

Scribble



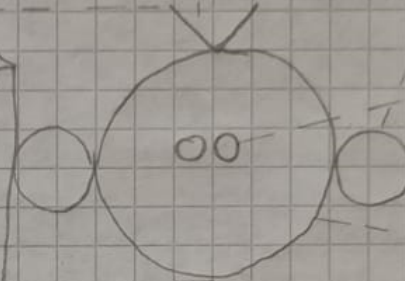
Strokecolor = black
moveTo and lineTo

Strokestyle = black
fillstyle = white
arc(-12, 0.3, 0, Math.PI * 2)

scale(this.random
Scale, this.random
Scale)
arc2.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 = turquoise
arc(3, 3, 12, 0,
2 * Math.PI)
fill

draw Cloud()
translate(posX, posY)

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 bezierCurves

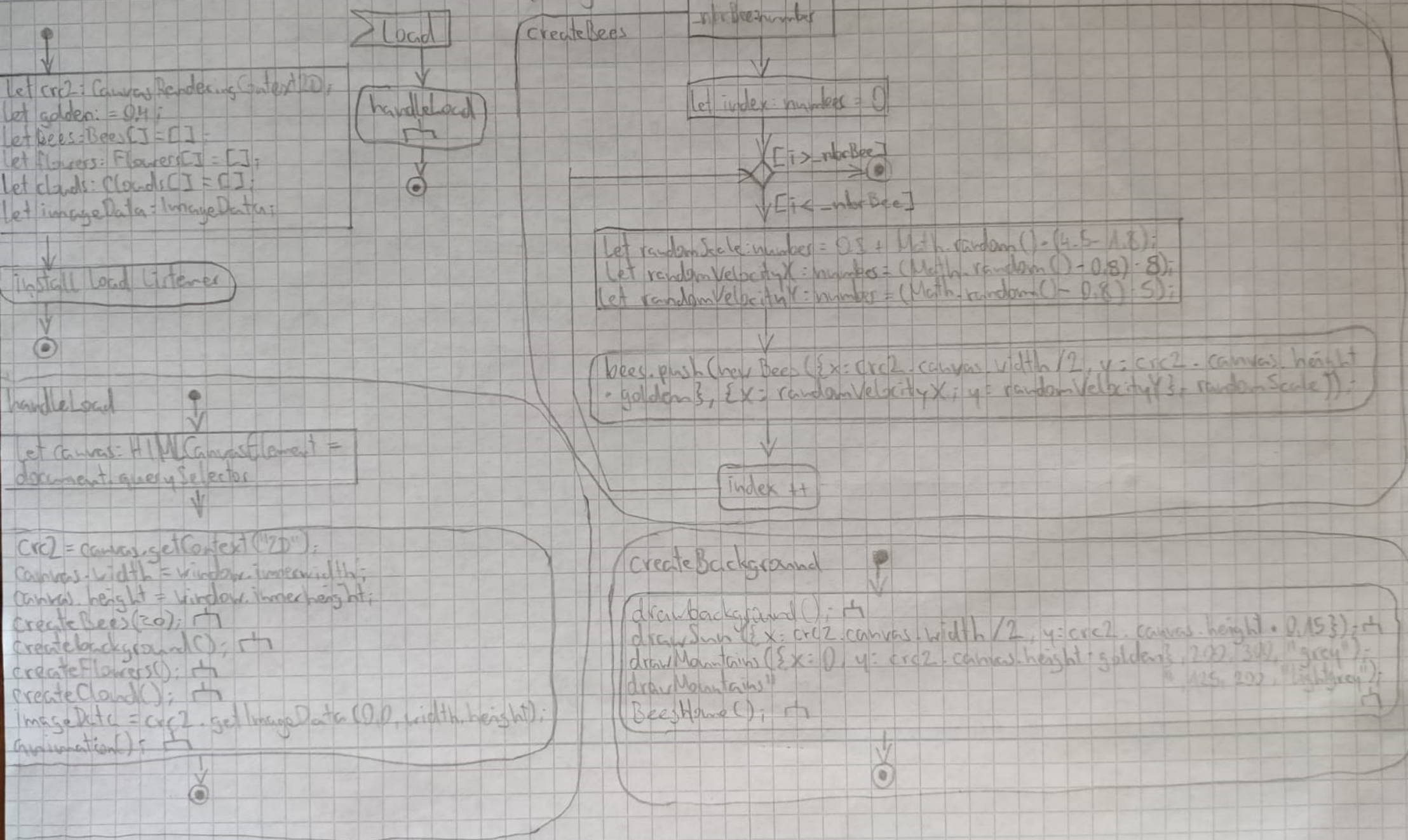
function BeechHome()
translate(canvas.width
/ 2, canvas.height * 0.35)



