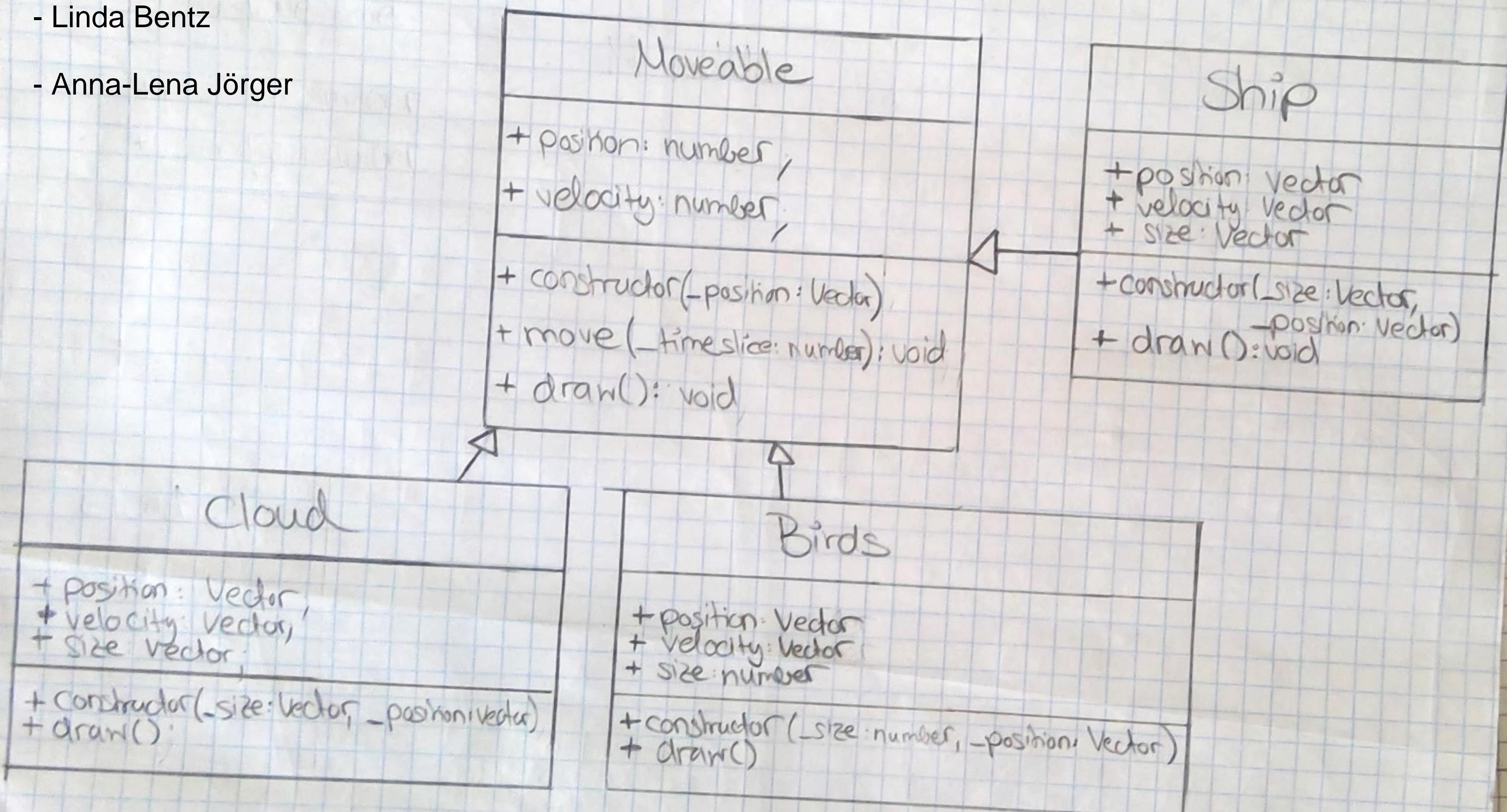


# Beach - Polymorphie: Classdiagram

In Zusammenarbeit mit :

- Linda Bentz
- Anna-Lena Jörger



# Beach - Polymorphie: Activity Diagram

Beach

```
let imageData: ImageData;
let moveables: Moveable[] = [];
crc2: CanvasRenderingContext2D;
num: number = 0.5;
```

install load listener

window.setInterval with update

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

load

handledLoad

createShip

createBirds

createClouds

createPeople

handledLoad

event: Event

create canvas for  
html and get  
rendering context

```
let bay: number = crc2 -  
canvas.height  
let posBay: Vector = {x, y}
```

drawSky

drawSand

drawWater

drawSun  
give x, y

drawBay  
give posBay, min,  
max, and color

In Zusammenarbeit mit:

drawSky

let gradient: CanvasGradient

Create color crossing  
with several addColorStop

Create color with fillStyle  
and define the color as  
property with fillRect

drawWater

let gradient: CanvasGradient

Create color crossing  
with several addColorStop

crc2.beginPath()

Create color with fillStyle  
and define the color as  
property with fillRect

drawSand

let gradient: CanvasGradient

Create color crossing  
with several addColorStop

crc2.beginPath()

