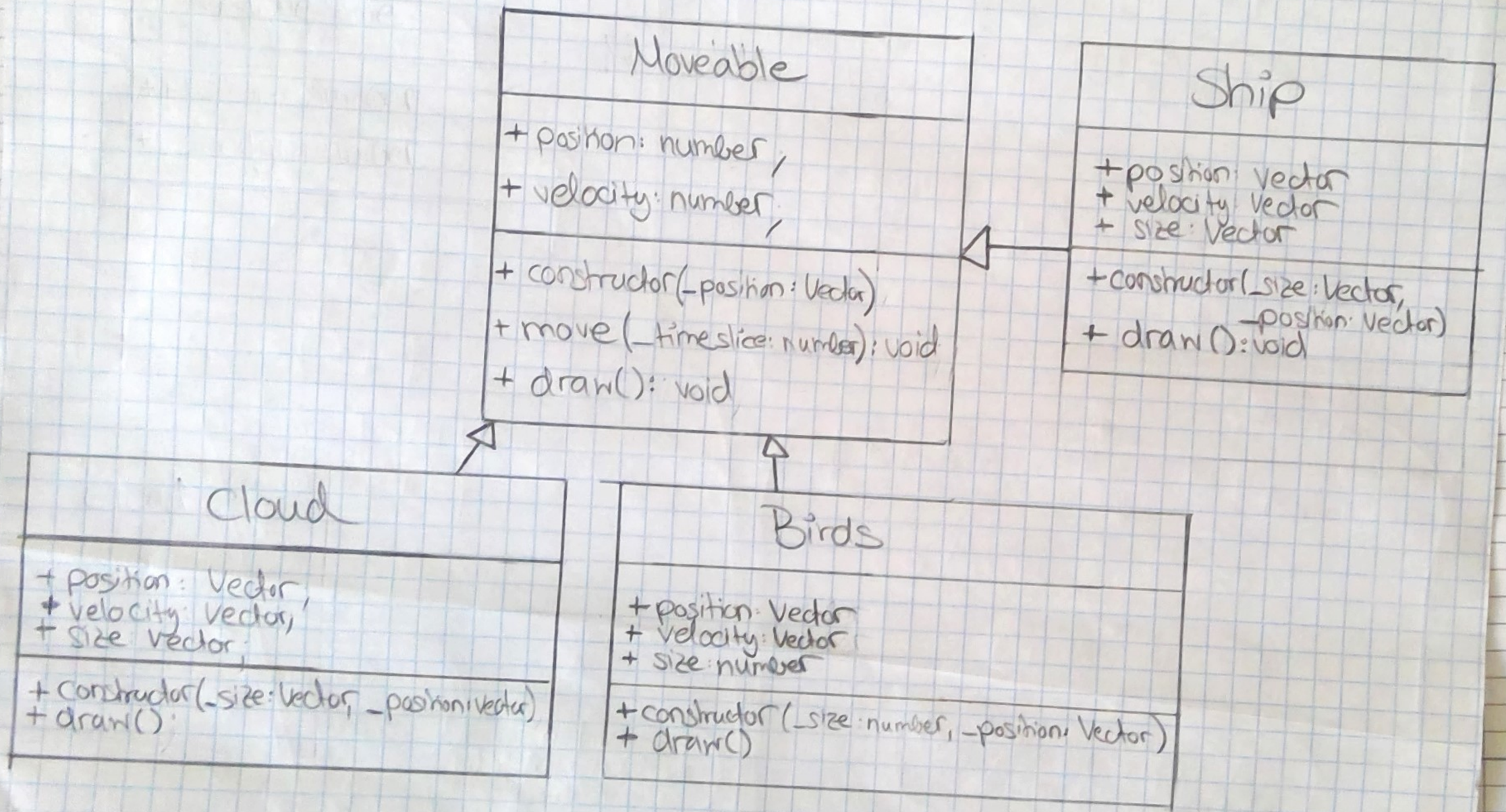
