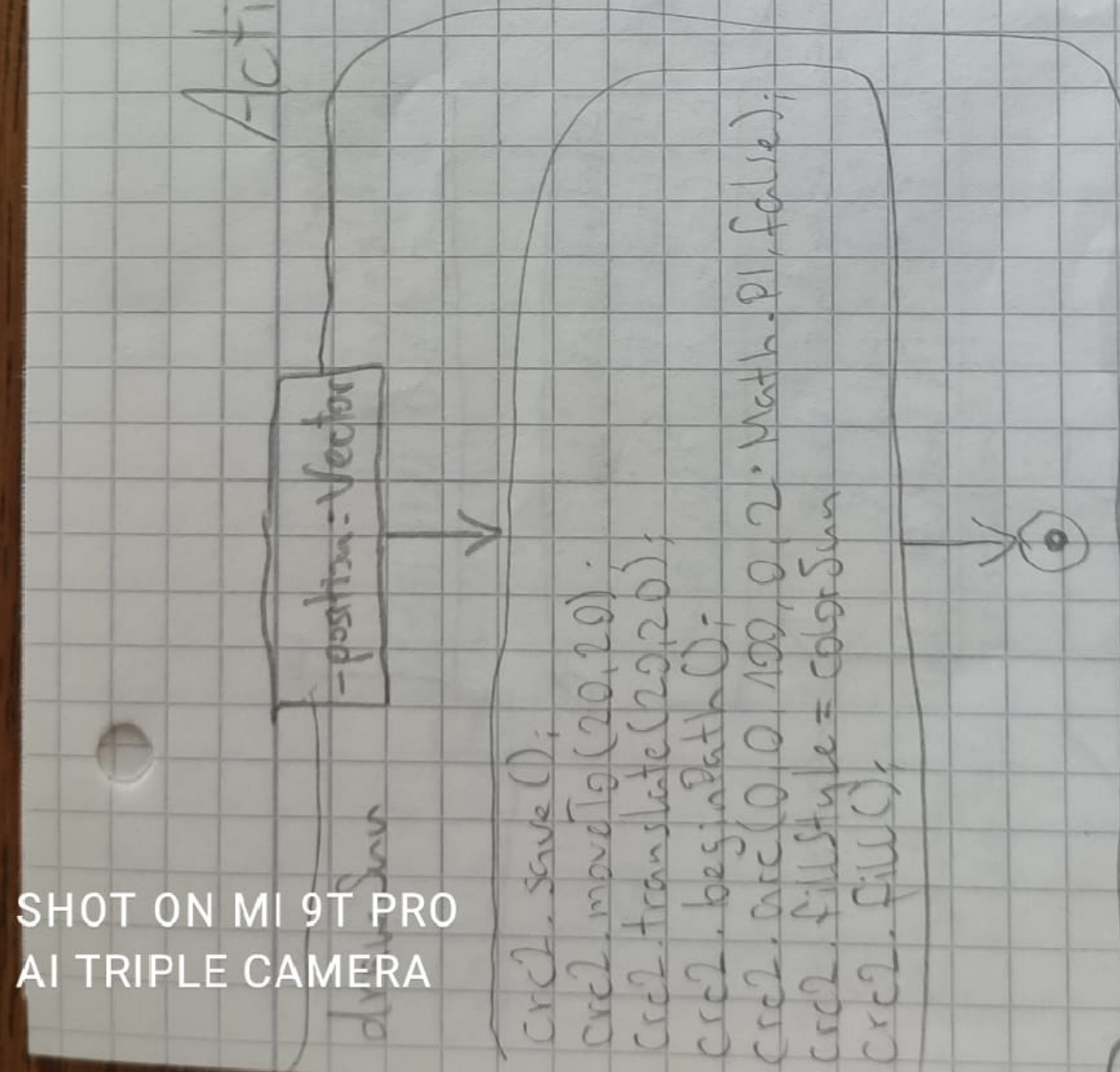
x += stepMin + Math.random() * (stepMax);
let y: number = -_height - Math.random() * (_width - _height);

lineTo(x, y)

