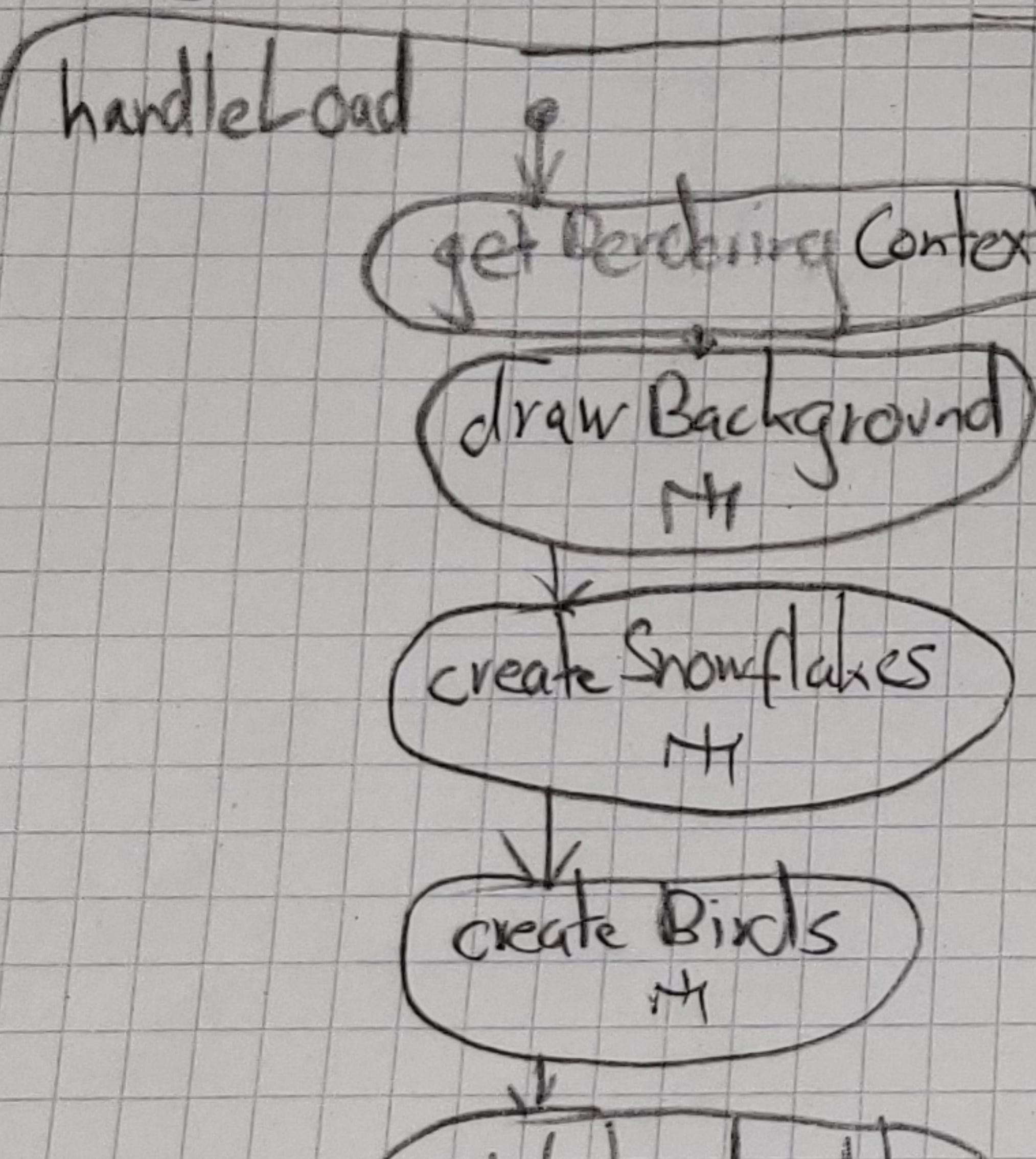
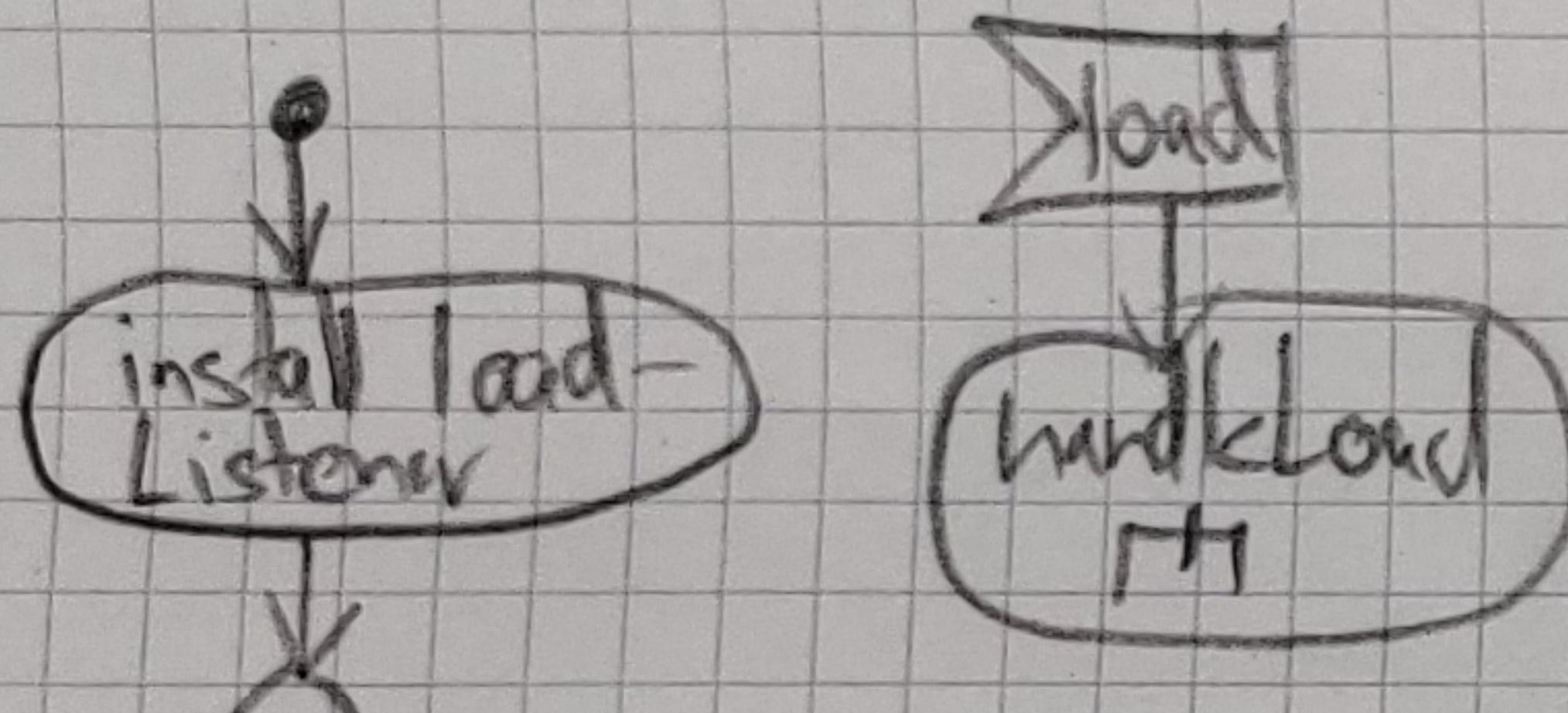