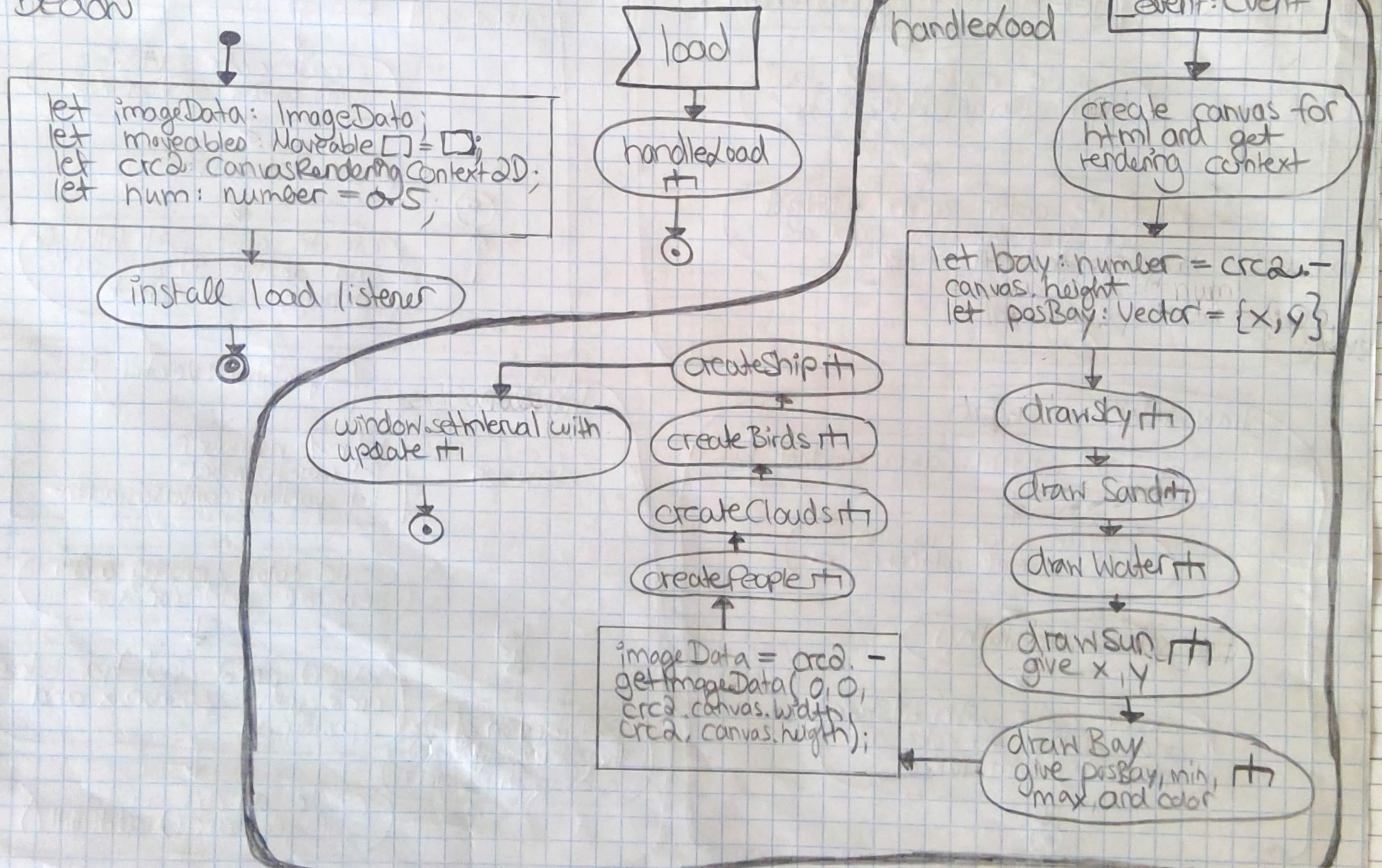
Beach_Polymorphie: Classdiagram