Create color with fillStyle  
and define the color as  
property with fillRect

drawSun

-position: Vector

let r1: number

let r2: number

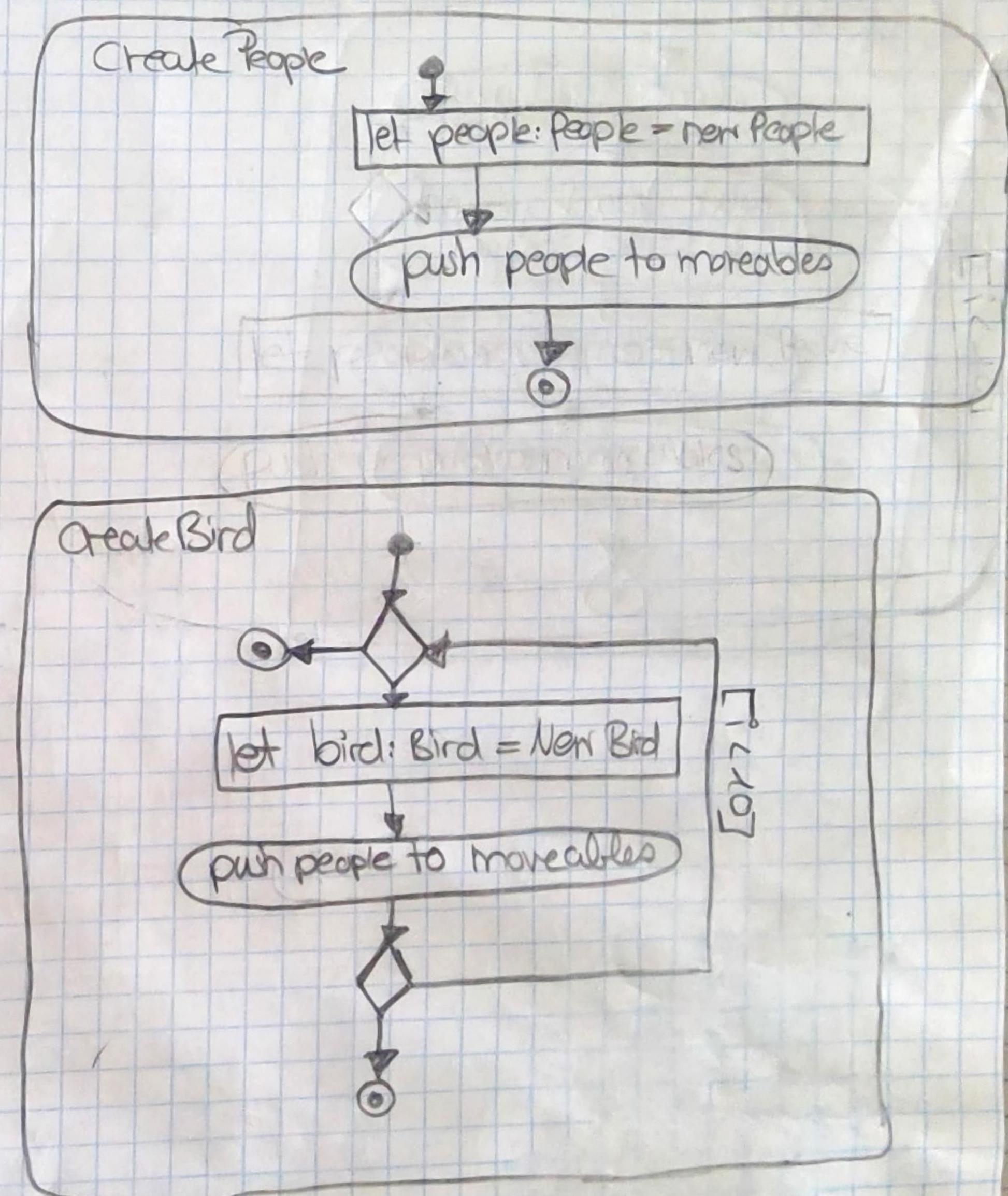
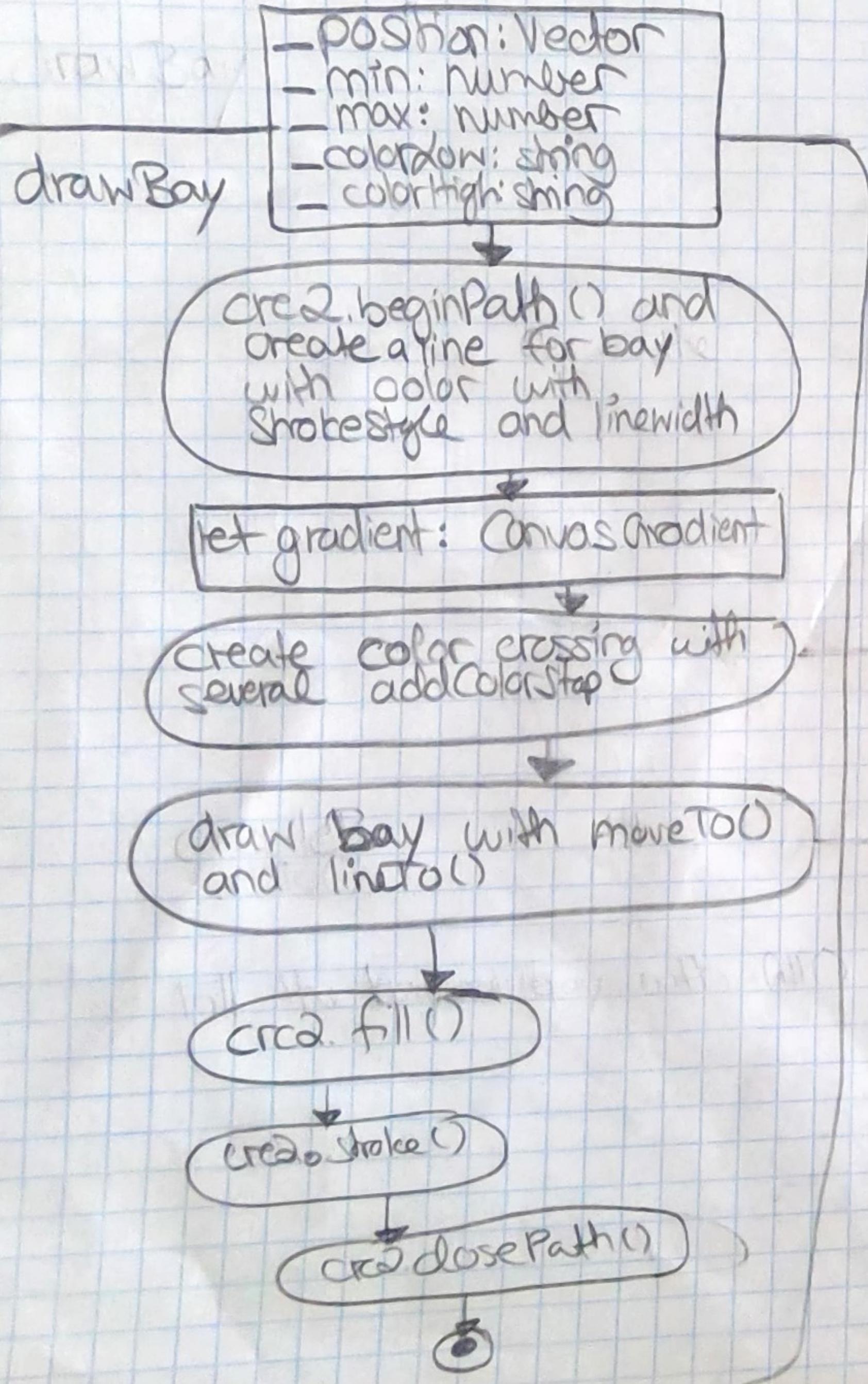
let gradient: CanvasGradient