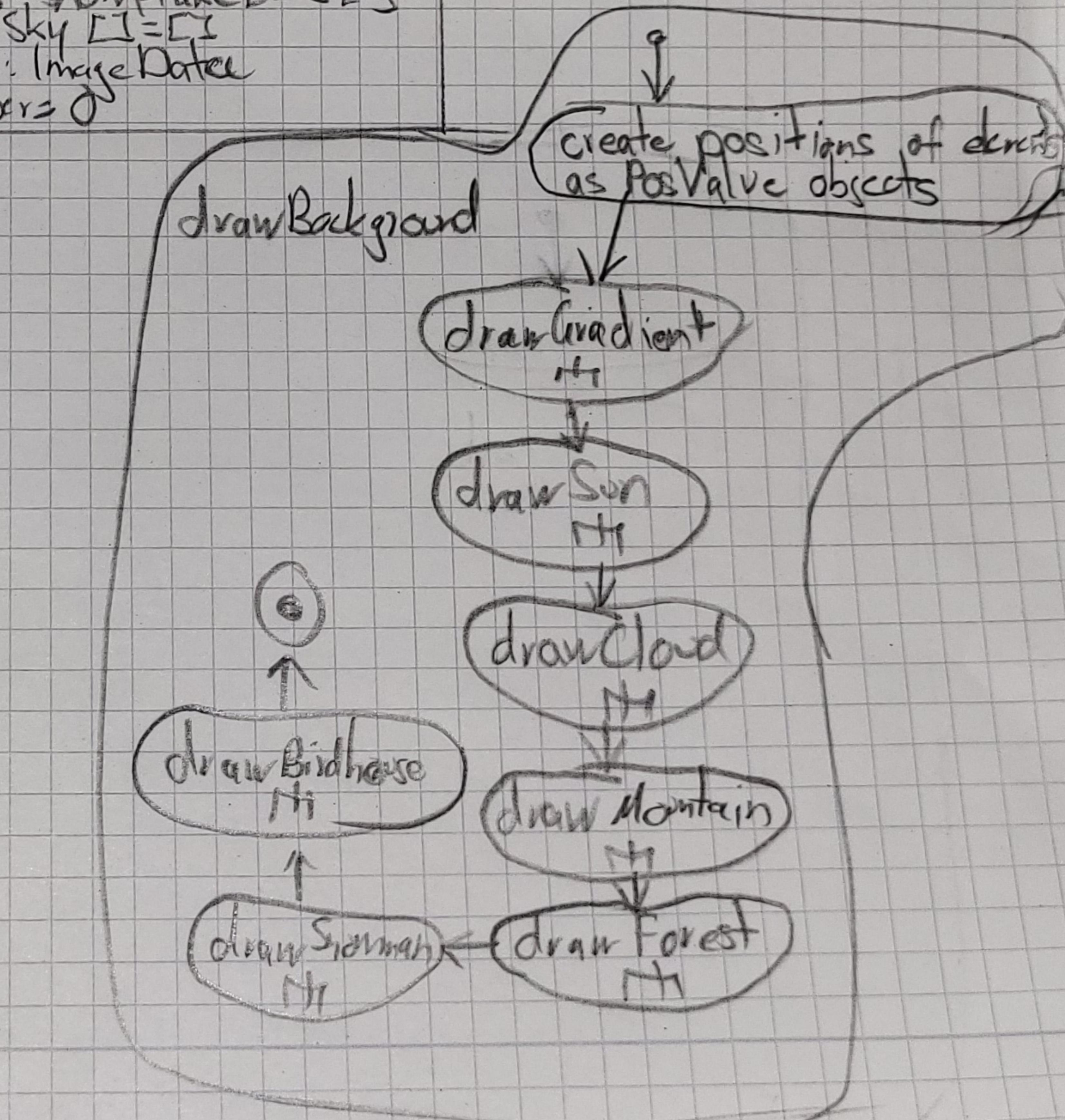


