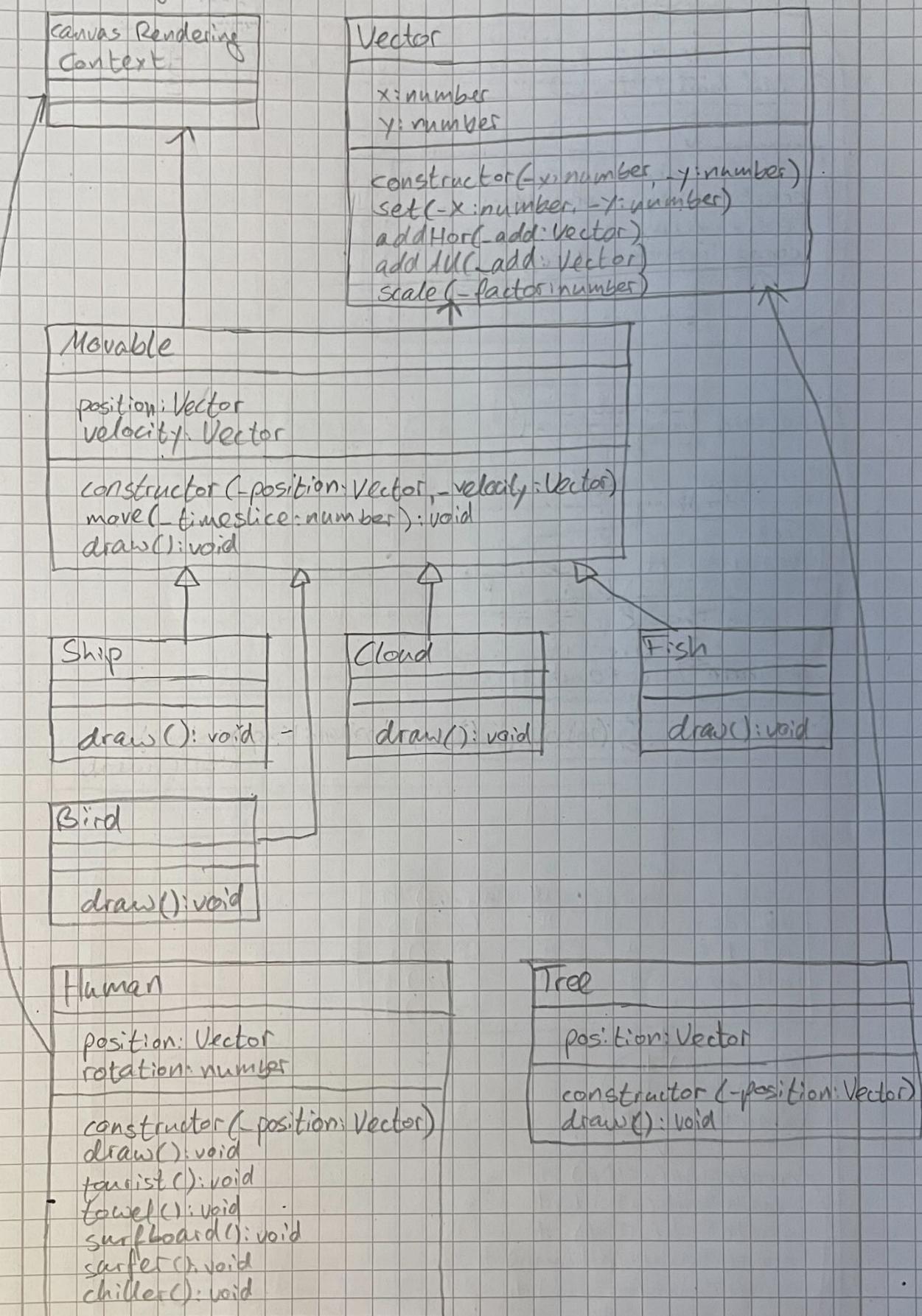


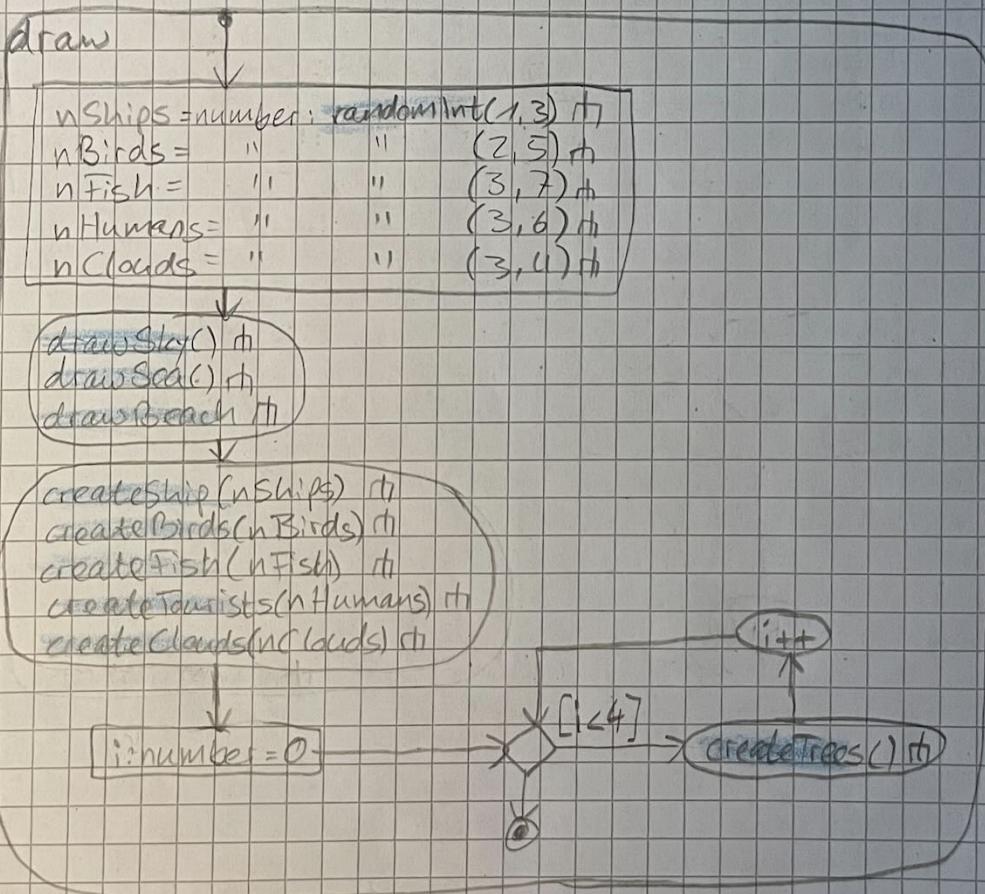
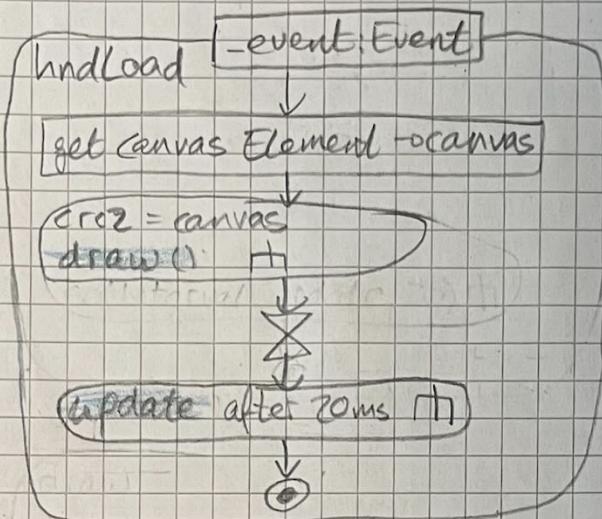
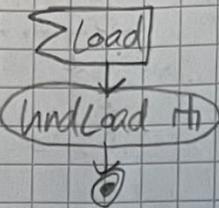
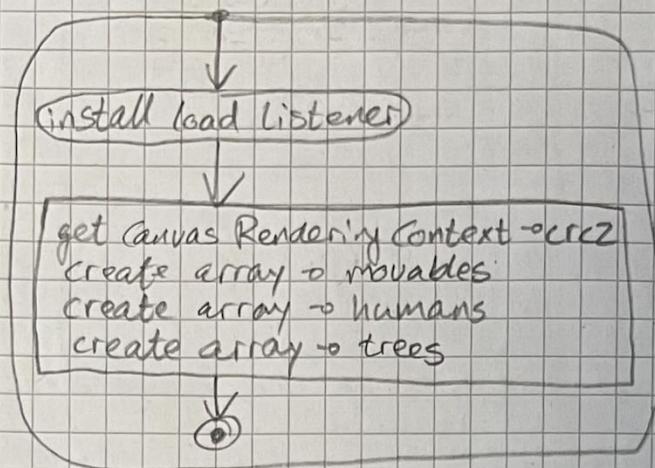
L10.2 Strand: Polymorphism

