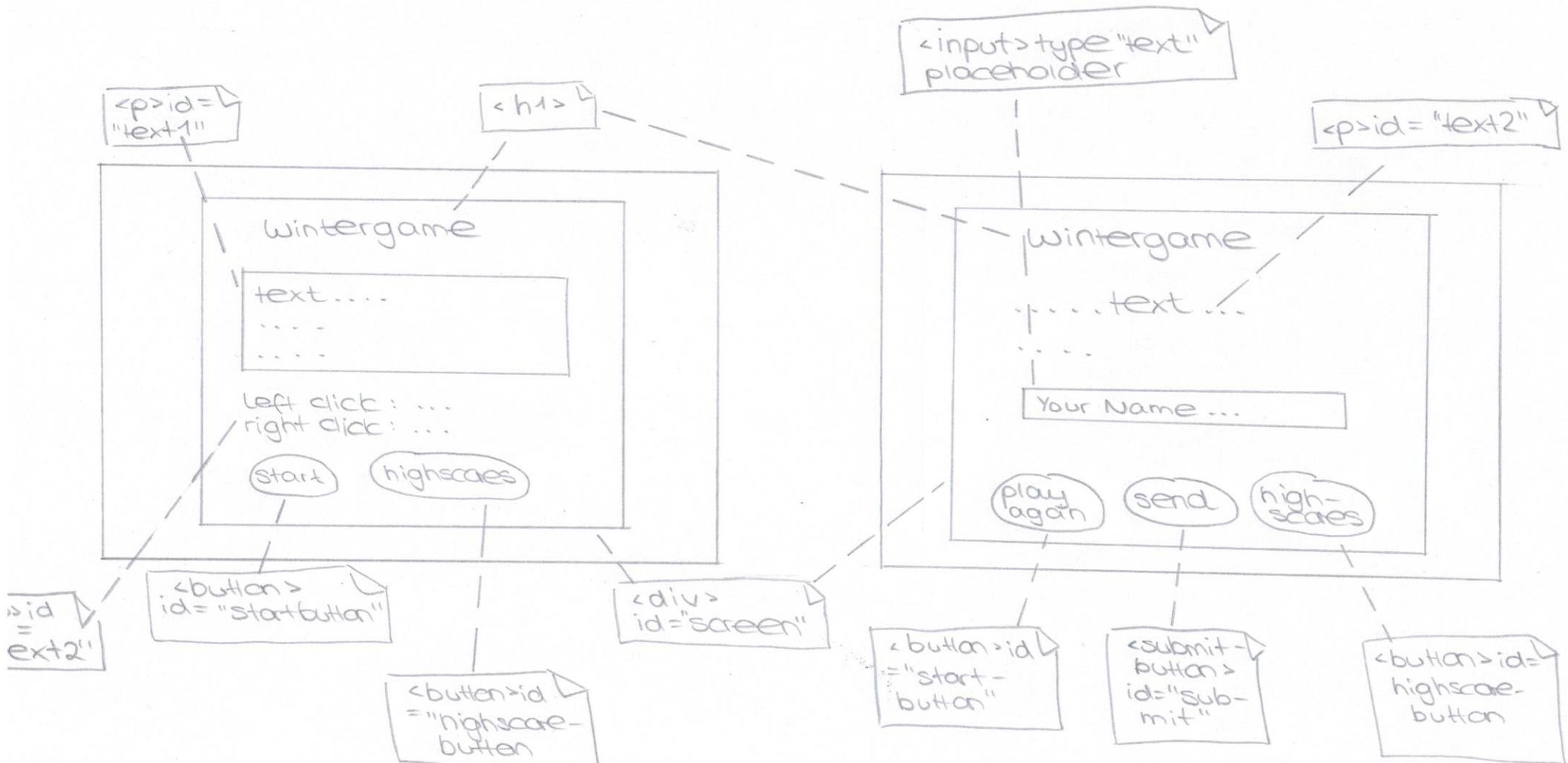
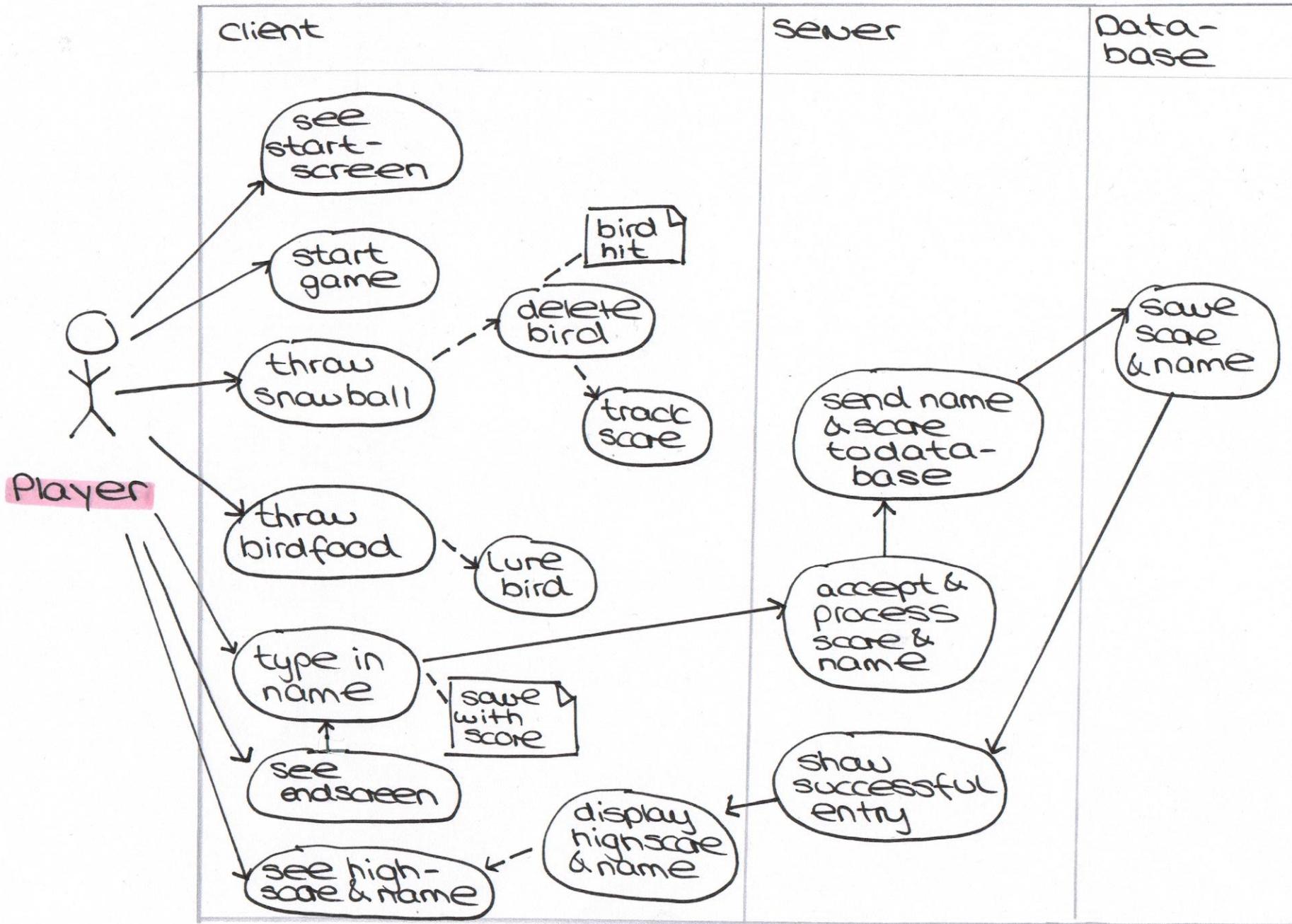




