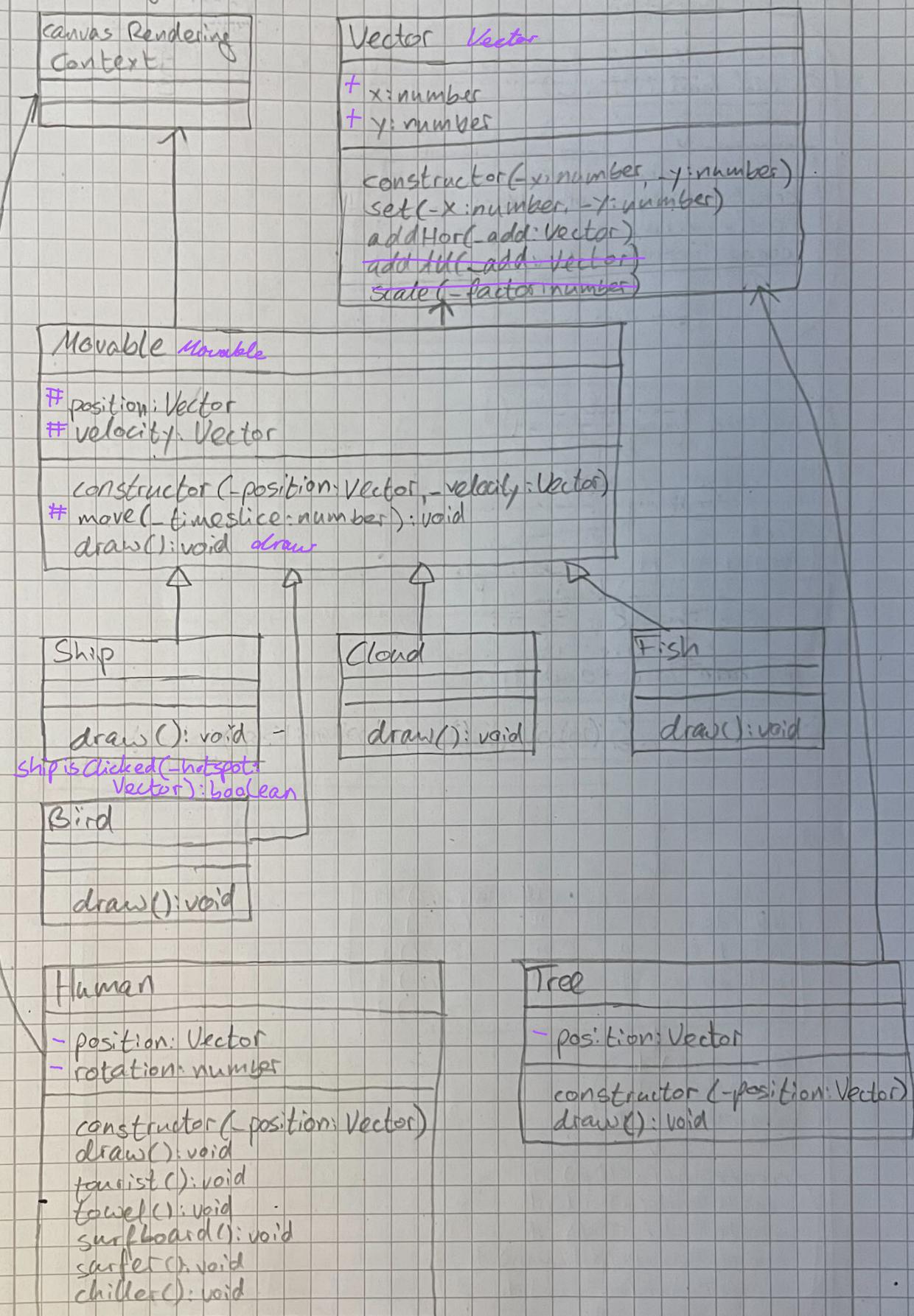


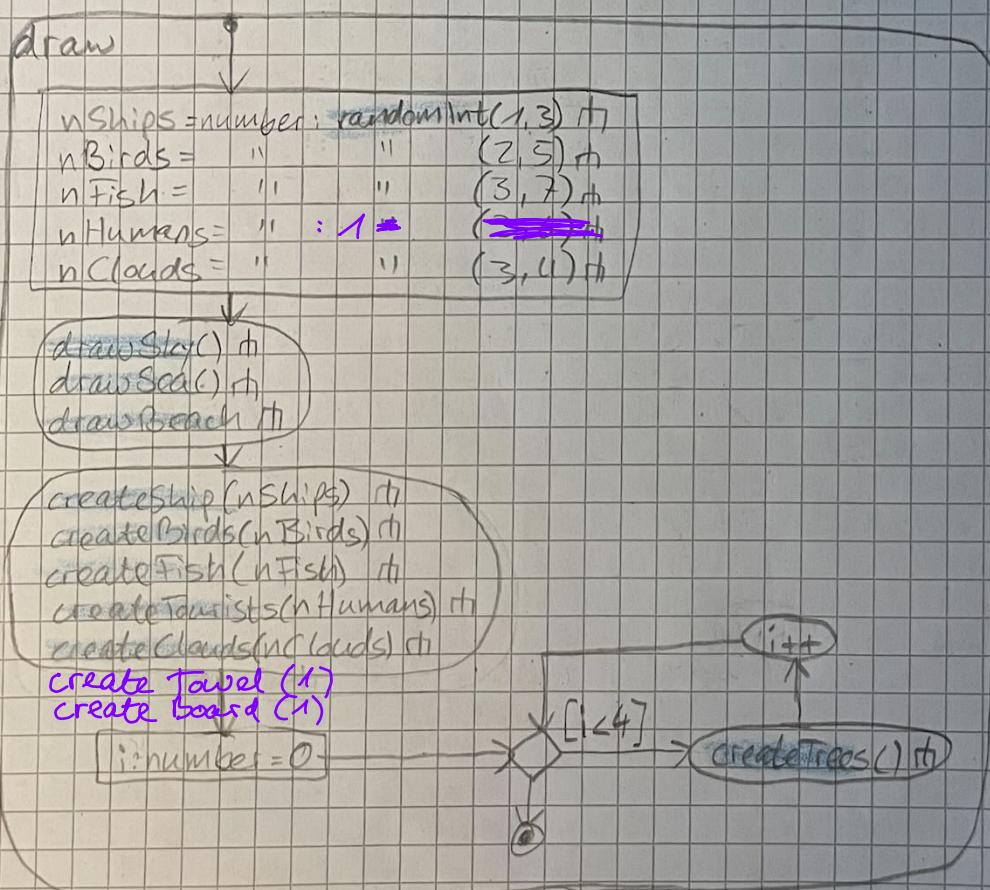
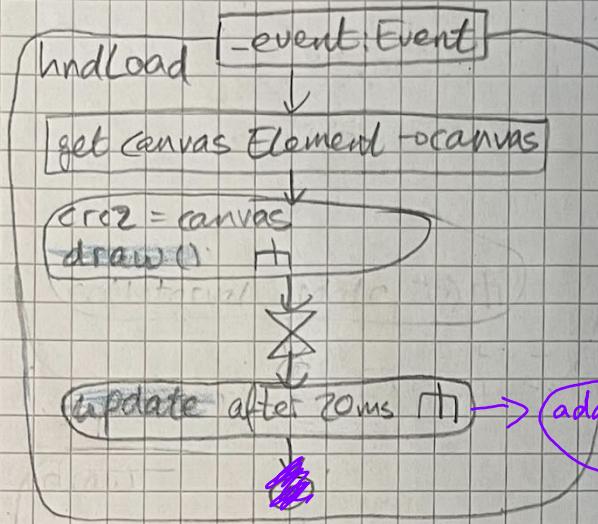
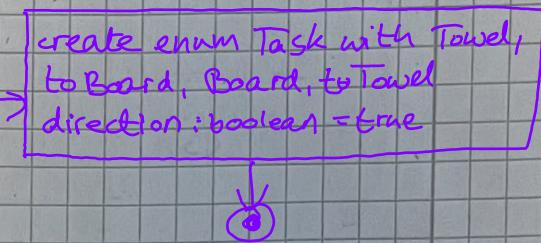
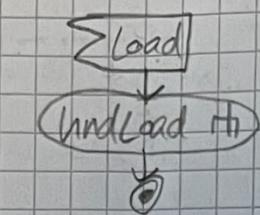
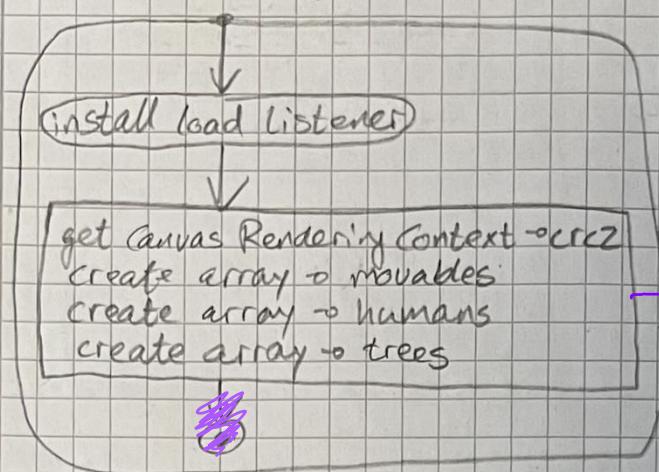
## L10.2 Strand: Polymorphism

### Class Diagram



# Activity Diagram

Main



Activity  
Diagram  
Main

randomInt

-min: number, max: number

return Math.floor(Math.random() \* (max - min)) + min

update

drawSky ()  
drawSea ()  
drawBeach ()

movable = first value of movables

movable.move(1/50)  
movable.draw()

movable = next  
value in movables

tourist = first value of humans

tourist.move(1/200)  
tourist.draw()

tourist = next  
value in humans

tree = first value of trees

(tree.draw())

(tree = next  
value in trees)