Class Diagram

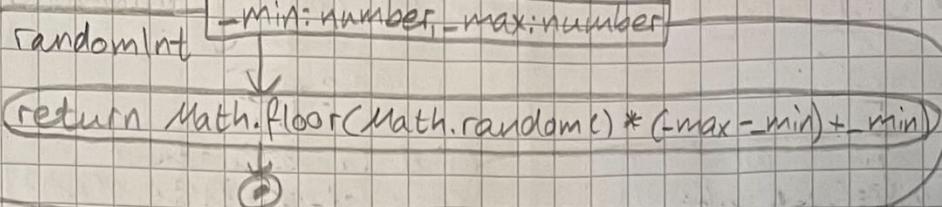


Activity Diagram

Main



Activity
Diagram
Main



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.draw())

tourist = next
value in humans

tree = first value of trees

(tree.draw())

tree = next
value in trees

drawSky

save crcz

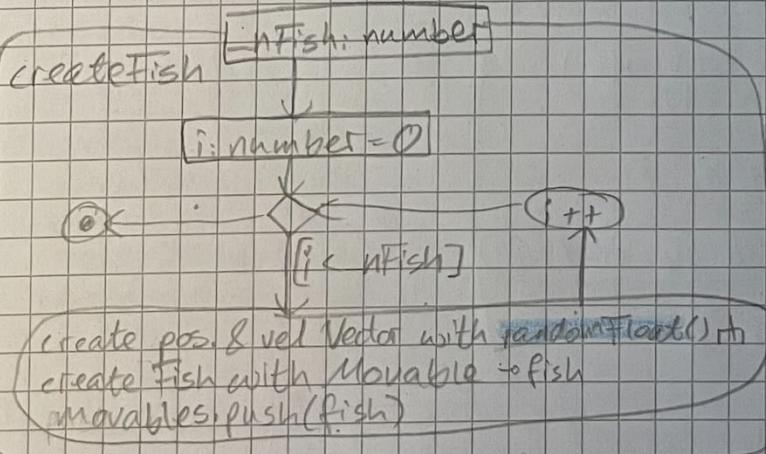
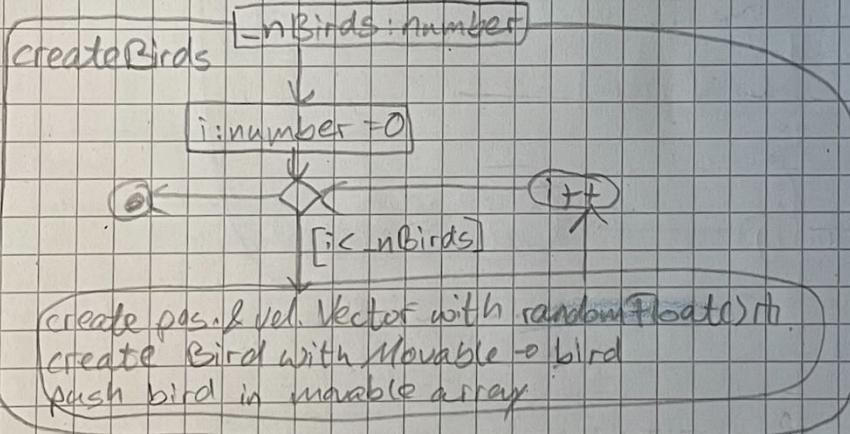
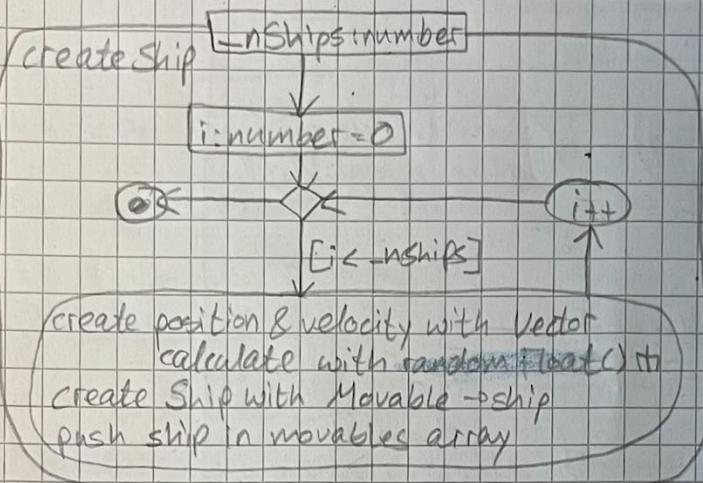
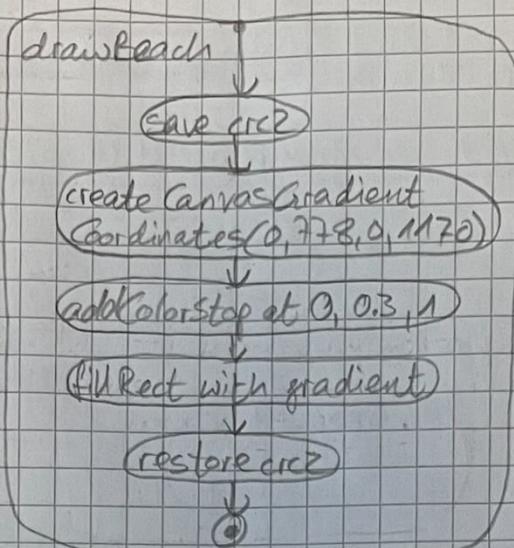
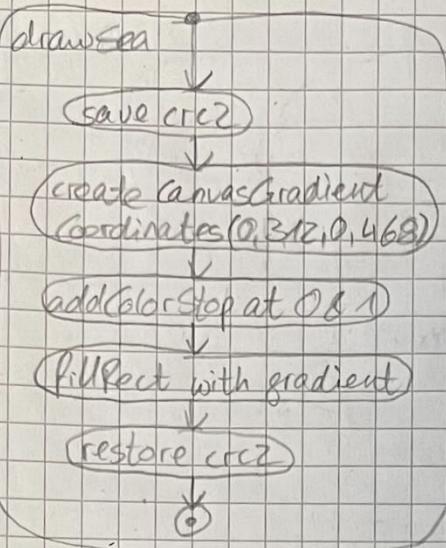
Create CanvasGradient with
coordinates (0,0,0,312)

