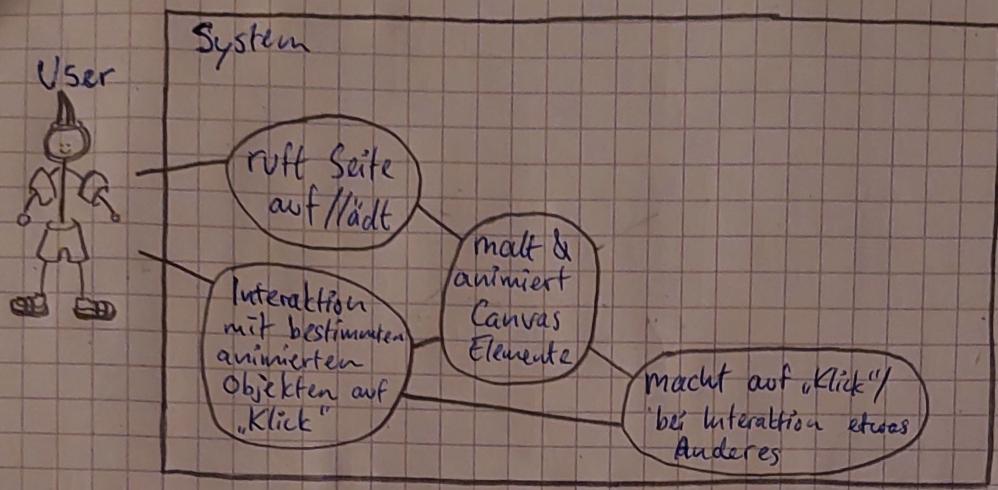
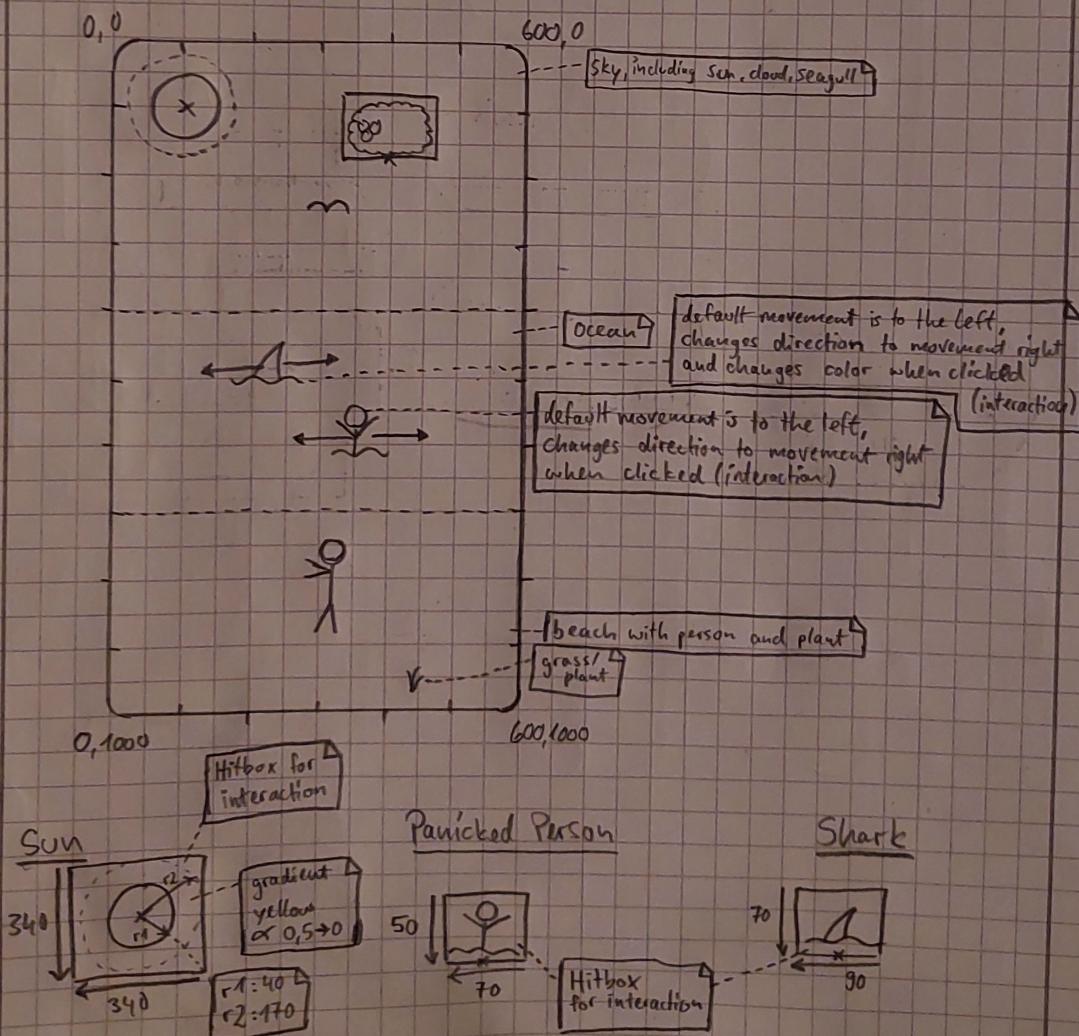