drawSky

save cccz

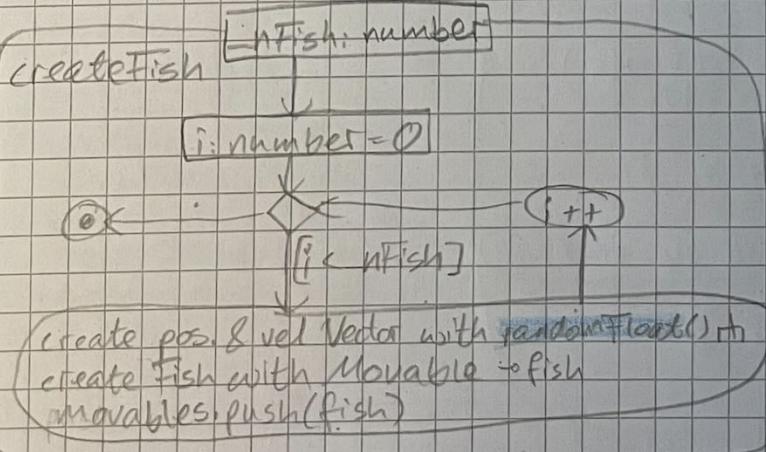
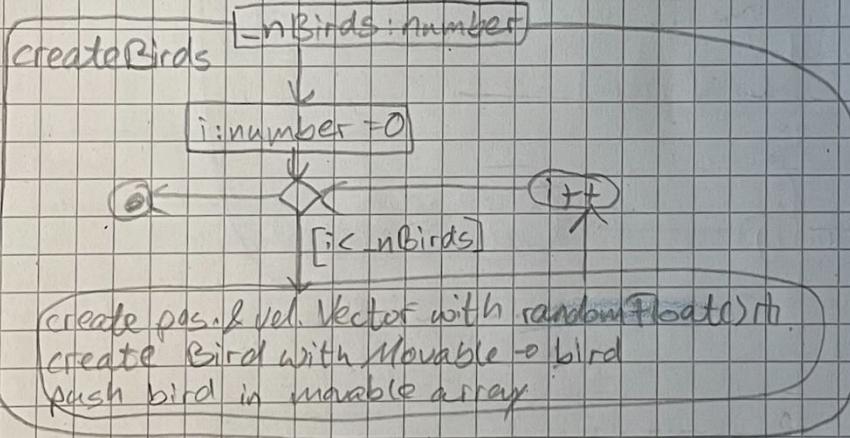
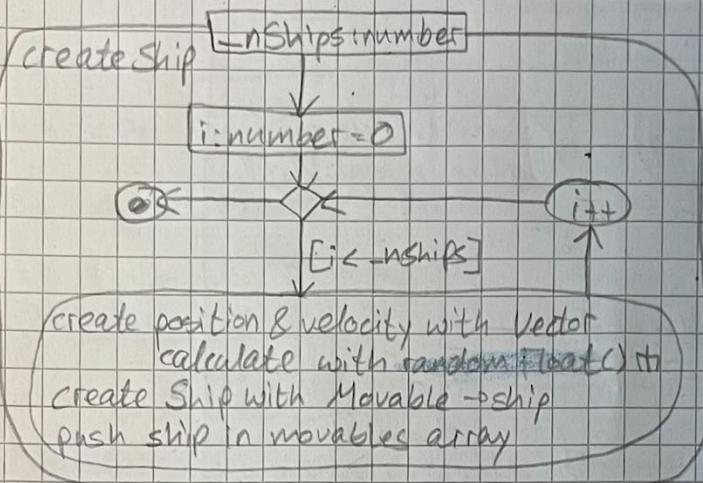
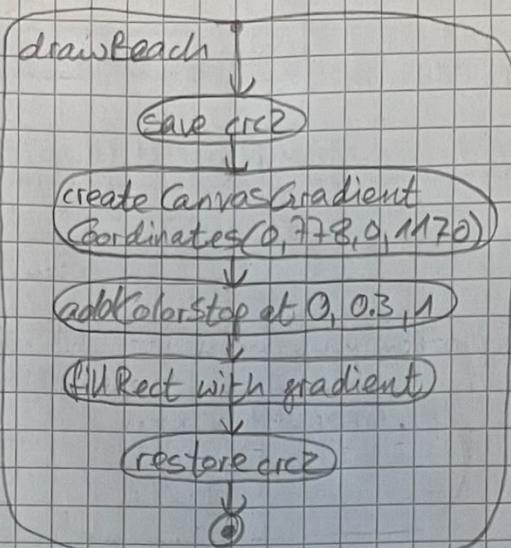
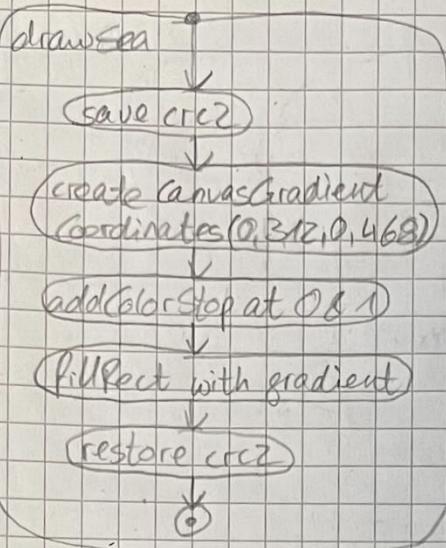
Create CanvasGradient with  