addColorStop at 0 & 1

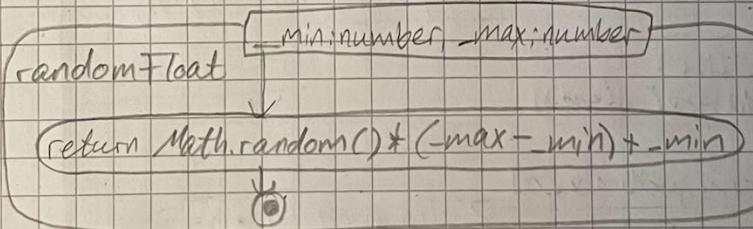
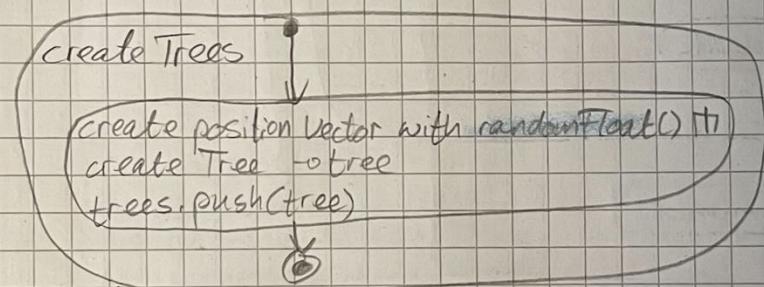
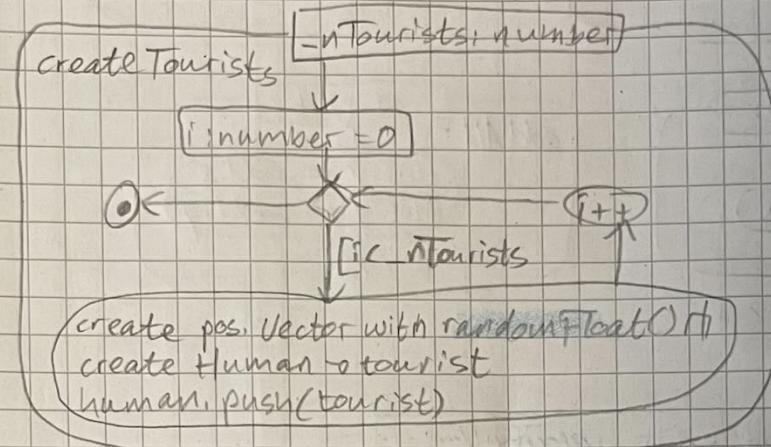
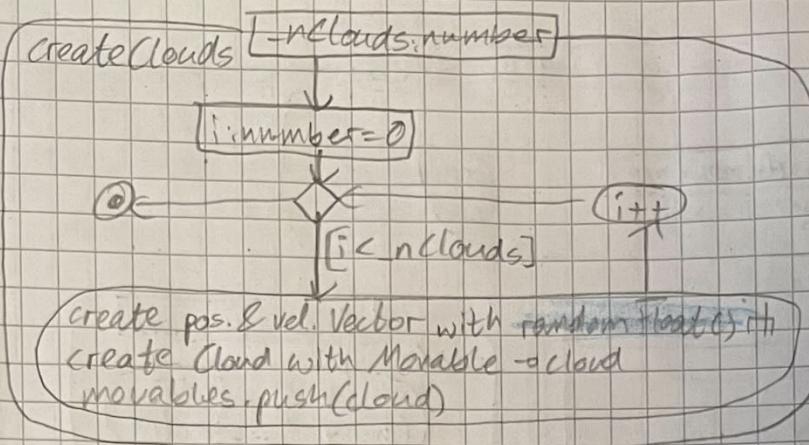
fillRect with fillStyle=gradient

