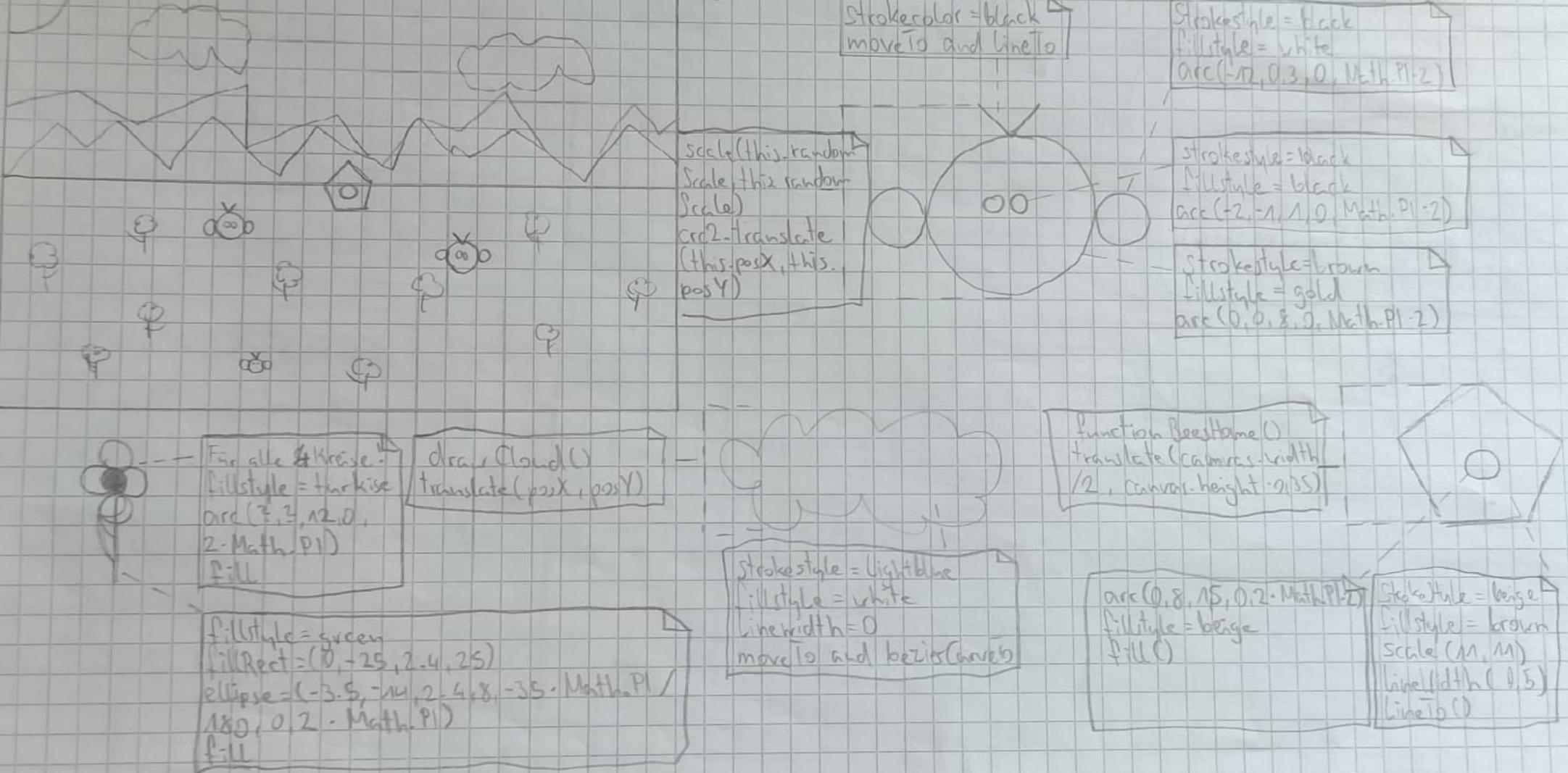
