

sun

12

Gradient
- bright yellow
- $a = 0$

Particles
random placed
In Area

cloud

A hand-drawn diagram on graph paper. It features a horizontal line with several circles connected by lines, representing a particle system. The word "Particle" is written next to the diagram.

Gradient
 $a = 0,5$ zu 0

Mountains

random step
to next point

Bird

A simple line drawing of a dog standing on all four legs. The dog has a large, round head with a small body. The word "orange" is written to the left of the head, with an arrow pointing to the dog's neck. The word "windish" is written to the right of the dog's back, with an arrow pointing to the dog's back. The word "stroke" is written at the bottom right, with an arrow pointing to the front right leg.

Bird house

- Gradient
- light brown

Gradient
-brown

Gradient
darkbrown

Gradient
- darkgreen

Gradient
- downward

100

X → random
y = 400px
place

snowflake

snowman

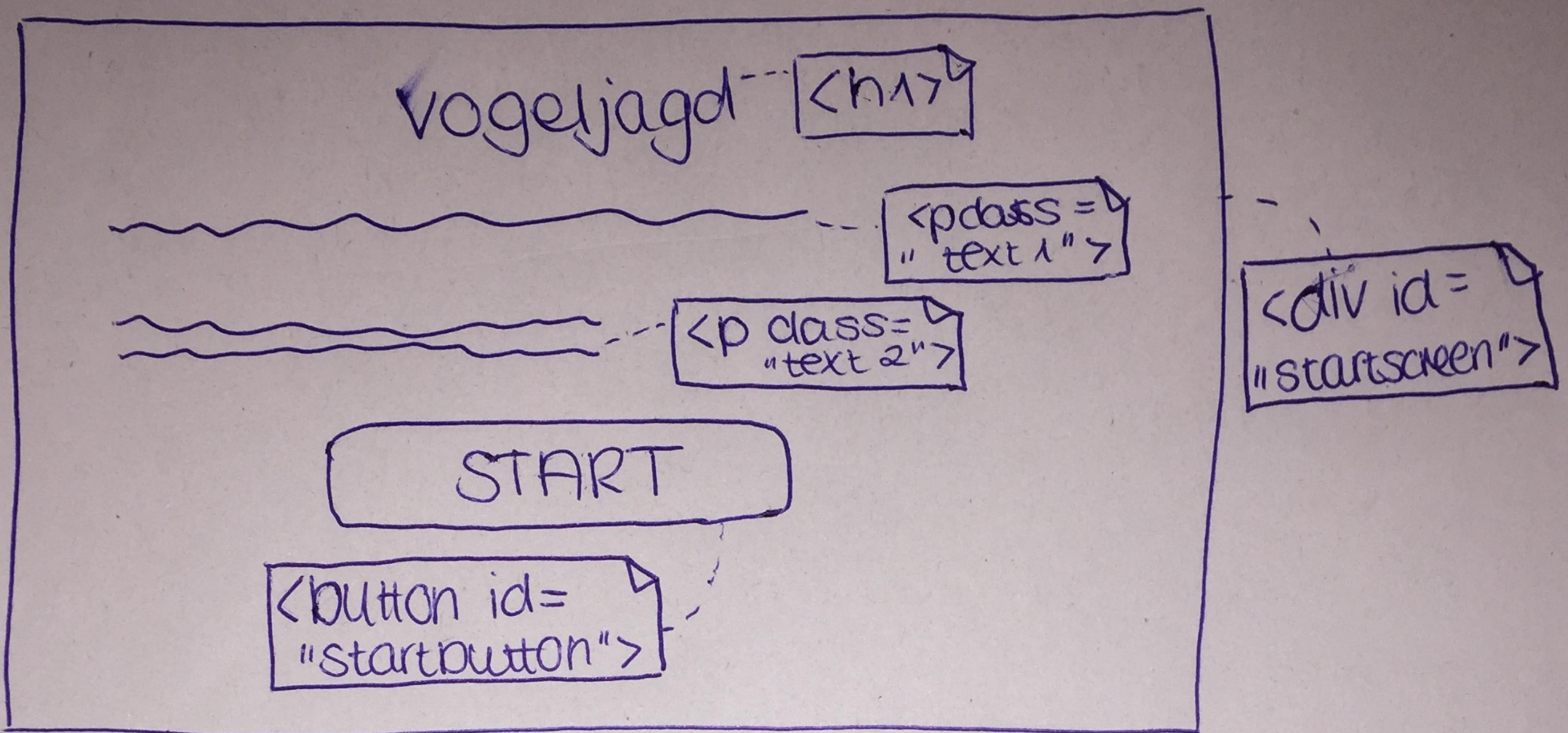
卷之三

particles
pieces random
in area

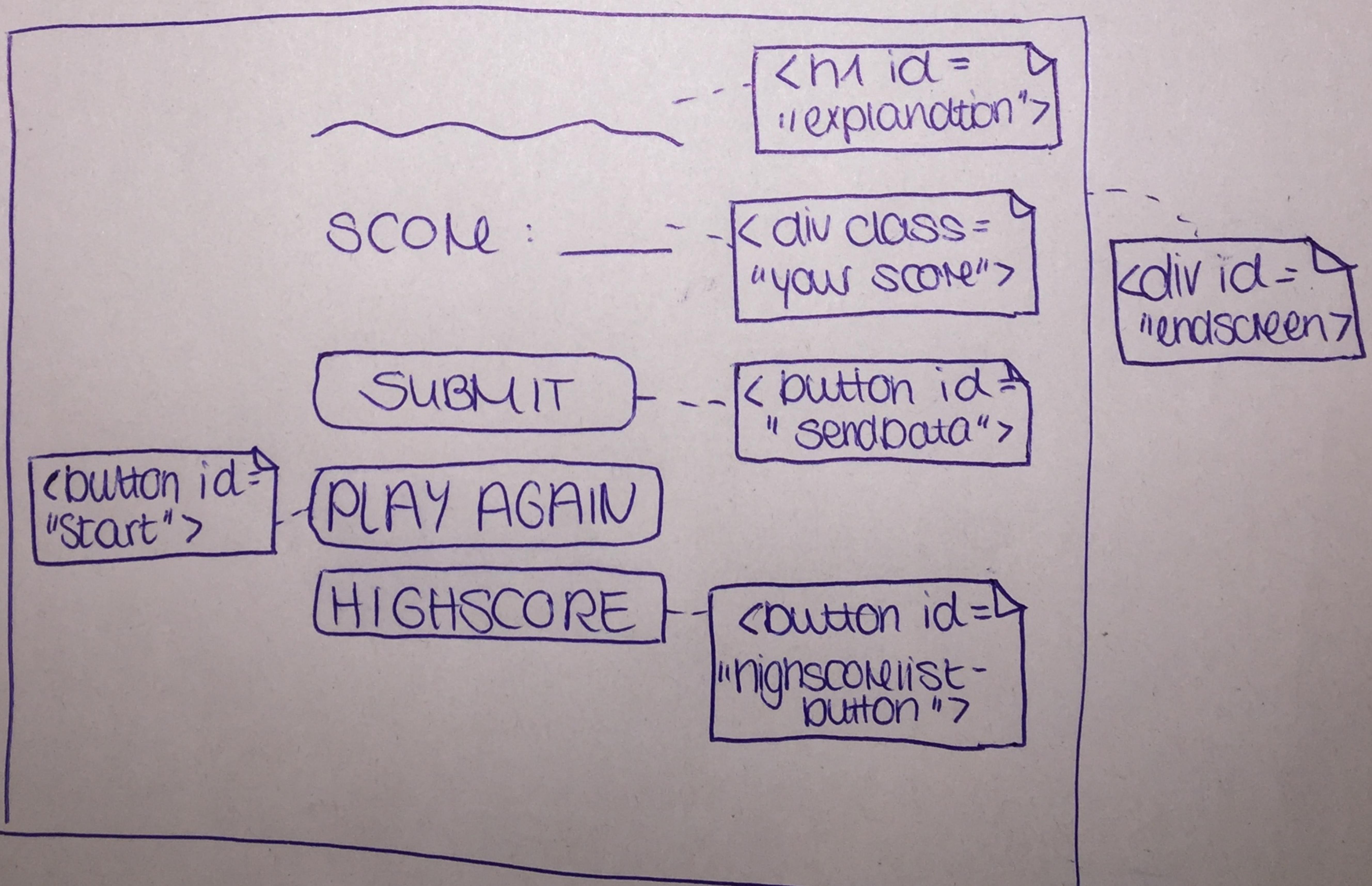
A simple line drawing of a snowman on a grid background. The snowman has three stacked circular body parts, two coal eyes, and a carrot nose. It is holding a single snowball in its right hand.