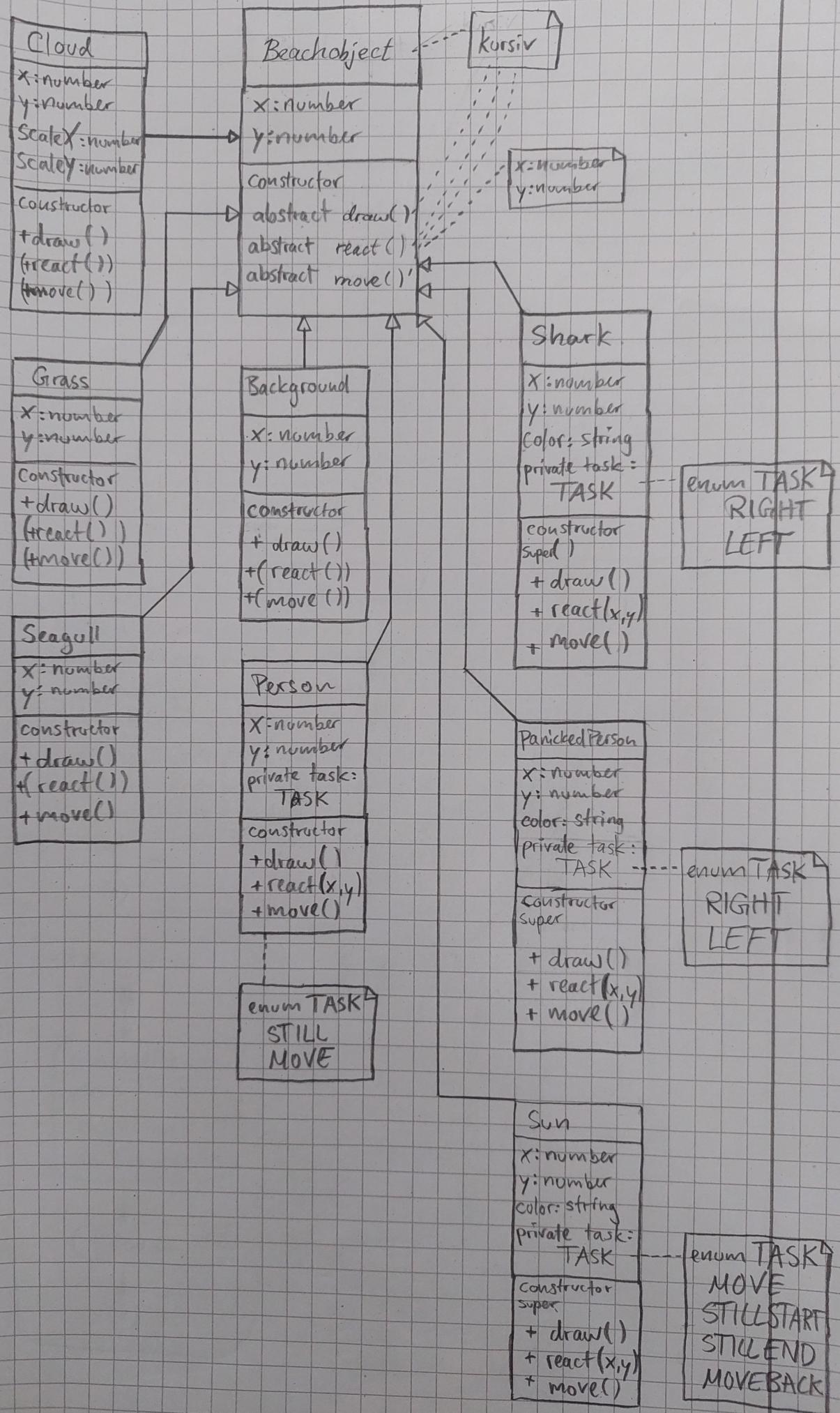
EIA 2 A11 : Interaction

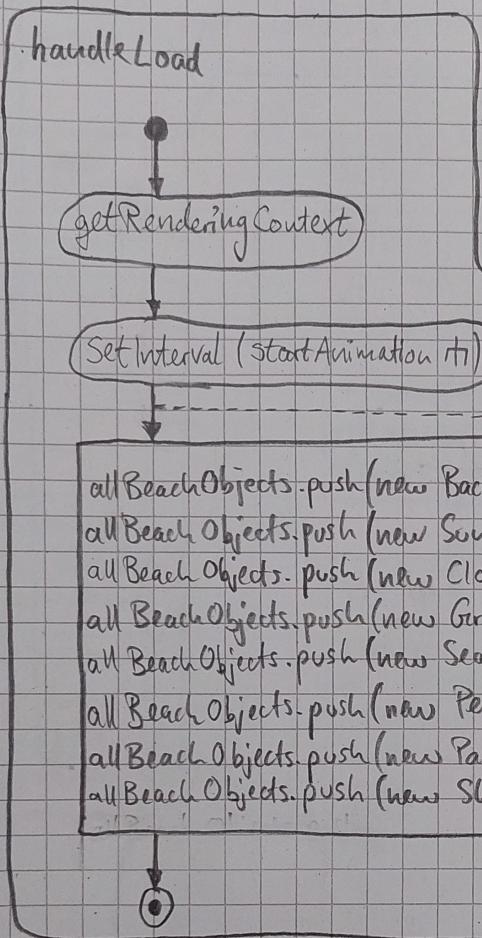
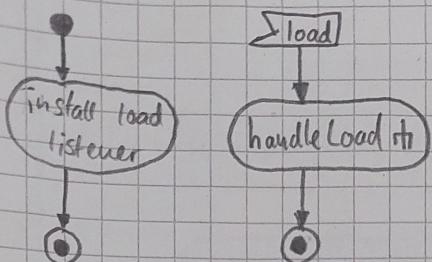