coordinates (0,0,0,312)

addColorStop at 0 & 1

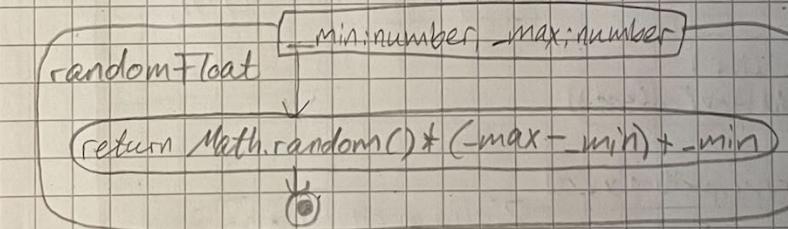
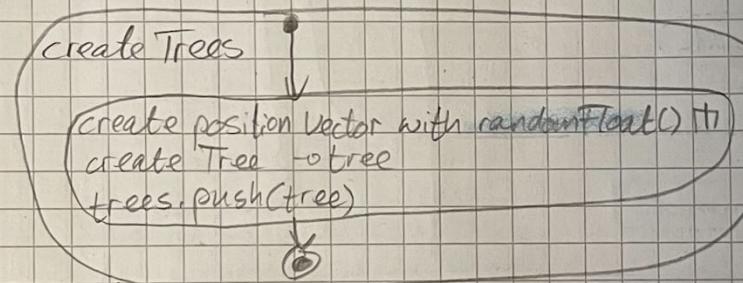
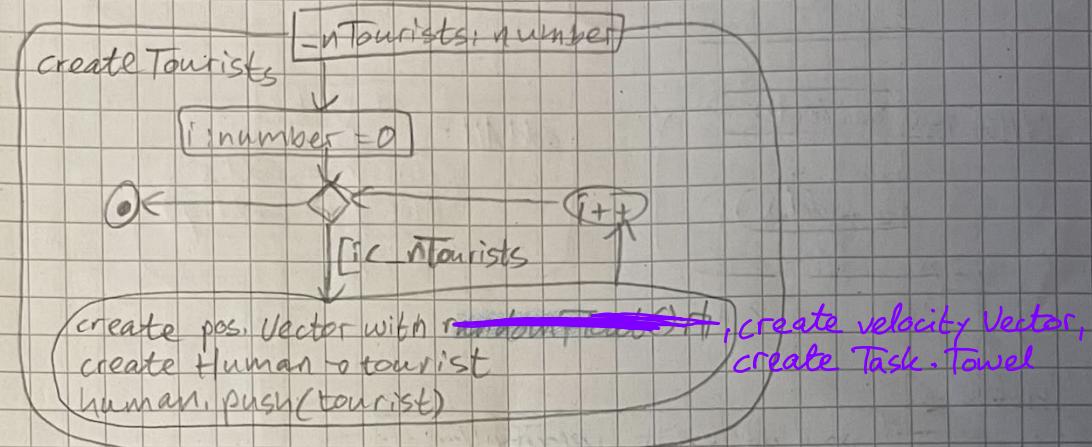
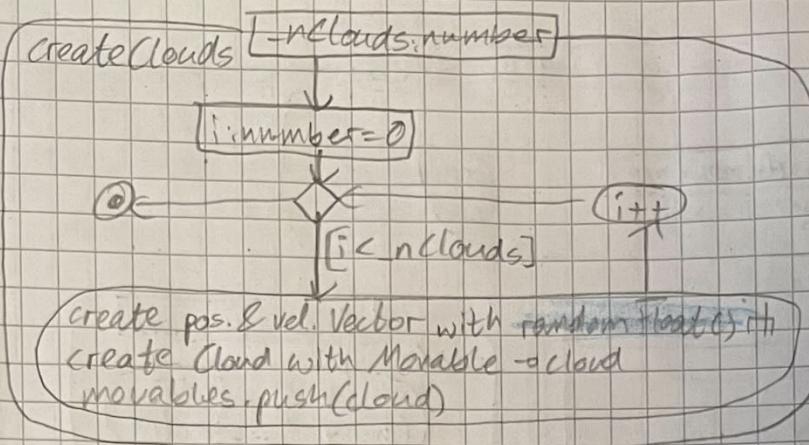
fillRect with fillStyle = gradient