Create color crossing with  
several addColorStop

Restoration of  
the storage of  
crc2 with restore()

Save() crc2 and translate()  
crc2 with -position.x and  
-position.y

Create an arc with  
fillStyle, arc() and fill()



update



clear background

put background  
to save image

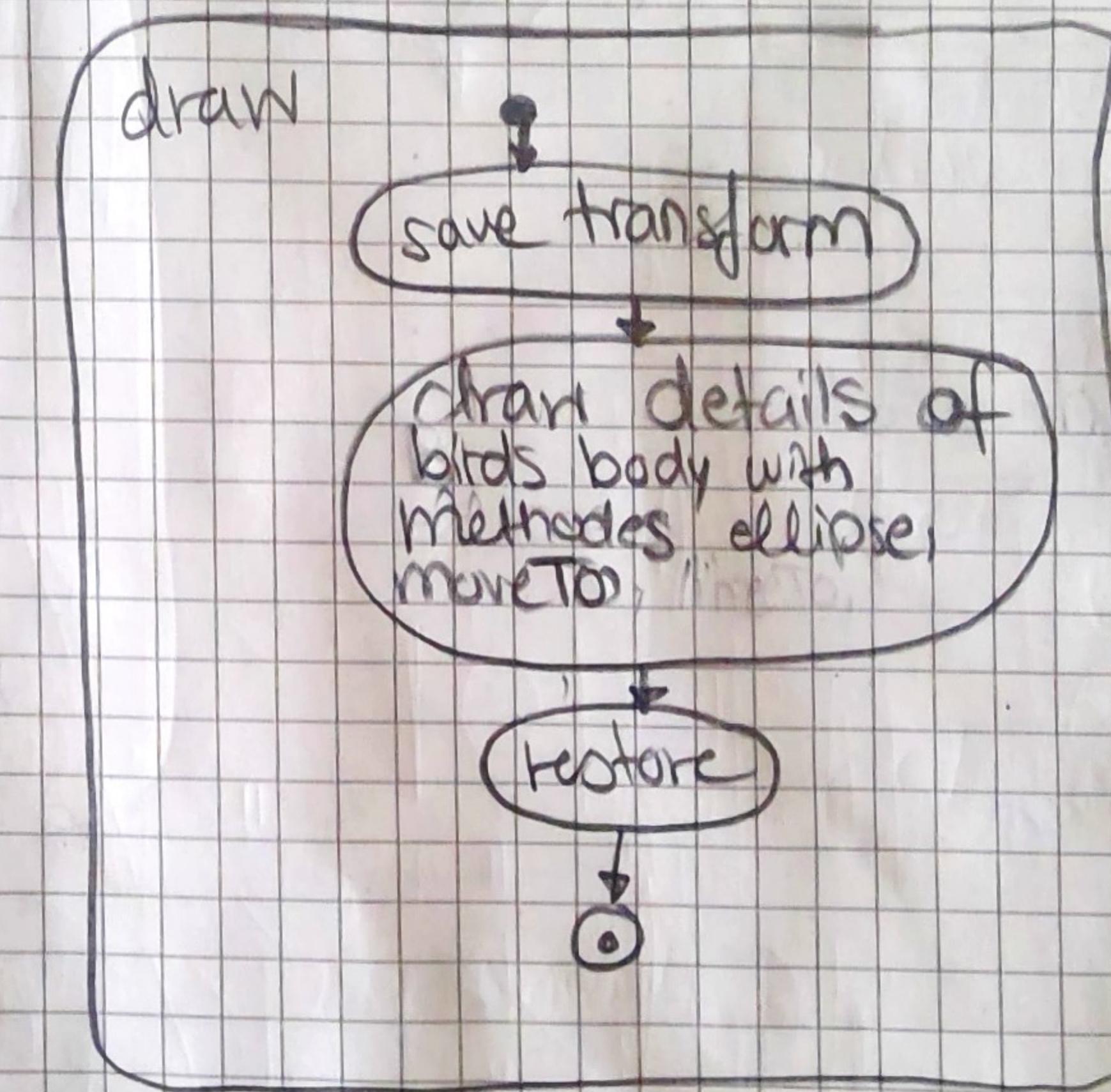
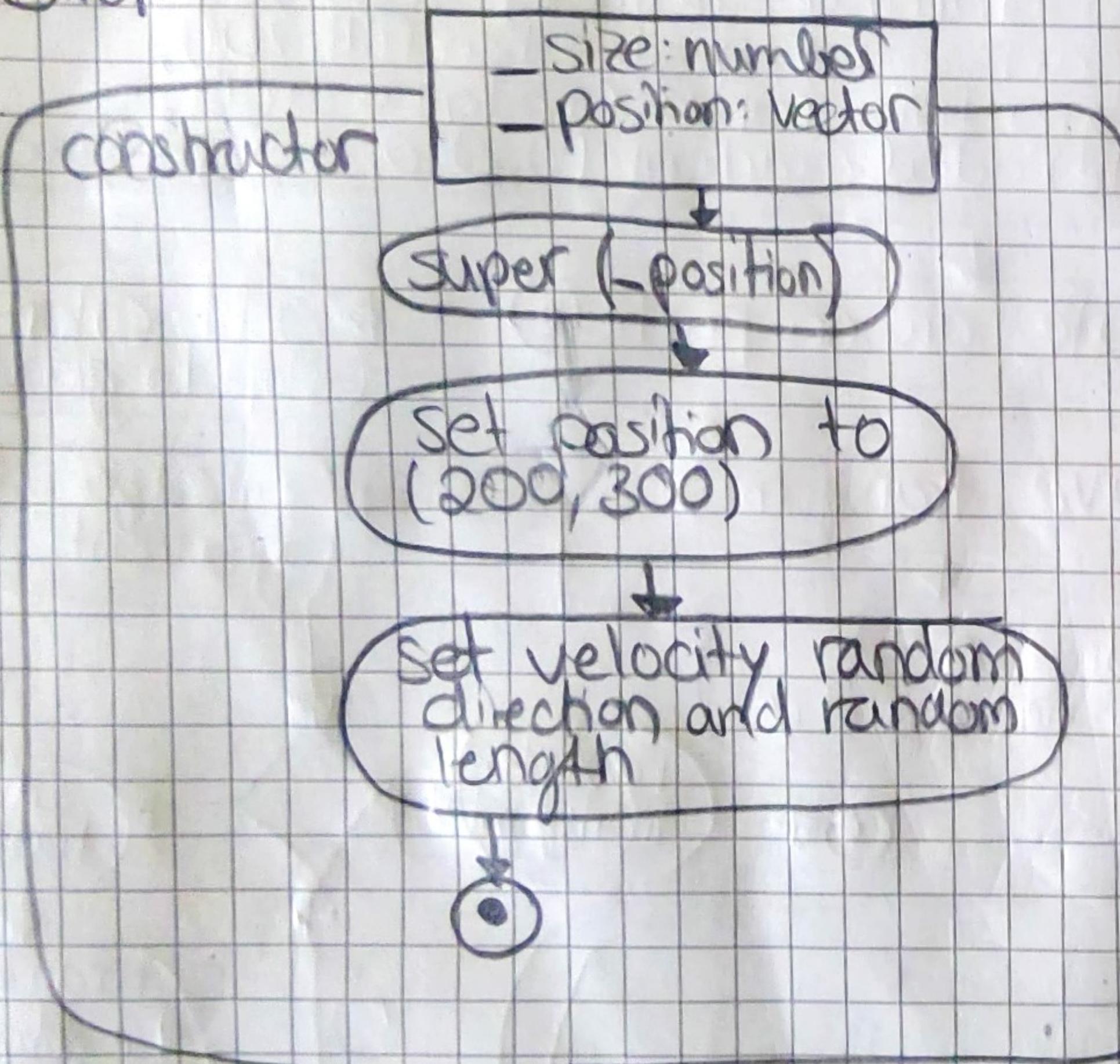
remove moveable