Use-Case Diagram



UI-Scribble







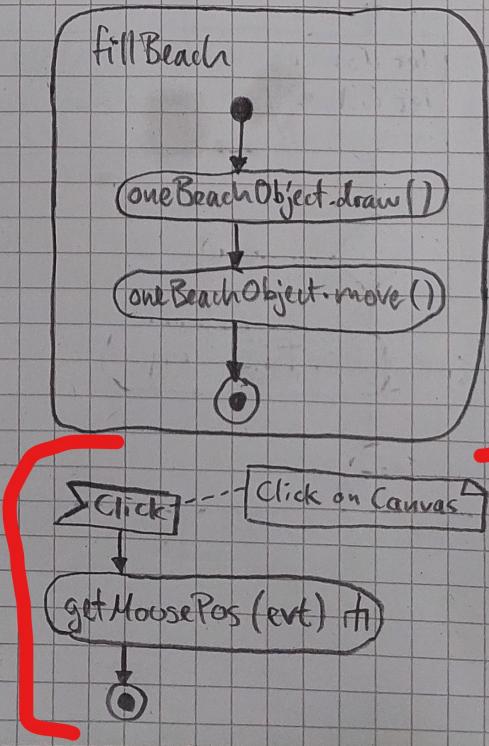
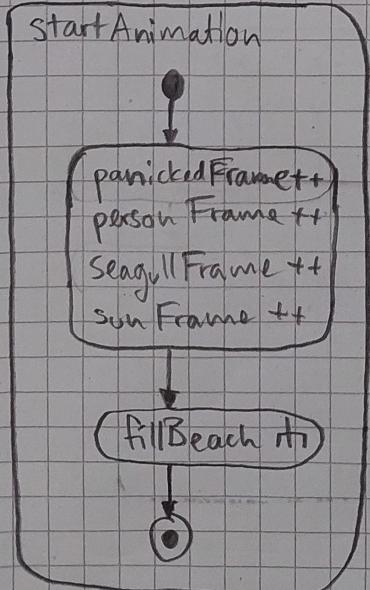
export