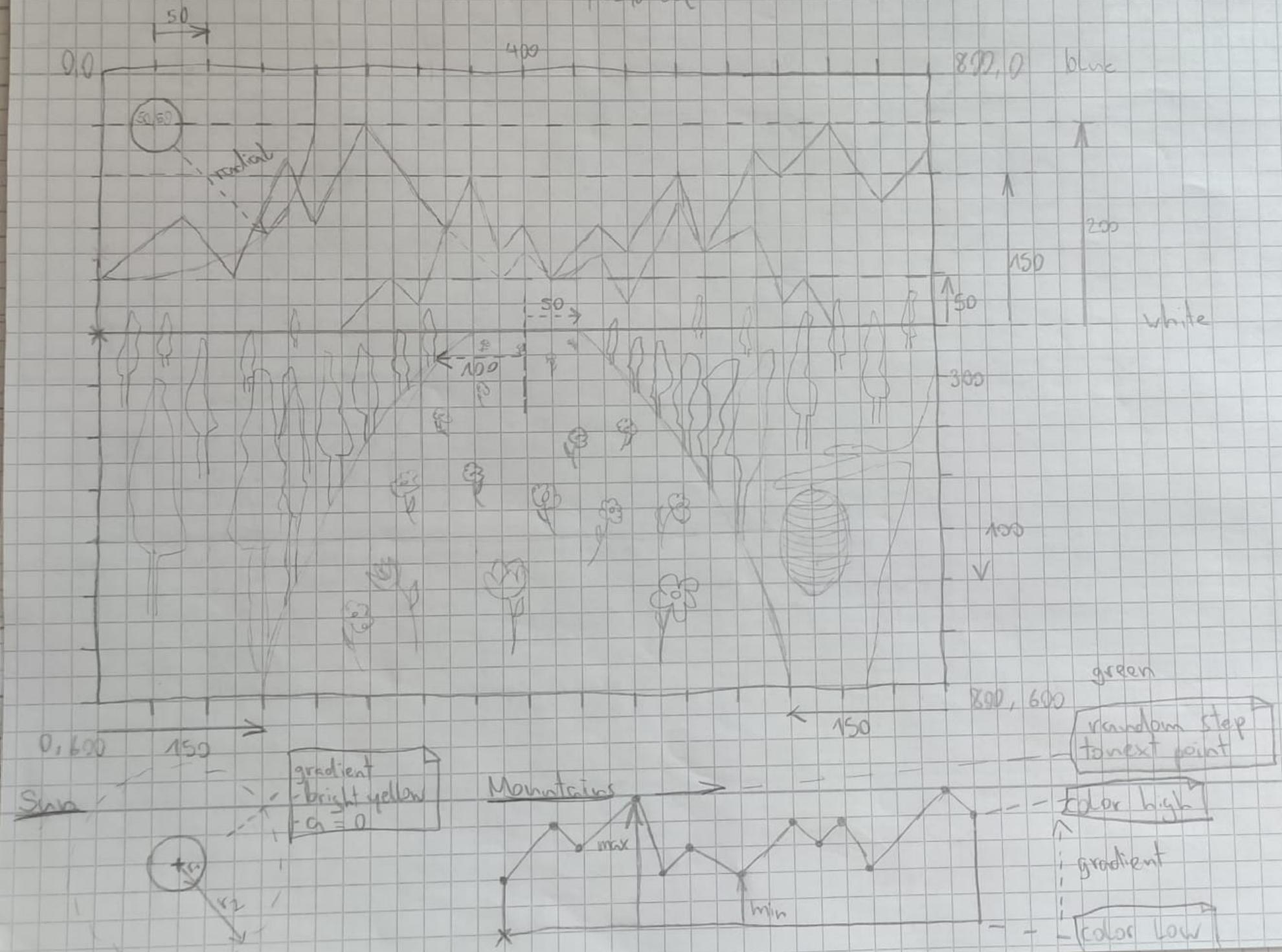