## SCRIBBLE2 - Endaufgabe



## Use Case Diagram



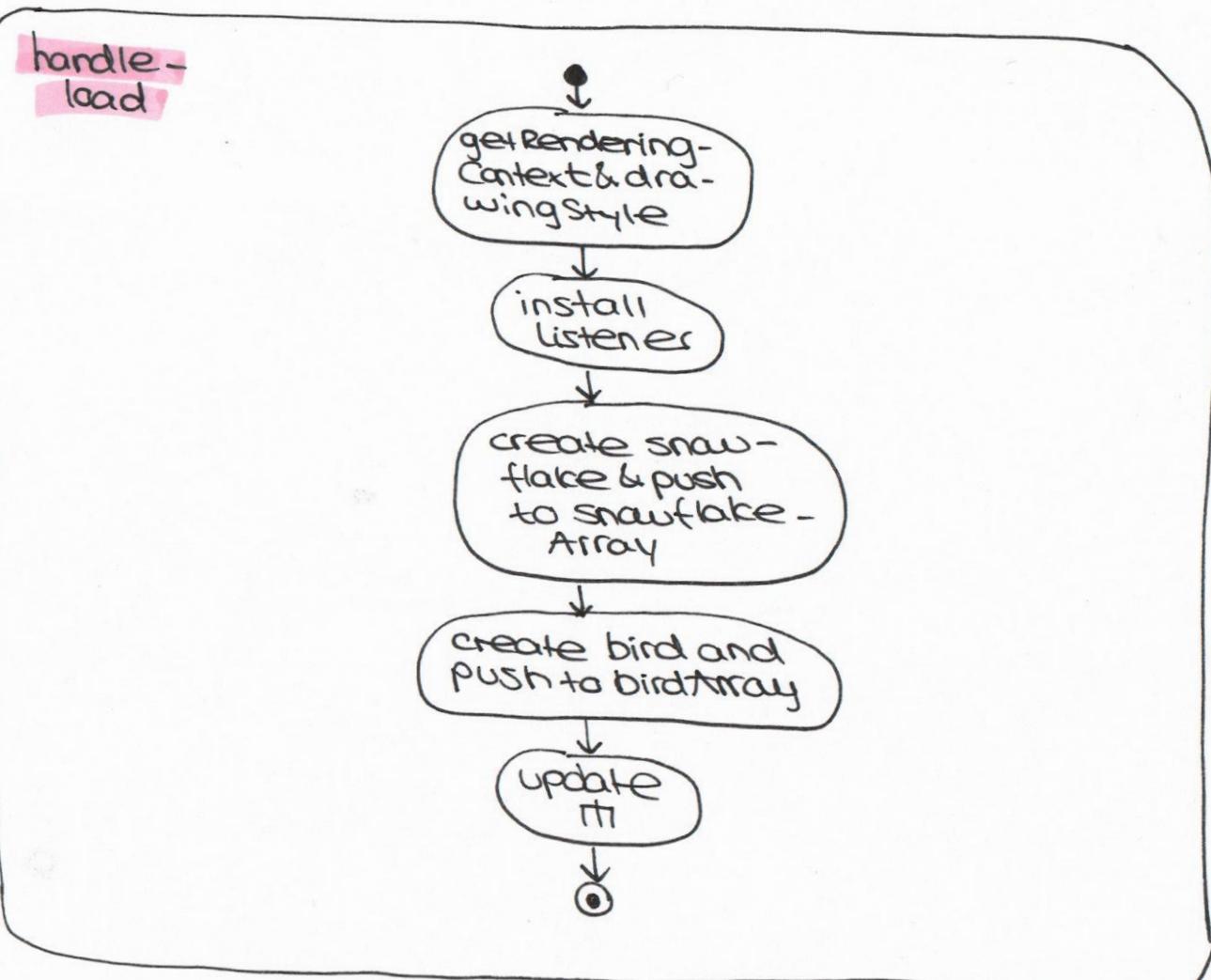
## Main



## Globals

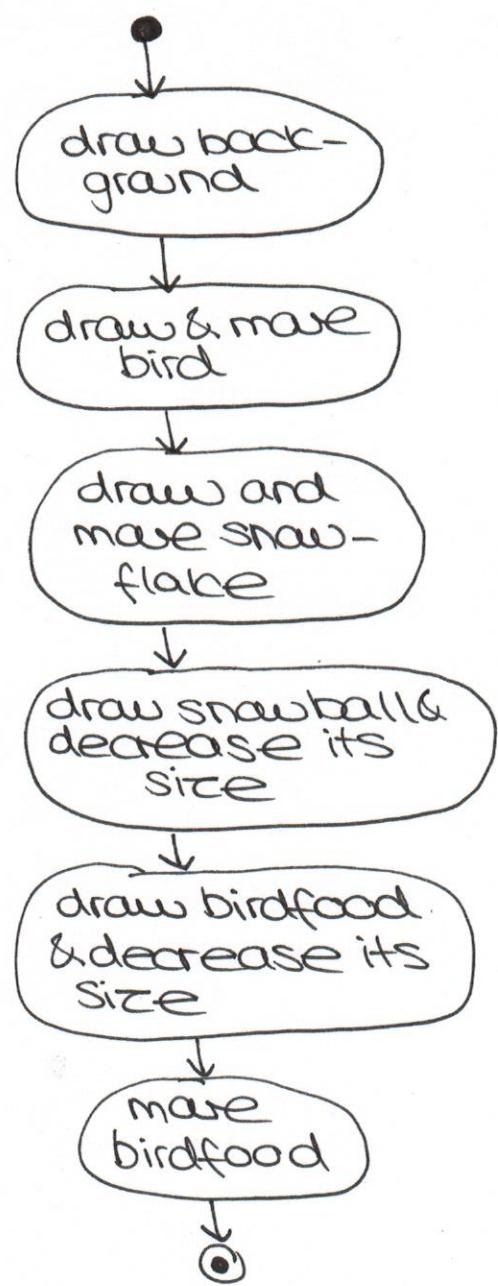
```

crc2 = CanvasRenderingContext2D
snowflakeArray: Snowflake[] = []
birdArray: Bird[] = []
snowball: Snowball
birdFood: BirdFood
fps: number
score: number
  
```



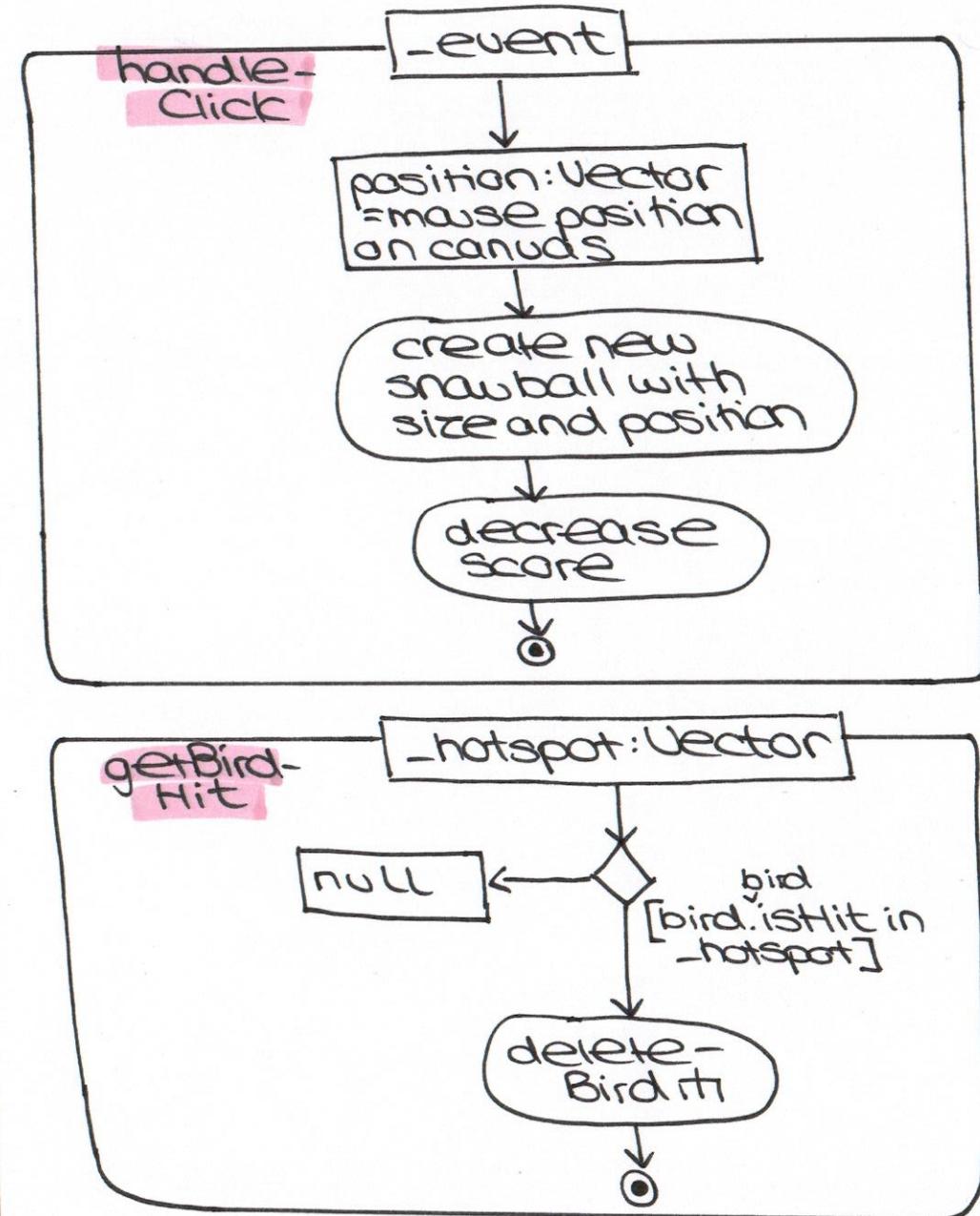
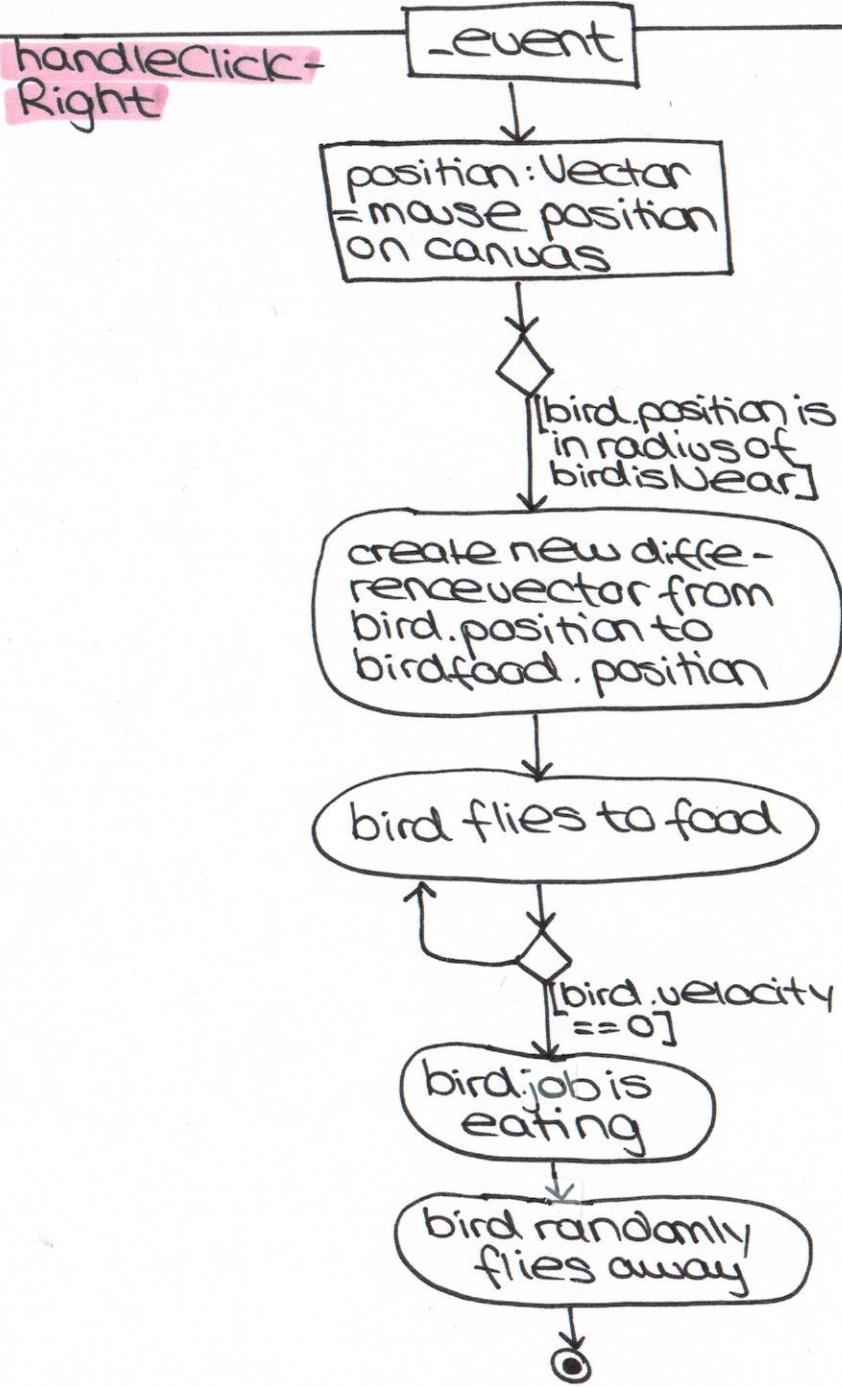
## Main