In Zusammenarbeit mit Linda Bentz & Evelin Sinner



Beach_Polymorphie: Activity Diagram

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

drawSand

let gradient: CanvasGradient

create color crossing with several addColorStop

crc2.beginPath()

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

drawSun

-position: Vector

let r1: number
let r2: number
let gradient: CanvasGradient

create color crossing with several addColorStop

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

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

Restoration of the storage of crc2 with restore()

drawBay

- position: Vector
- min: number
- max: number
- colorLow: string
- colorHigh: string

ctx2.beginPath() and create a line for bay with color with strokeStyle and lineWidth

let gradient: CanvasGradient

create color crossing with several addColorStop()

drawBay with moveTo() and lineTo()

ctx2.fill()

ctx2.stroke()

ctx2.closePath()

createPeople

let people: People = new People

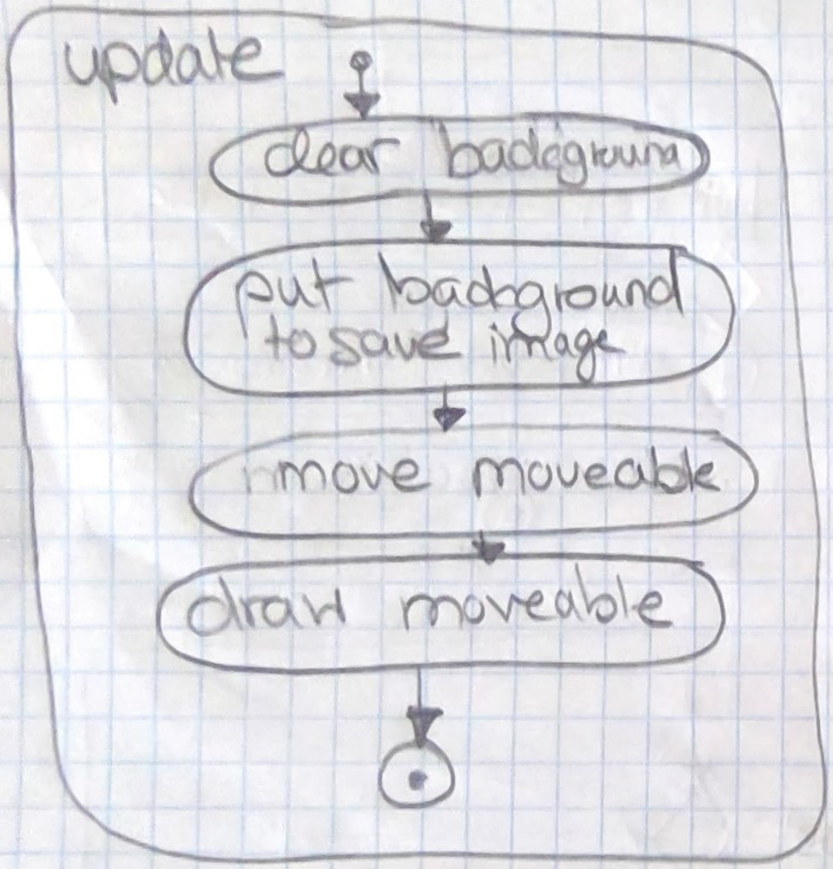
push people to moveables

createBird

let bird: Bird = New Bird

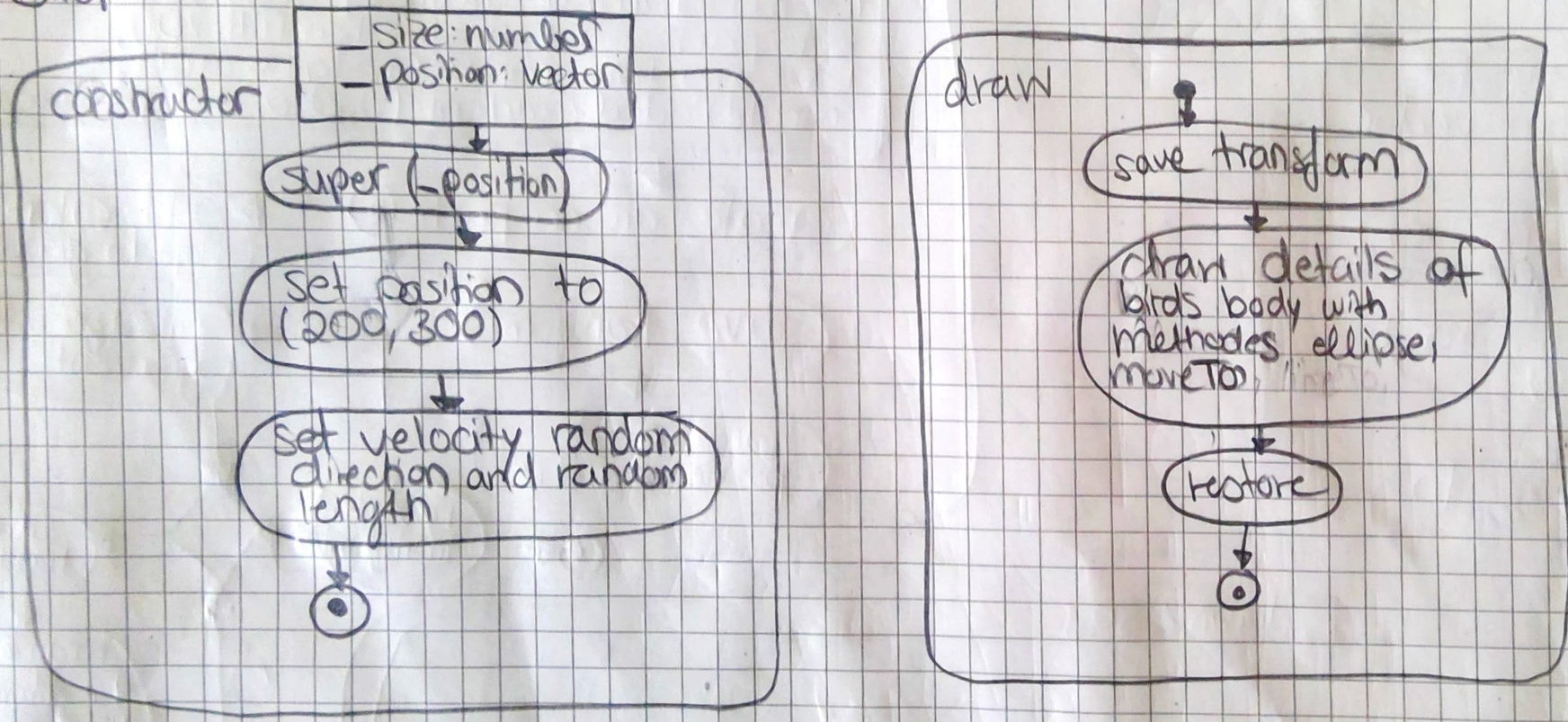
push people to moveables

[for]