draw moveable



# Beach Polymorphic : Activity Diagramm

Bird



# Cloud

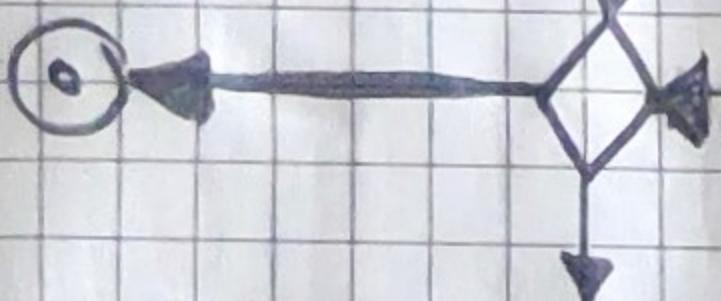
constructor

- size: Vector  
- position: Vector

super(-position)

Set position to  
(50, 100)

Set velocity to  
(30, 0)



[drawn < 50]

let x: number = (Math.random() - 0.5) \* size.x  
let y: number = -Math.random() \* size.y

let position: Vector = new Vector(x, y)

draw

```
let radiusParticle: number = 50;
let particle: Path2D = new Path2D();
let gradient: CanvasGradient = createRadialGradient(0, 0, 0, 0, radiusParticle);
```

Create arc on particle

Add color stop to gradient

Save transform

Fill style with gradient

Transition of  
position x and y

Restore  
transform

Save  
transform

Translate  
to drawnx  
and drawny

[drawn  
of positions]