Activity Diagram

WWL-Classes

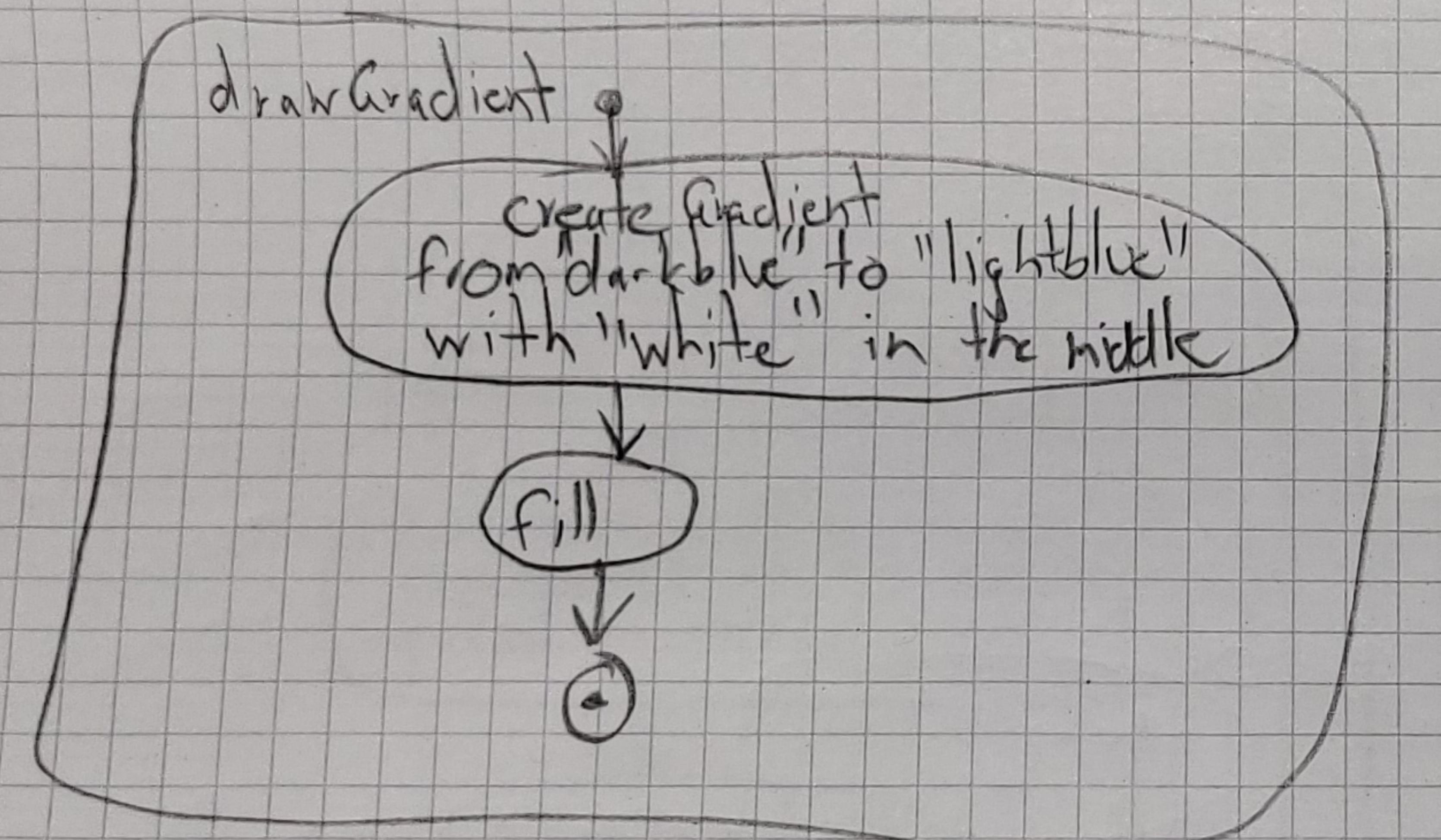


```
let crc2: CanvasRenderingContext  
let canvas: HTMLCanvasElement  
let snowflakes: snowflake[] = []  
let birds: BirdSky[] = []  
let background: ImageData  
let step: number = 0
```



Activity Diagram

WWL-classes



Activity Diagram

Birds / WNL_classes

drawSun

-position: PosValue

r1: number = 40
r2: number = 80
gradient = Radial Gradient

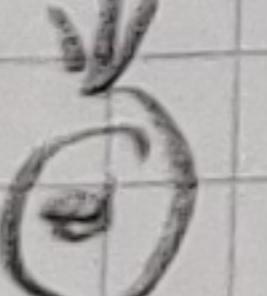
set colour stops for gradient
transparent at 10 and bright
yellow at $r1/r2$

Save transform

translate to -position

draw full circle

restore transform



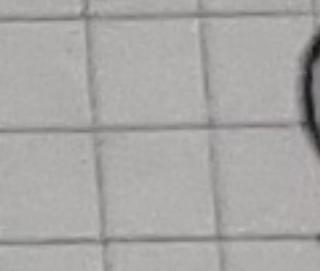
drawCloud

-position: PosValue
-size: PosValue

numParticles: number = 50;
radParticles: number = 20;
particle = path with full circle
with radParticles
gradient: Radial Gradient
from a=0.5 to a=0.1

Save Transform

translate to -position



restore transform

[index nparticles]

