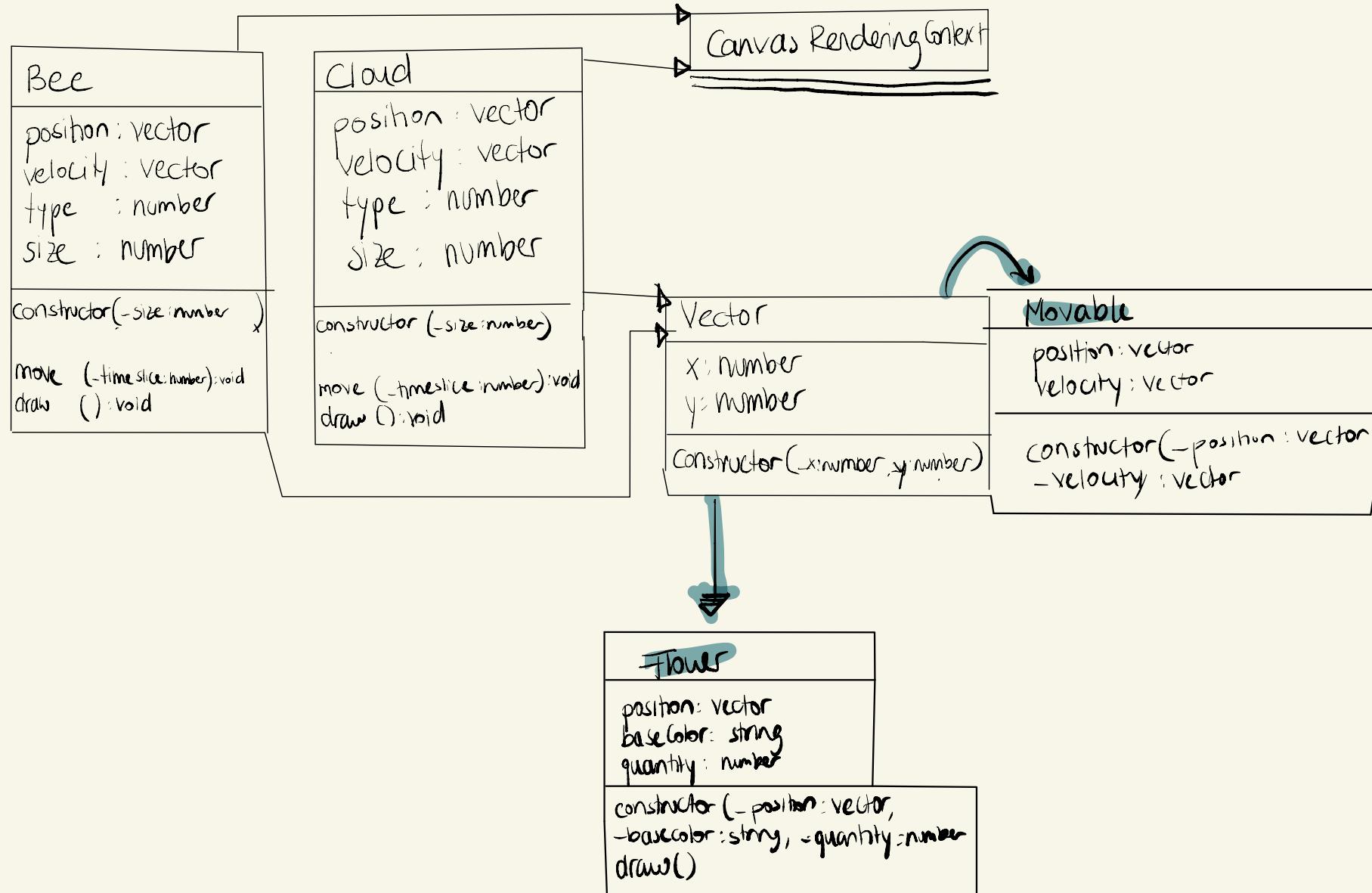
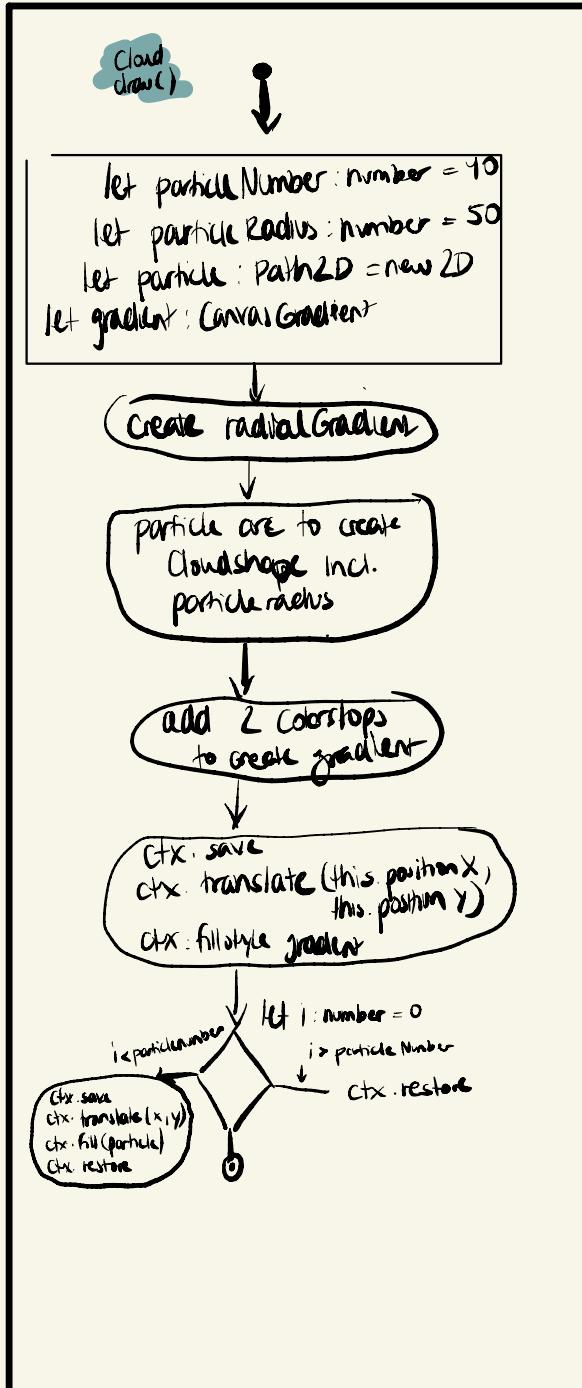