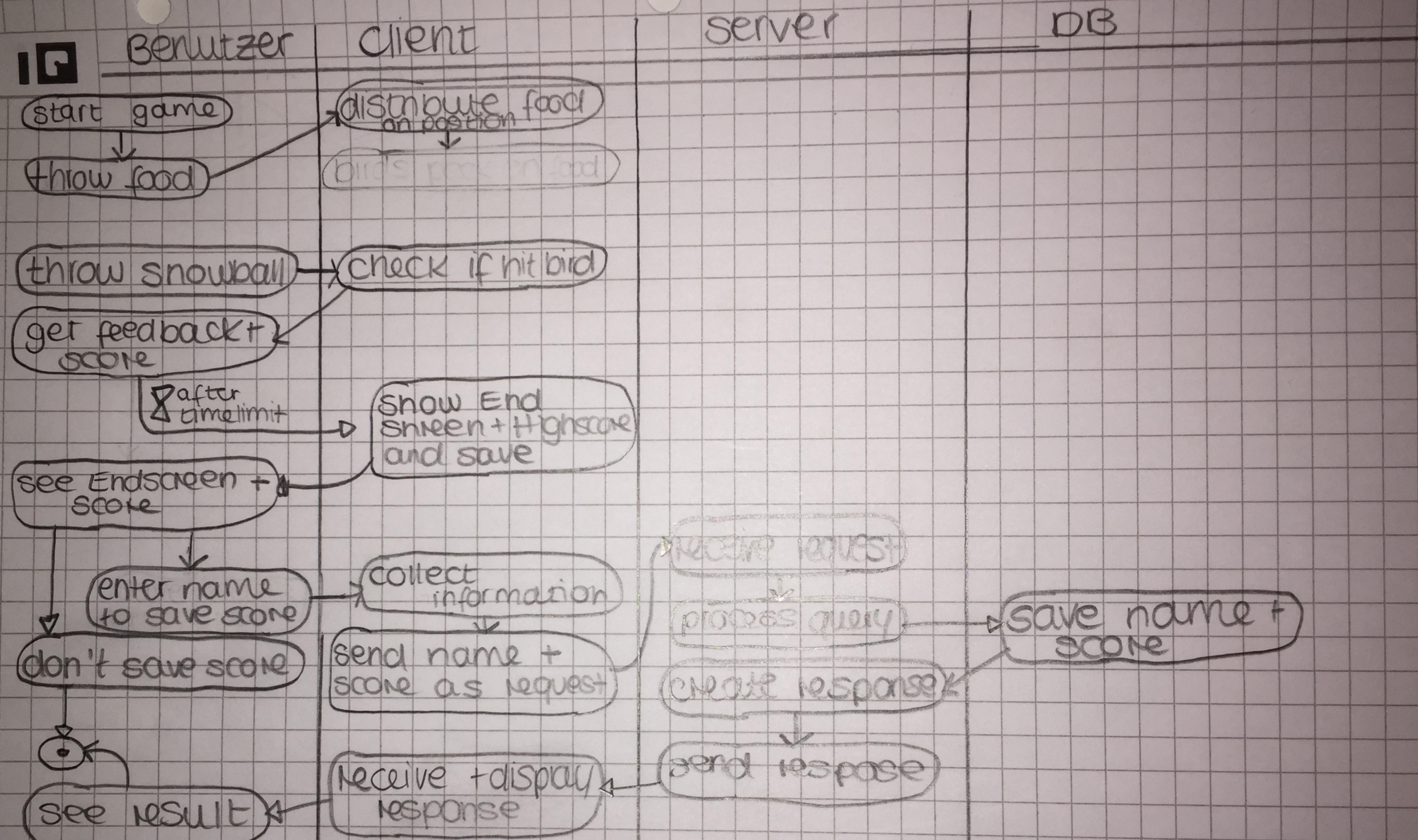
Godent
- white


Scribble Start



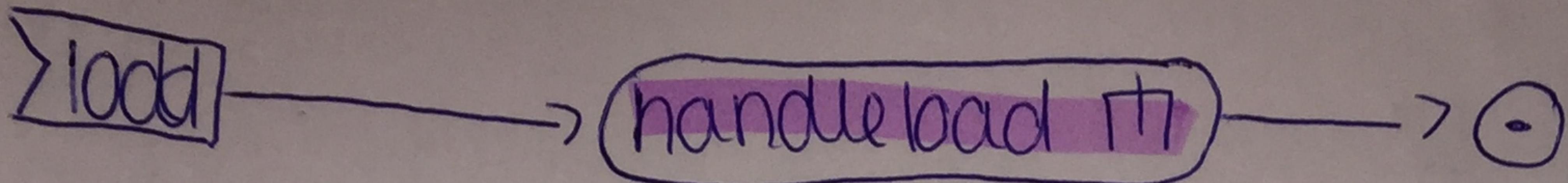
Endscreen







main



handleLoad

get Rendering Context &
setup drawing style

install listeners → handleclick,
handleRightclick

generateBird

