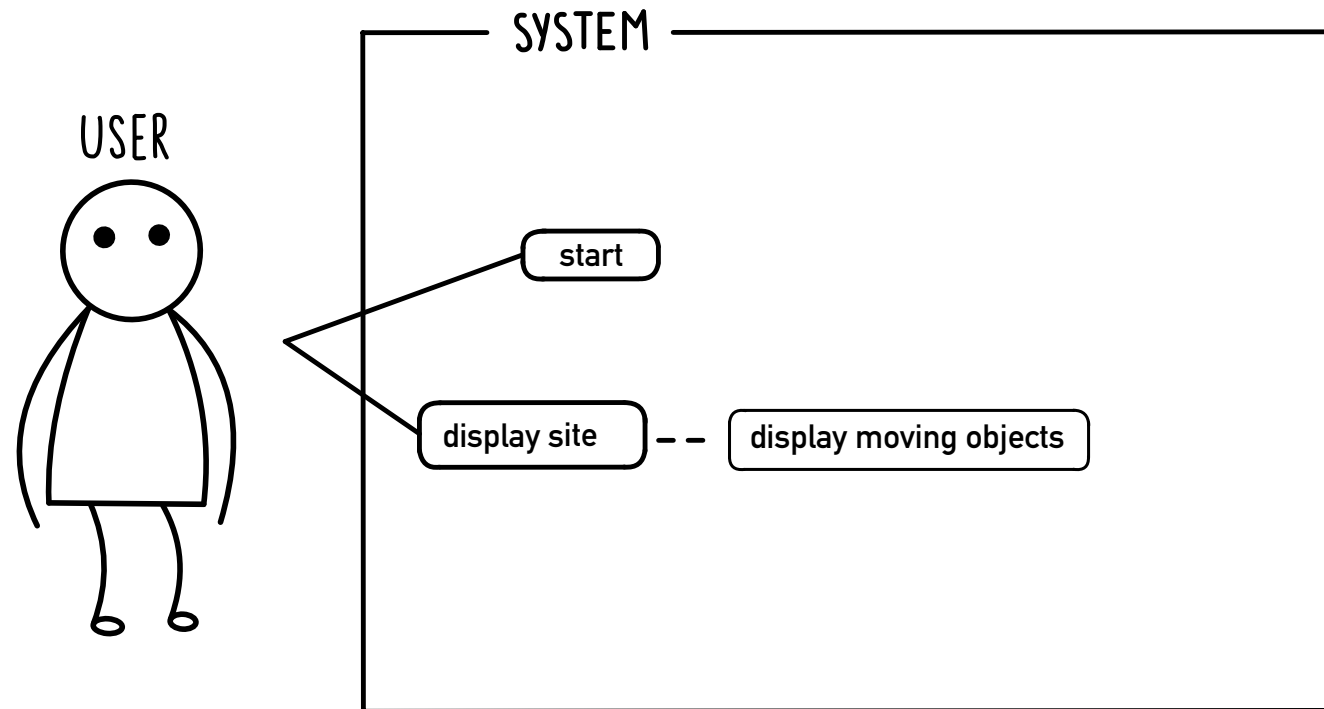