x: number = random size.x

y: number = random size.y

restore transform

Save transform

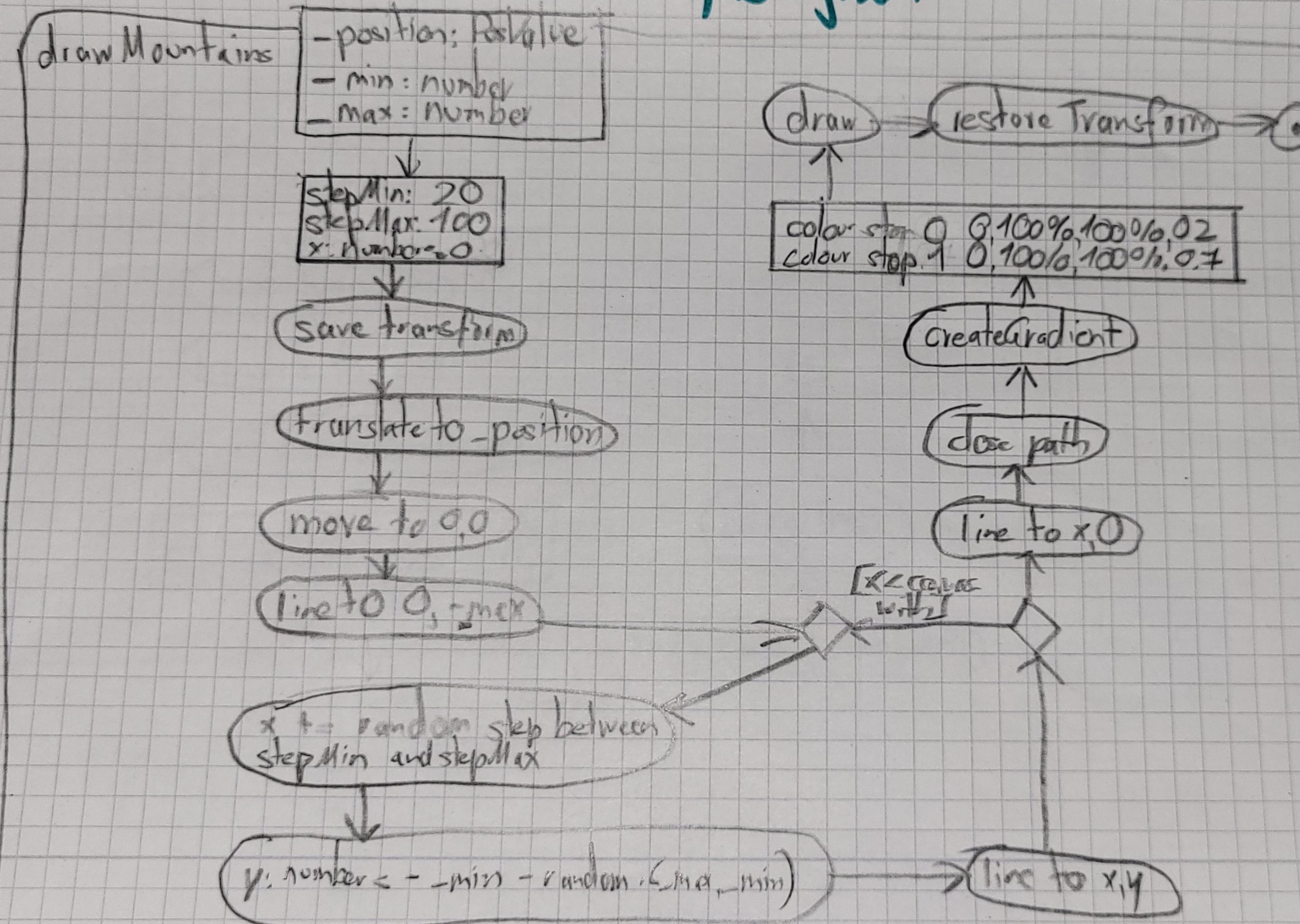
translate to x,y,P

draw