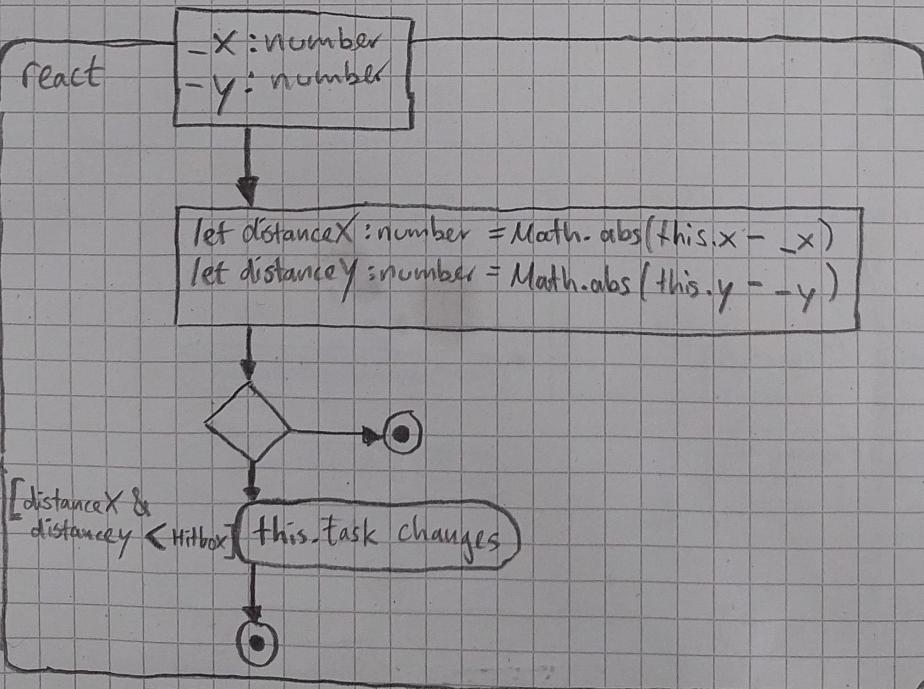
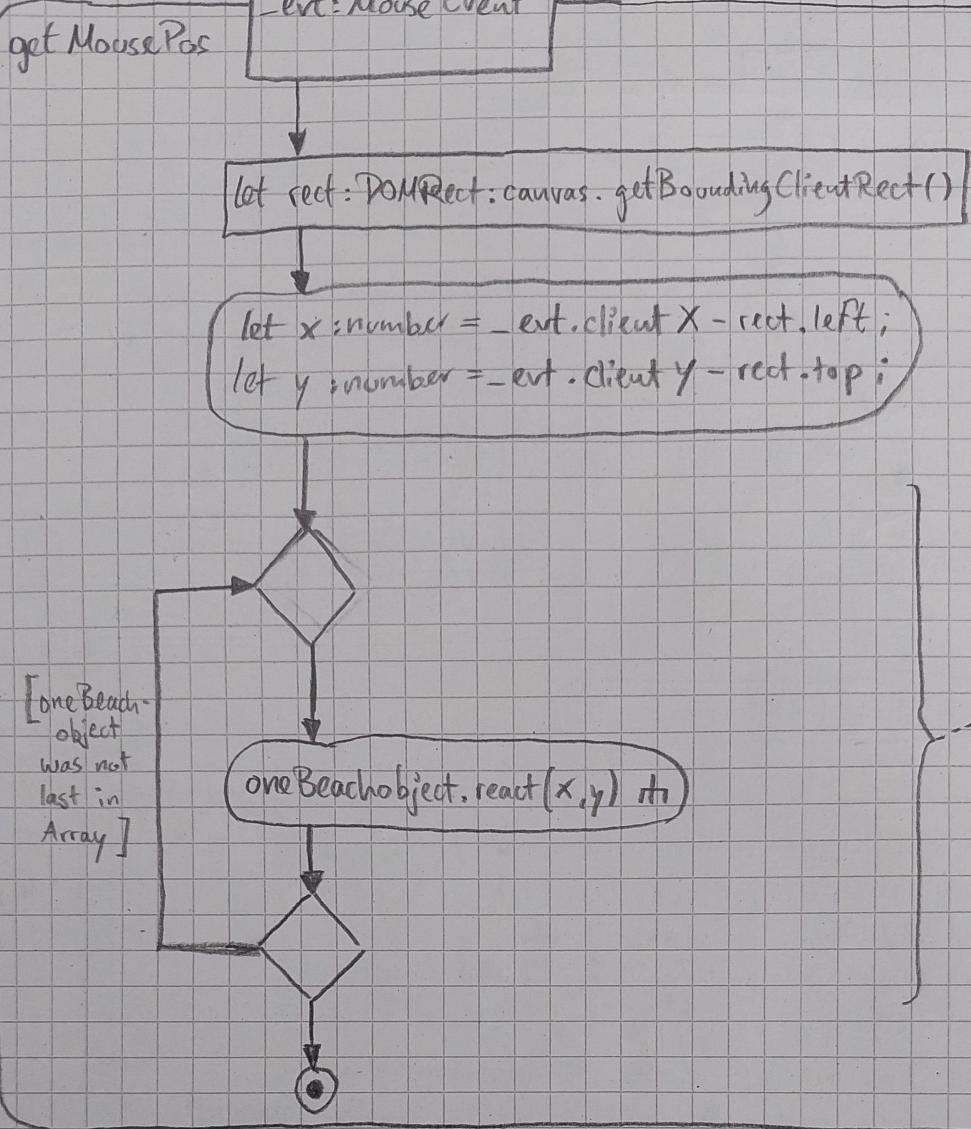
interface Vector	• let canvas: HTMLCanvasElement
x: number	• let cr2: CanvasRenderingContext
y: number	