generate Snowflakes

generate score

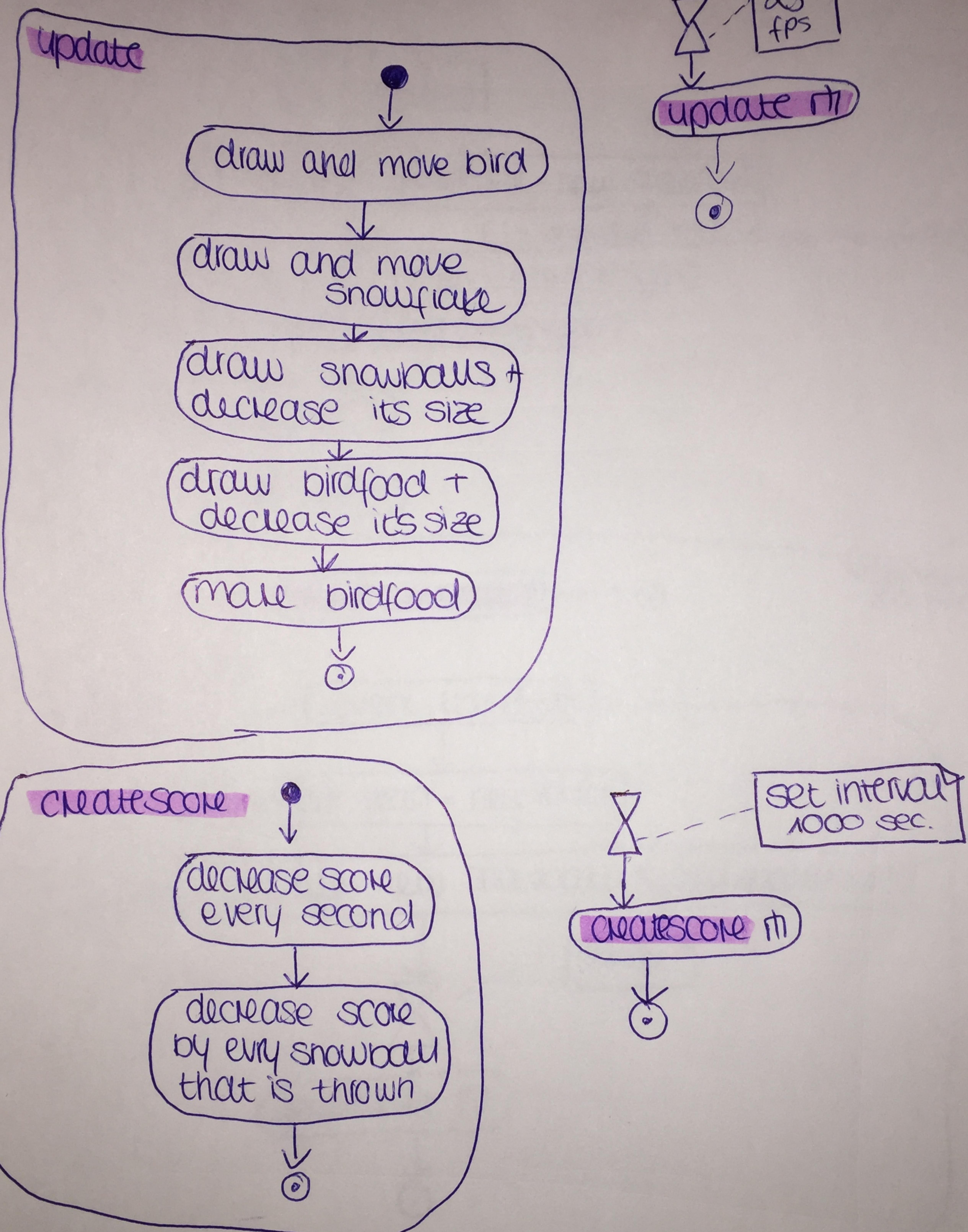
generateScore

let SCORE: number = 1000

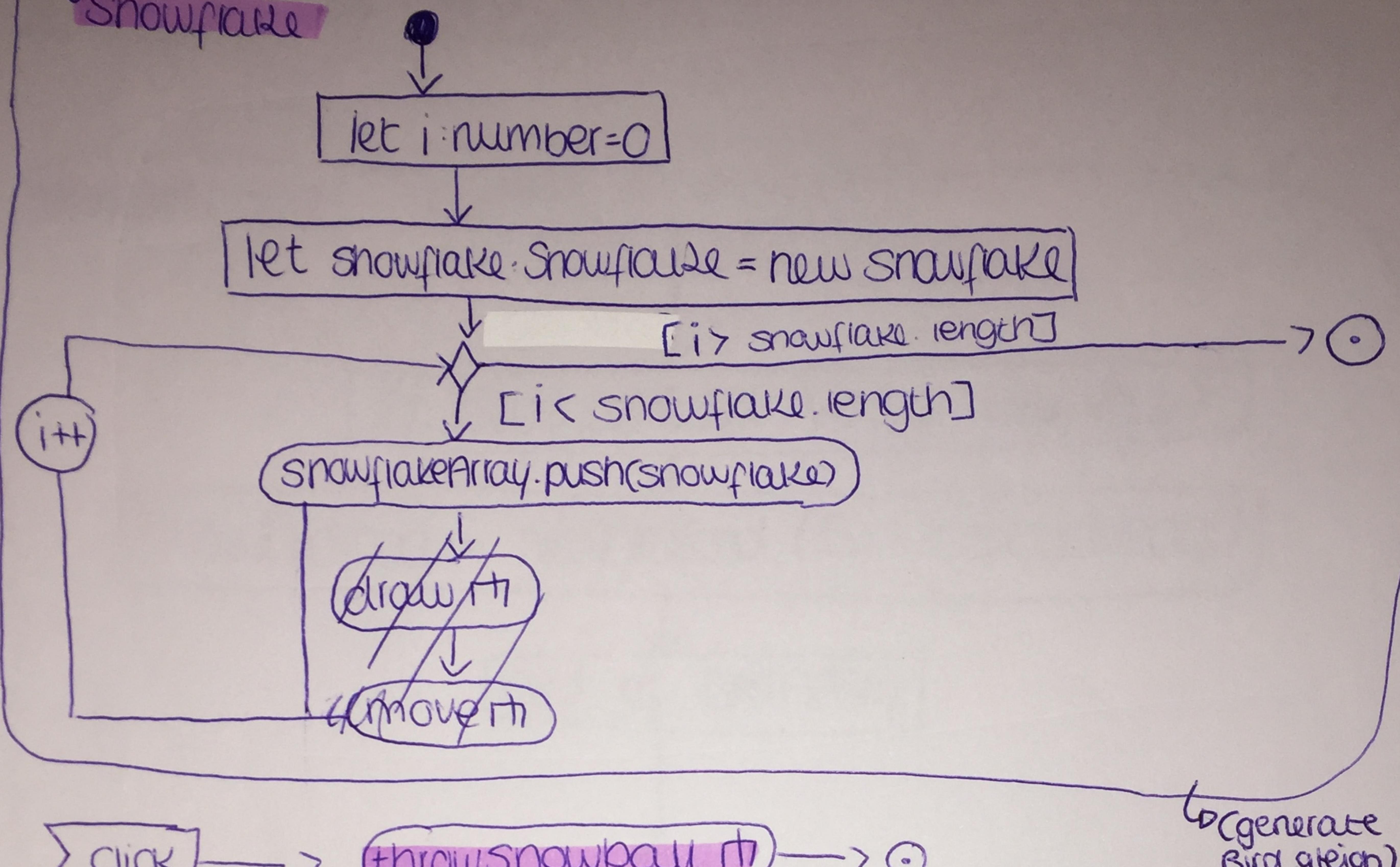
festlegung von Intervall

score --

main



generate Snowflake

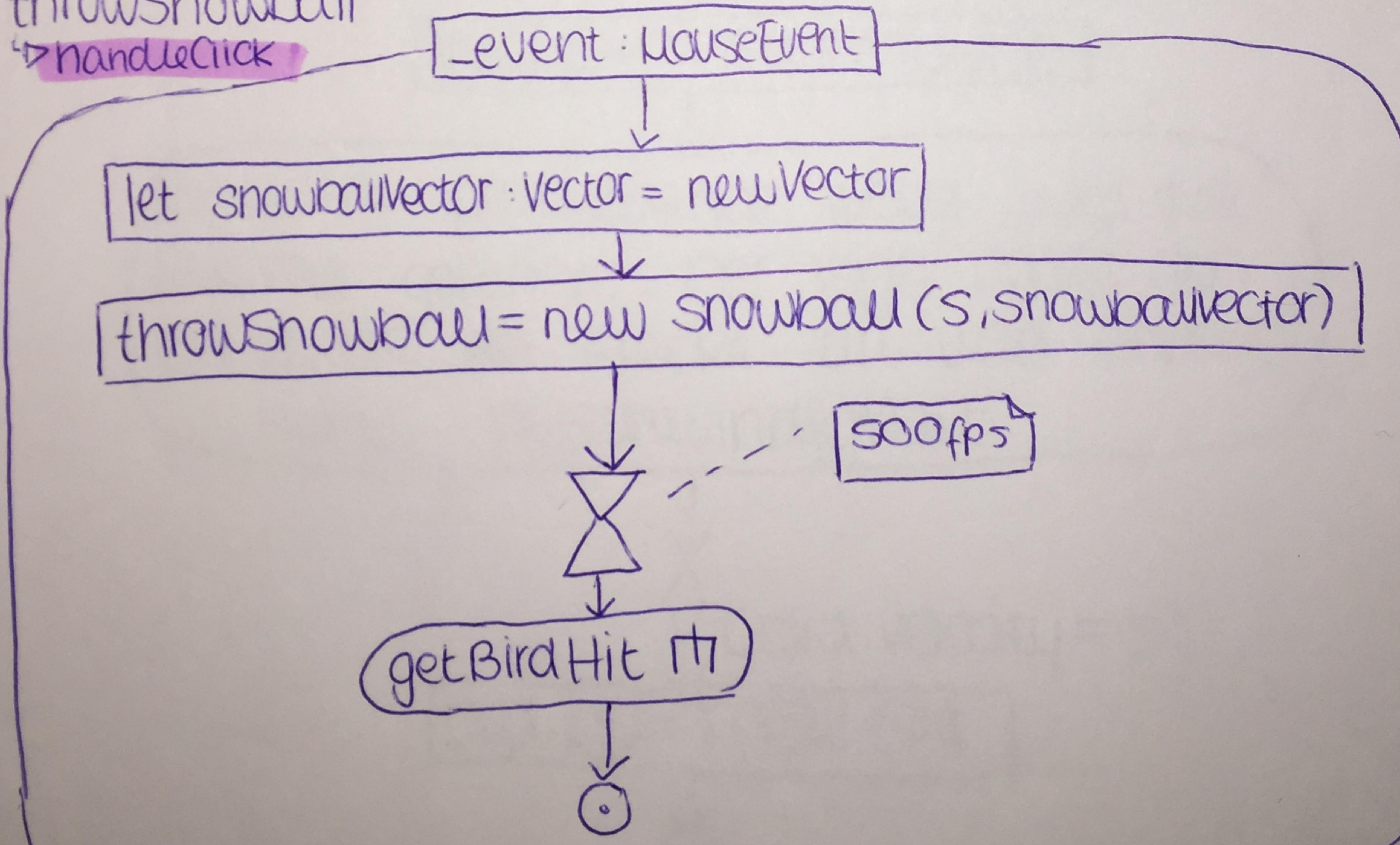