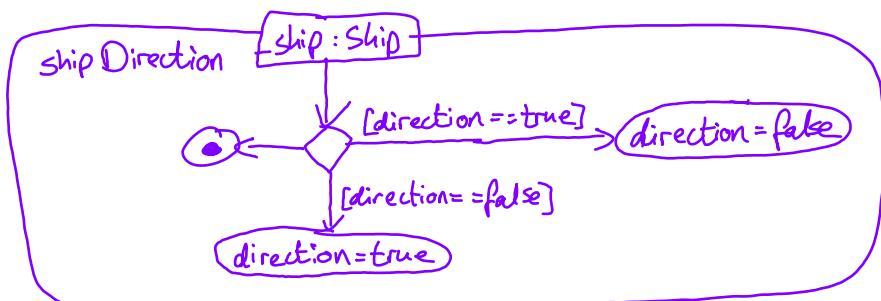
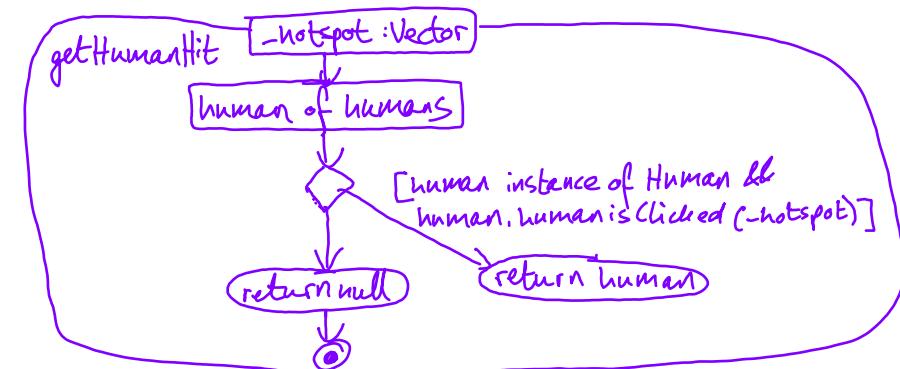
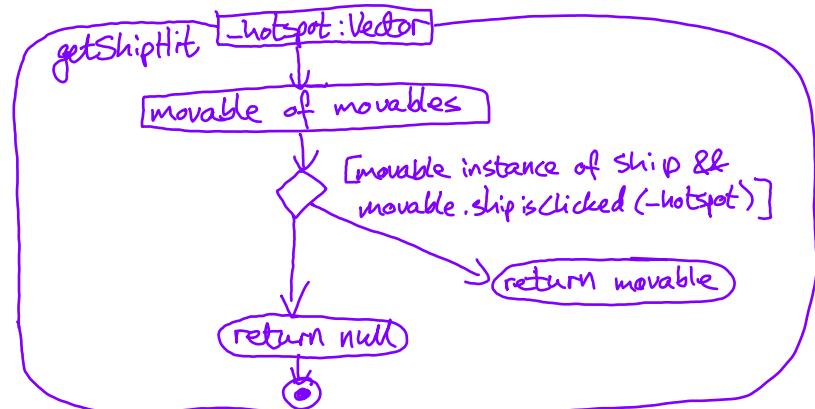
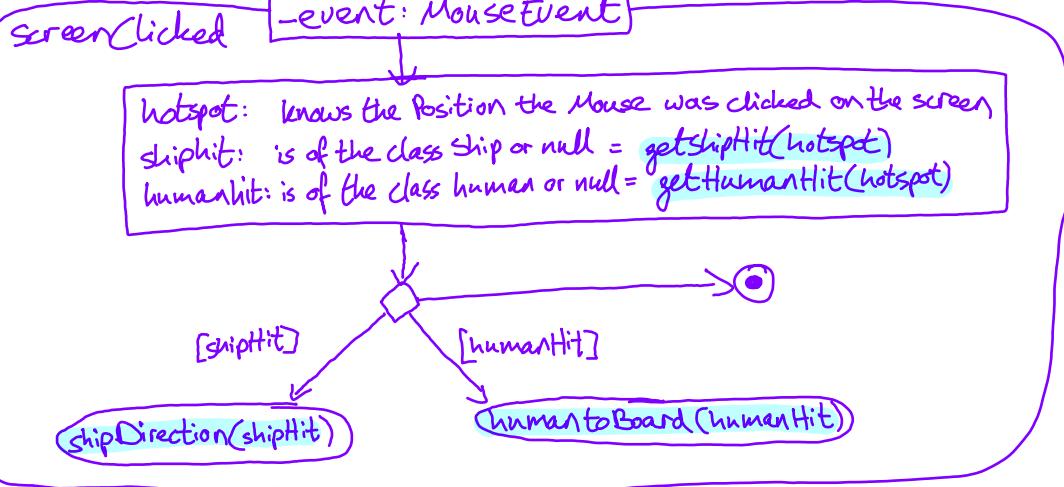
restore cccz