- let sharkFrame: number = 0
- let panickedFrame: number = 0
- let personFrame: number = 100
- let seagullFrame: number = 300
- let sunFrame: number = 0

let allBeachObjects: Beachobject [] = []

install Canvas Click
listener





Background (drawBackground)

```
let gradient: CanvasGradient = crc2.createLinearGradient()  
gradient.addColorStop(0)  
gradient.addColorStop(0.5)  
gradient.addColorStop(0.6)  
gradient.addColorStop(1)  
crc2.fillStyle = gradient  
crc2.fillRect()
```

drawCloud (draw function in class Cloud)

```
let nparticles: number  
let radiusParticles: number  
let particle: Path2D = new Path2D()  
let gradient: CanvasGradient
```

```
particle.arc  
gradient.addColorStop(0)  
gradient.addColorStop(1)
```

Save

translate

restore transform

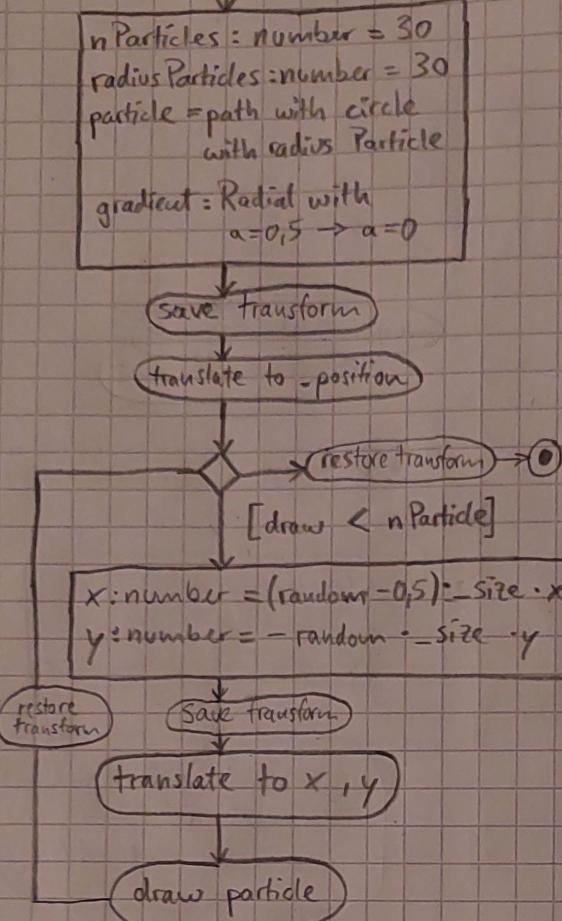
x: number(Math.random())
y: number(Math.random())

Save transform

draw particle

translate x,y

draw Clouds



draw Seagulls

X : getRandomNumber
Y : getRandomNumber

for (var s = 0; s < 4;
s++)

draw Person

position X : getRandomNumber
position Y : getRandomNumber

Save

translate (-position x
-position y)

Restore

panickedPerson

position X : getRandomNumber
position Y : getRandomNumber

reset Transform

Save

translate (-position x
-position y)

Restore

draw SharfinFull

position X : getRandomNumber
position Y : getRandomNumber
let path = new Path2D
var

fillStyle
+ fill(path)

Restore

Wave Position

Save

translate

restore

- position x
- position y