update

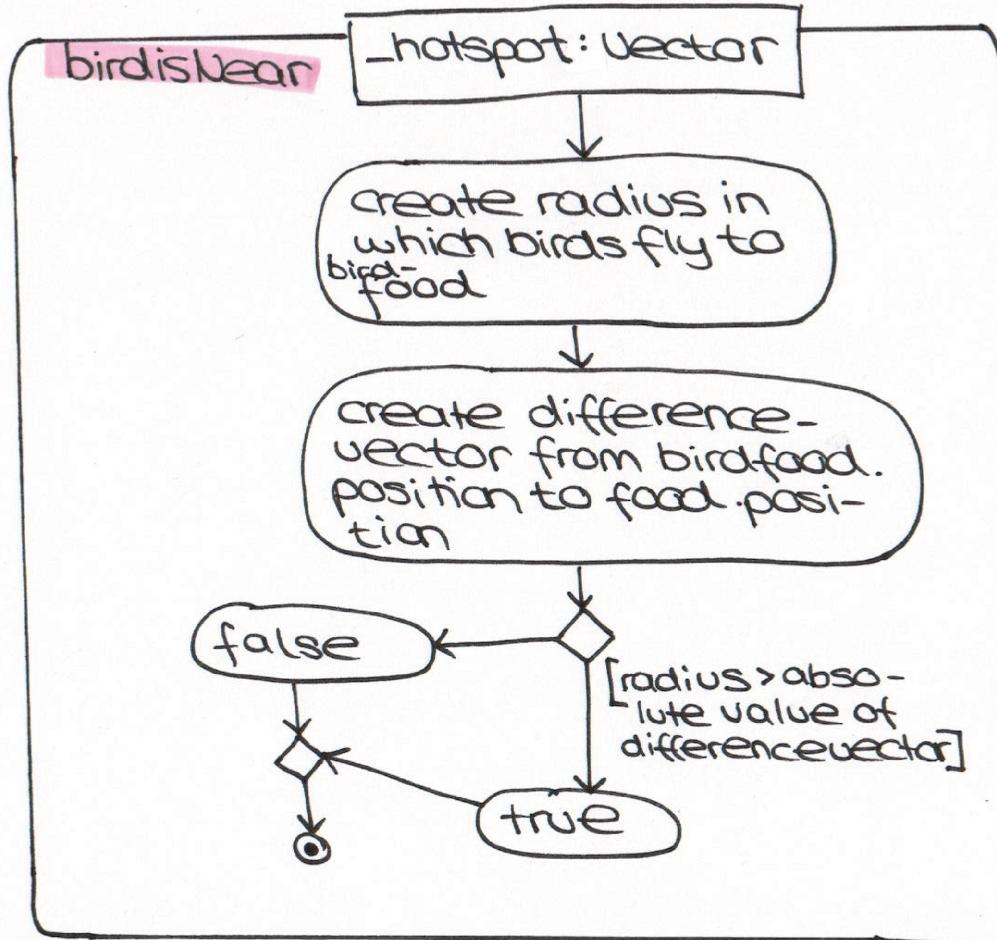


Klasse  
 Funktion

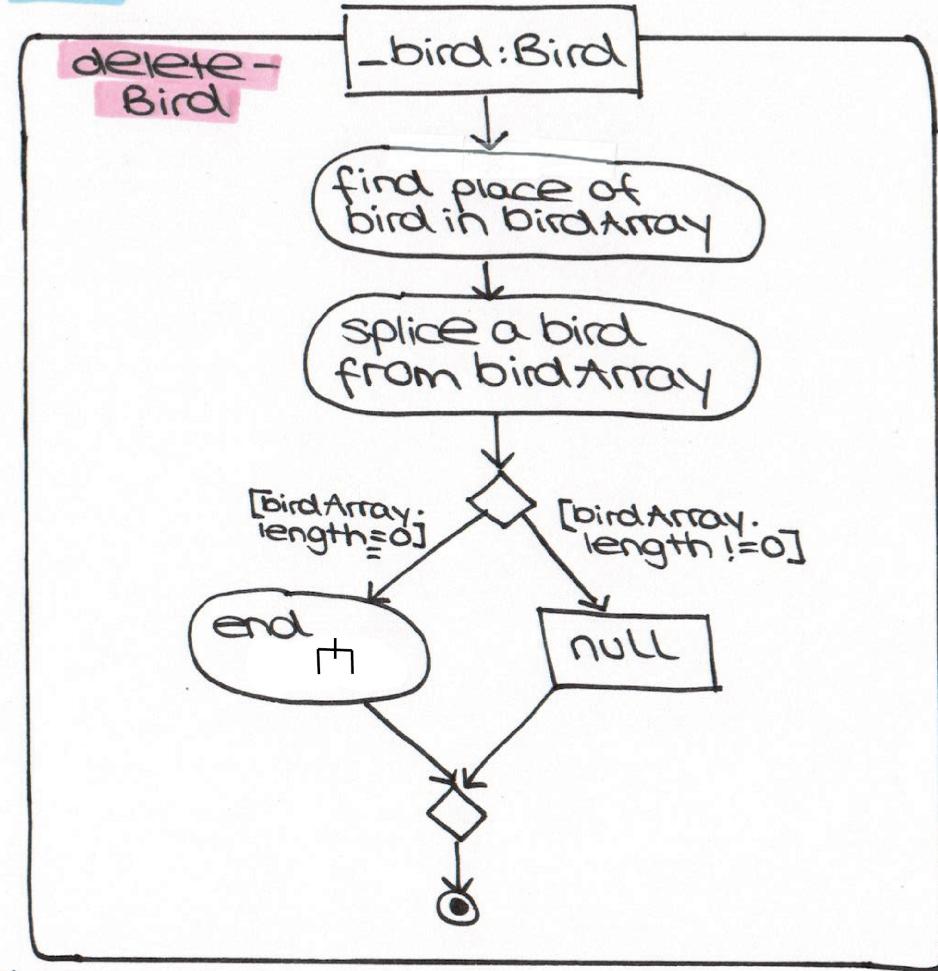
# Main



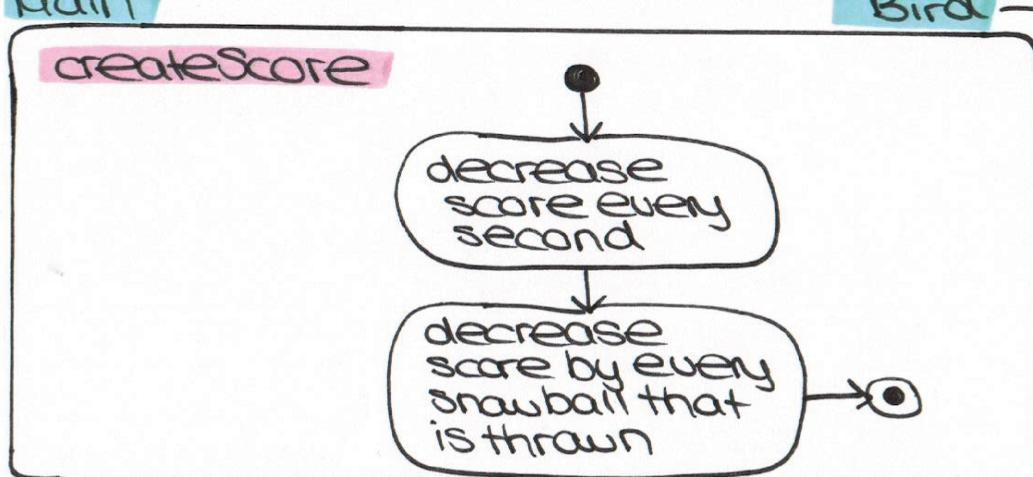