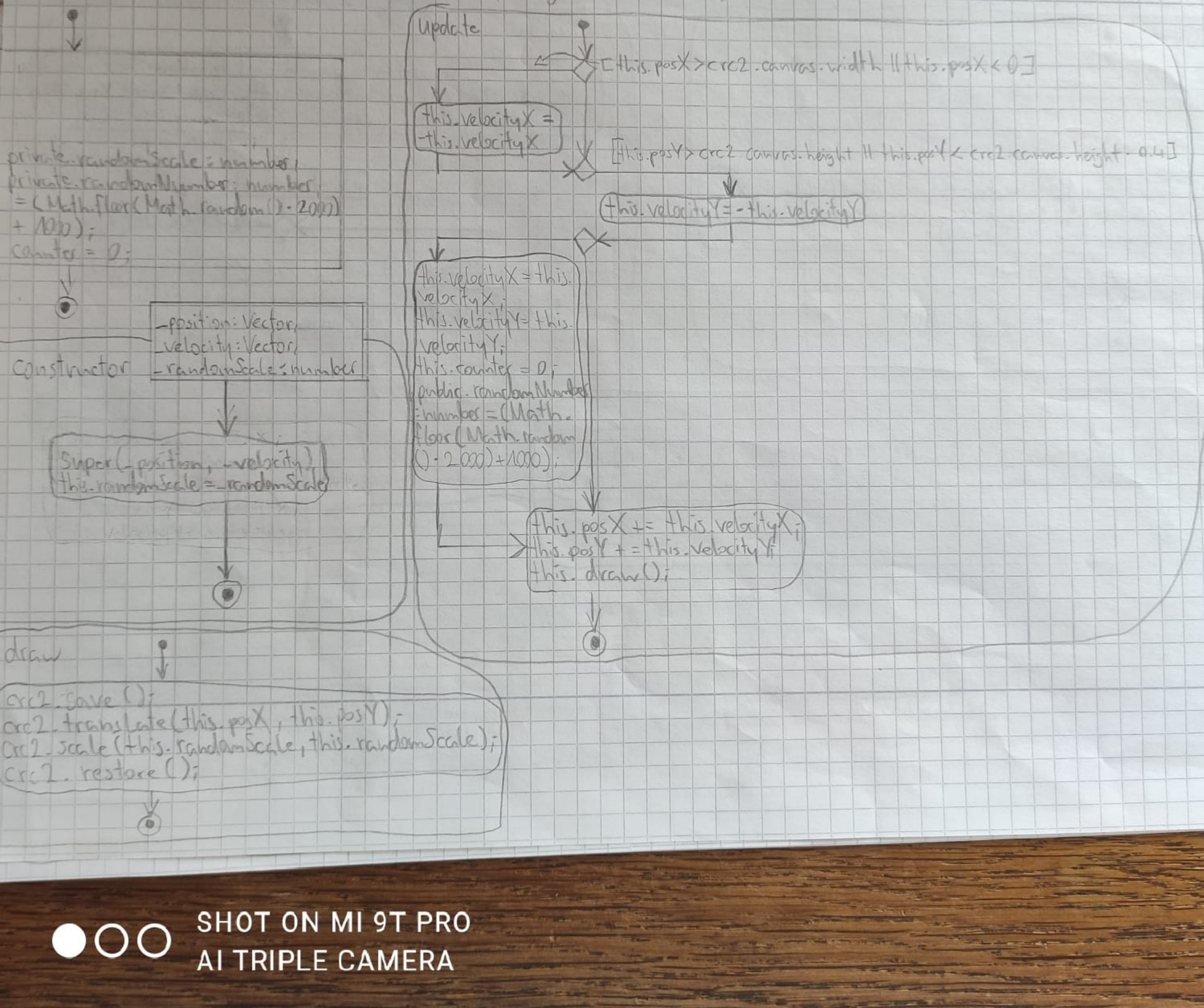
[x < crc2.canvas.width]

lineTo(x, 0);
closePath();
crc2.fillStyle = _color;
crc2.fill();
crc2.restore();

Activity diagram: background



Activitydiagram: bees11



```
Bee.Home
ctc2.save();
ctc2.translate(ctc2.canvas.width/2, ctc2.canvas.height-0.85);
ctc2.scale(M,M)
ctc2.lineWidth = 0.5;
ctc2.strokeStyle = "beige"
Beeshive kind gazebo
ctc2.restore();
```

```
Super(-xPos,-yRandomMin,
      -yRandomMax)
```

```
public xPos: number;
public yRandomMax: number;
public yRandomMin: number;
```

```
Constructor(
    -xPos: number;
    -yRandomMax: number;
    yRandomMin: number;
```

```
Architectural Icegrow: Flowers[]
draw
Let RandomScale: number = 0.3 + Math.random() * (A3-0.5);
Let y: number = this.yRandomMin + Math.random() * (this.yRandomMax-this.yRandomMin);
ctc2.save()
ctc2.translate(this.xPos, y);
ctc2.scale(RandomScale, RandomScale);
[flower Type == A] [flower Type != A]
draw Flower A
draw Flower 2
```