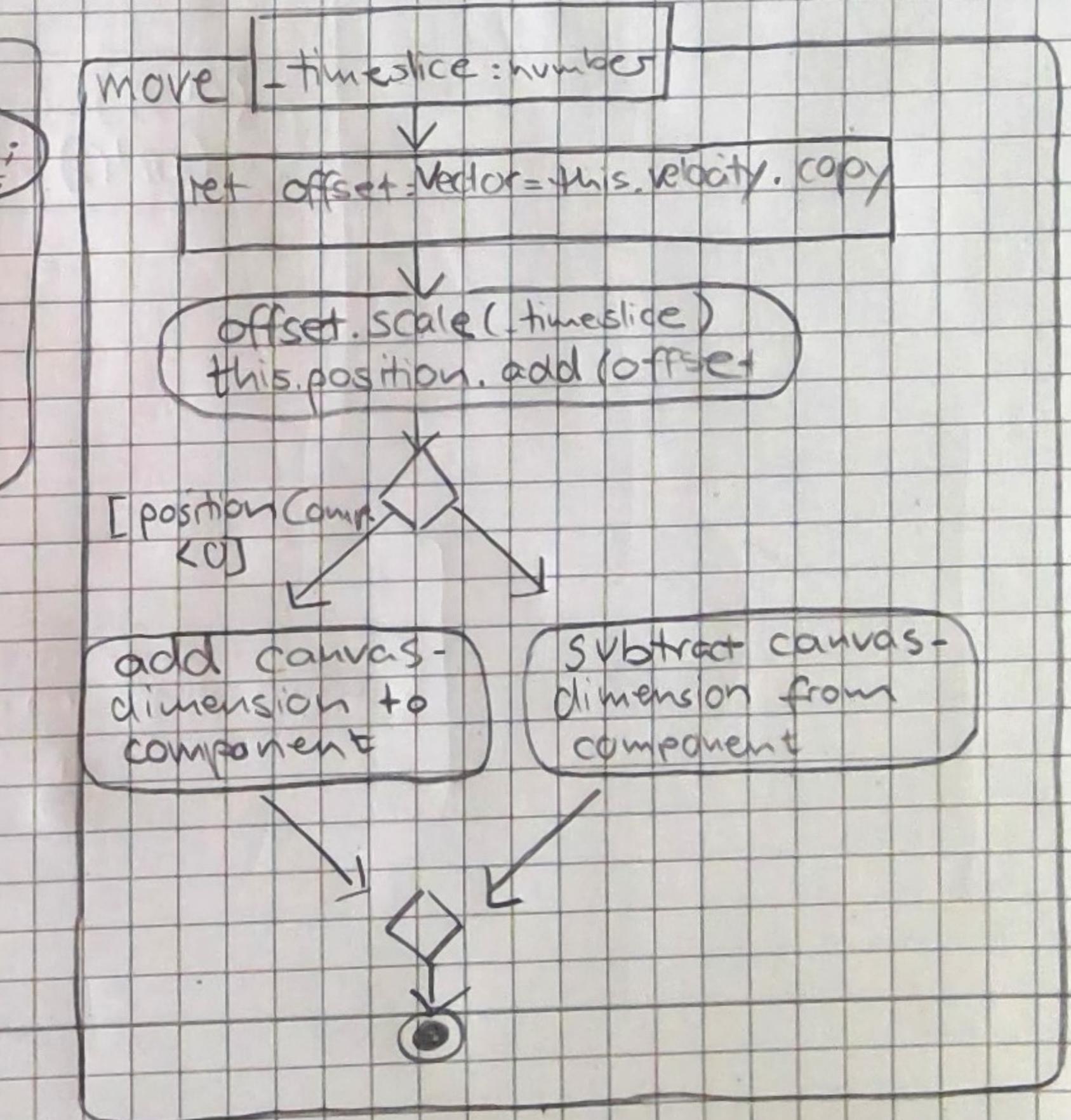
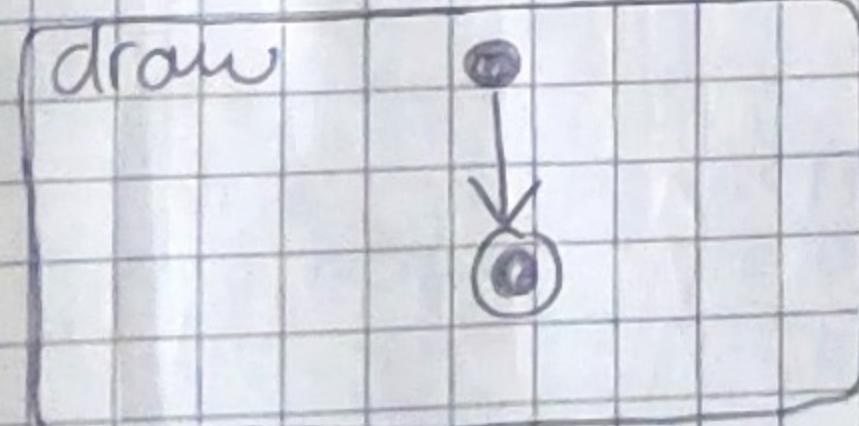
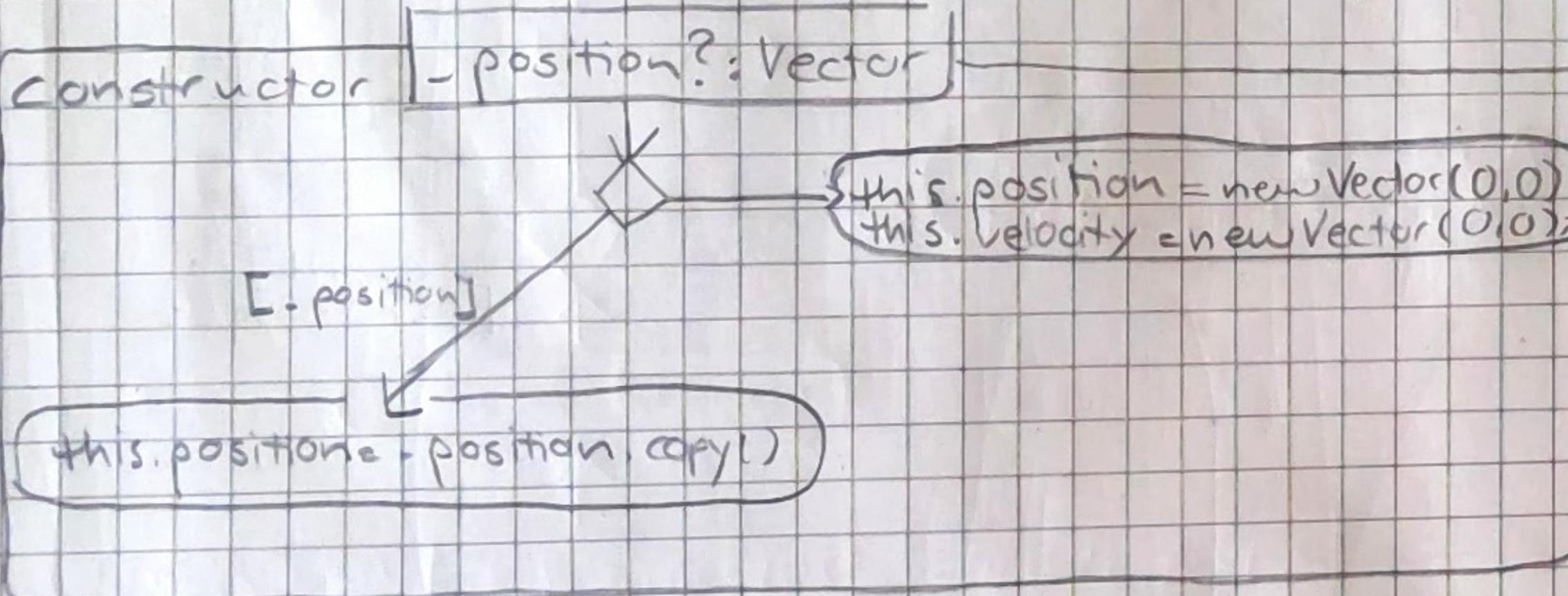
Draw  
particle

Restore  
Transform

Push position  
to particlePosition

# Moveable - Activity Diagram

position : Vector  
velocity : Vector



## Ship : Activity Diagram

```
position : Vector  
velocity : Vector  
size : Vector
```

```
constructor [-size?: Vector, -position?: Vector]
```

```
super(-position)
```

```
[ -position ]
```