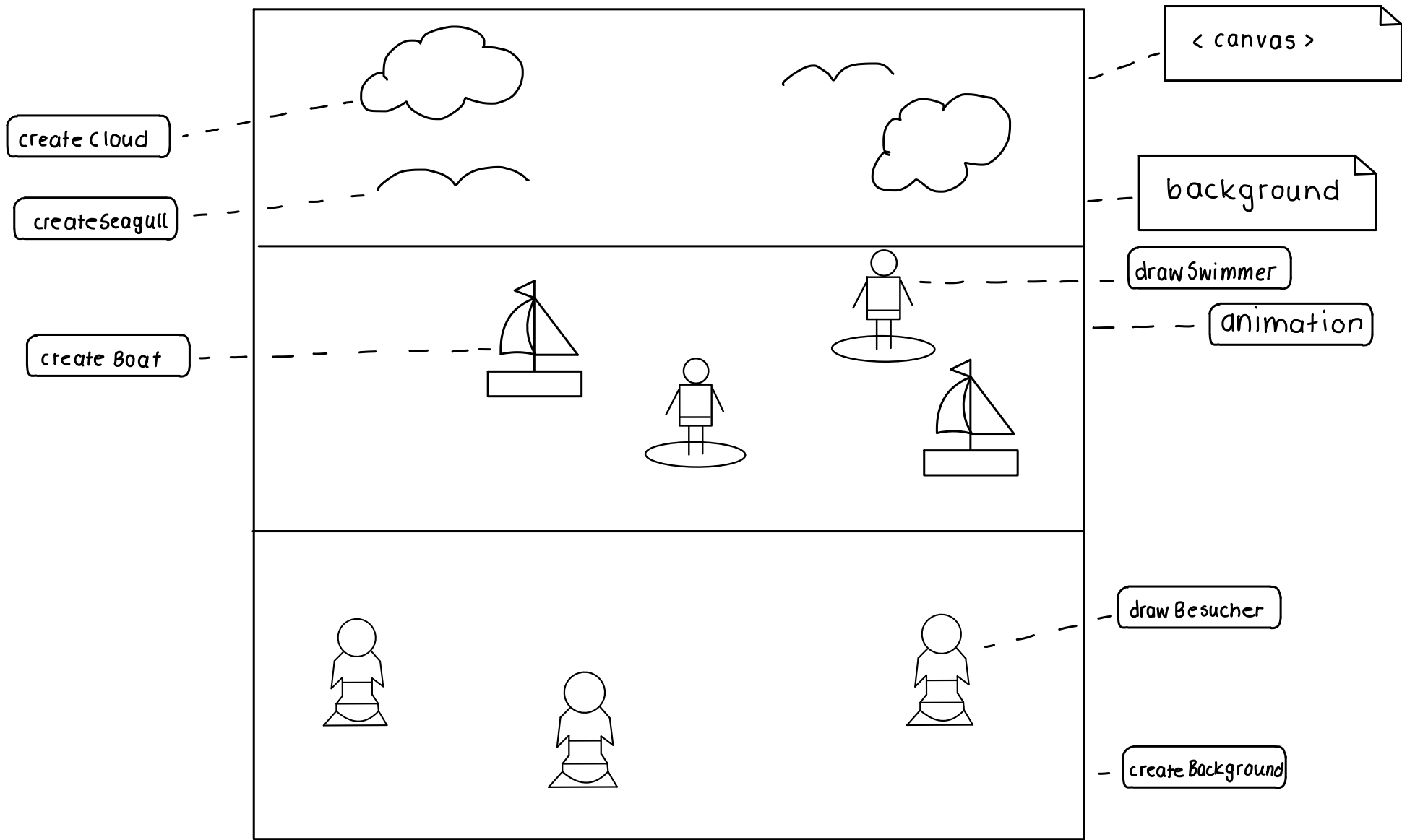