## Main



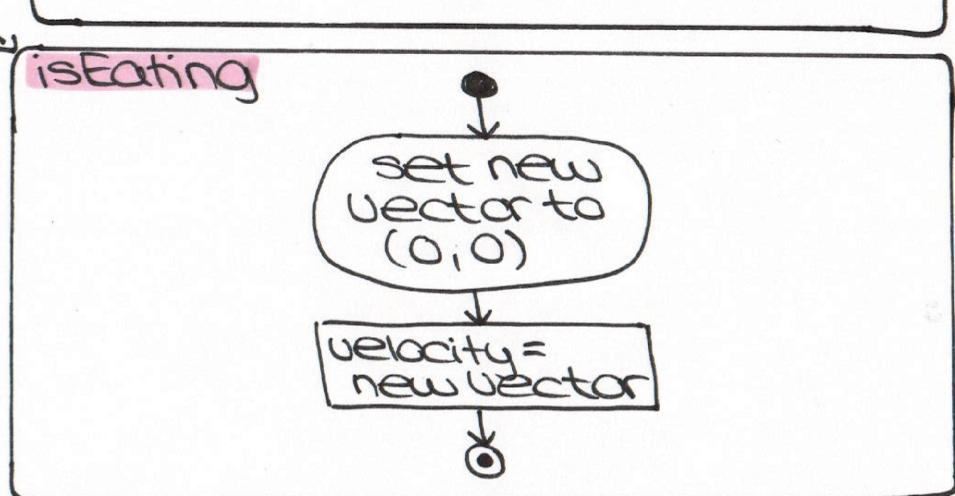
## Main



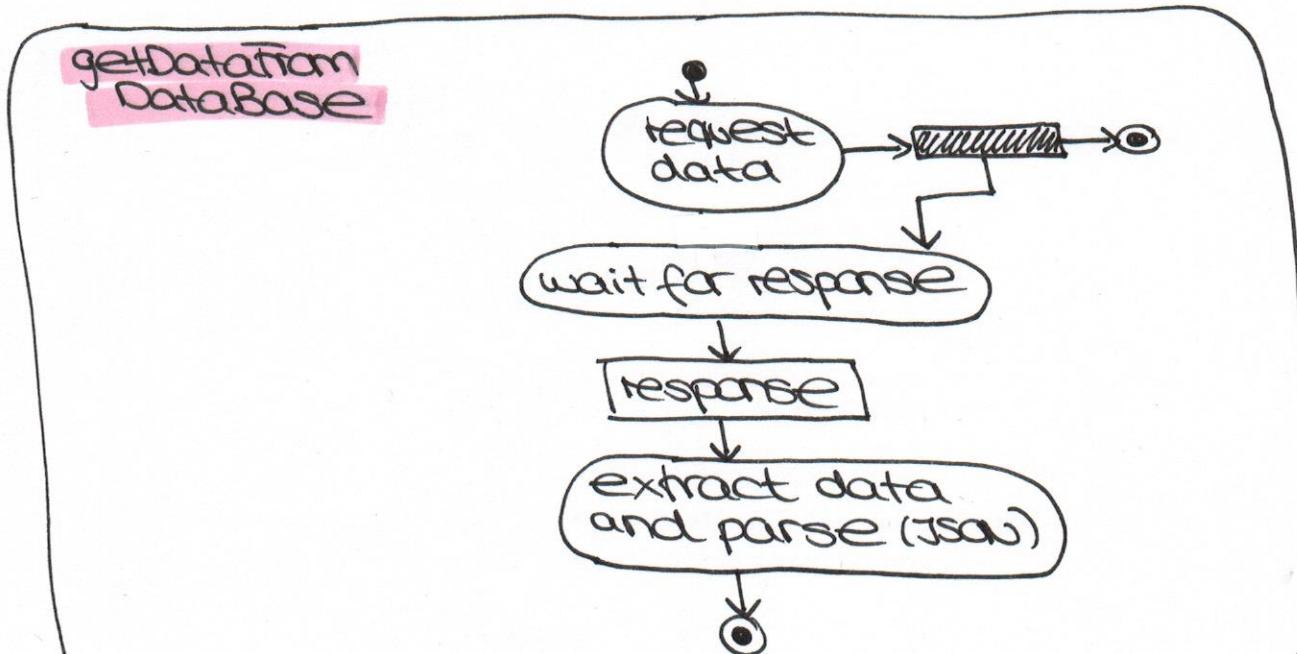
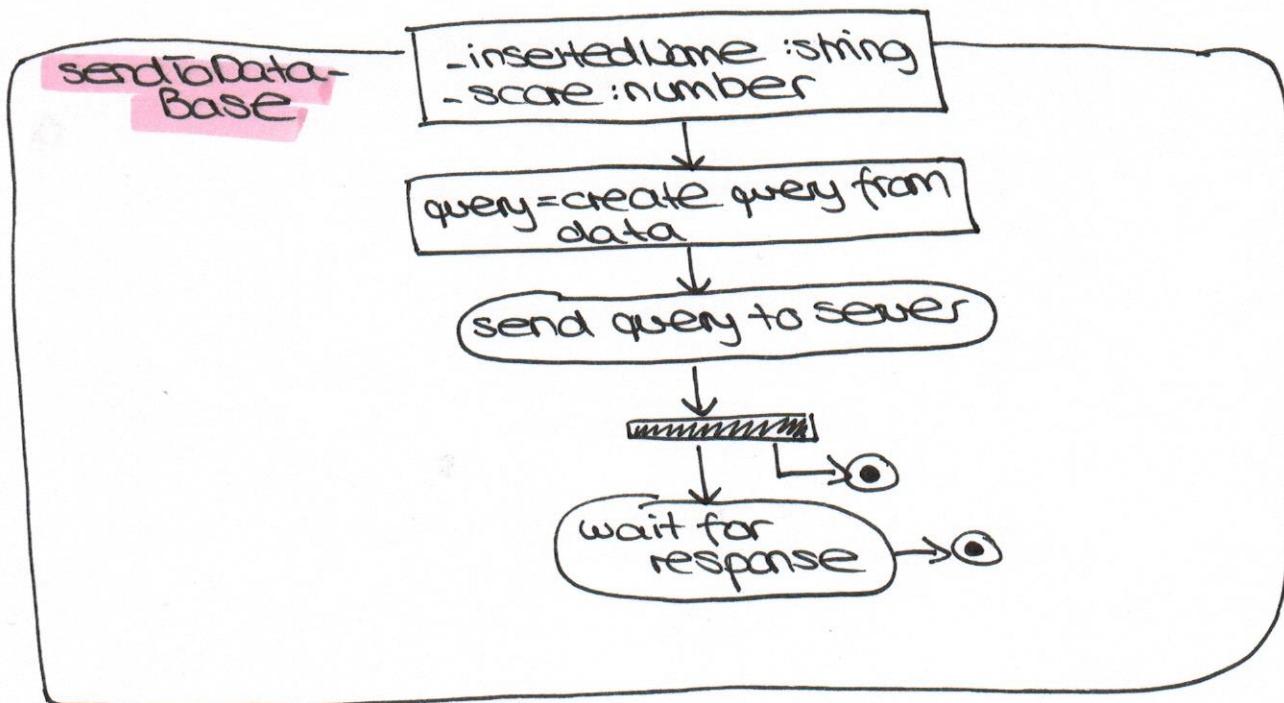
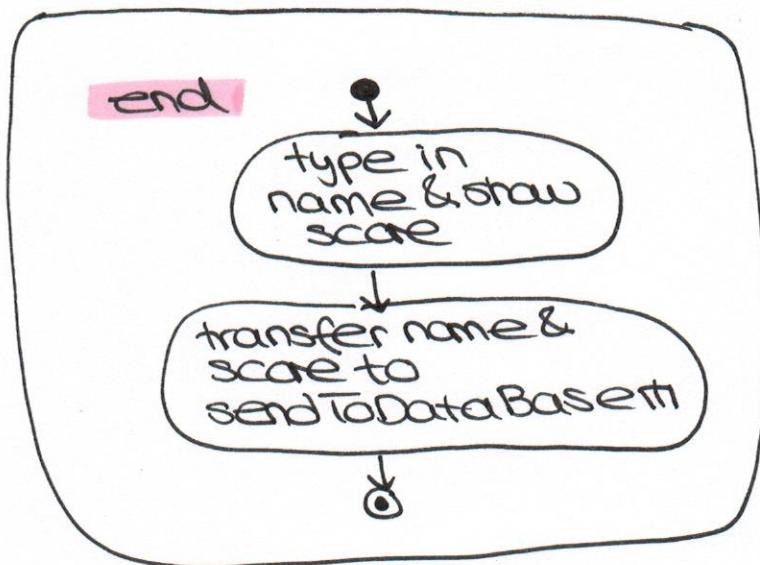
## Main