# Scribble



# Landschaftsbild: Scribbler



SHOT ON MI 9T PRO  
AI TRIPLE CAMERA

# Class Diagrams

Clouds

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

update(): void

draw(): void

Bee

randomNumber: number = (Math.floor(Math.random() \* 2000) + 1000);

randomScale: number;

counter: number;

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

draw()

update()

Vector

x: number;

y: number;

Move

posX: number,

posY: number,

velocityX: number,

velocityY: number,

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

update(): void

draw(): void

# Class Diagrams

● OO

AI TRIPLE CAMERA  
SHOT ON MI 9T PRO

FlowerM

---

-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

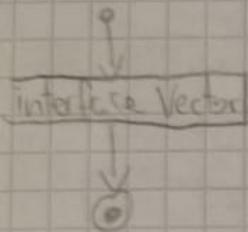
---

constructor(-xPos: number, -yRandomMin: number, -yRandomMax: number)

public draw(): void

public updateNectar(): void

## Activity diagram: background10



drawBackground

```
let gradient: CanvasGradient =  
  crc2.createLinearGradient()
```

gradient Eigenschaften

```
crc2.fillStyle = gradient  
crc2.fillRect()
```

```
- position: Vector;  
- min: number  
- max: 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);
```

Line to (x, y)

x &lt; canvas.width

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

# Activity diagram: background10

drawSun

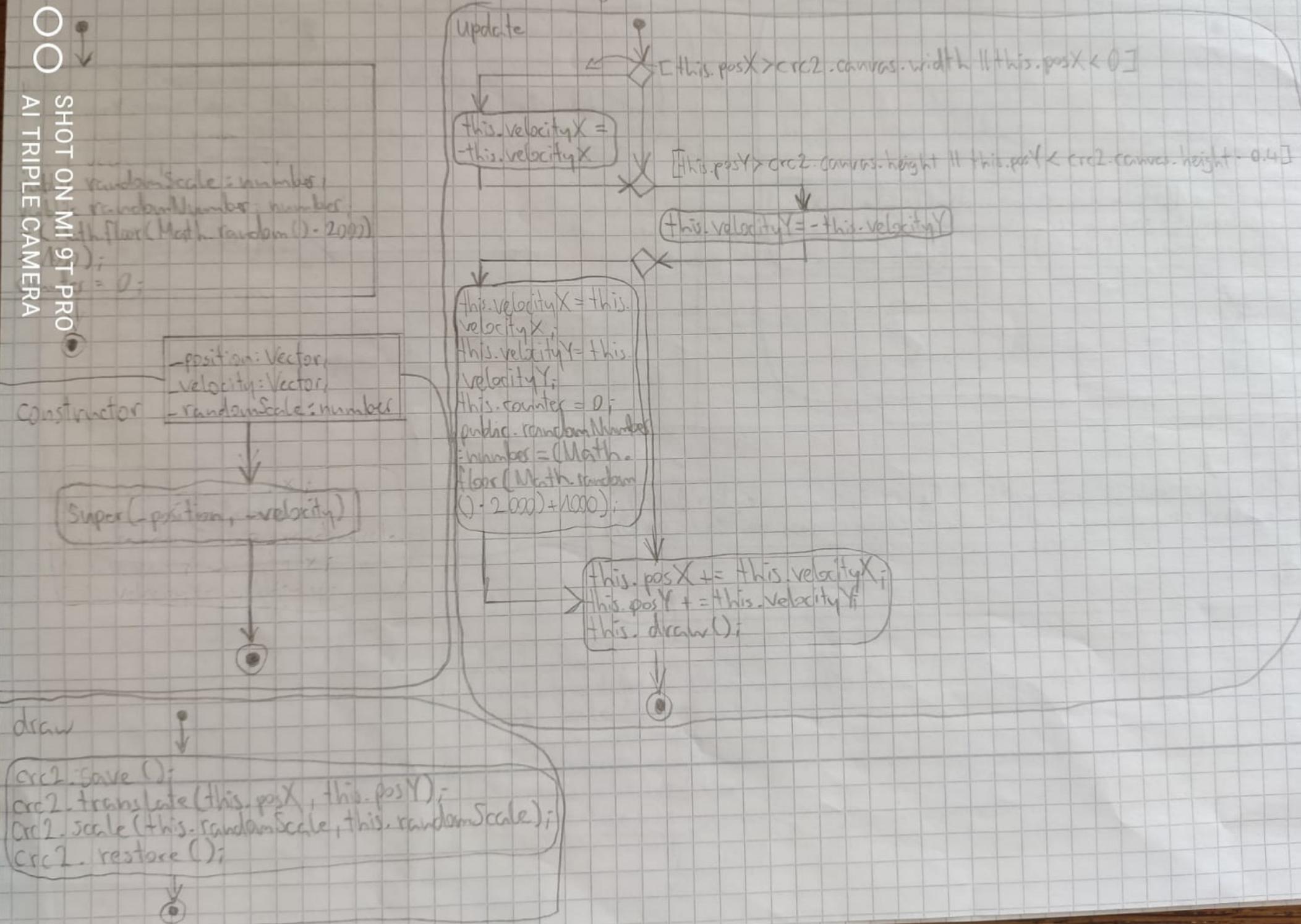
-position: Vector



```
crc2.save();
crc2.moveTo(20,20);
crc2.translate(20,20);
crc2.beginPath();
crc2.arc(0,0,100,0,2*Math.PI, false);
crc2.fillStyle = colorSun;
crc2.fill();
```