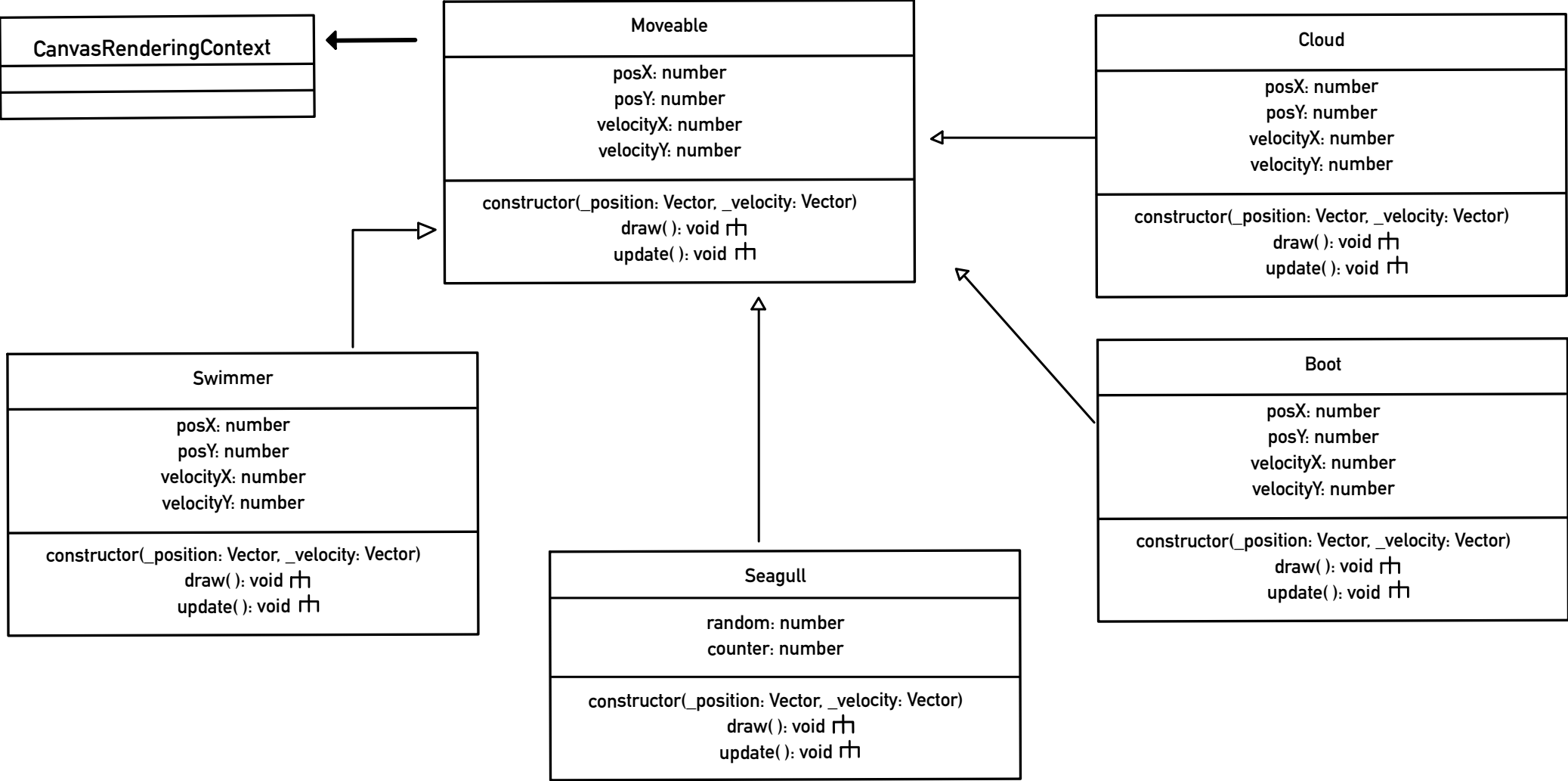