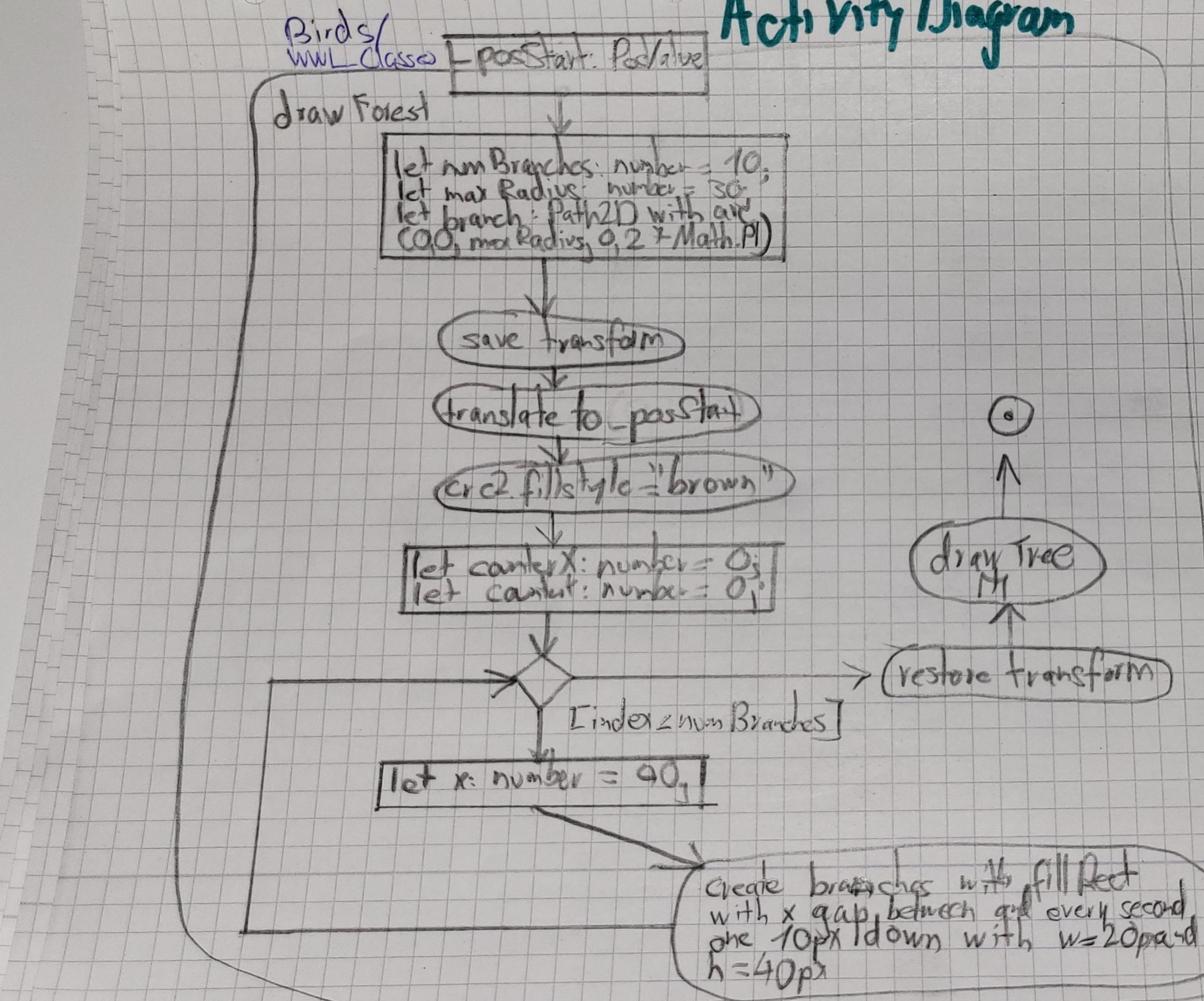


Birds / WWL classes

Scribble Activity Diagram

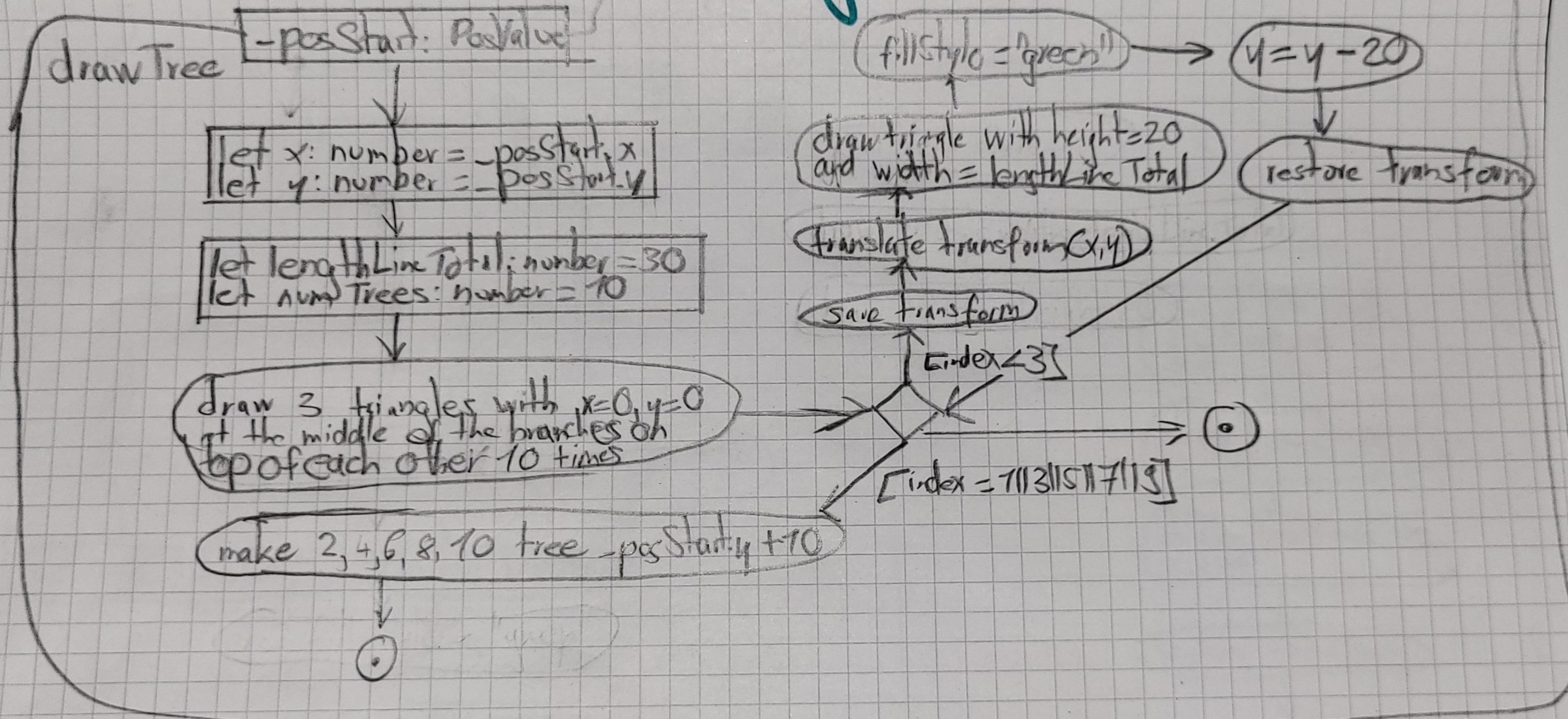


Activity Diagram