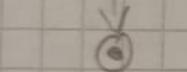


# Activity diagram: bsp10



# Activity diagram: Flowers10

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



constructor

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

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

draw

```
Let randomScale: number = 0.3 + Math.random() * (1.3 - 0.3);  
Let y: number = this.yRandomMin + Math.random() * (this.yRandomMax - this.yRandomMin);
```

```
crc2.save();  
crc2.translate(this.xPos, y);  
crc2.scale(randomScale, randomScale);
```

[flowerType == 1] [flowerType != 1]

draw Flower 1

draw Flower 2

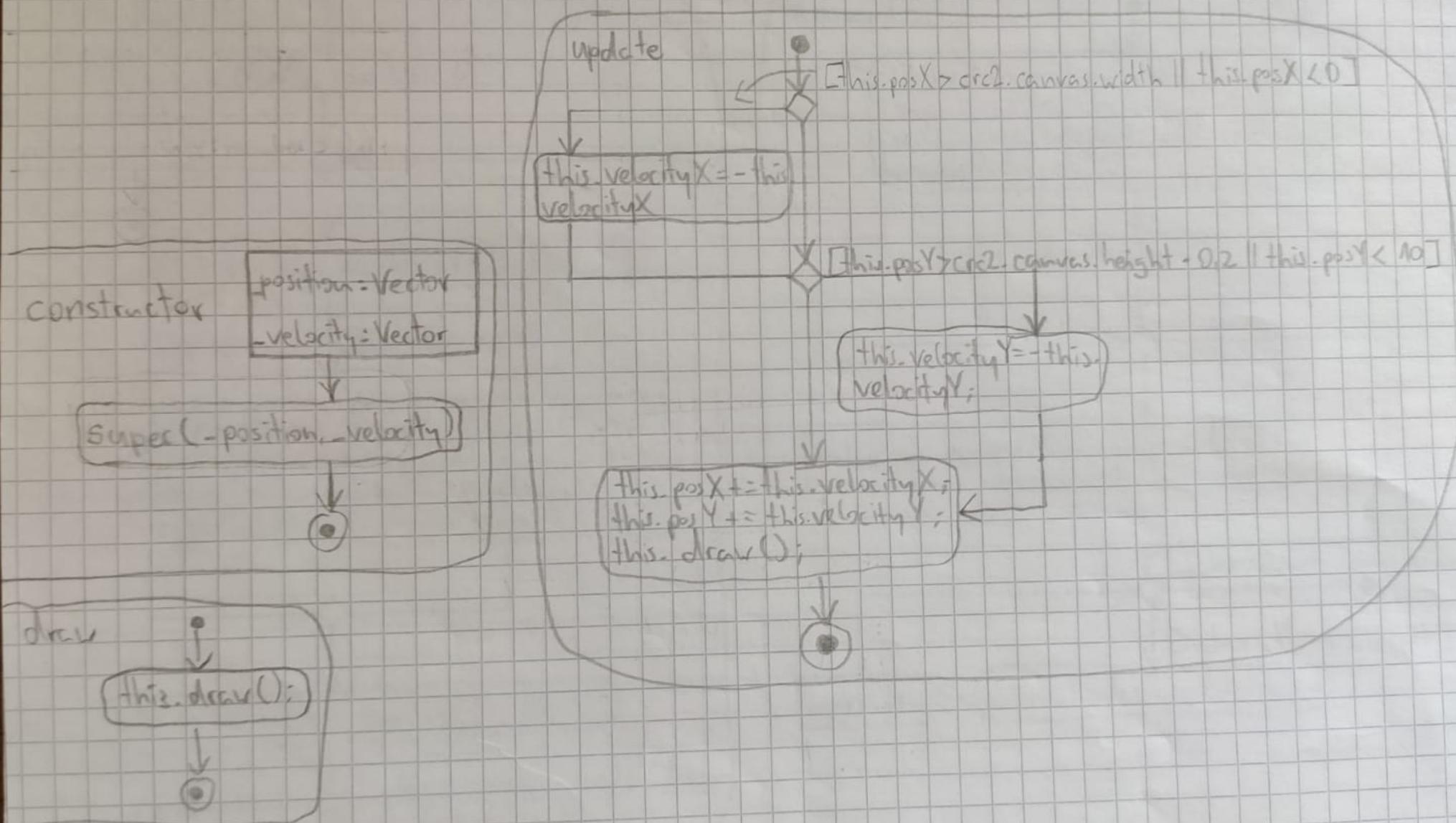
BeeHome

```
crc2.save();  
crc2.translate(crc2.canvas.width / 2, crc2.canvas.height - 0.35);  
crc2.scale(M, M)  
crc2.lineWidth = 0.5;  
crc2.strokeStyle = "beige"  
BeeHome wird gezeichnet  
crc2.restore();
```