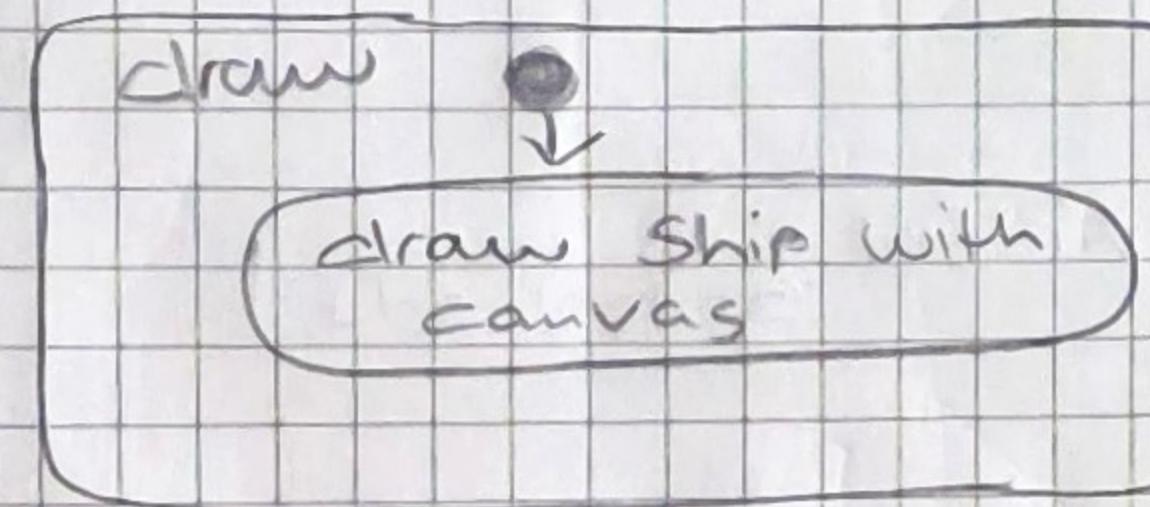
```
this.position = -position
```

```
this.position = new Vector(0, 20)  
this.velocity = new Vector(30, 0)
```

```
[ size ]
```