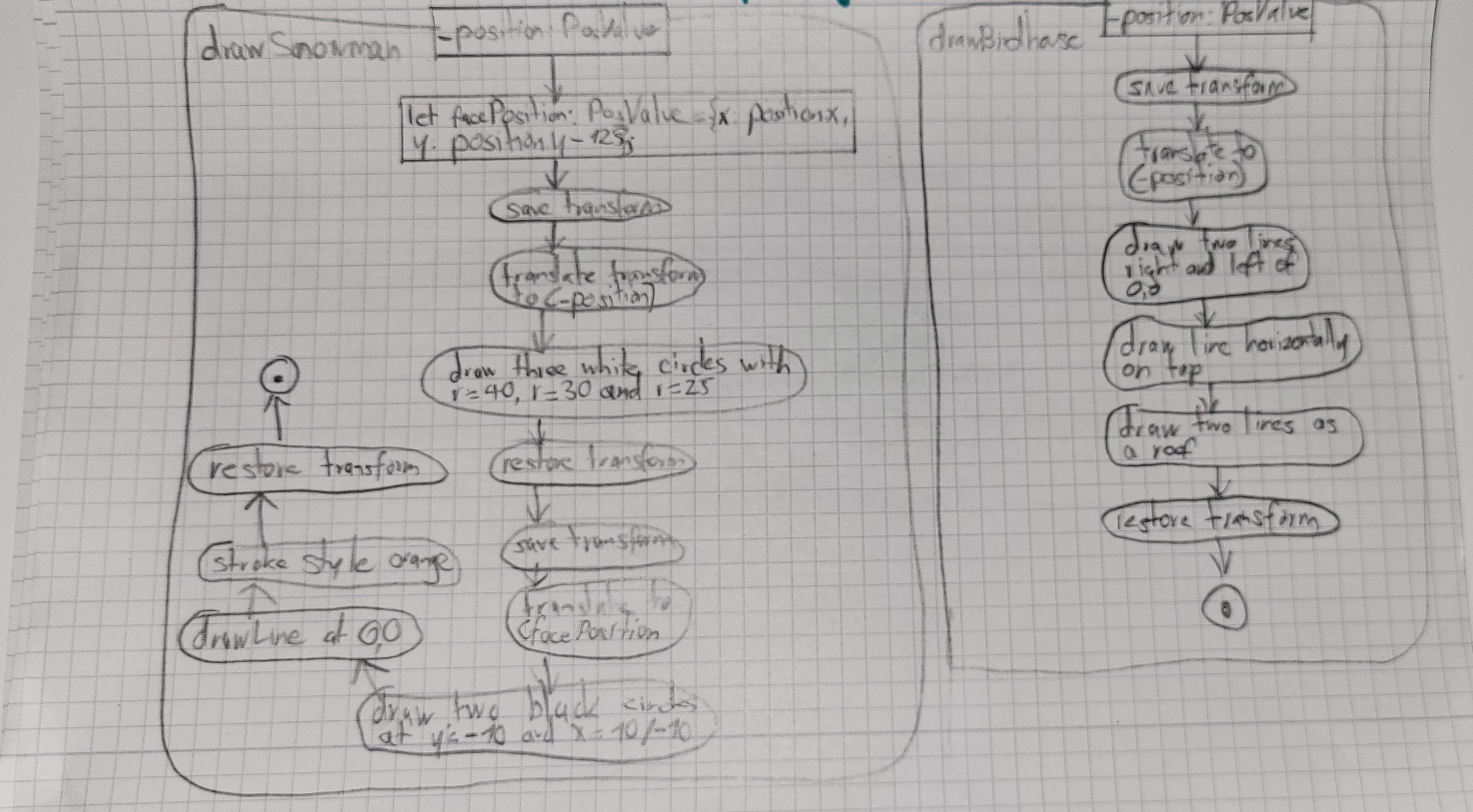
Birds / wwl_classes

Activity Diagram



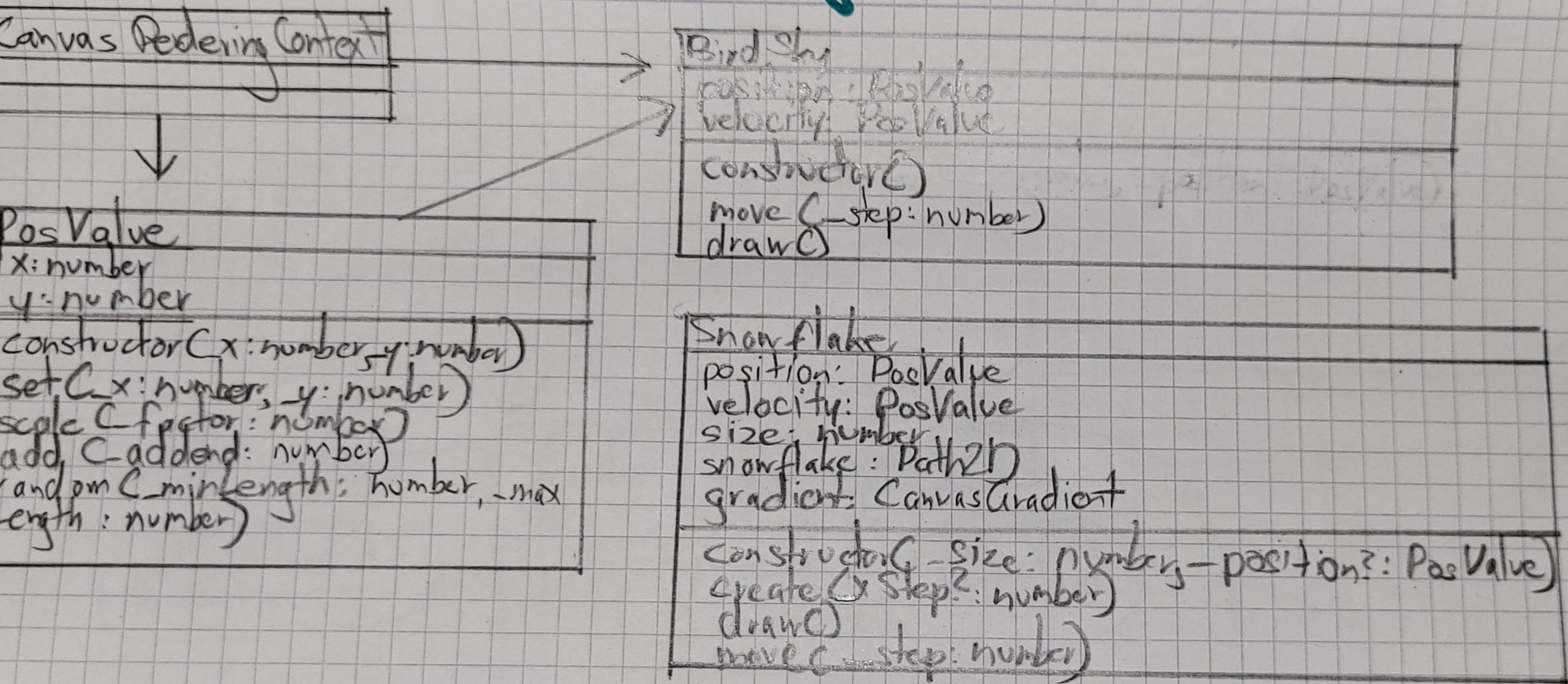
Birds/WWL classes

Activity Diagram