click → throwSnowball rit → ○

↳ (generate Bird gteion)

throwSnowball

↳ handleClick



main

click → handleRightclick (t) → ⊙

handleRightclick

-event: MouseEvent

let birdFoodVector: Vector = NewVector

throwBirdFood = newBirdFood(5, birdFoodVector)

bird of birdArray

[(isNear(bird.position))]

this.job = TASK.FLYTOFOOD

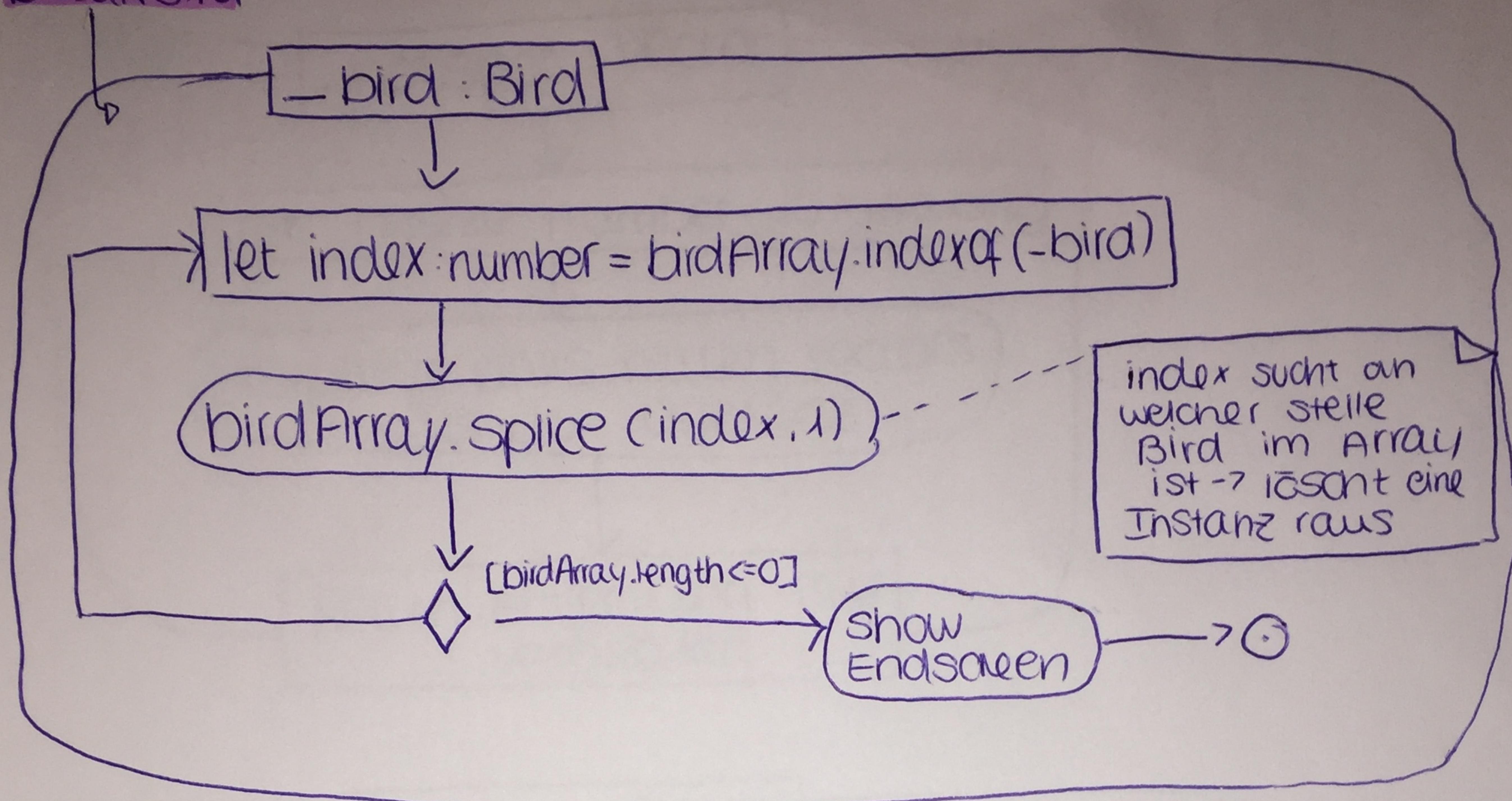
aus dem getDifference vector wird eine Strecke generiert. Der Vogel passt sich an diese ~~Strecke~~ Strecke an und ändert seine Geschwindigkeit