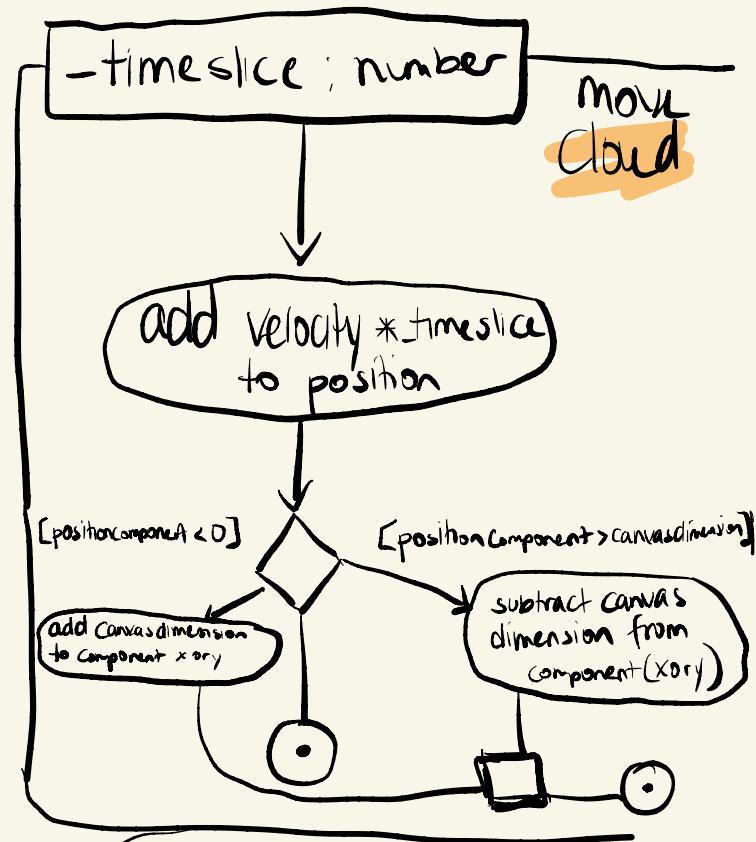