Use Case Diagramm





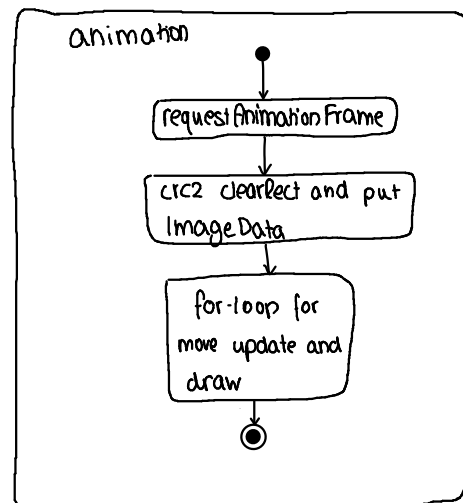
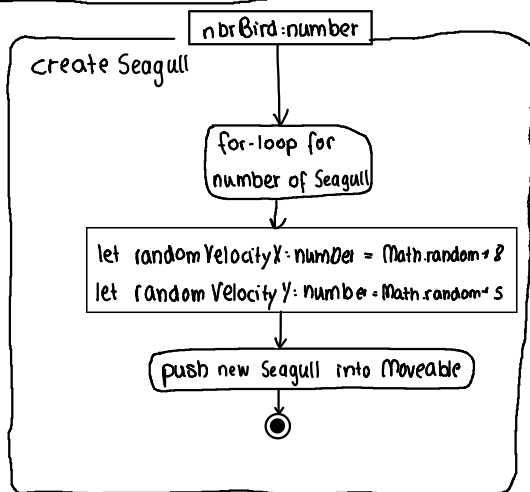
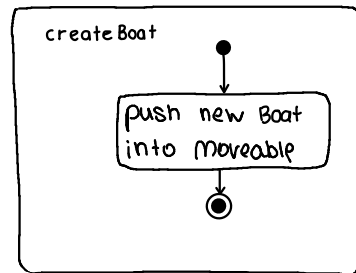
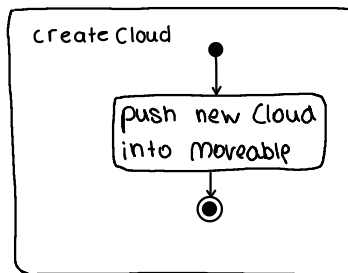
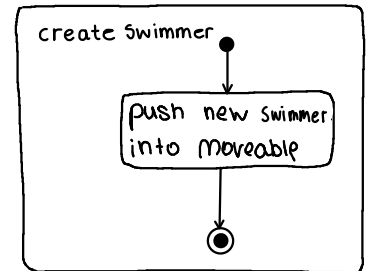
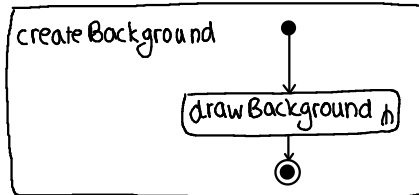
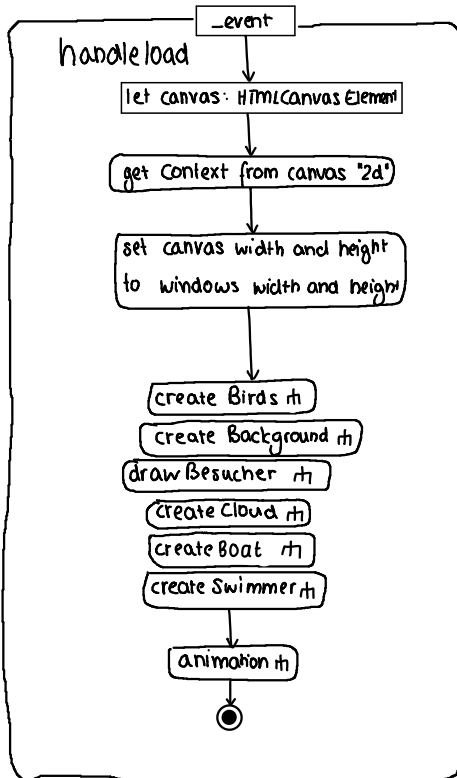
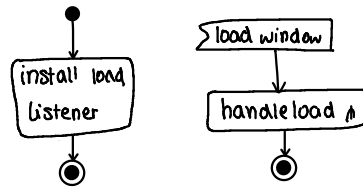
Class Diagramm



# Activity - Diagram

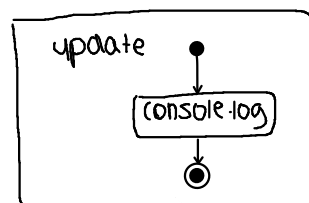
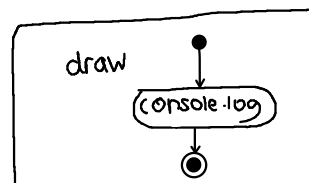
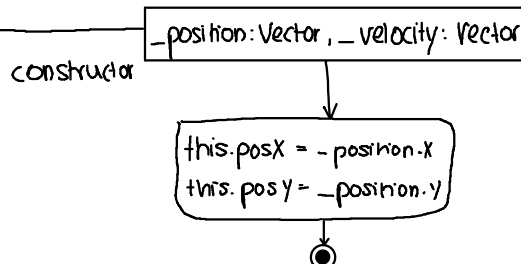
## main

```
export let crc2: CanvasRenderingContext2D;
export let rotate: number;
let move: Movable[] = [];
let ImageData: ImageData;
```



## Movable

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



Seagull extends Moreable

randomNumber: number  
counter: number = 0

constructor

-position: Vector, -velocity: Vector

get from superclass  
-position and -velocity

draw

let scale: number

save crc2 and draw with  
beziercurve and restore

update

this.velocityY  
= -this.velocityY

if posY > canvas.height

if posX > canvas.width

this.velocityX  
= -this.velocityX

if this.counter = this.RandomNumber

this.velocityX = -this.velocityX  
this.velocityY = -this.velocityY  
this.counter = 0

this.posX += this.velocityX  
this.posY += this.velocityY  
this.counter++

Boat extends Moreable

declare posX: number  
declare posY: number  
velocityX: number  
velocityY: number