arc(0.8, 15, 0.2 * Math.PI * 2)
fillstyle = beige
fill()

strokestyle = beige
fillstyle = brown
scale(11, 11)
linewidth(1.5)
lineTo()

Activity diagram: main 09



Activity diagram: main 09

Create Flowers

Let xPos: number = 0;



Flowers.push(new Flowers(Math.floor(Math.random() * 2) + 1, xPos, 5 + (crc2.canvas.height * golden), crc2.canvas.height * 0.8));
xPos += 10 + Math.random() * (50 - 10);

[xPos < canvas.width]

[xPos >= canvas.width]

Create Cloud

clouds.push(new Cloud(x: crc2.canvas.width * 5, y: crc2.canvas.height * 0.1));
clouds.push(new Cloud(x: crc2.canvas.width * 1, y: crc2.canvas.height * 0.3));



animation

requestAnimationFrame(Animation);
crc2.clearRect(0, 0, crc2.canvas.width, crc2.canvas.height);
crc2.putImageData(imageData, 0, 0);

Let index: number = 0;



index++ ← bees[index].update();

Let index: number = 0;

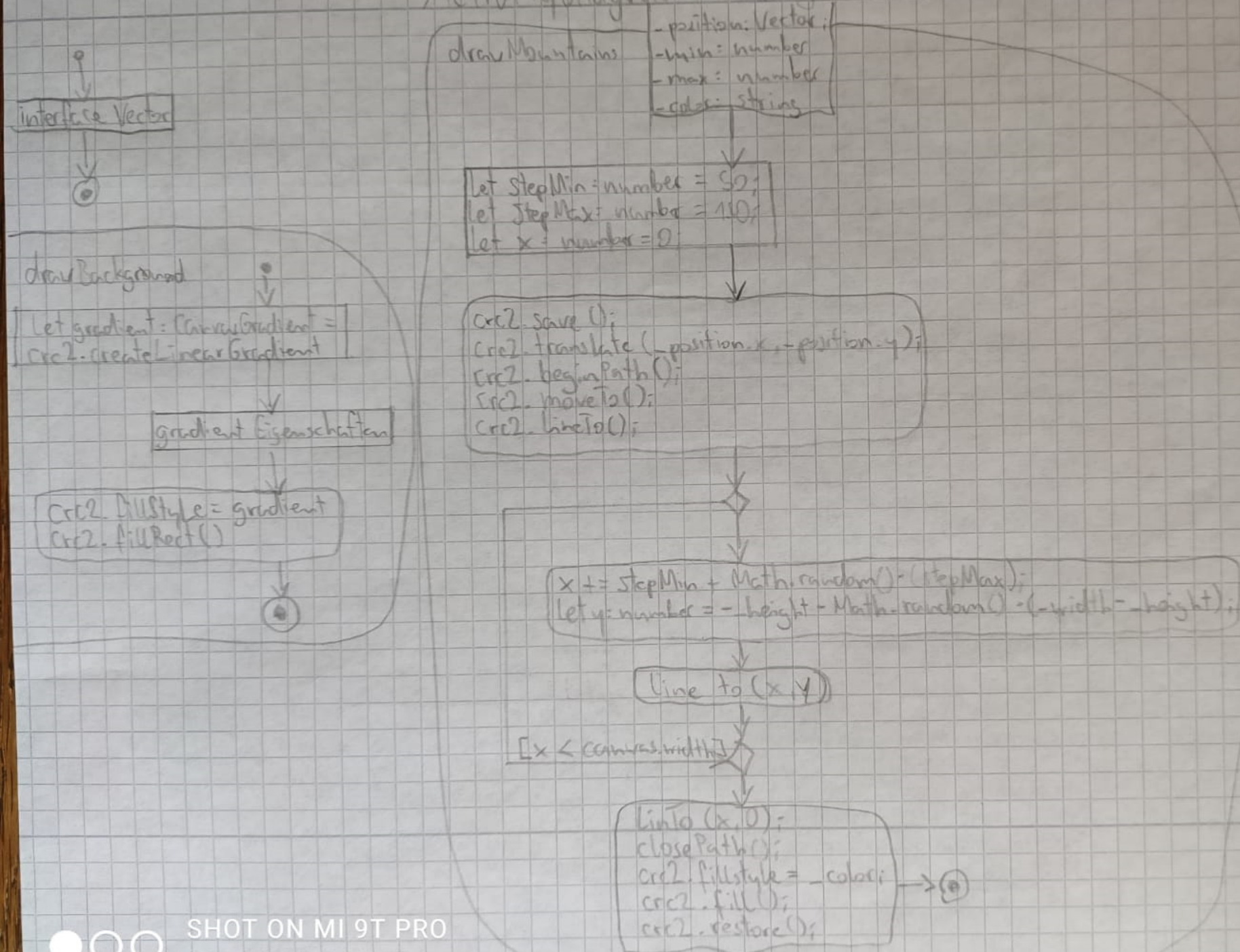
[index > clouds.length]