restore crcz

Activity
Diagram
Main

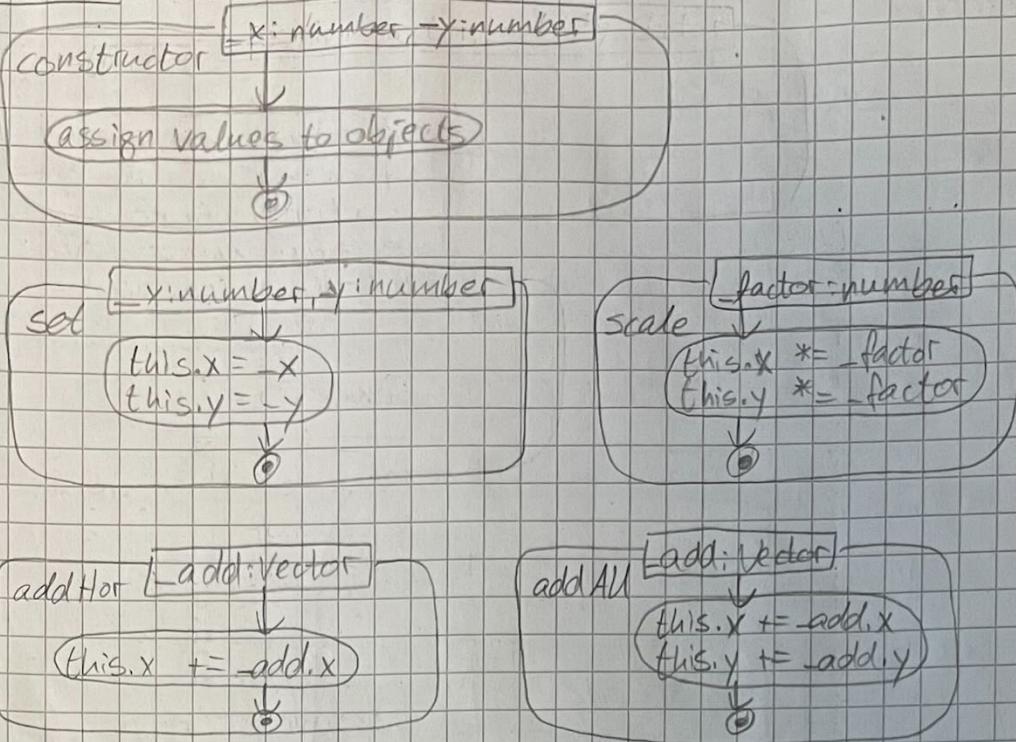


Activity
Diagram
Main

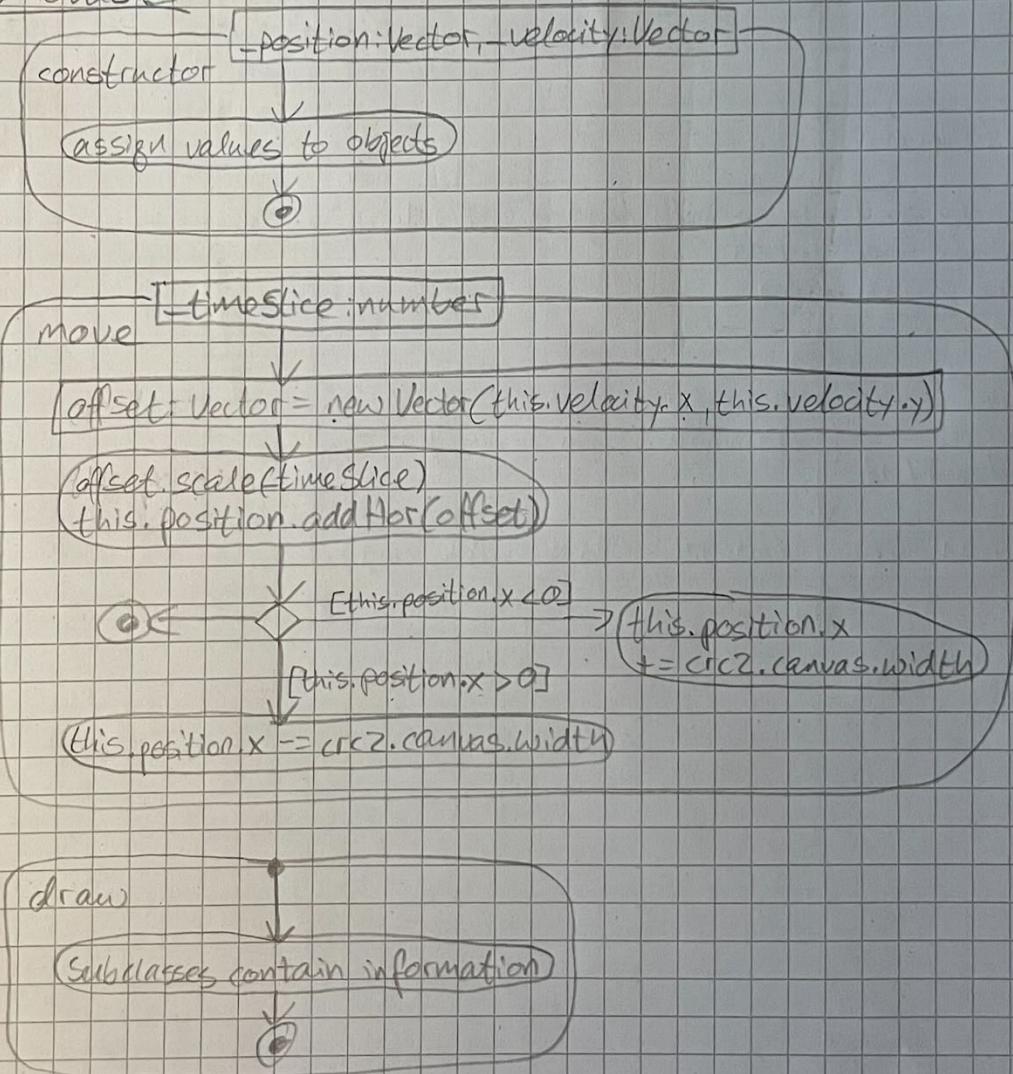


Activity Diagram

Vector



Movable



Activity
Diagram

Ship

draw

x: number = this.position.x
y: number = this.position.y

ship = Path2D

draw a half circle

flag = Path2D

draw a triangle

save cr2C
translate to (x, y)

ship.fillStyle = brown
flag.fillStyle = red

restore cr2C

Bird

draw

x: number = this.position.x
y: number = this.position.y
rotation = number[] = [10, 15, ...]
rotNum: number = Math.floor(
Math.random() * (rotation.length
- 1) + 1) - 1

wing = Path2D

draw 2 arcs for the wings

save cr2C
translate to (x, y)
rotate with (rotation[rotNum])

stroke(wing)

