



# SpaceWorld: Scribble



# Space: Activity Diagram I



## Space : Activity Diagram II

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

drawRandom  
01

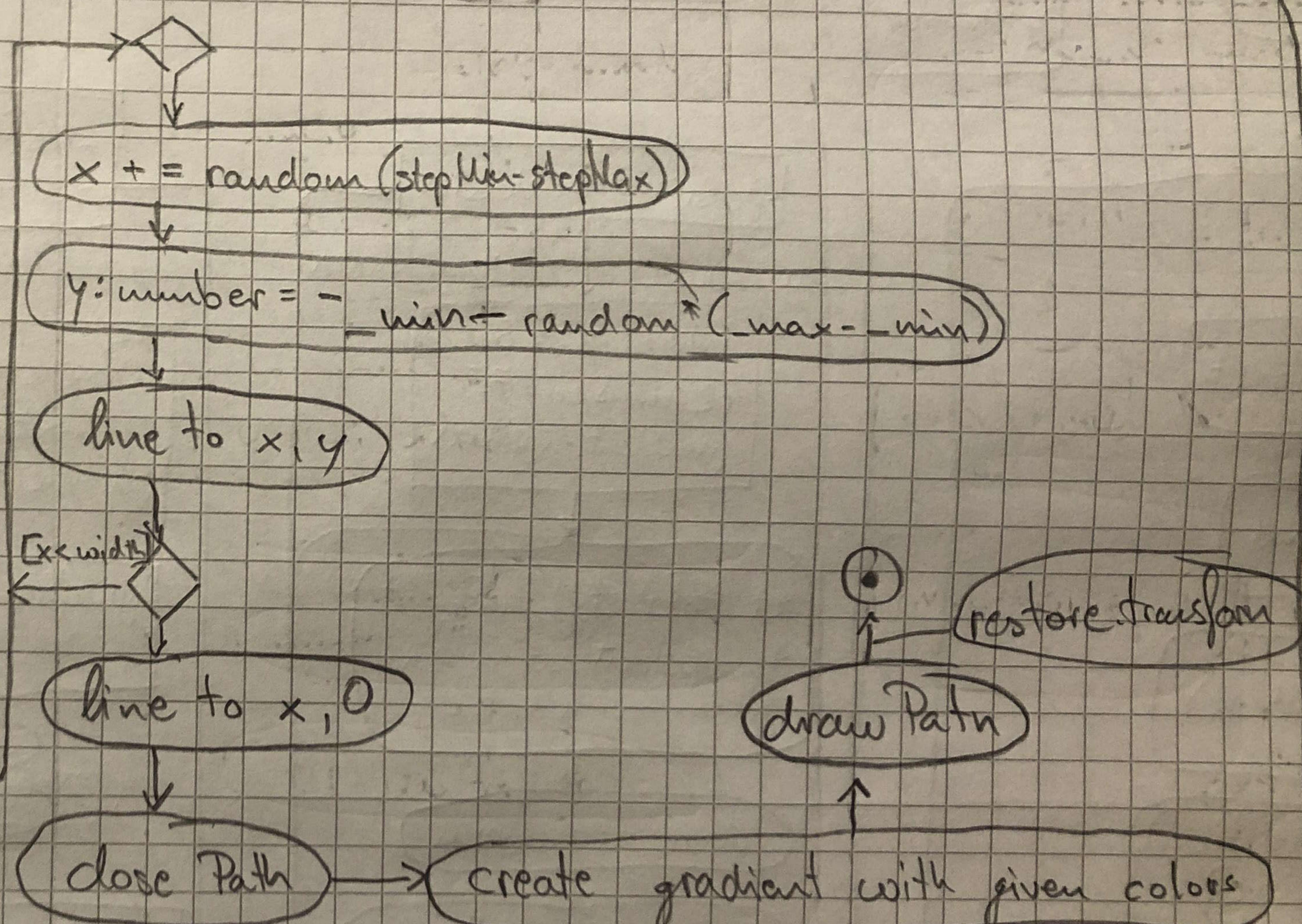
stepMin: number = 10  
stepMax: number = 50  
x: number = 0

Save Transform


translate to -position

moveto 0,0

line to 0,±max



# Space : Activity Diagram III



## U.F.O.s : Class Diagram I

Canvas Rendering Context

U.F.O

position: Vector  
velocity: Vector  
type: number  
size: number

constructor (-size:number)  
move (-timeslice :number): void  
draw (): void

Vector

x: number  
y: number

constructor(x: number, y: number)  
set (-x: number, -y: number): void  
scale (-factor: number): void  
add (-addend: Vector): void