[index < clouds.length]

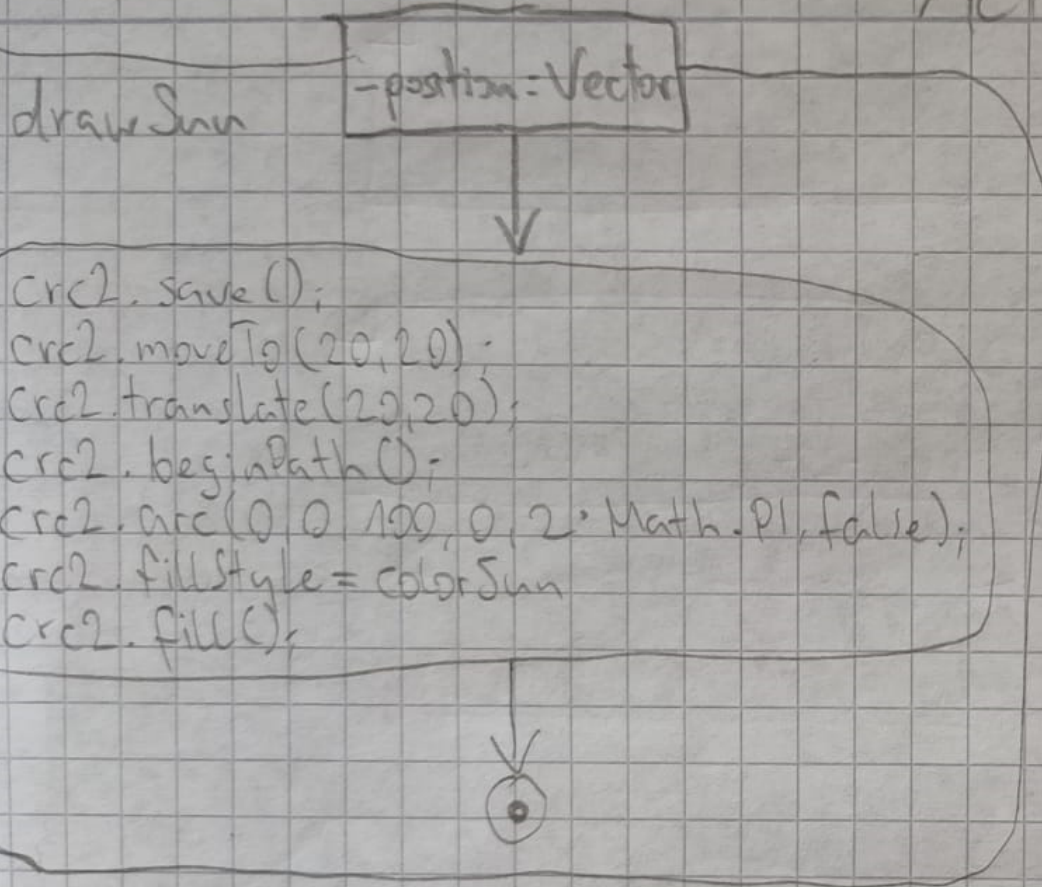
index++

clouds[index].update();

Activitydiagram: background 09