SHOT ON MI 9T PRO  
AI TRIPLE CAMERA

# Activity diagram: cloud10



SHOT ON MI 9T PRO  
AI TRIPLE CAMERA

# Activity Diagram: move11 / updateNectar

```
public posX: number
public posY: number
public velocityX: number
public velocityY: number
```

Constructor

```
-position: Vector
velocity: Vector
```

this posY = position.y  
 this posX = position.x  
 this.VelocityY = velocity.y  
 this.VelocityX = velocity.x

draw

update

UpdateNectar

nectarCounter ++

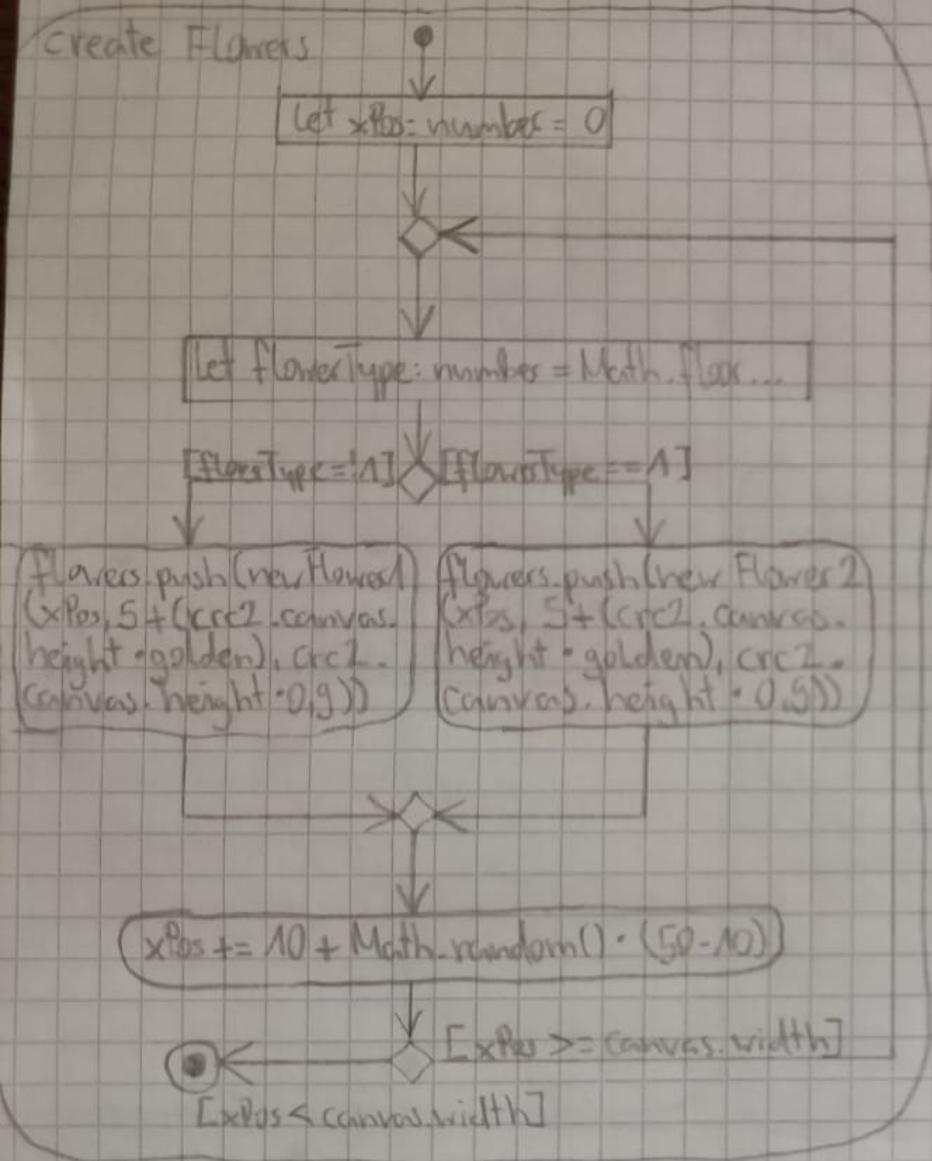
if [nectar.length < 15] & [nectar.value == nectarCounter]