## Bird



## Main



# Bird

## constructor

-size : number  
-position : Vector

super(-position)

create new Vector with random value to position

create new Vector with velocity and direction

bodycolor = get - RandomColor()

## draw

[job.TASK == EAT]

set new Vector for velocity

draw bird in flying - position

[velocity <= -0.1]

forward

[velocity >= 0.1]

back - word

draw bird in eating - position

save translation

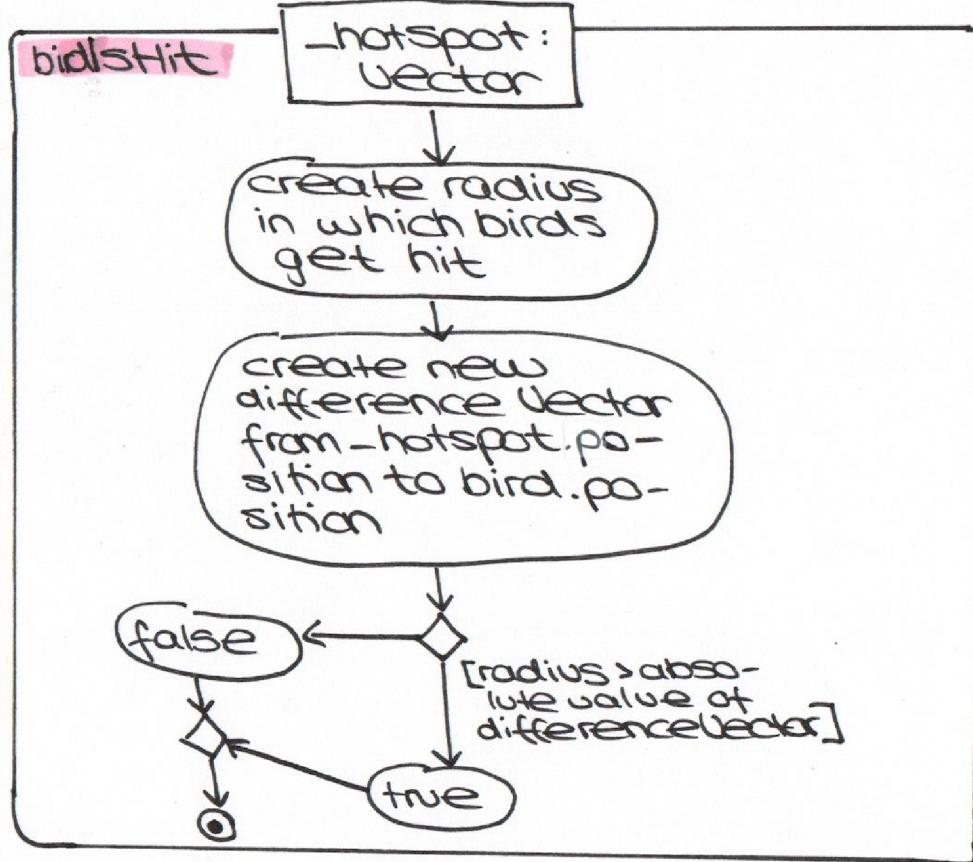
translate to position

scale to size

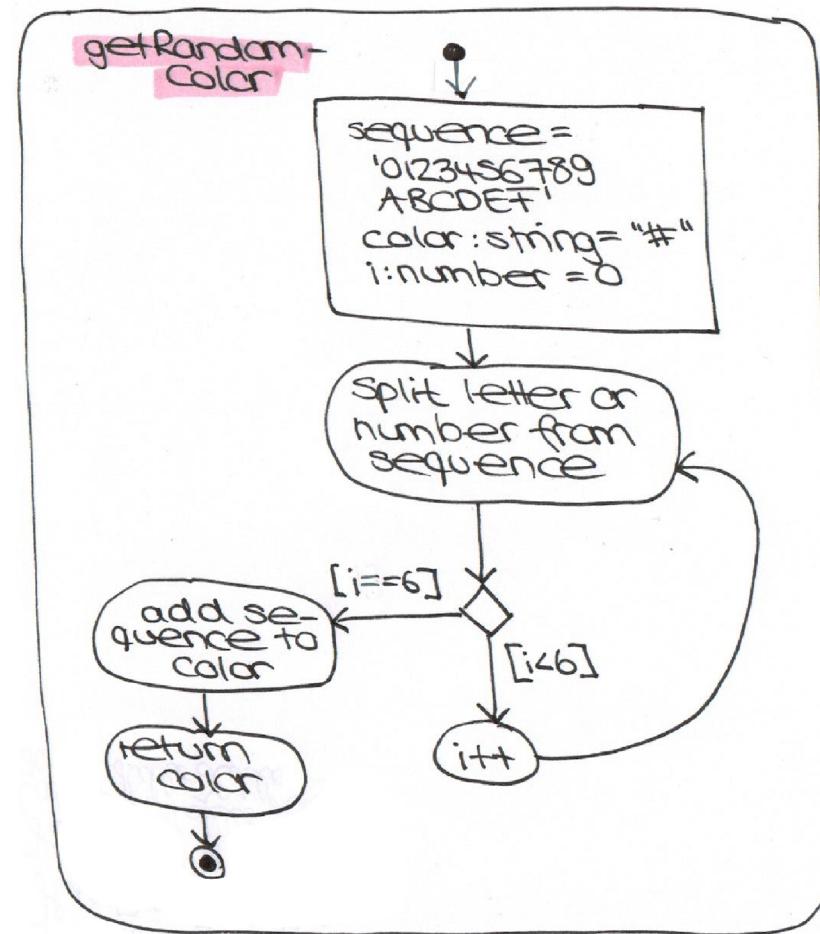
draw path representing type

restore translation

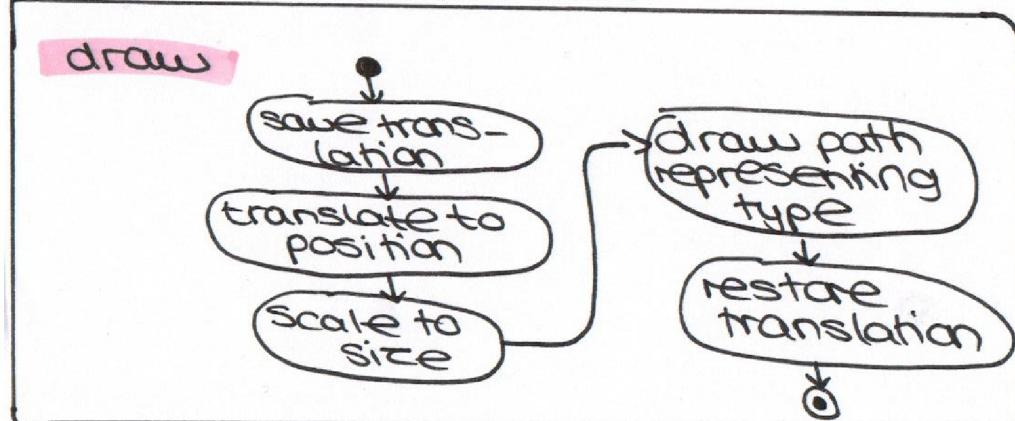
## Bird



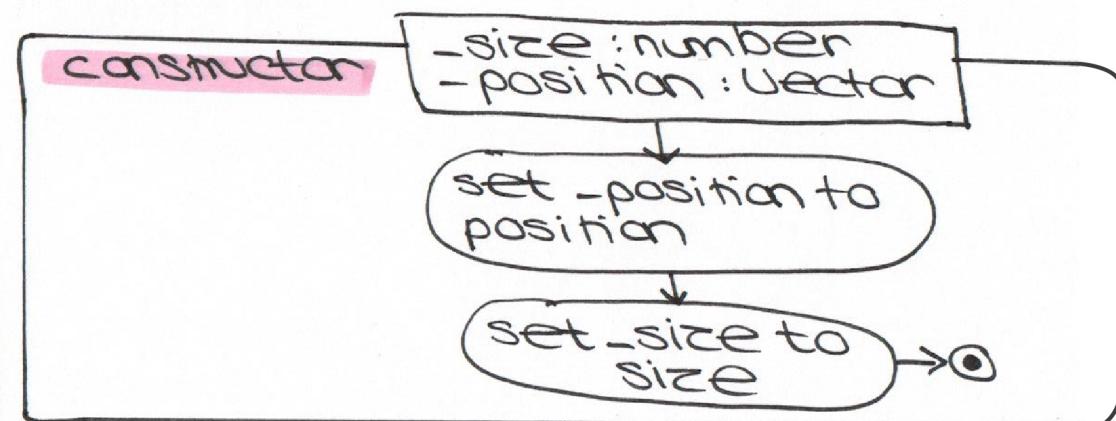