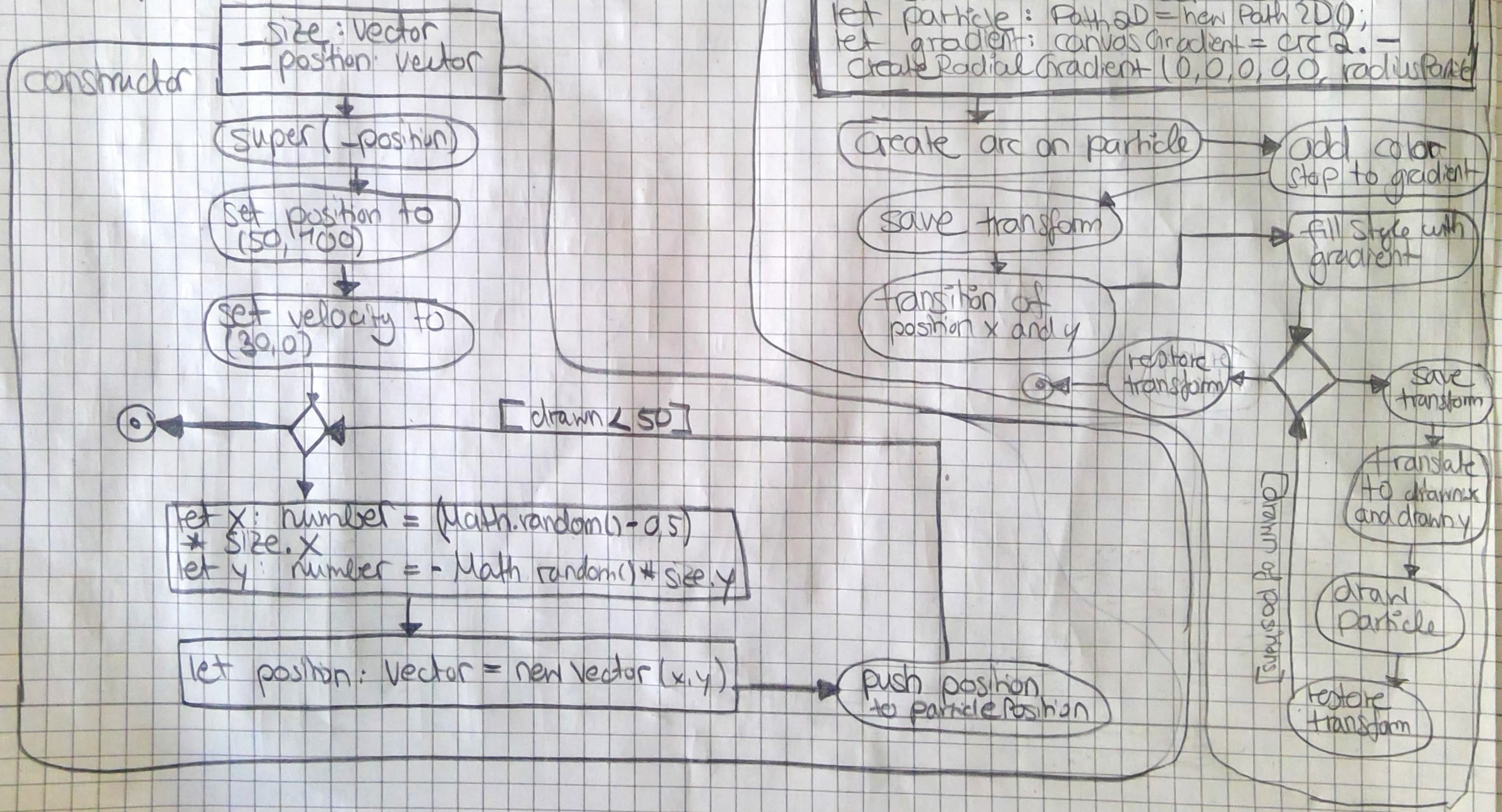

update
clear background
put background to save image
move moveable
draw moveable

Beach Polymorphie : Activity Diagramm

Bird

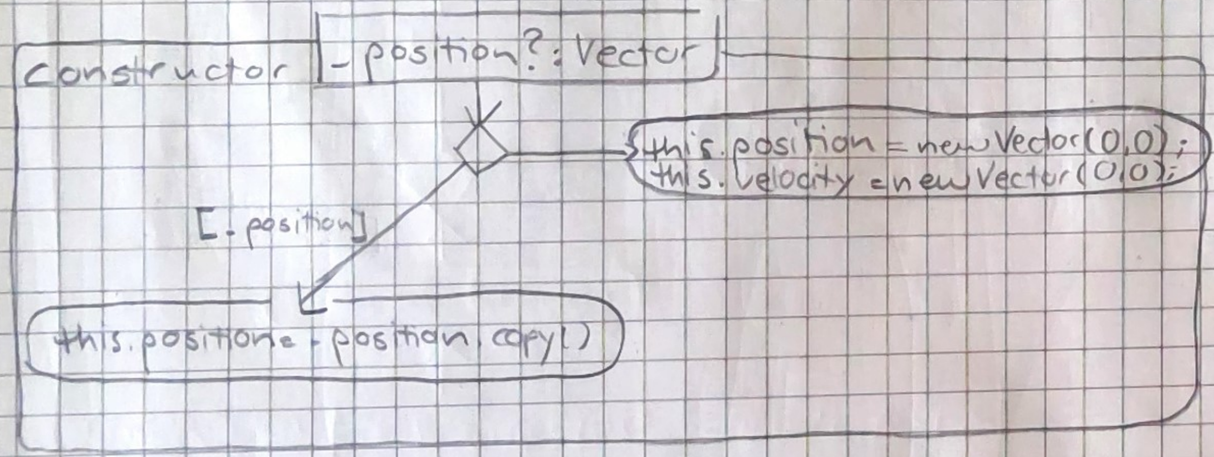


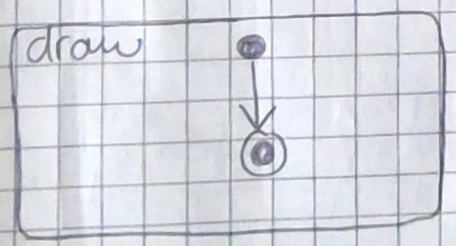
Cloud

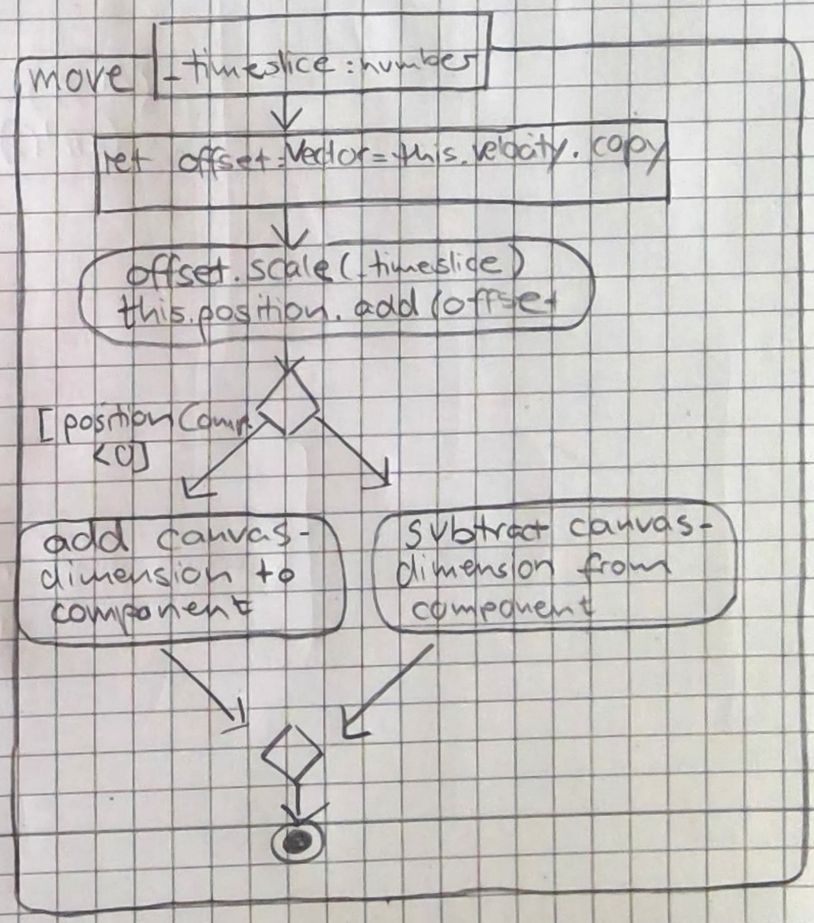

constructor
_size : Vector
_position: Vector
super(_position)
set position to (150, 100)
set velocity to (30, 0)
drawn < 50
let x: number = (Math.random() - 0.5) * size.x
let y: number = - Math.random() * size.y
let position: Vector = new Vector(x, y)
push position to particlePosition
draw
let radiusParticle: number = 50;
let particle: Path2D = new Path2D();
let gradient: canvasGradient = crc2.-
createRadialGradient(0,0,0,0,0, radiusPart
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 drawn.x and drawn.y
draw particle
restore transform
drawn of positions

Moveable - Activity Diagram

position : Vector
velocity : Vector







Ship : Activity Diagram

position : Vector
velocity : Vector
size : Vector

draw

draw ship with canvas

constructor | _size?:Vector, _position?:Vector

super(_position)