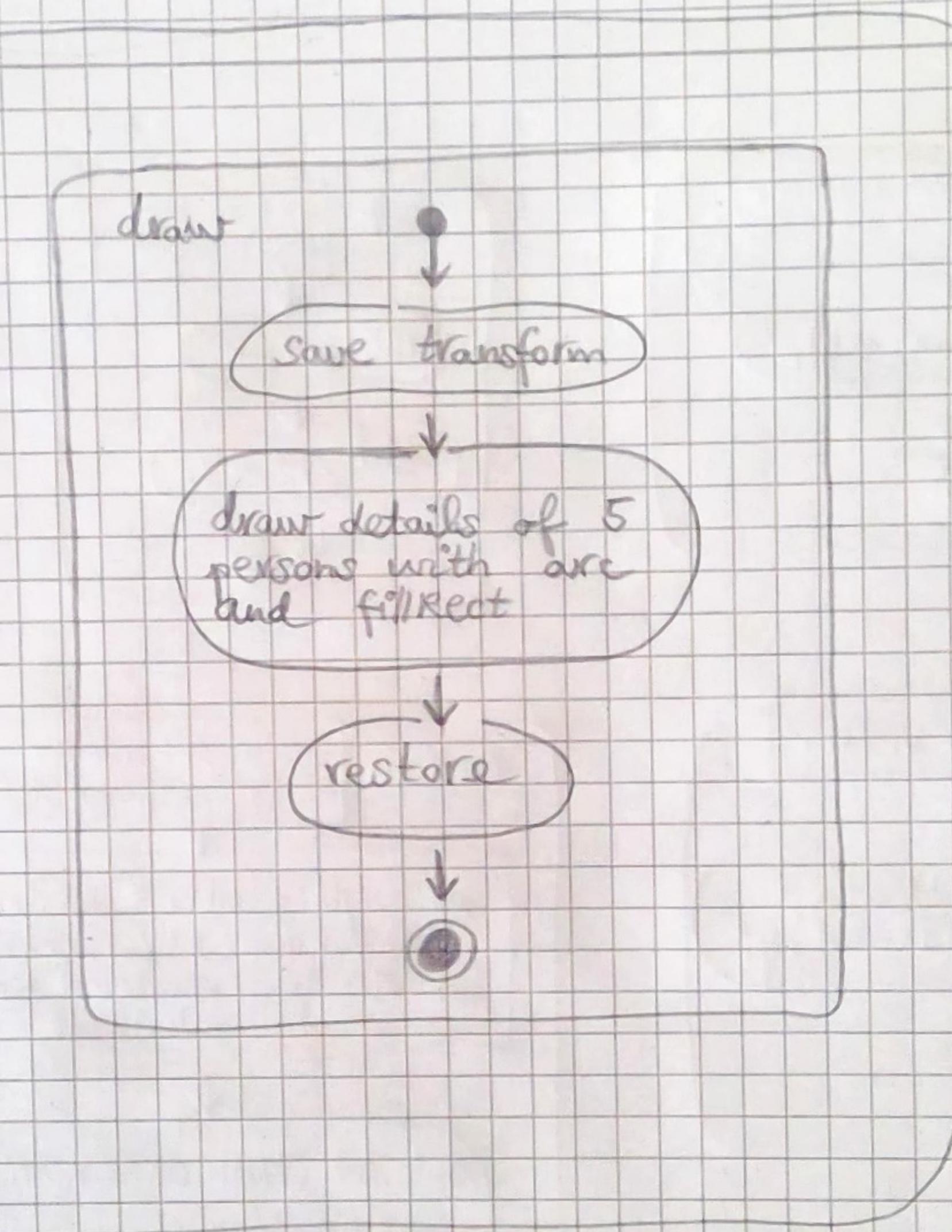
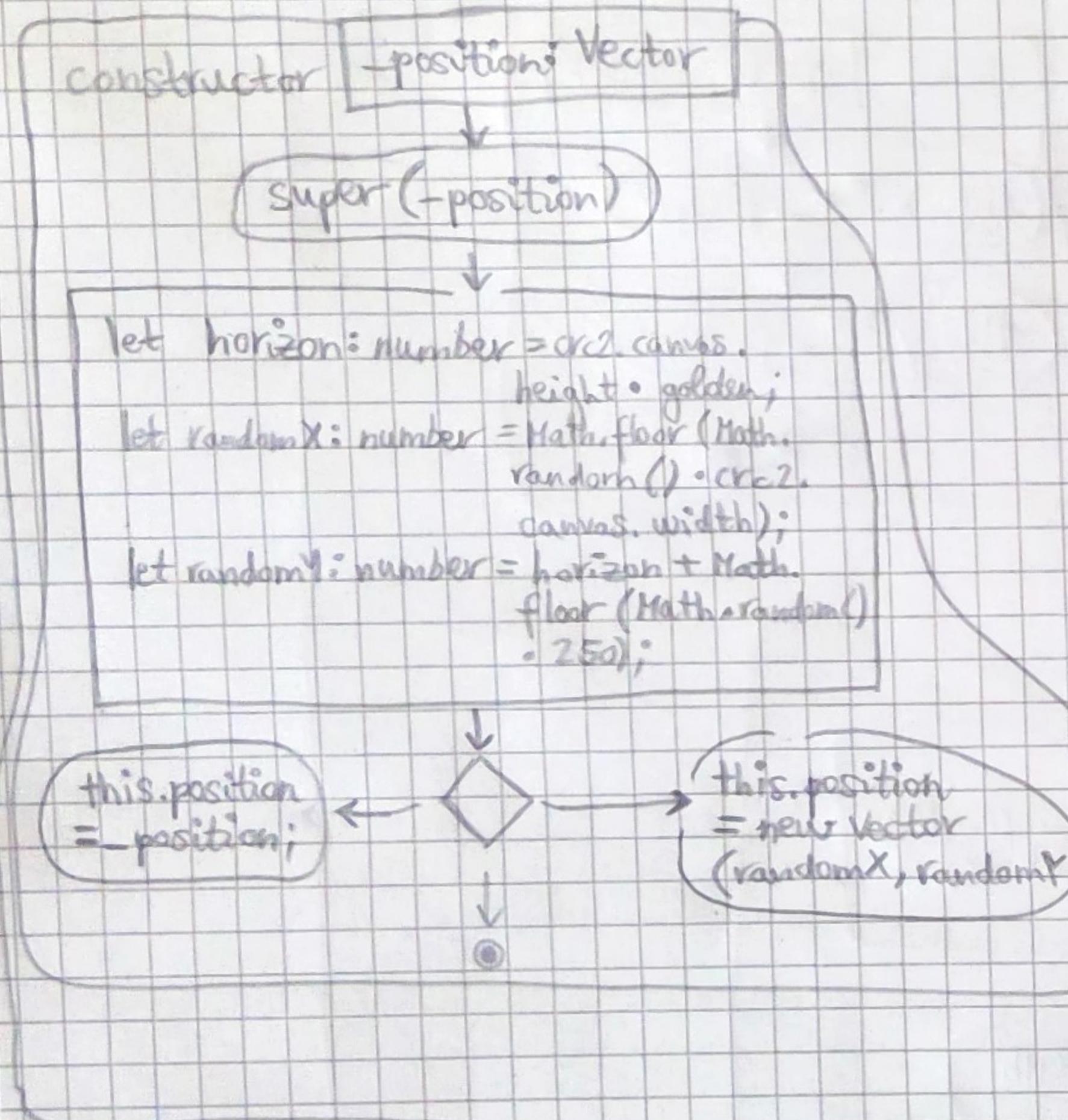
```
this.size = new Vector(270, 75)
```

```
this.size = -size
```



## Beach Polymorphie: Activity Diagramm

Person



# Vector

constructor

```
-x: number;  
-y: number;
```

```
this.set(-x, -y);
```

set

```
-x: number;  
-y: number;
```

```
this.x = -x;
```

```
this.y = -y;
```

scale [-factor: number]

```
this.x *= -factor;
```

```
this.y *= -factor;
```

add

```
-addend: Vector
```

```
this.x += addend.x;  
this.y += addend.y;
```

Random

```
-minlength: number  
-maxlength: number
```

```
let length: number = minlength +  
Math.random() * (maxlength - minlength);  
let direction: number = Math.random() * 2  
* Math.PI;
```

```
this.set(Math.cos(direction), Math.  
sin(direction));
```

```
this.scale(length);
```