# Polymorphie

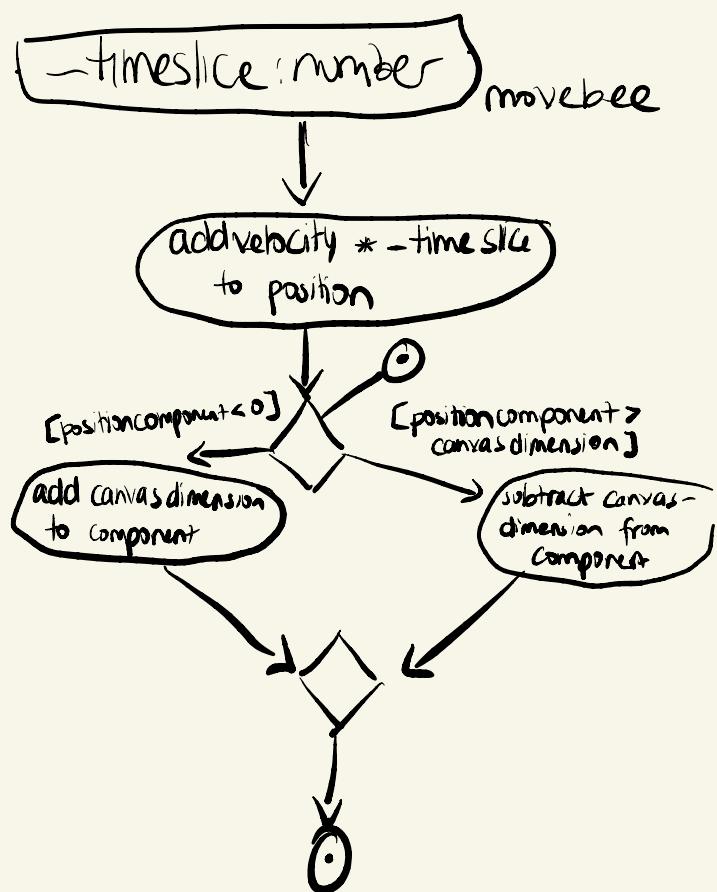
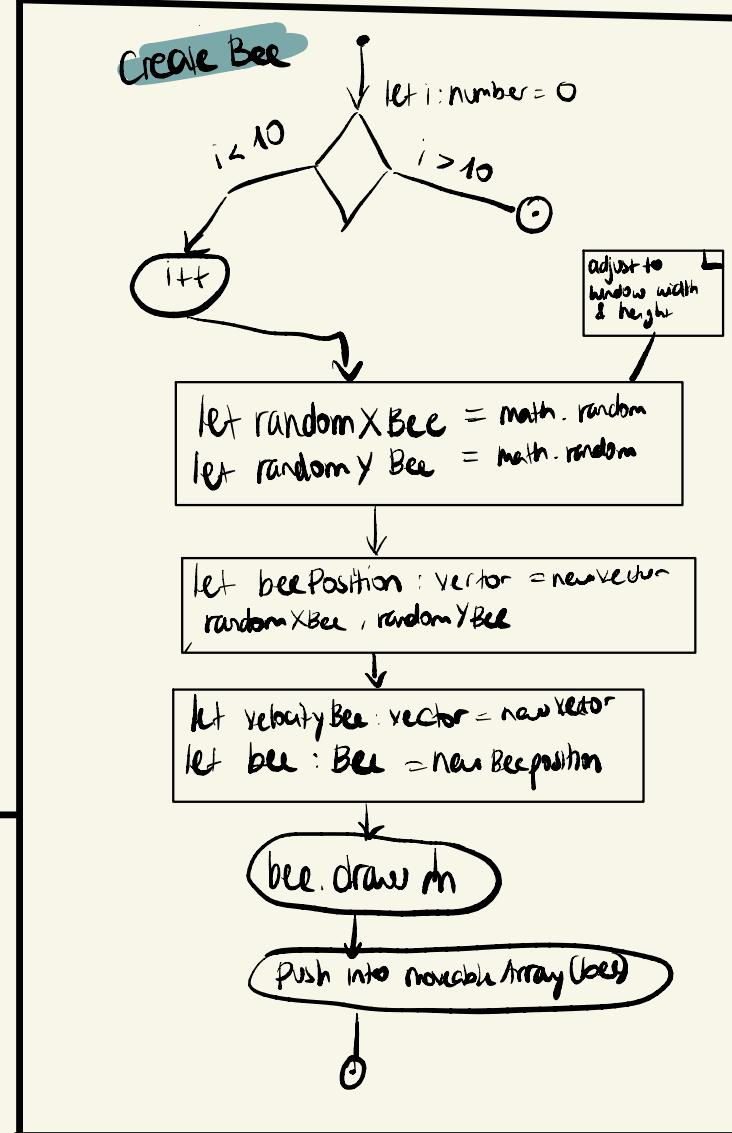
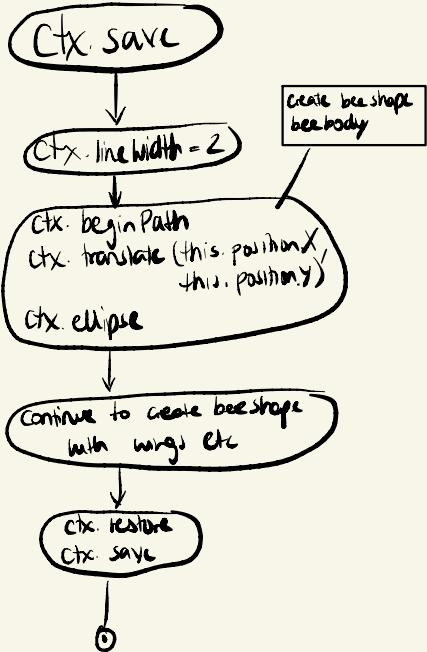
## Blumenwiese