[_position]

this.position = _position

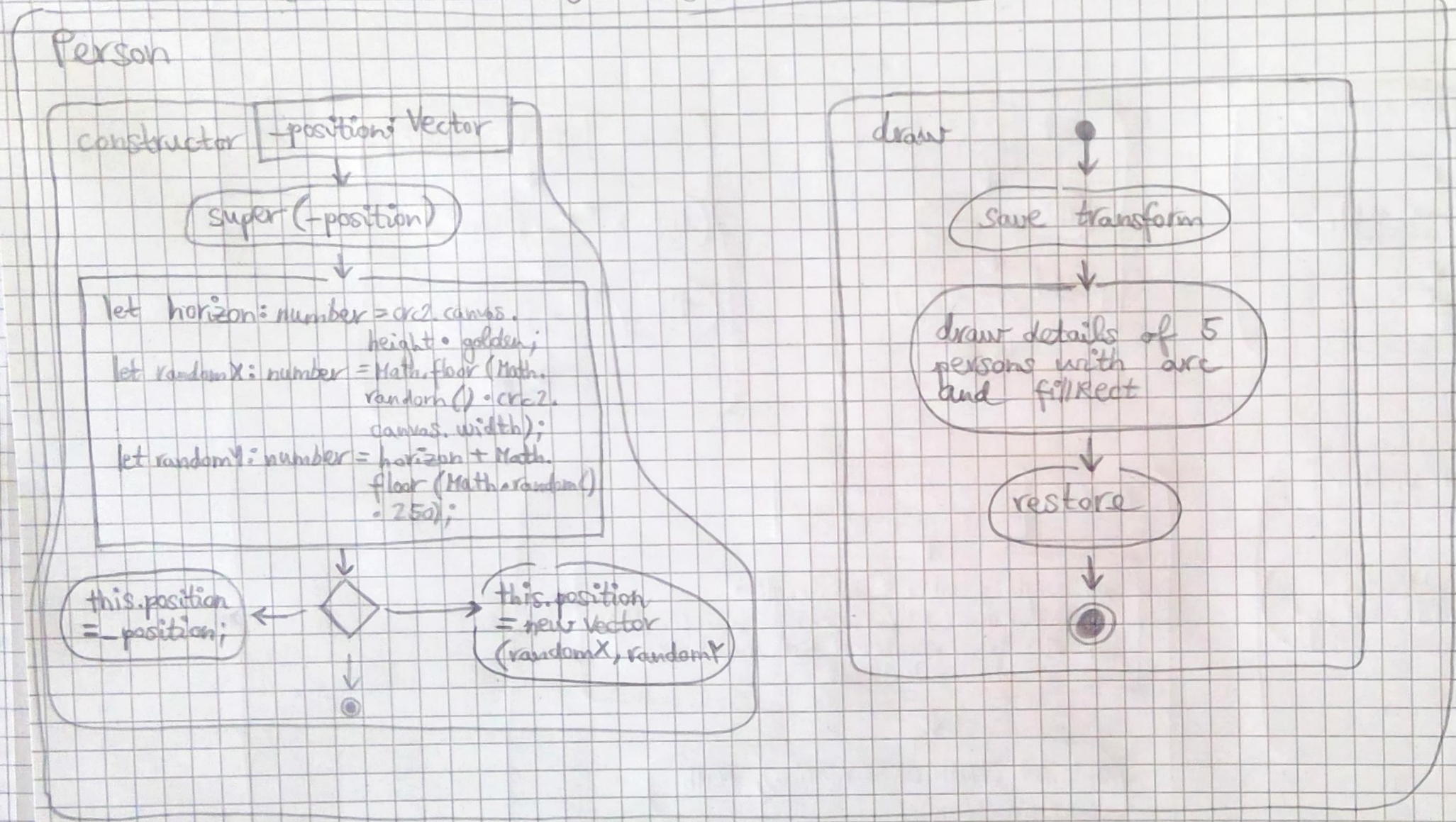
this.position = new Vector(0,20)
this.velocity = new Vector(30,0)

[size]

this.size = _size

this.size = new Vector(270,175)

Beach Polymorphie: Activity Diagramm



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);