constructor

-position: Vector, -velocity: Vector

get from superclass  
-position and -velocity

draw

let scale: number

save crc2 and draw with  
lines/path and restore

update

this.velocityY  
= -this.velocityY

if posY > canvas.height

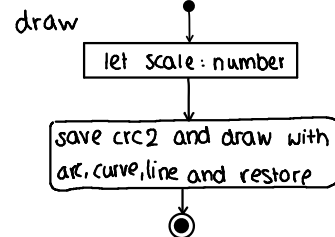
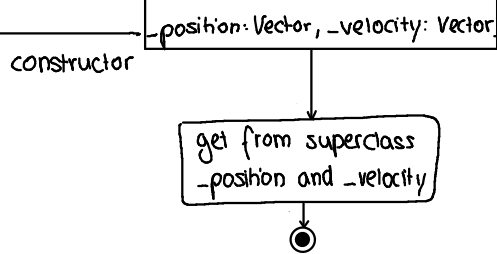
if posX > canvas.width

this.velocityX  
= -this.velocityX

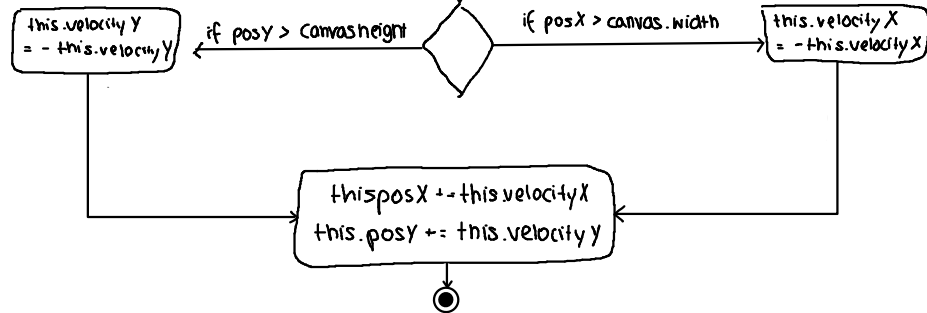
this.posX += this.velocityX  
this.posY += this.velocityY

Swimmer extends Moreade

declare posX: number  
declare posY: number  
velocityX: number  
velocityY: number

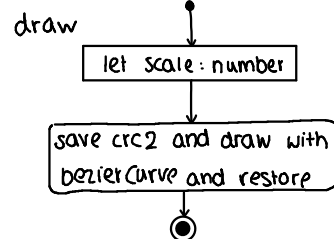
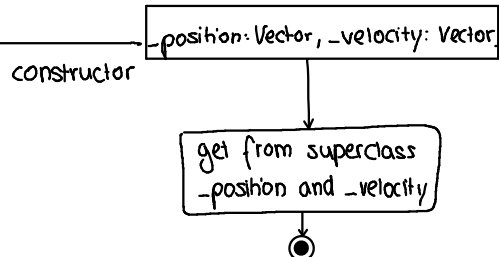


update

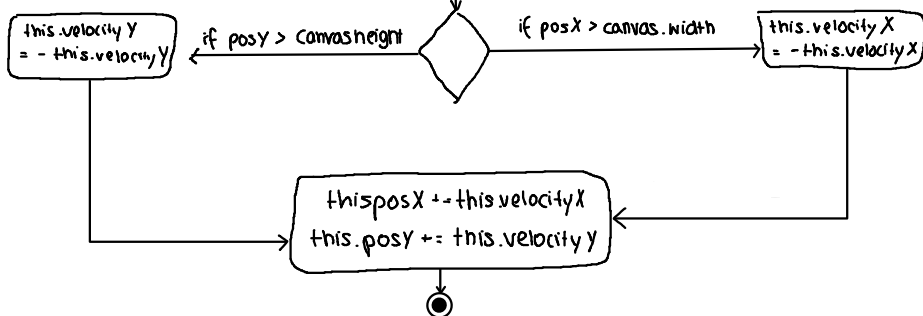


Cloud extends Moreade

declare posX: number  
declare posY: number  
velocityX: number  
velocityY: number



update



## background

export interface Vector

x: number  
y: number

export drawBackground

fill crc2 with colors  
and fillRect

export drawBesucher

\_x: number, \_y: number

let scale: number

save crc2 and draw  
with lines, curve and path  
and fillStyle and  
restore crc2.