restore cr2C

Fish

draw

x: number = this.position.x
y: number = this.position.y
rotation = number[] = [0, 25, ...]
rotNum: number = Math.floor(
Math.random() * (rotation.length
- 1) + 1) - 1

fish = Path2D

draw an ellipse & a triangle

save cr2C
translate to (x, y)
rotate(rotation[rotNum])

stroke(fish)
fillStyle = orange

restore cr2C

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]

restore crc2

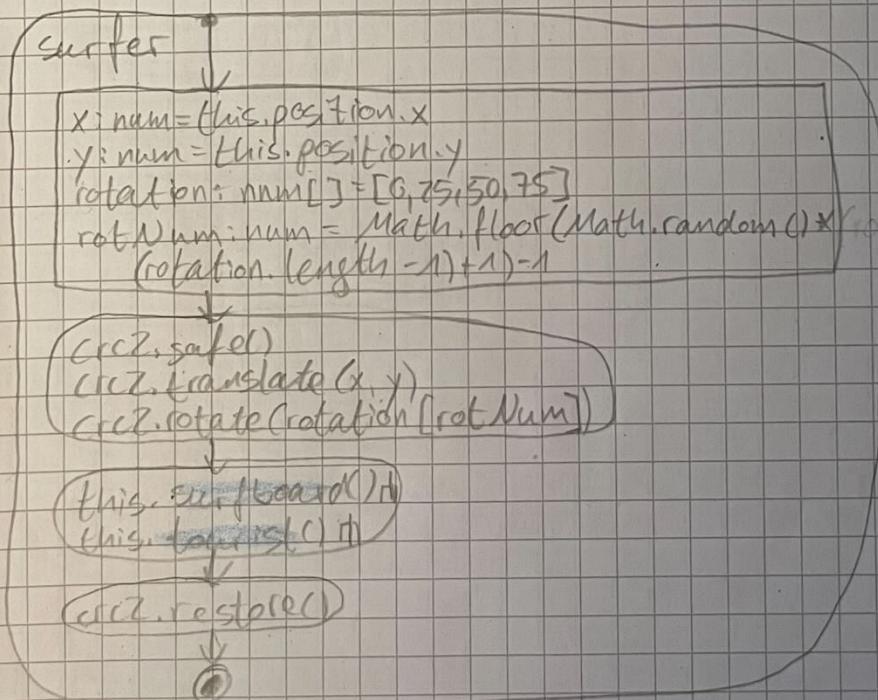
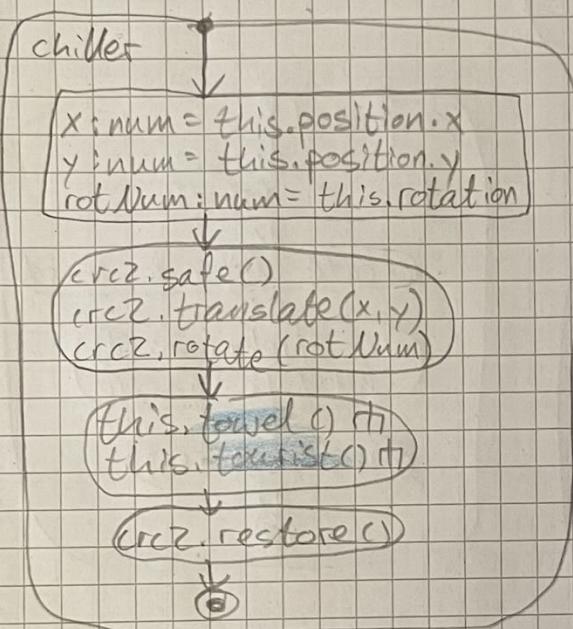
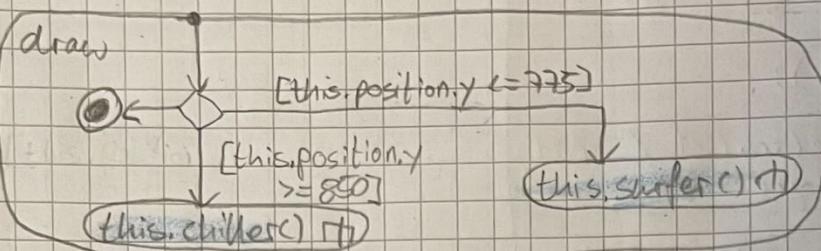
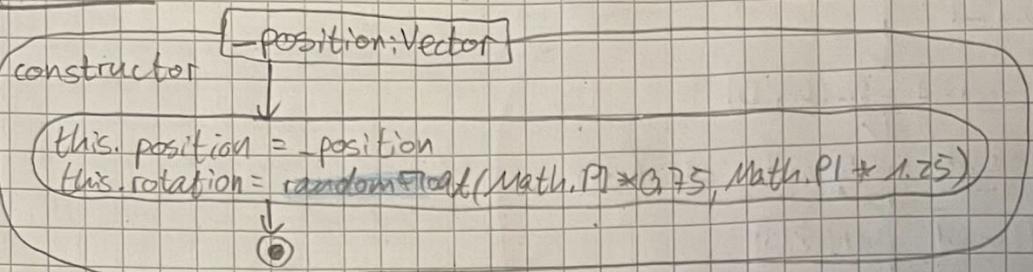
Save crc2