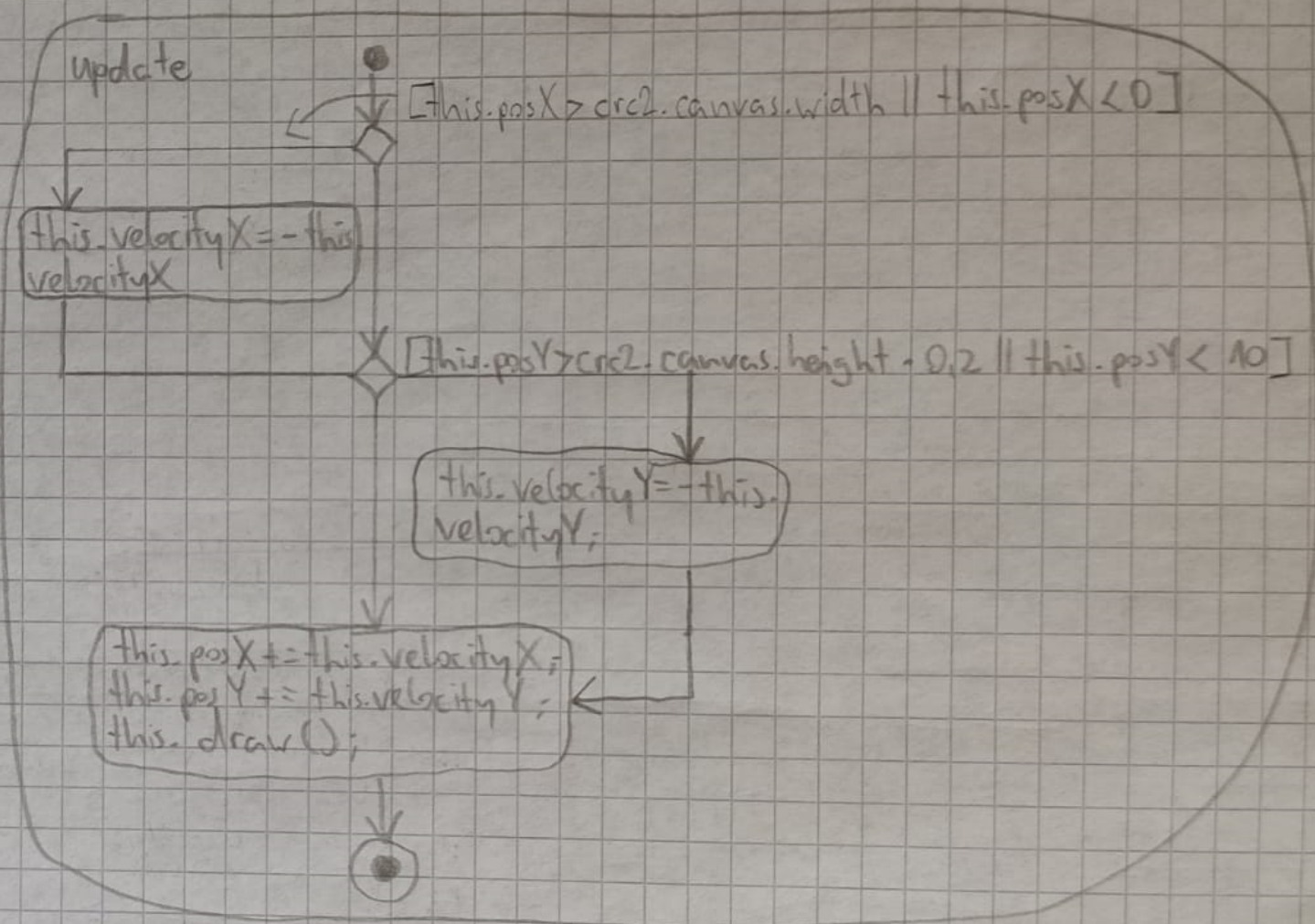
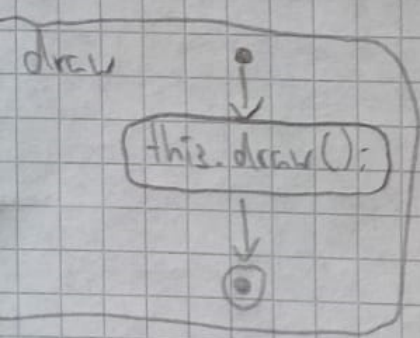
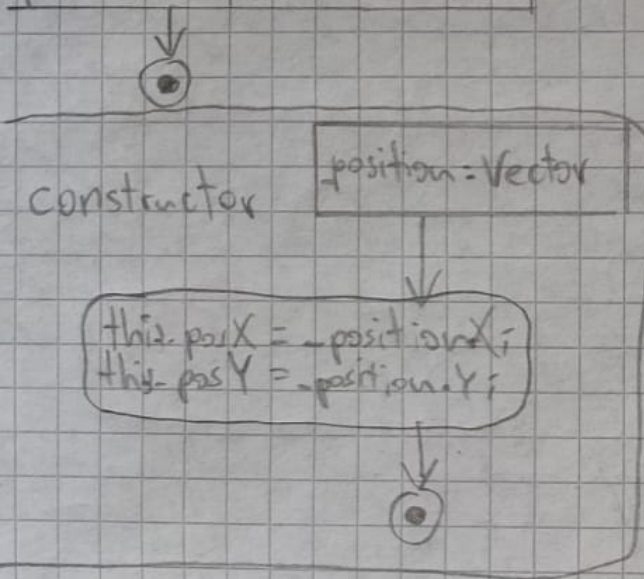
Activity diagram: background