Clouds

position: Vector  
velocity: Vector  
x: number  
q: number  
constructor (-x: number; -velocity: Vector(6, ))  
draw Cloud

## U.F.O.s : Activity Diagram I

- position: Vector  
- size: number

constructor

position to -position

Set velocity to random direction and random length:  $l \geq \min < \max$

randomly choose type

Set size to -size

draw

save. transform

- timeslice: number

move

add velocity \* -timeslice to position

position component(s)

subtract canvas dimension from component

add canvas dimension to component

position component > canvas dimension

draw

translate to position


scale to size

draw path representing type


restore transform

fill style

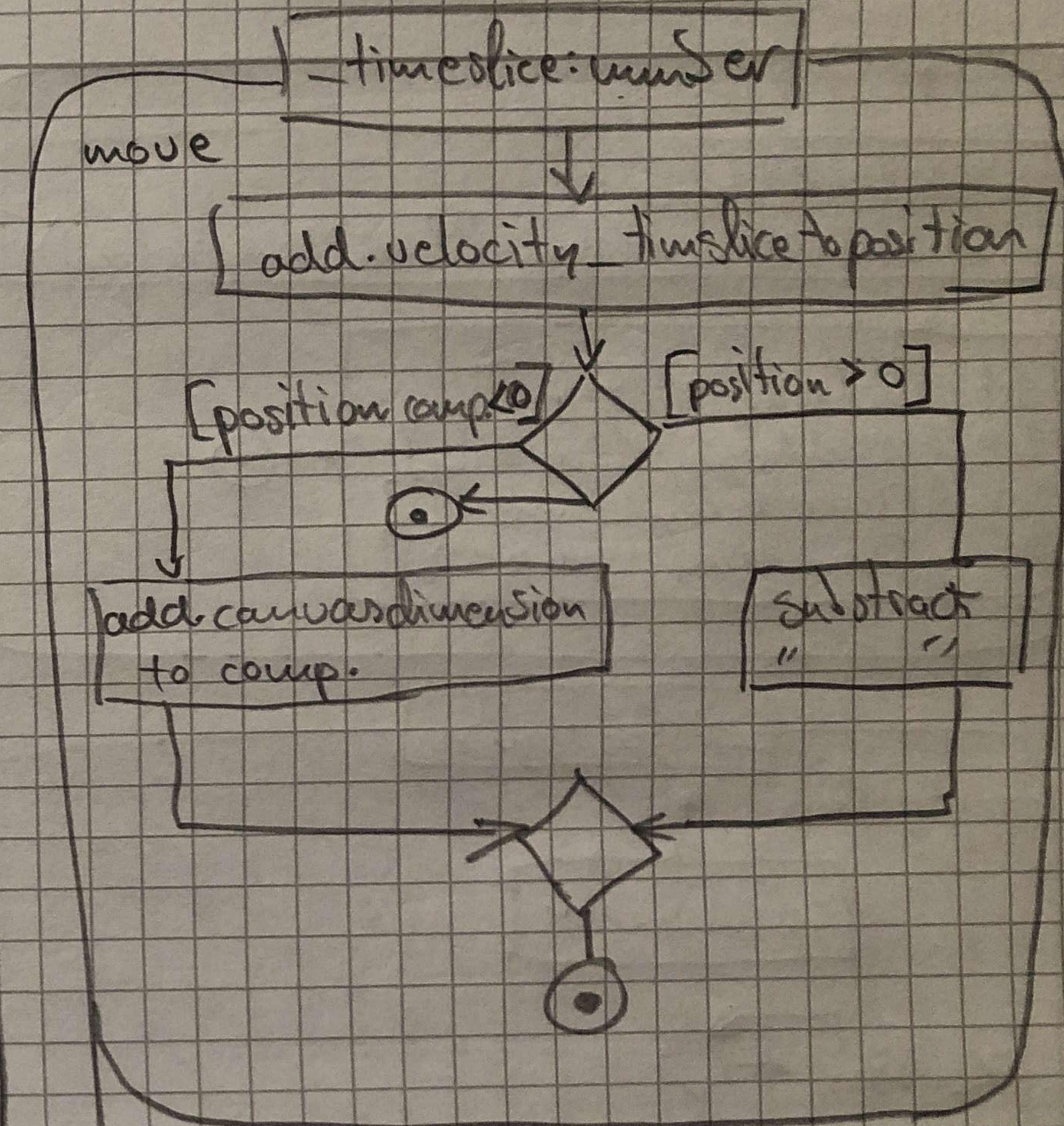
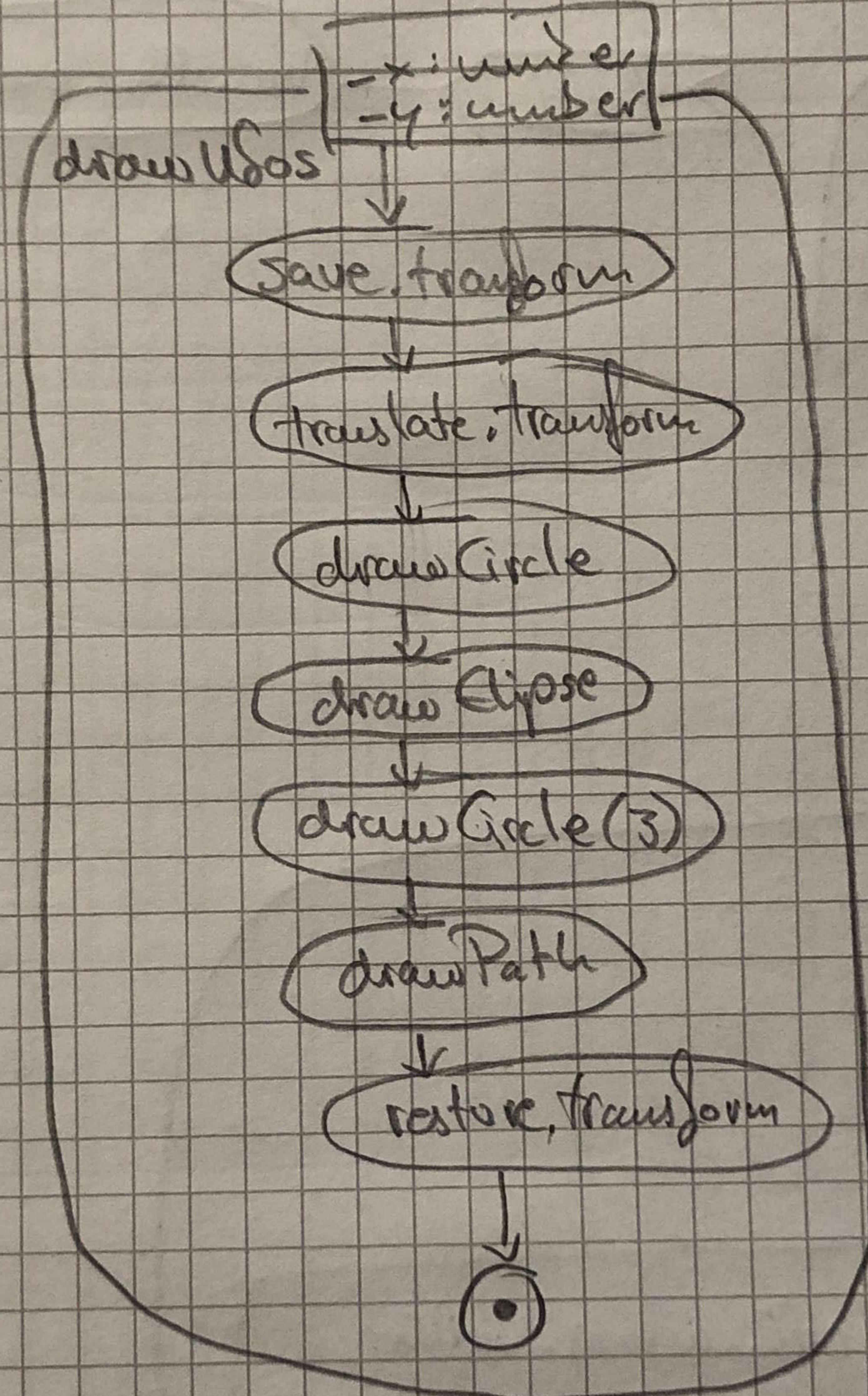
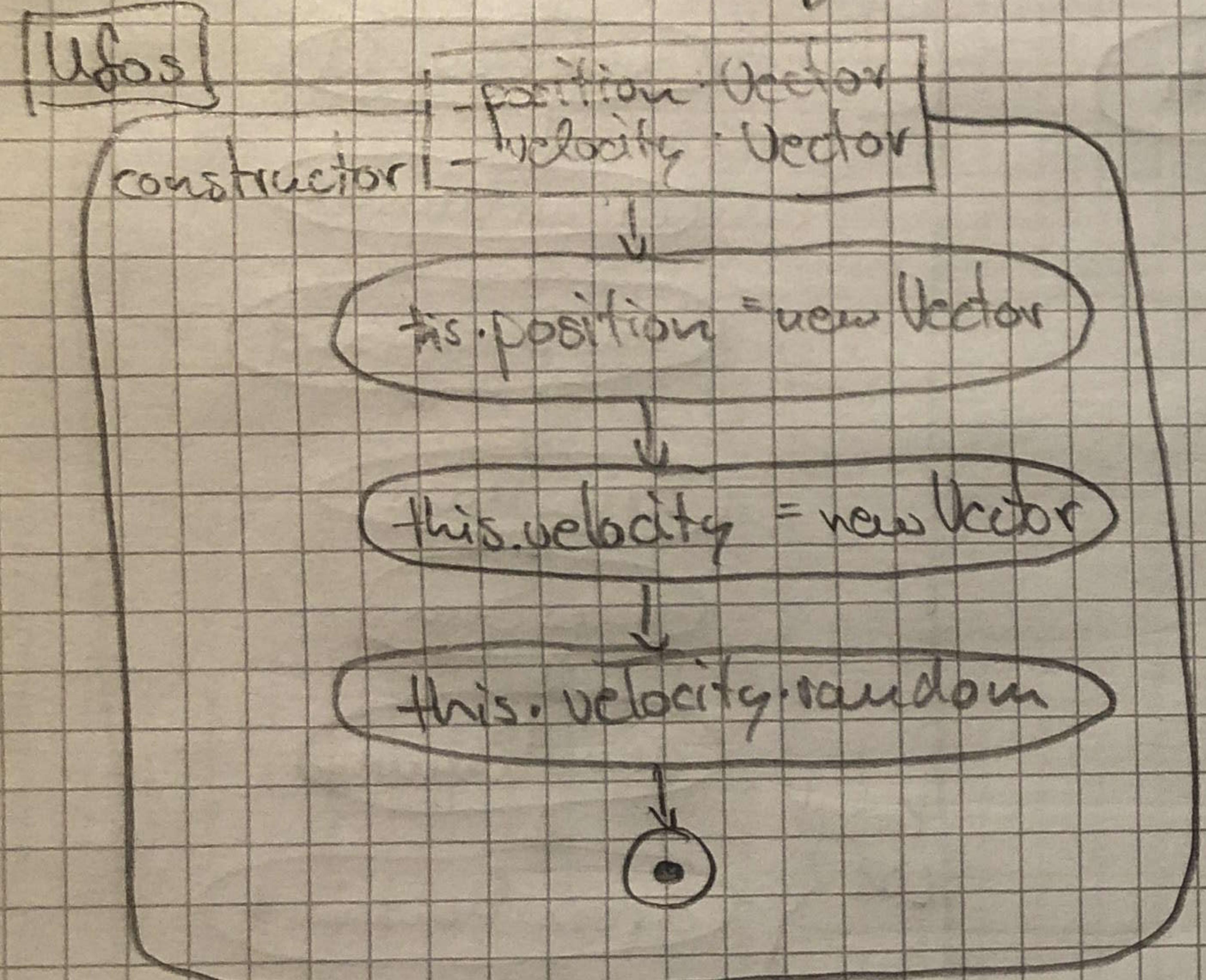
# Space: Anwendungsdiagramm



# Space : UI - Scribble



## Space: Activiti diagram I



## Space : Activity Diagram II (LOG2.ts)

install load listener:

```
window.addEventListener("load", handleLoad)
```

```
let ufoArray : Ufos []
let cloudArray : Cloud []
get xCloudArray : number []
let yCloudArray : number []
```

Ufos : number

createUfos

```
i : number = 0
```

for i in Ufos

```
let randomXUfo : number = math.random(1) * canvas.width
let randomYUfo : number = math.random(1) * canvas.height
let ufoPosition : Vector = new Vector(randomXUfo, randomYUfo)
let ufoVelocity : Vector = new Vector(70, 0)
let ufo : Ufo = new Ufo(ufoPosition, ufoVelocity)
```

ufoArray.push(ufo) → 0

> load → handleLoad (t)

handleLoad

createCloud (t)

createUfos (5)

drawBackground

drawMars

drawSun

drawRunway

Kontakt  
L08

drawMountain1

drawMountain2

drawLights

createCloudxy()

createCloud(t)

createUfos(t)

(60/sec)

window.setInterval(60)