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

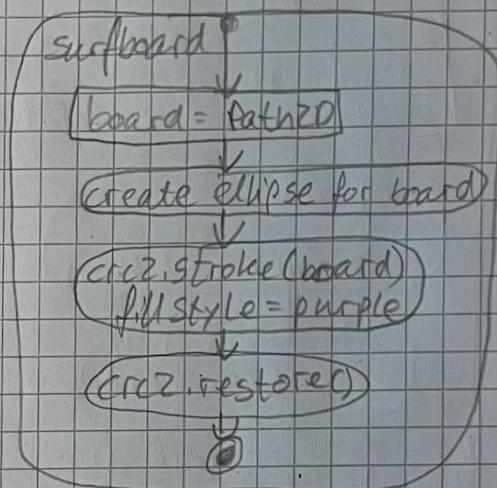
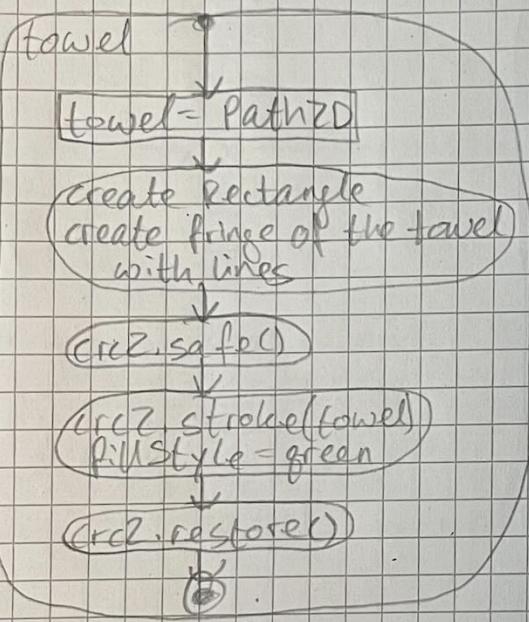
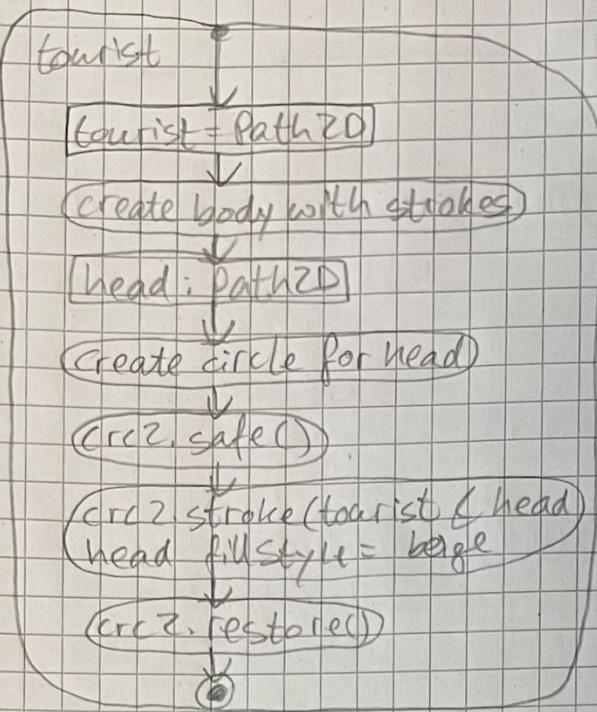
translate to (x,y)
crc2.fill(circles)

Activity
Diagram

Human



Activity
Diagram
Human



Activity Diagram

Tree