## Bird



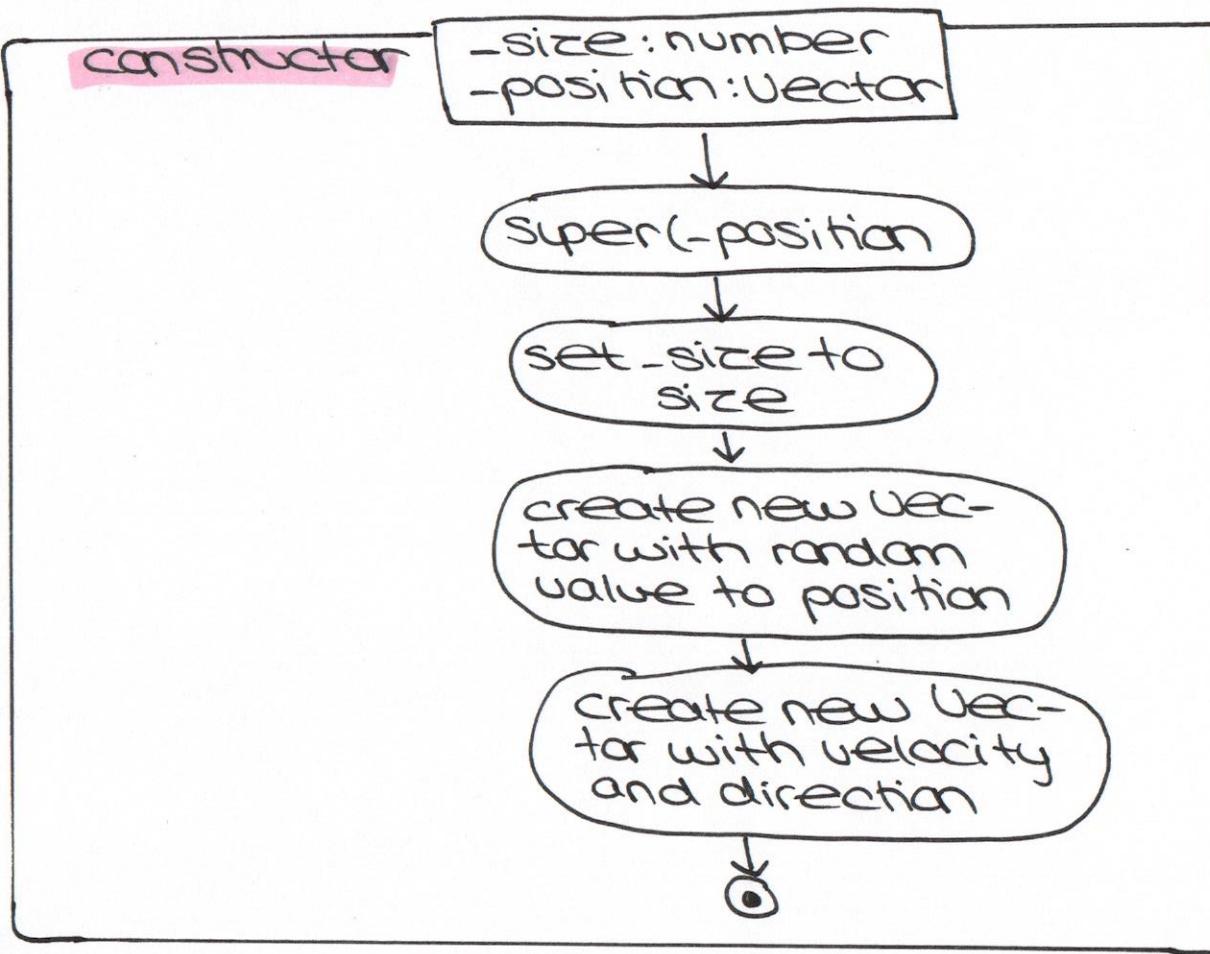
## Snowball & Snowflake



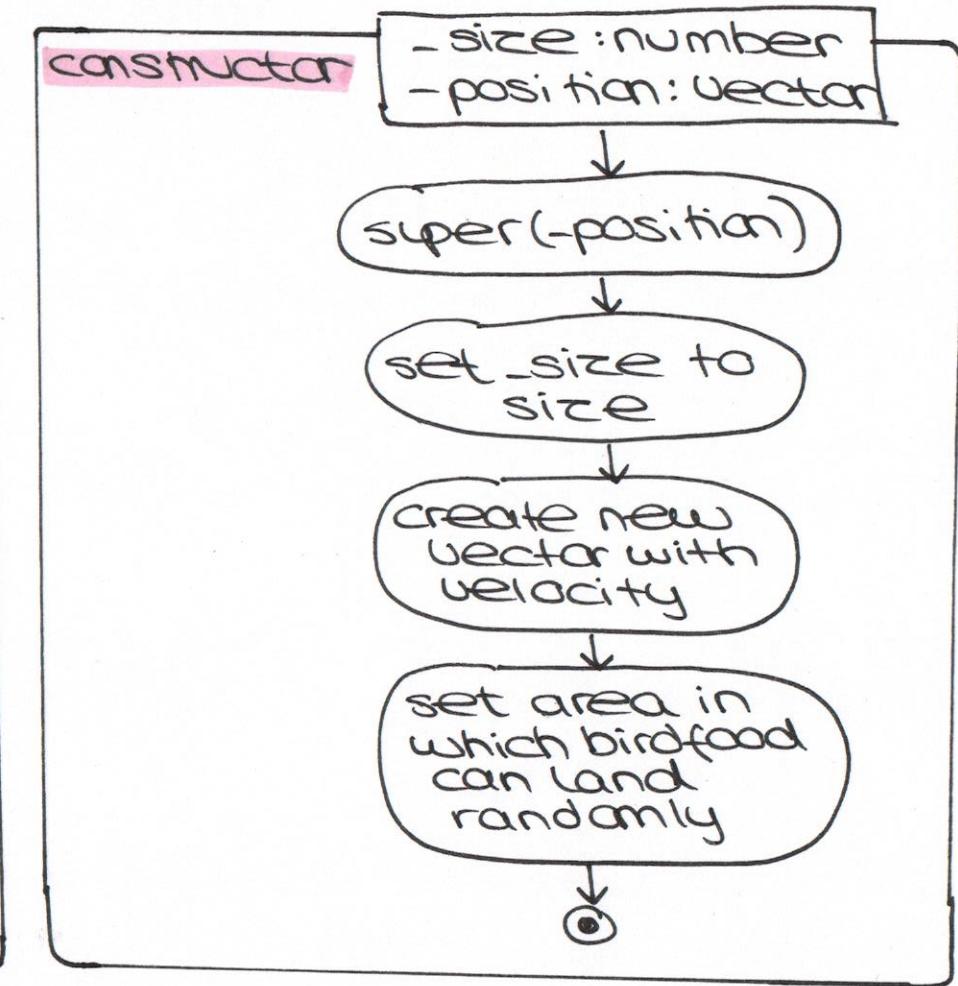
## Snowball



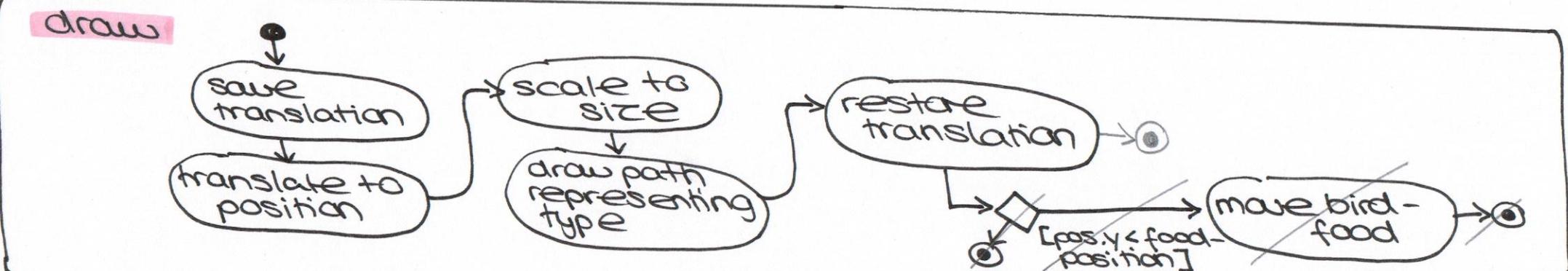
## Snowflake



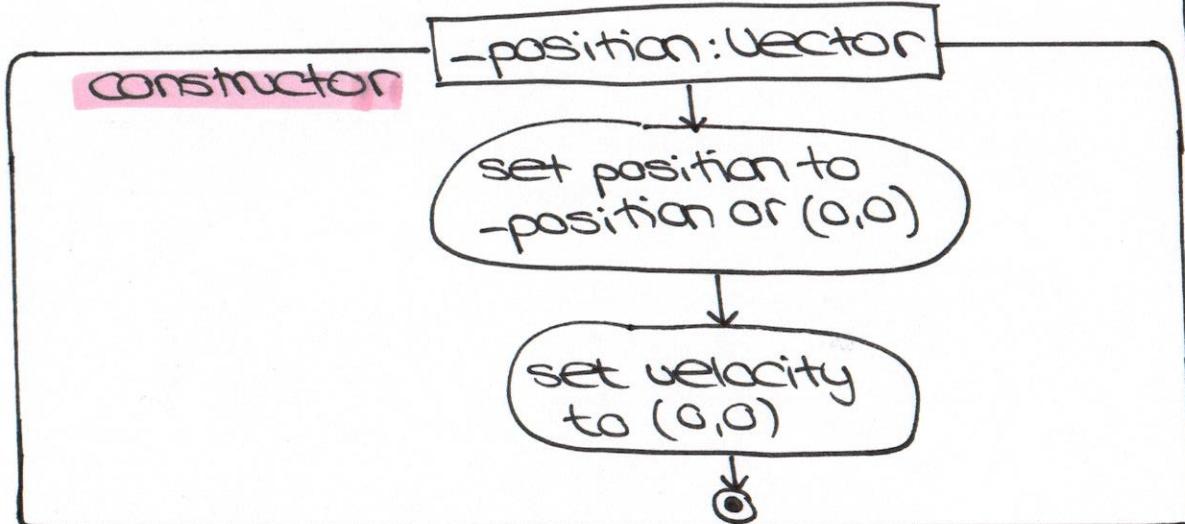
## Birdfood



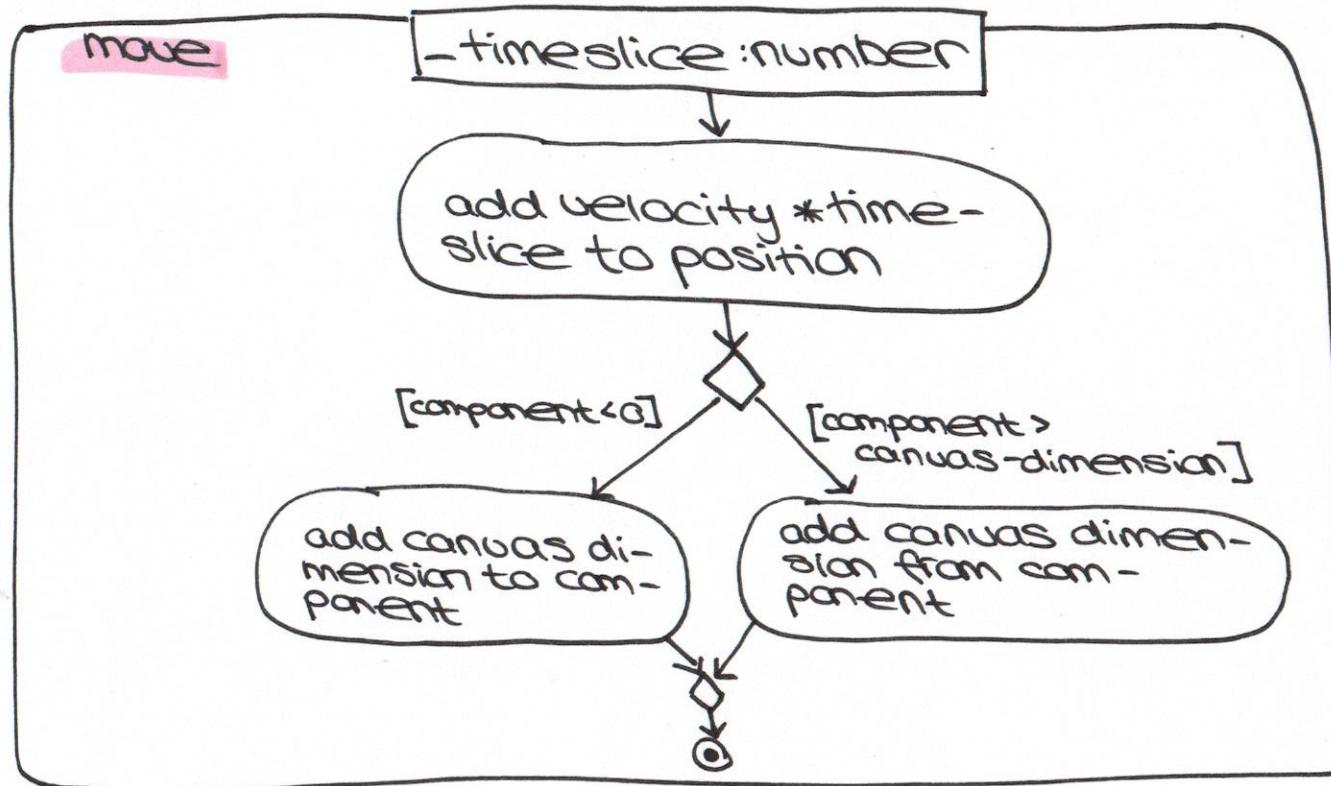
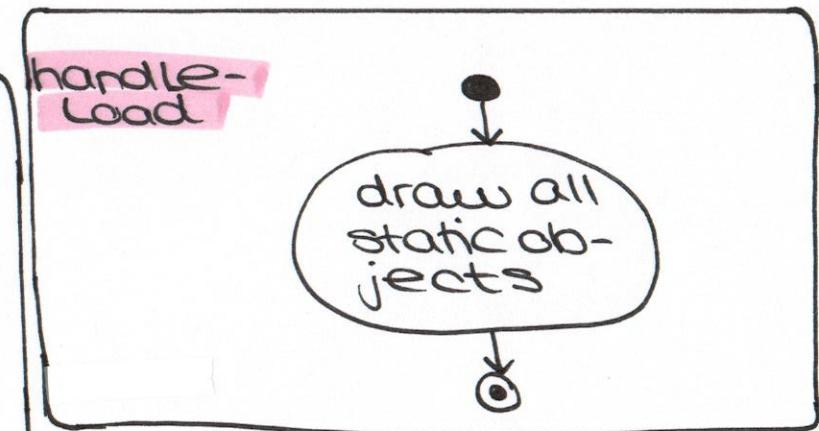
## Birdfood