[bird.velocity = 0]

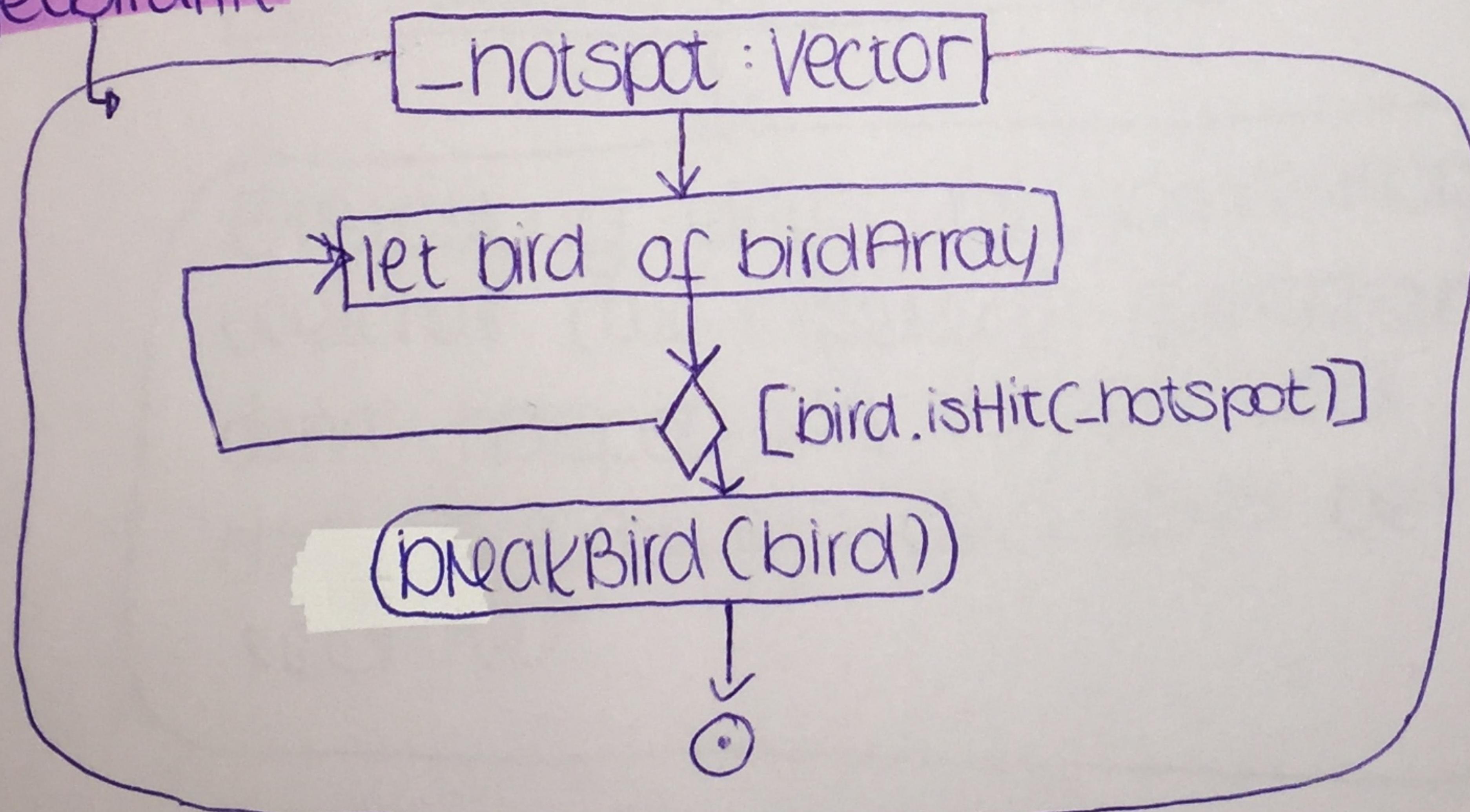
bird.job = TASK.EAT

main

breakBird



getBirdHit



main

isHit

-hotspot : vector

let hitsize: number = 10 * this.size

Erstellung eines neuen Vektors aus der Differenz