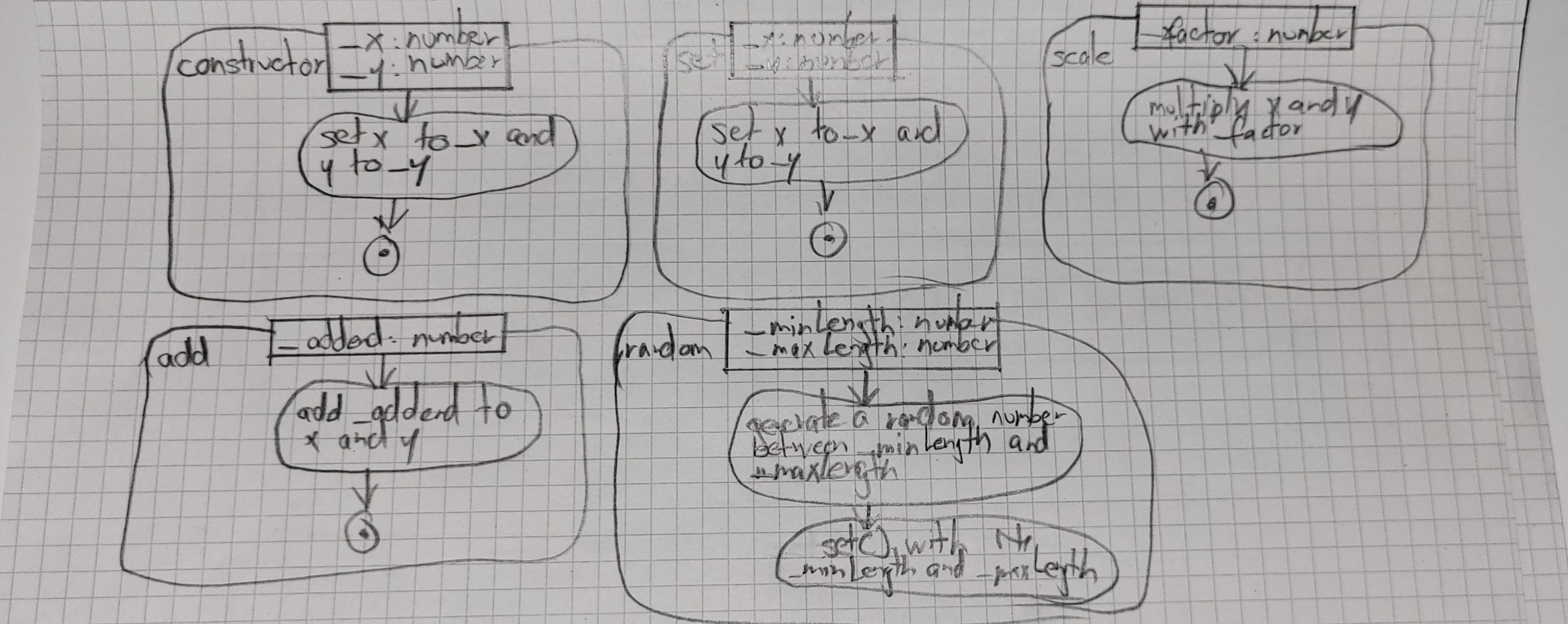
Class Diagram

WWL-classes



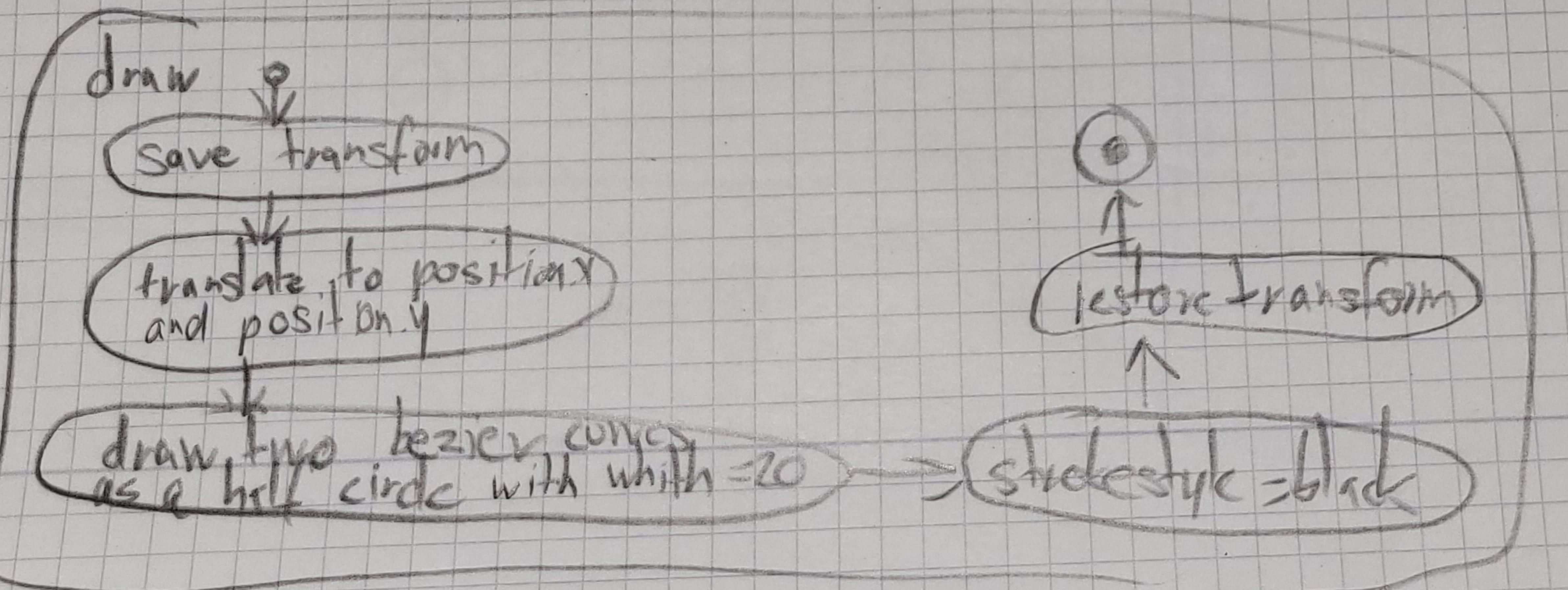
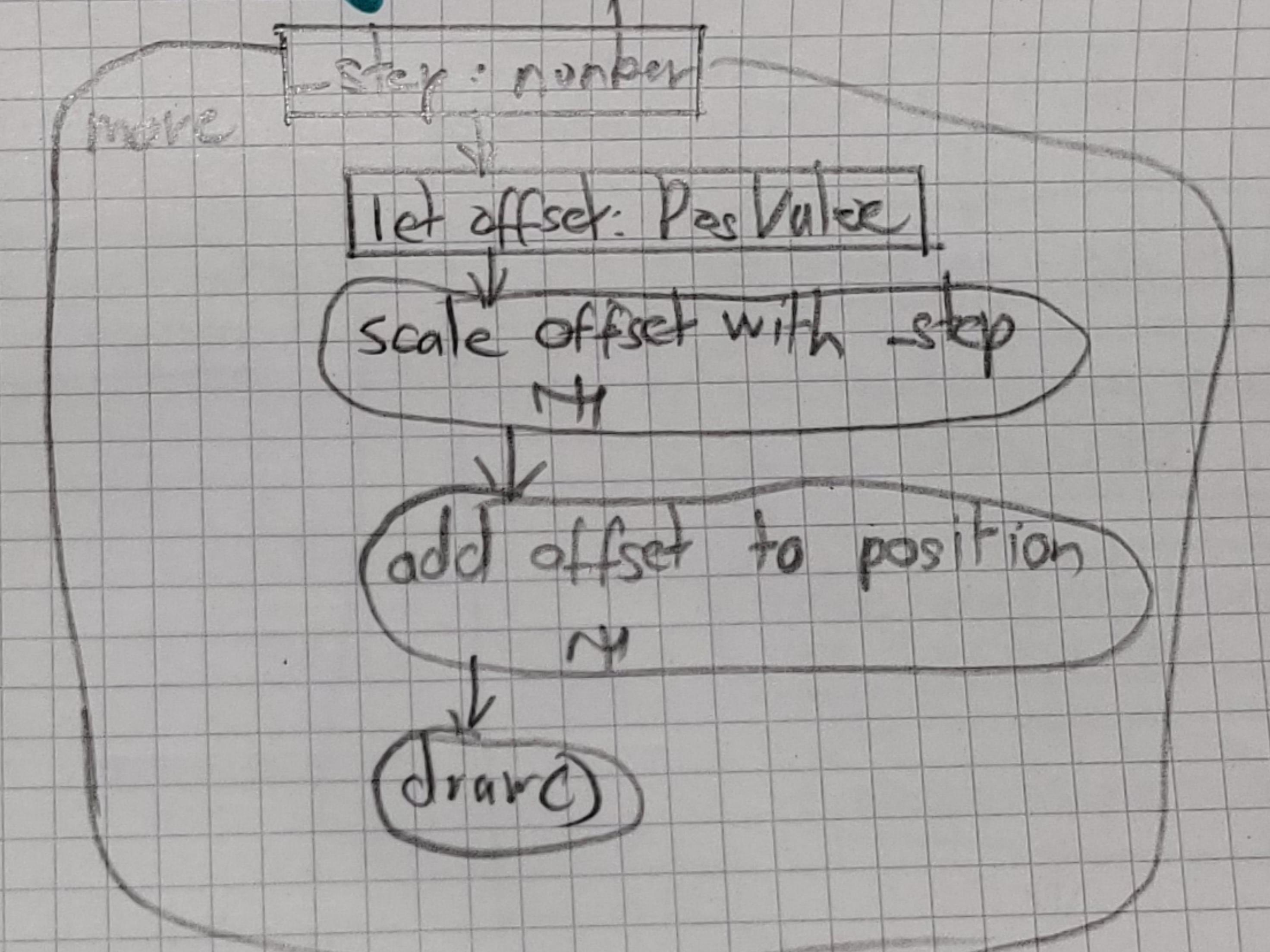