nectar.length += 2  
 nectar.value = Math.random()

draw Nectarcirclebox for  
 Flower1 & Flower2

## Activitydiagramm: Main

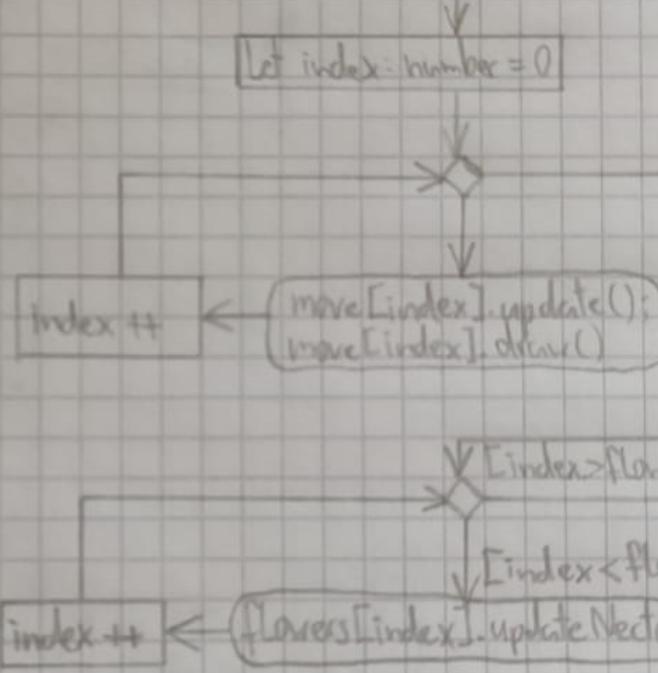
create Flowers



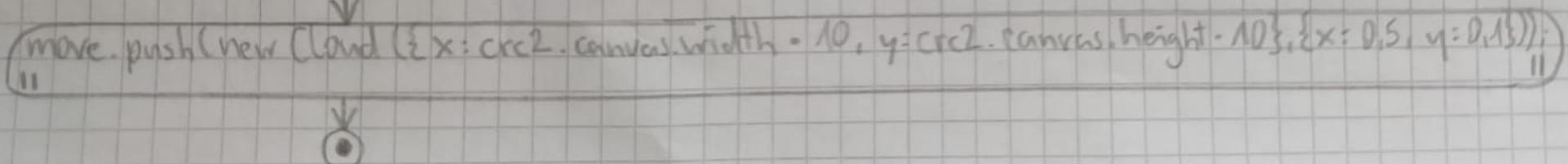
animation

```

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



CreateCloud



## Activity diagram: mainAD

 $\sum$  Local

createBees

nbcBees

handleLoad

Let index: number = 0

[ $i \rightarrow nbcBee$ ]  $\Rightarrow$ [ $i < nbcBee$ ]

```
let randomScale: number = 0.8 + Math.random() * (4.5 - 1.8);
let randomVelocityX: number = (Math.random() - 0.8) * 8;
let randomVelocityY: number = (Math.random() - 0.8) * 5;
```

```
move.push(new Bees({x: crc2.canvas.width / 2, y: crc2.canvas.height * golden}, {x: randomVelocityX, y: randomVelocityY}, randomScale));
```

index ++

createBackground

drawBackground();

drawSun({x: crc2.canvas.width / 2, y: crc2.canvas.height \* 0.15});

drawMountains({x: 0, y: crc2.canvas.height \* golden}, 200, 300, "grey");

, 125, 225, "lightgrey");

BeesHome();

```
let crc2: CanvasRenderingContext2D;
let golden: number = 0.4;
let move: Move[] = [];
let flowers: Flowers[] = [];

let imageData: ImageData;
```

let imageData: ImageData;

Install Load Listener

V

handleLoad

V

```
let canvas: HTMLCanvasElement =
document.querySelector("canvas");
```

V

```
crc2 = canvas.getContext("2d");
canvas.width = window.innerWidth;
canvas.height = window.innerHeight;
createBees(20);
```

V

createBackground();

V

createFlowers();

V

createClouds();

V

imageData = crc2.getImageData(0, 0, width, height);

V

animation();

V