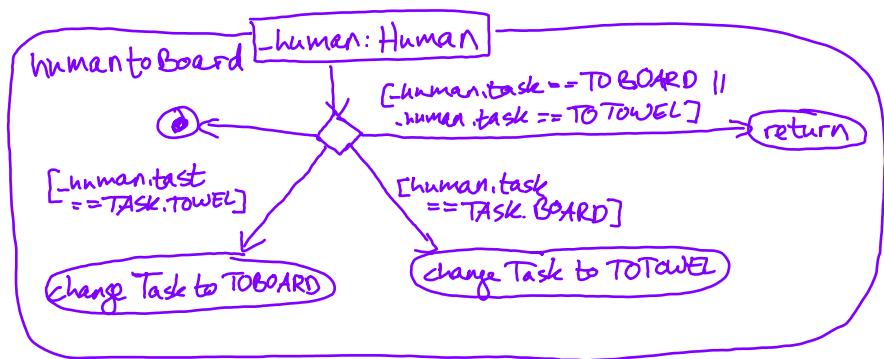
Activity  
Diagram  
Main



Activity  
Diagram  
Main

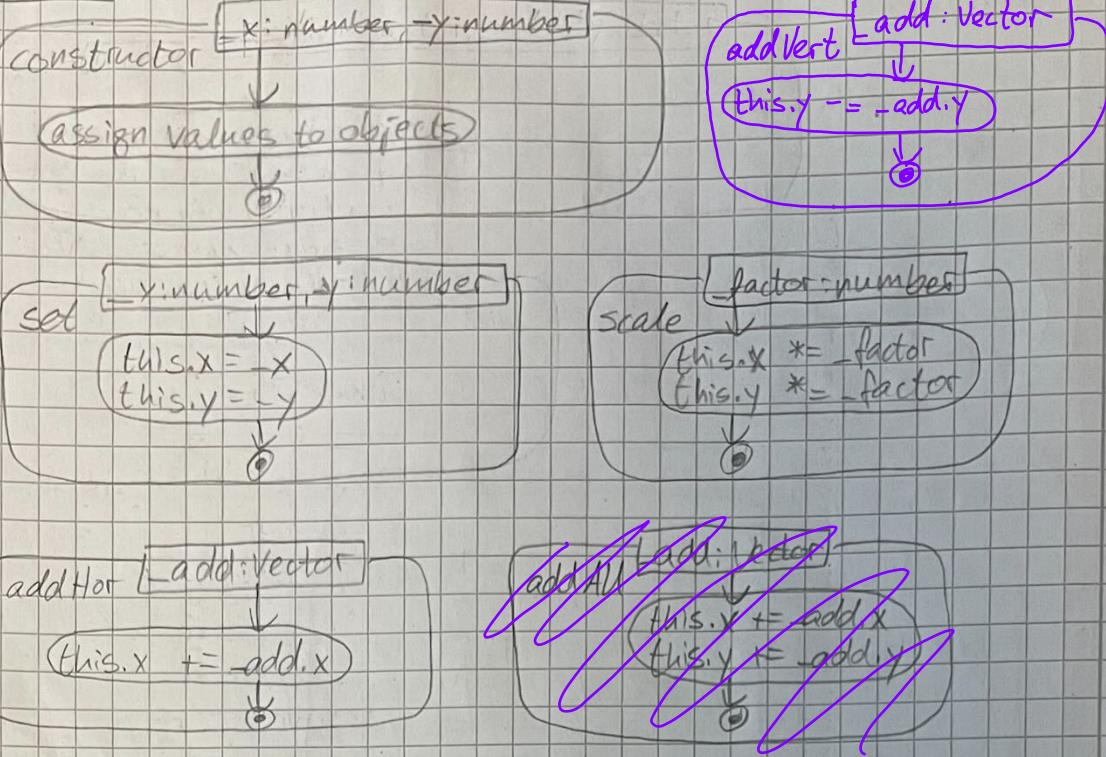




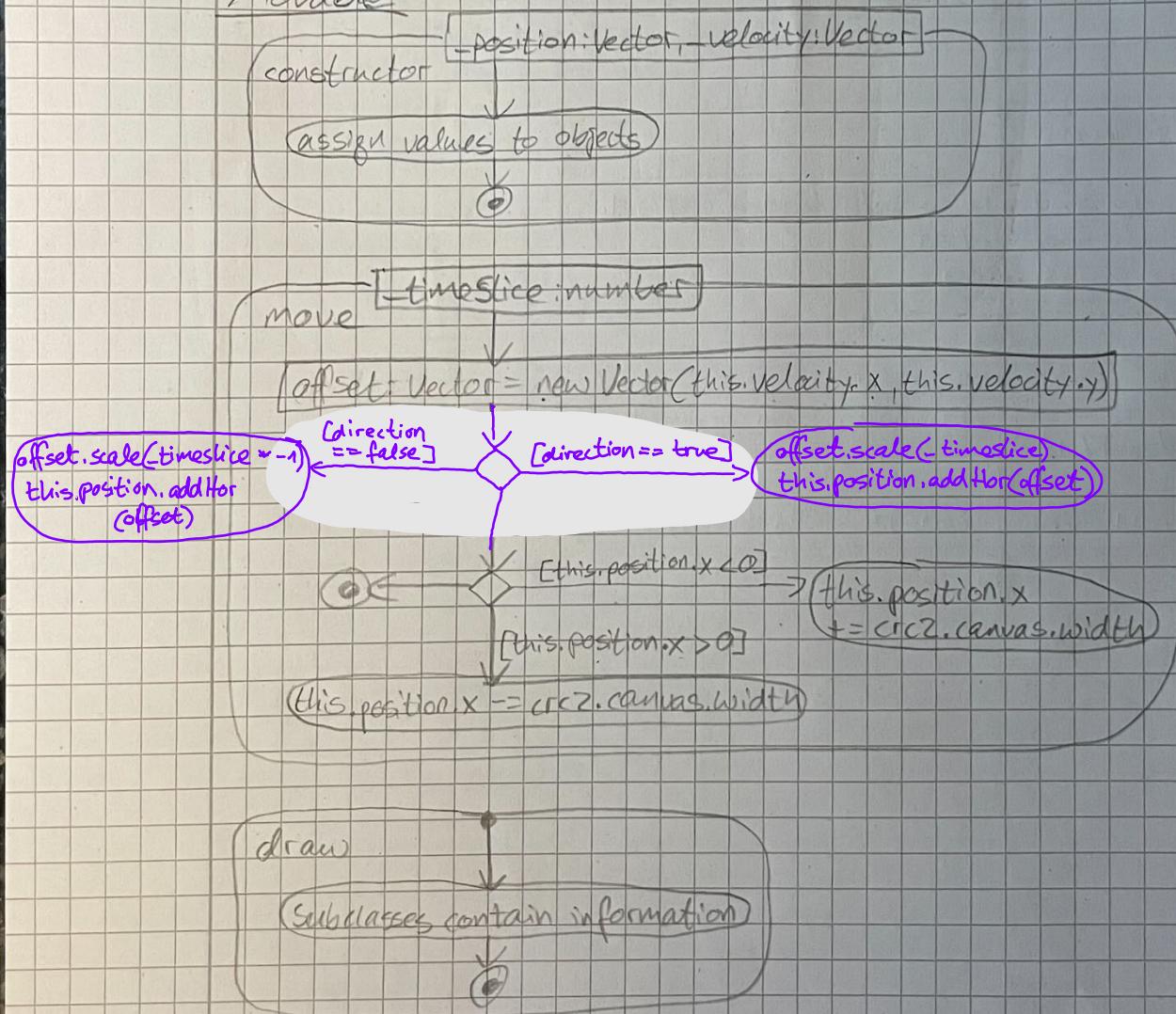


# Activity Diagram

## Vector

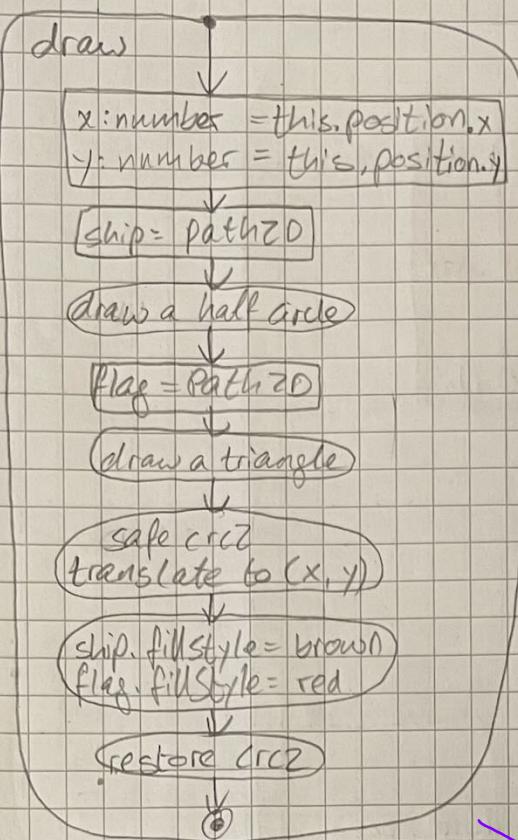


## Movable

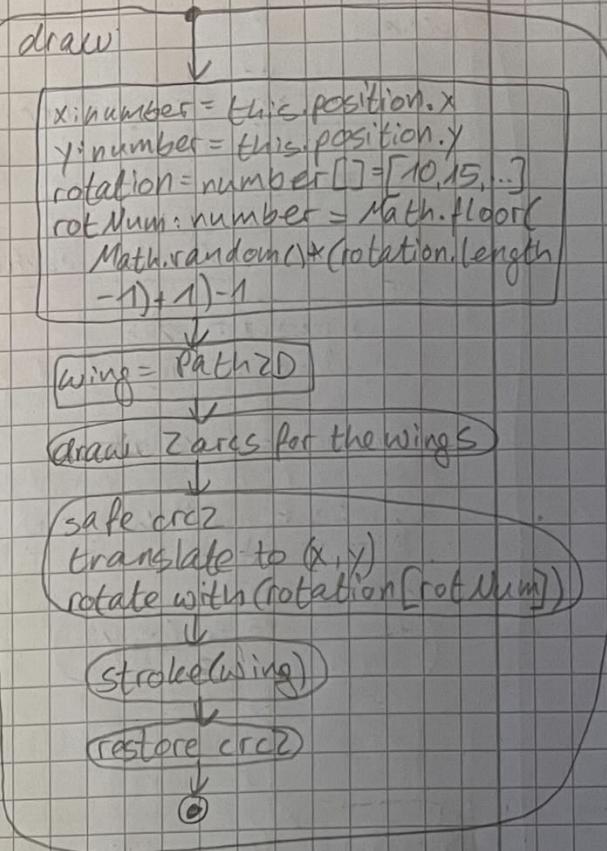


# Activity Diagram

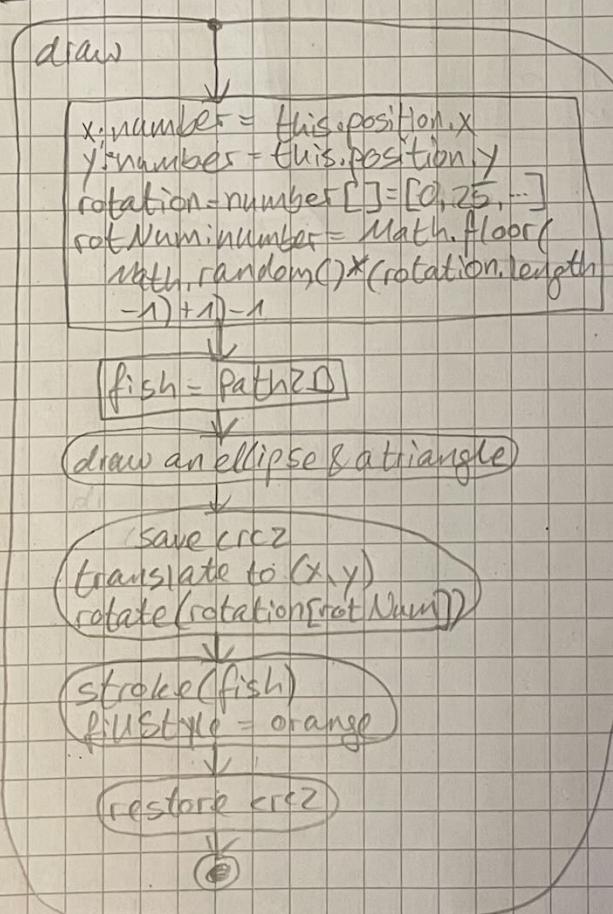
## Ship



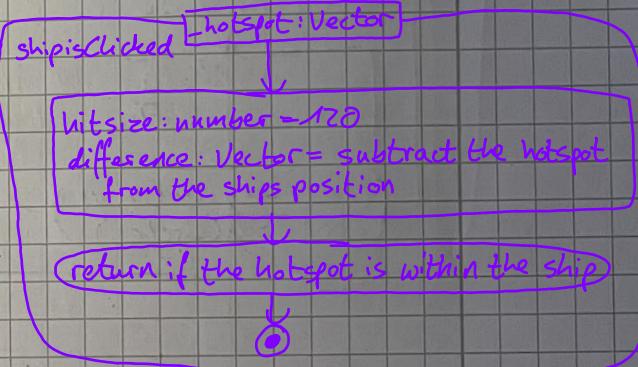
## Bird



## Fish



## Ship



Activity  
Diagram

Cloud

draw

```
x:number = this.position.x  
y:number = this.position.y  
ncirclesmin:num = 8  
ncirclesmax:num = 15  
ncircles = Number inbetween min & max  
r:num = 65  
circles: Path2D  
gradient: canvasGradient = (0,0,0,0,i)
```

draw arc for circle  
gradient.addColorStop at 0.  
gradient.addColorStop at 1

Save crc2  
translate to (x,y)  
fillStyle = gradient

[drawn:number = 0]

(drawn++

↓ [drawn < ncircles]

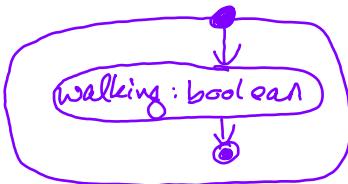
Save crc2

```
x:num = (Math.random() - 0.5) * 200  
y:num = (Math.random()) * 10
```

translate to (x,y)  
crc2.fill(circles)

restore crc2

## Human



constructor - position: Vect., - velocity: Vect., - task: TASK

this.position = -position  
this.velocity = -velocity  
this.task = -task

draw

x:num = this.pos.x  
y:num = this.pos.y

crcz.save()  
crcz.translate(x,y)

[Walking == true] → crcz.save()  
crcz.rotate from left to right

Tourist.Path2D

Create body with strokes

head : Path2D

Create circle for head

crcz.save()  
crcz.stroke  
.fillStyle(beige)  
.fill  
.restore()