= neu hinzugefügt

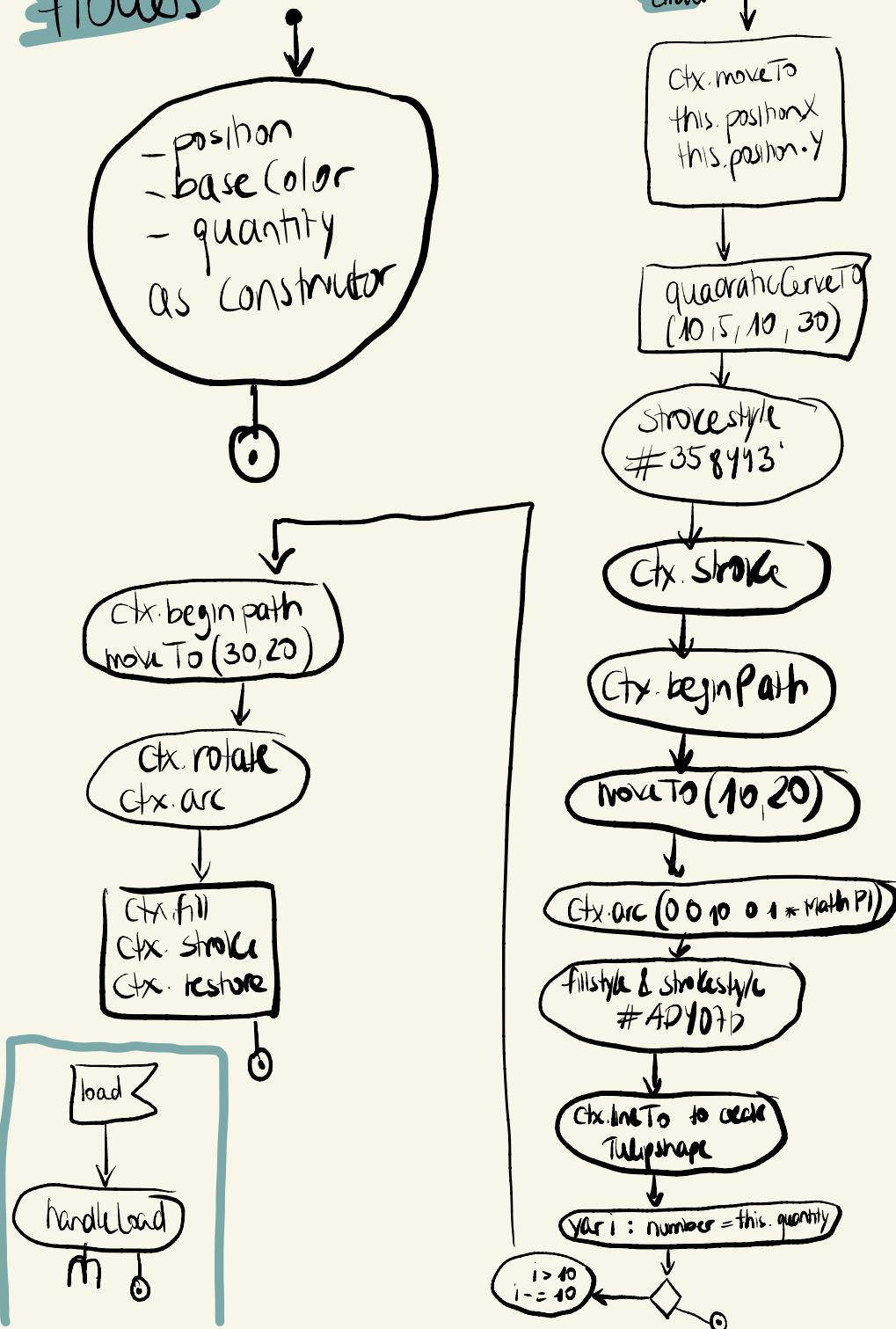




draw  
Bee



# flowers



main =

```

let horizon: number
let flowerArray: number []
let xCloudArray: number []
let yCloudArray: number []
let moveableArray: number []

```

export

```

let VectorMountains
x: number
y: number

```

export let ctx: CanvasRenderingContext2D

handleLoad

```

canvas.width = window.innerWidth
canvas.height = window.innerHeight

```

```

horizon = window.innerHeight * 0.6

```

drawTulip

CreateBee

setInterval  
to 100

window.setInterval

DrawBackground

## drawBackground

Create linear gradient  
(0,0,0,ctx.canvas.height)

Add 4 gradient  
Colorstops

ctx.fillStyle = gradient  
ctx.fillRect(0,0,canvas.width  
          , canvas.height)

for let i: number = 0  
    ↓  
    i < flowerArray.length  
    i++  
    draw()  
    ↓

for let moveable of moveableArray

moveable.draw  
    ↓

Draw Moon  
    ↓

## Draw Moon

```
let r1: number = 30  
let r2: number = 180
```

```
let gradientSun: CanvasGradient  
= ctx.createRadialGradient
```

Add 3 Colorstops  
with gradientSun

```
let X: number = ctx.canvas.width  
let Y: number = ctx.canvas.height / 2 - 100
```

ctx.SAVE

ctx.translate  
ctx.fillStyle = gradientSun  
ctx.arc

ctx.fill  
ctx.restore

## moveable

Export class  
Moveable

position: vector  
velocity: vector

Create different move patterns  
based on:

this.position.X  
this.positionY

window.innerWidth  
window.innerHeight

Create 2 situations:  
true & false  
(if loop)