Activity diagram: cloud 29

```

public posX: number;
public posY: number;
public velocityX: number = 0.9;
public velocityY: number = 0.1;
    
```



Activity diagram: bees09

```

public posX: number;
public posY: number;
public velocityX: number;
public velocityY: number;
public randomScale: number;
public randomNumber: number;
= (Math.floor(Math.random() * 2000)
+ 1000);
counter = 0;
    
```

Constructor

```

- position: Vector;
- velocity: Vector;
- randomScale: number
    
```

```

this.posX = position.x;
this.posY = position.y;
this.randomScale = randomScale;
this.velocityX = velocity.x;
this.velocityY = velocity.y;
    
```

draw

```

crc2.save();
crc2.translate(this.posX, this.posY);
crc2.scale(this.randomScale, this.randomScale);
crc2.restore();
    
```

update

```

[!this.posX > crc2.canvas.width || !this.posX < 0]
    
```

```

this.velocityX =
-this.velocityX
    
```

```

[!this.posY > crc2.canvas.height || !this.posY < crc2.canvas.height - 0.4]
    
```

```

this.velocityY = -this.velocityY
    
```

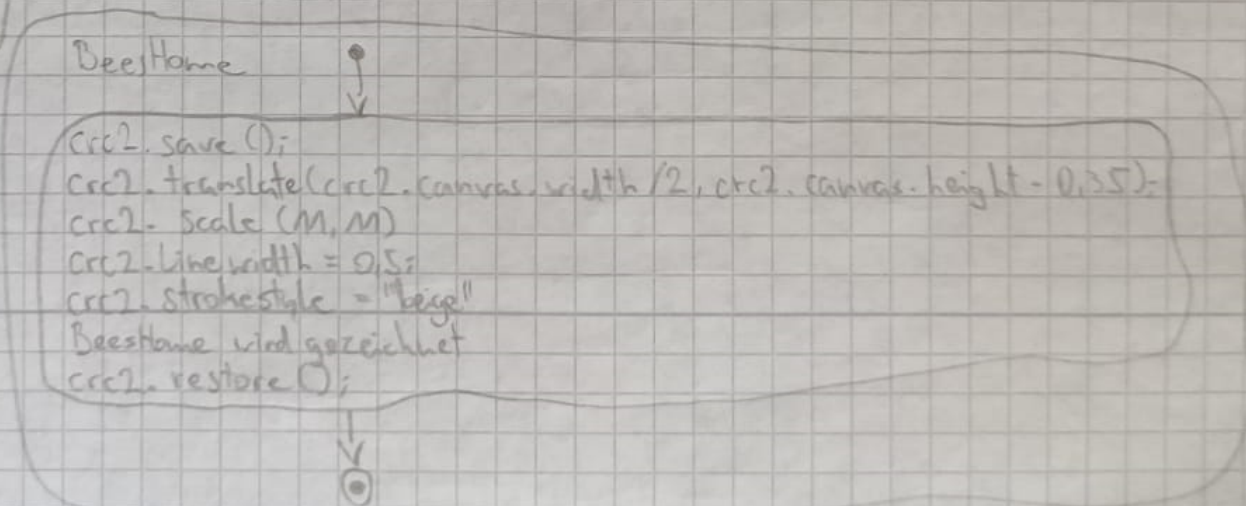
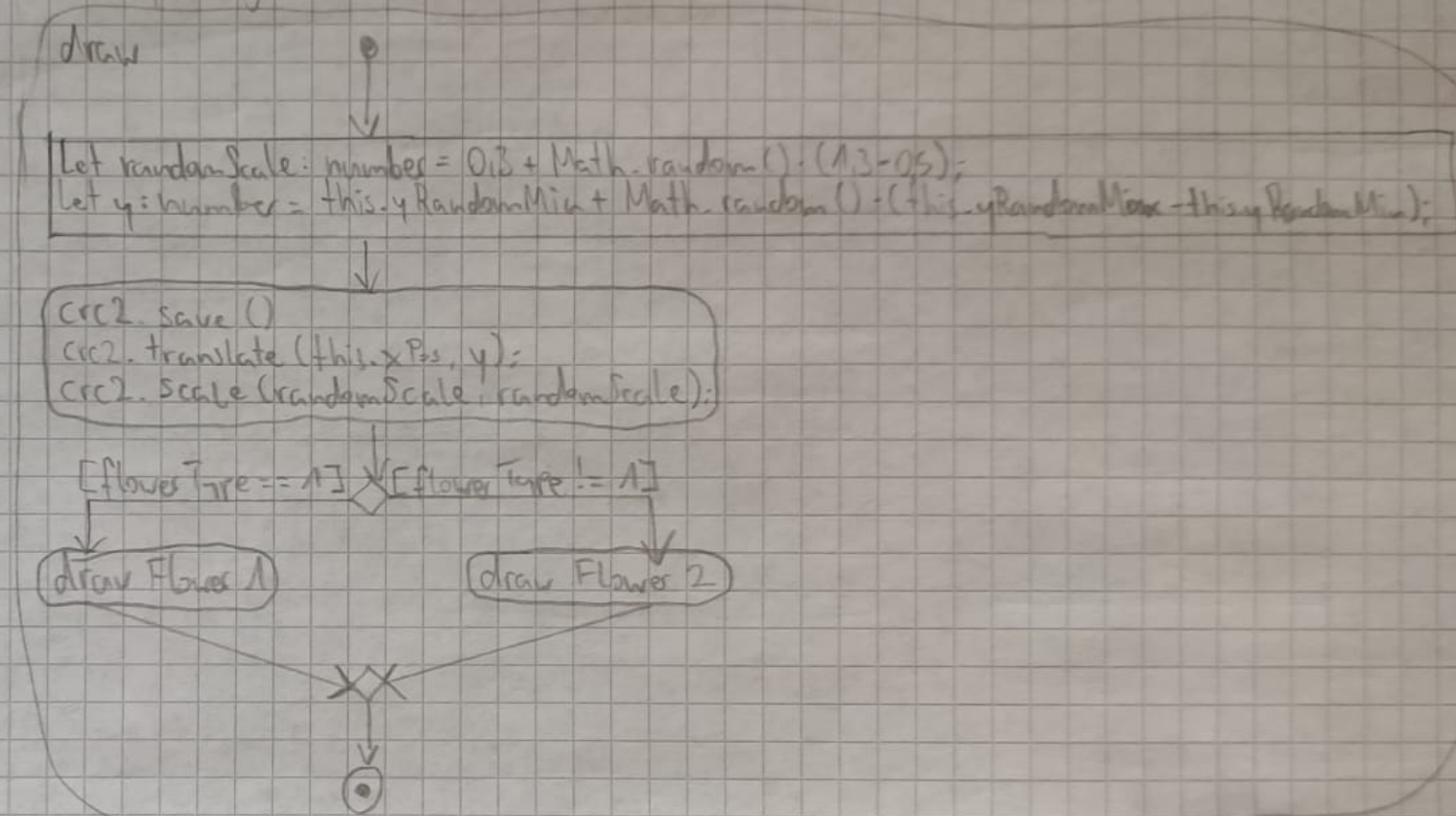
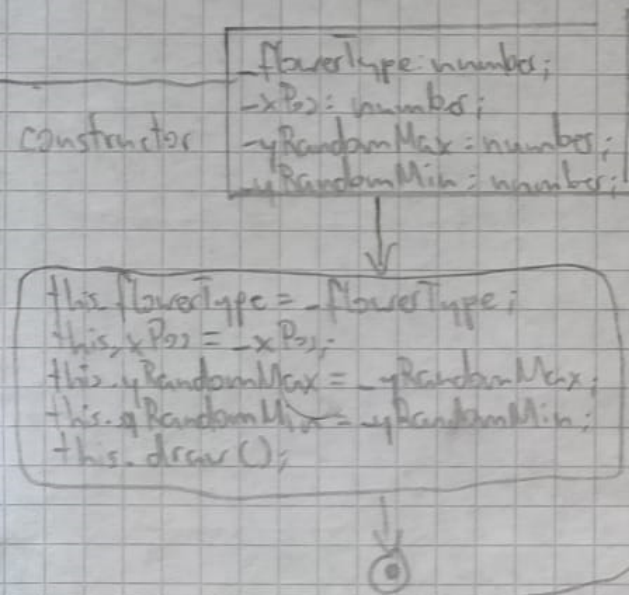
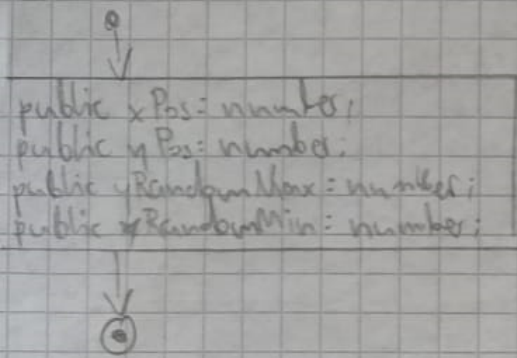
```

this.velocityX = this.
velocityX;
this.velocityY = this.
velocityY;
this.counter = 0;
public randomNumber
: number = (Math.
floor(Math.random
() * 2000) + 1000);
    
```

```

this.posX += this.velocityX;
this.posY += this.velocityY;
this.draw();
    
```

Activity diagram: flowers09



Class Diagrams

Bees

posX: number;
posY: number;
velocityY: number;
velocityX: number;
randomNumber: number = (Math.floor(Math.random() * 2000) + 1000);
randomScale: number;
Counter: number;

constructor (-position: Vector, -velocity: Vector, -randomScale: number)

draw()

update()

Cloud

posX: number;
posY: number;
velocityY: number = 0.1;
velocityX: number = 0.9

constructor (-position: Vector)

draw();

update();

Vector

X: number;