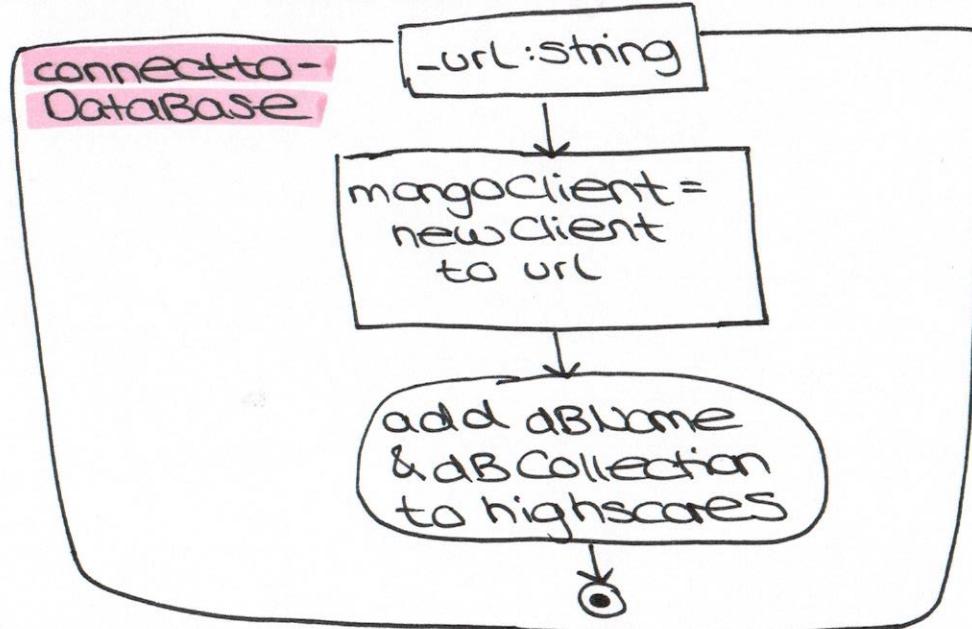
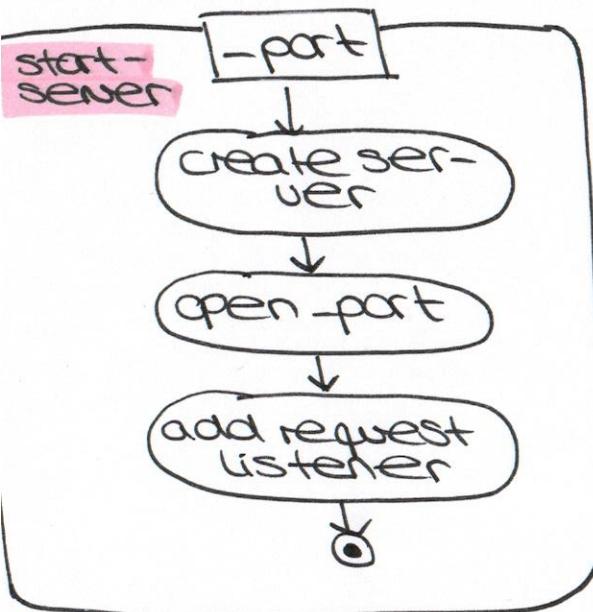
## Moveable



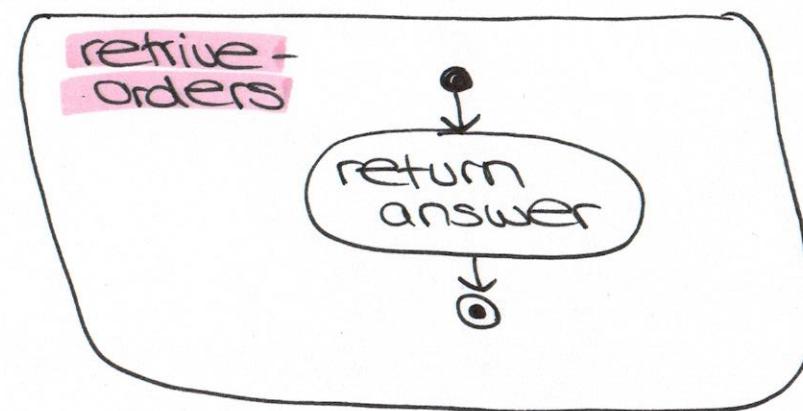
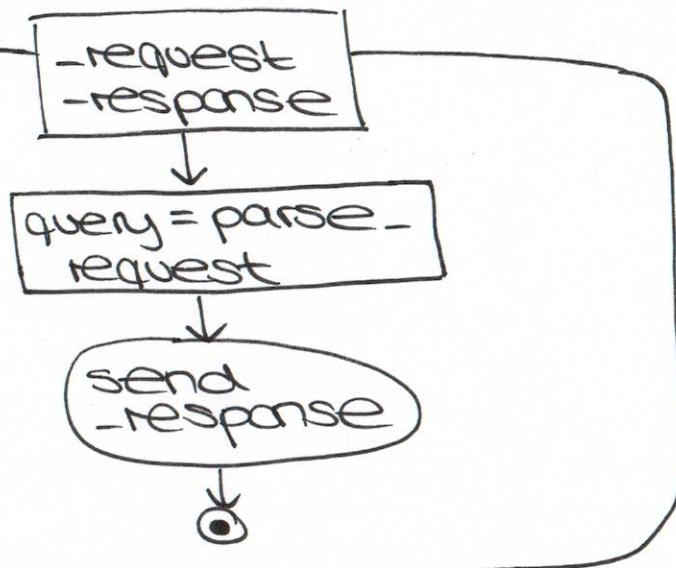
## Canvas

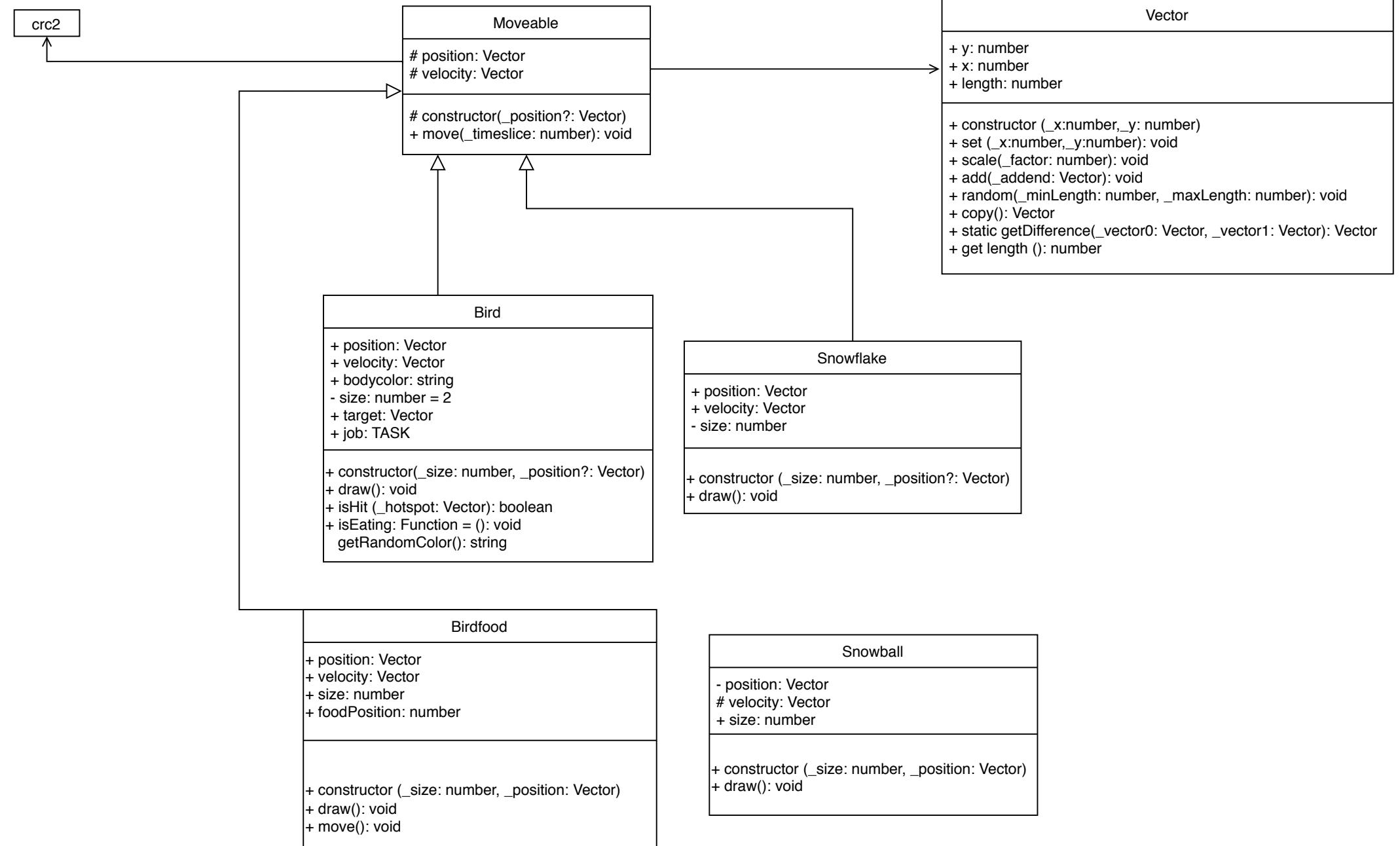


## Database



## handle-request





# Class Diagram