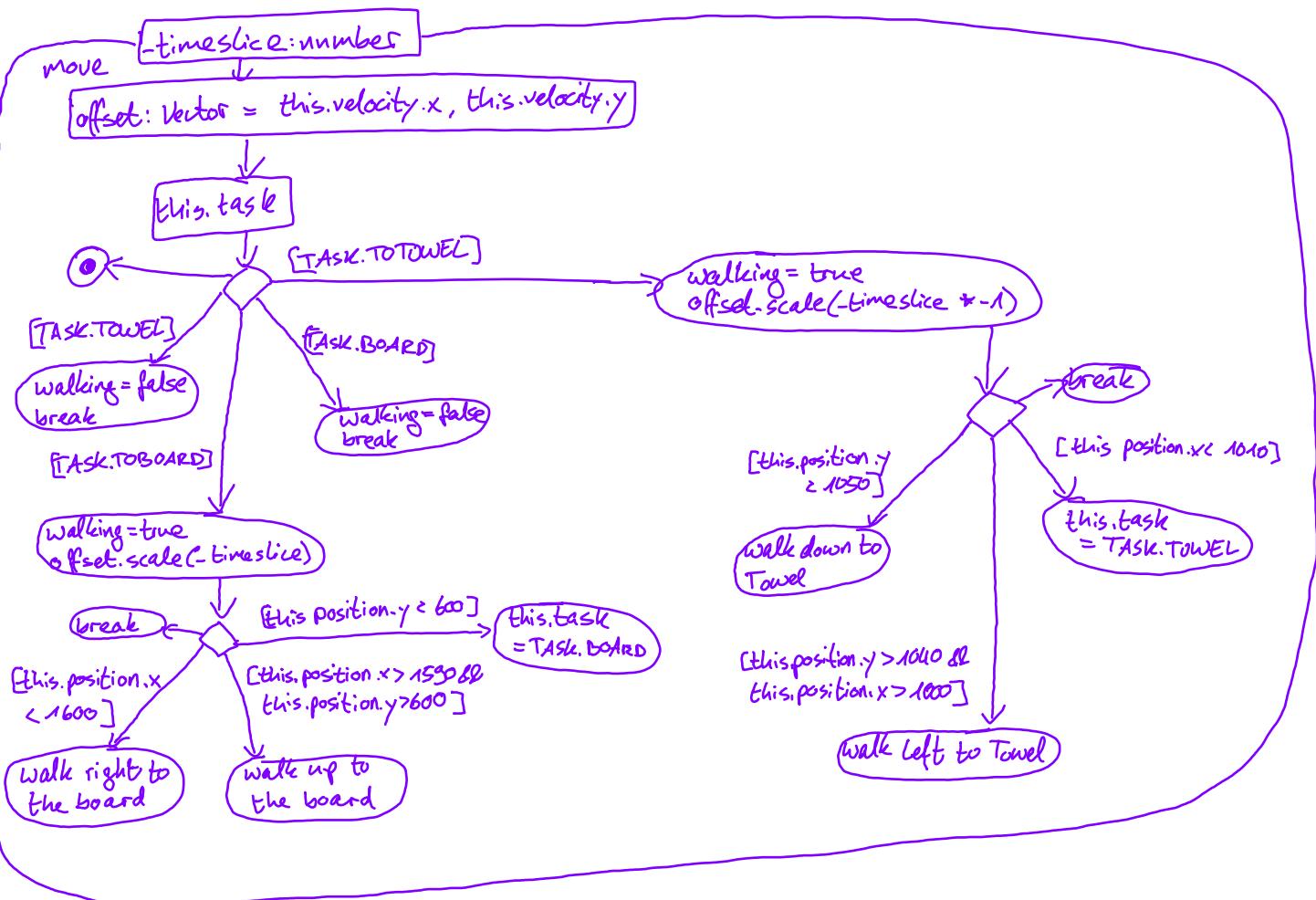
human is Clicked

hotspot: Vector

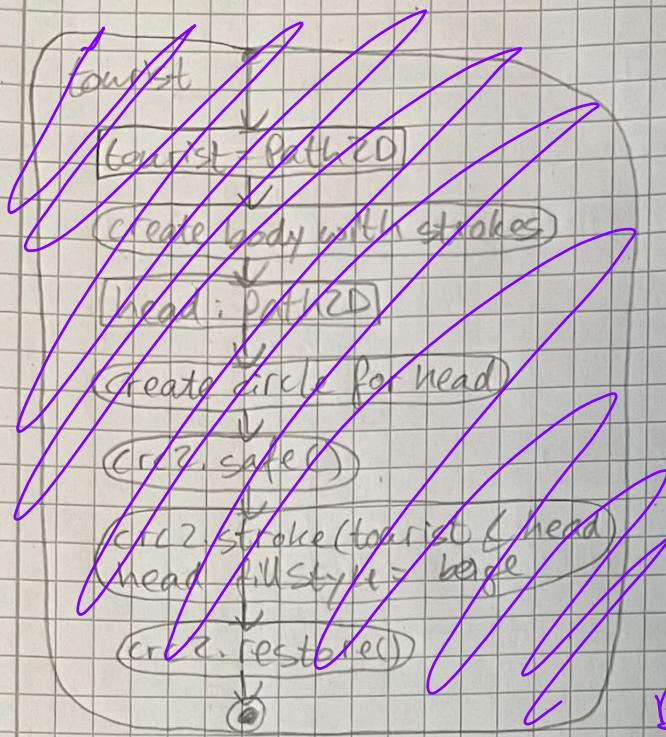
hitsize: num = 150

difference: Vector = subtract the hotspot from  
the humans position

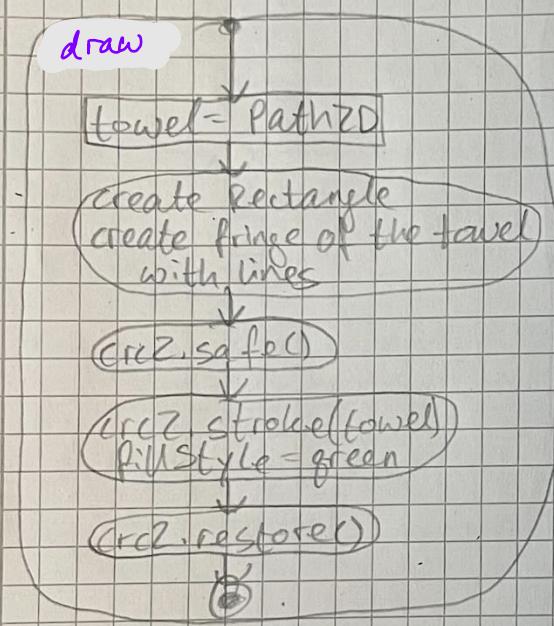
return if the hotspot is within the human



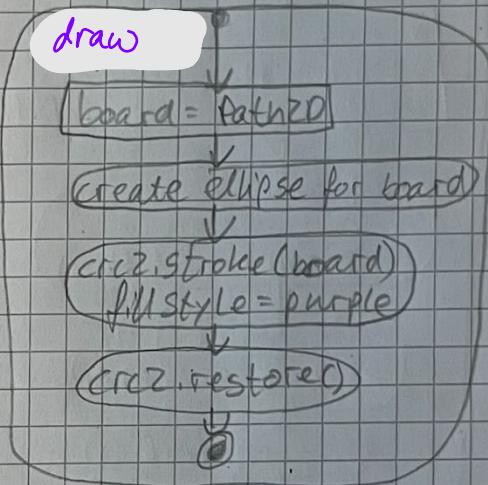
Activity  
Diagram  
Human



Towel



Board



# Activity Diagram

Tree