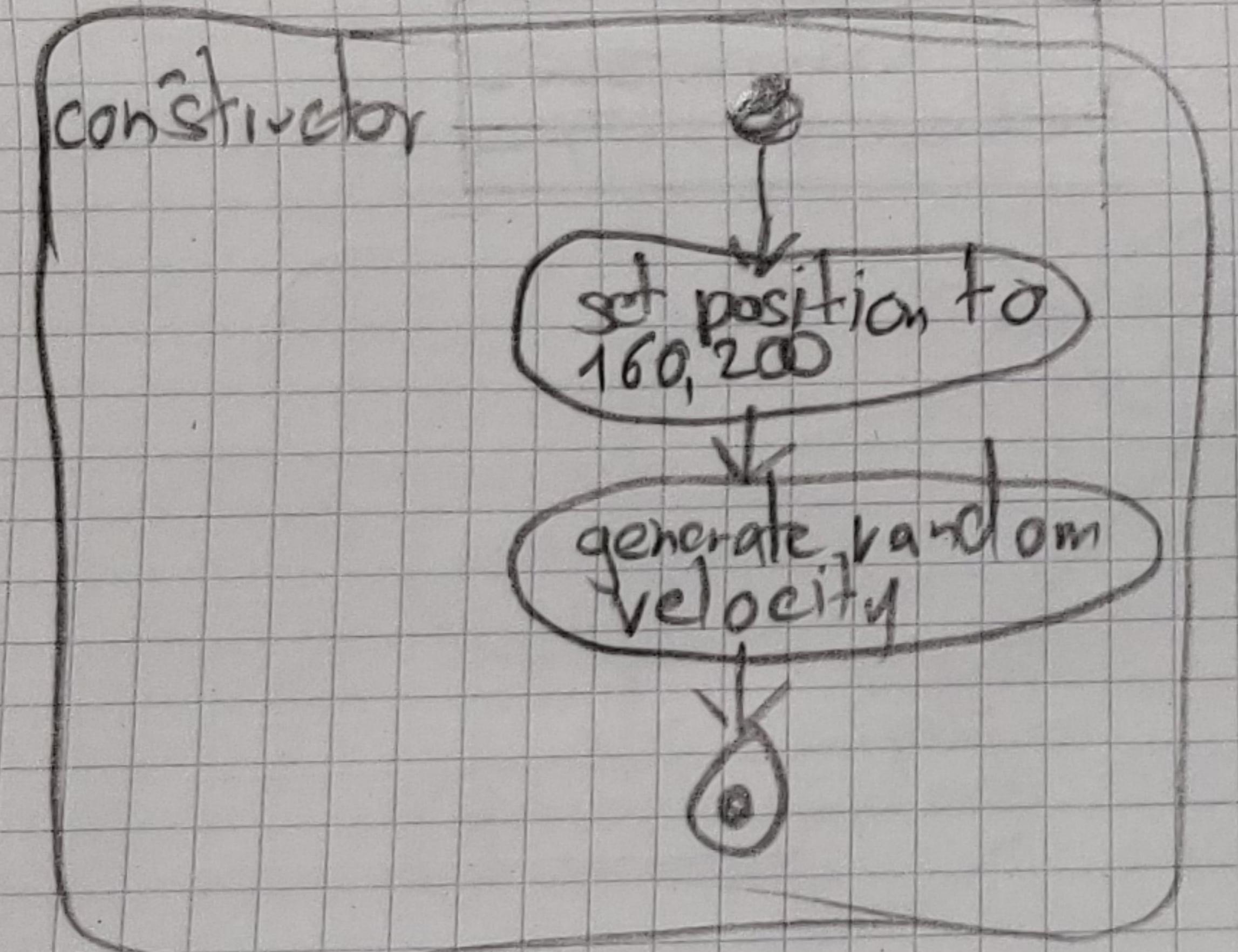
Activity Diagram

WWL=classes - PosValue



Activity Diagram

VML Classes - Bird Sky



Activity Diagram

WWL classes - Snowflake