Return Entfernung zur verbikalen

Bird

isNear

-hotspot : vector

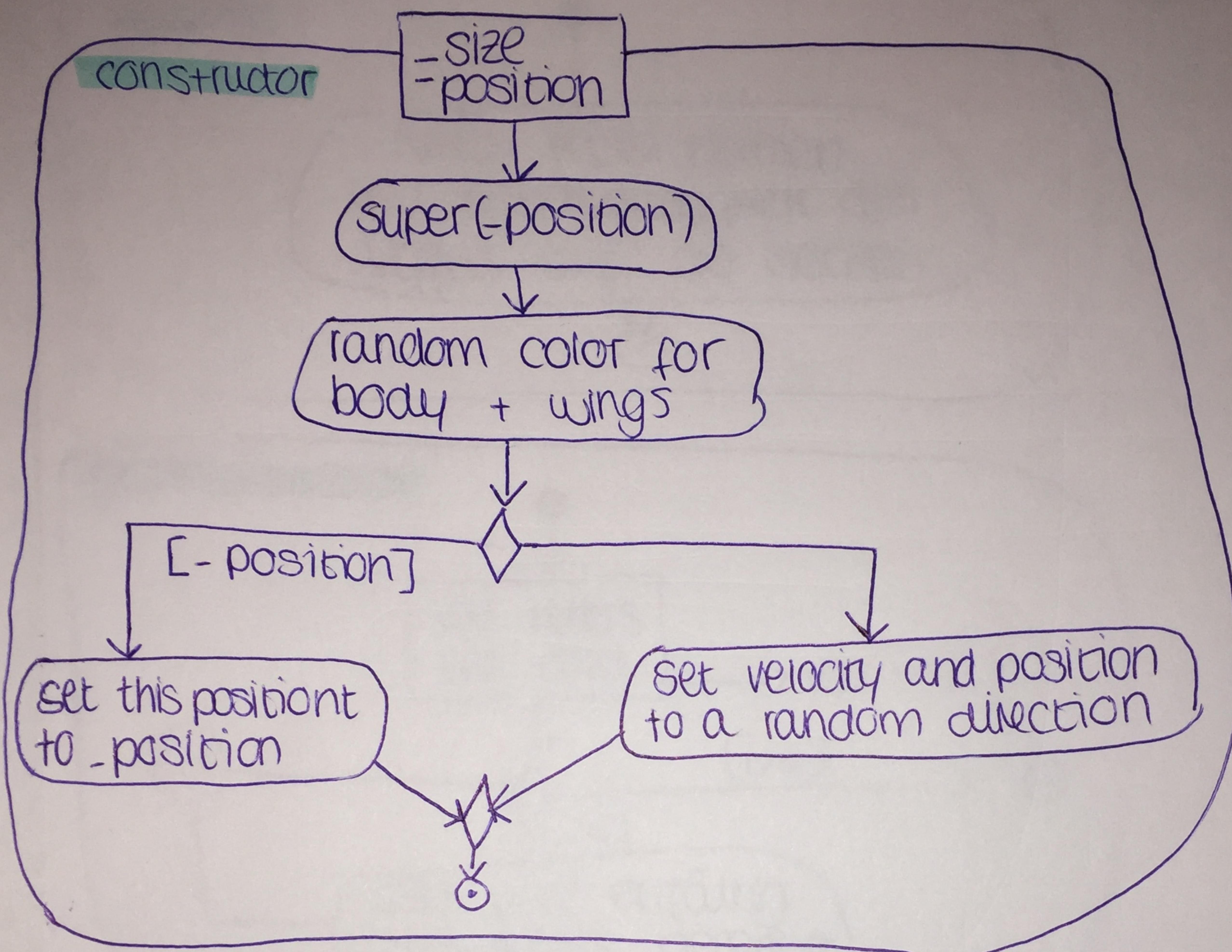
let nearsize: number = 100;

Erstellung eines neuen Vektors, welcher die Differenz zwischen dem hotspot des Vogels und der Endposition des Keks berechnet.

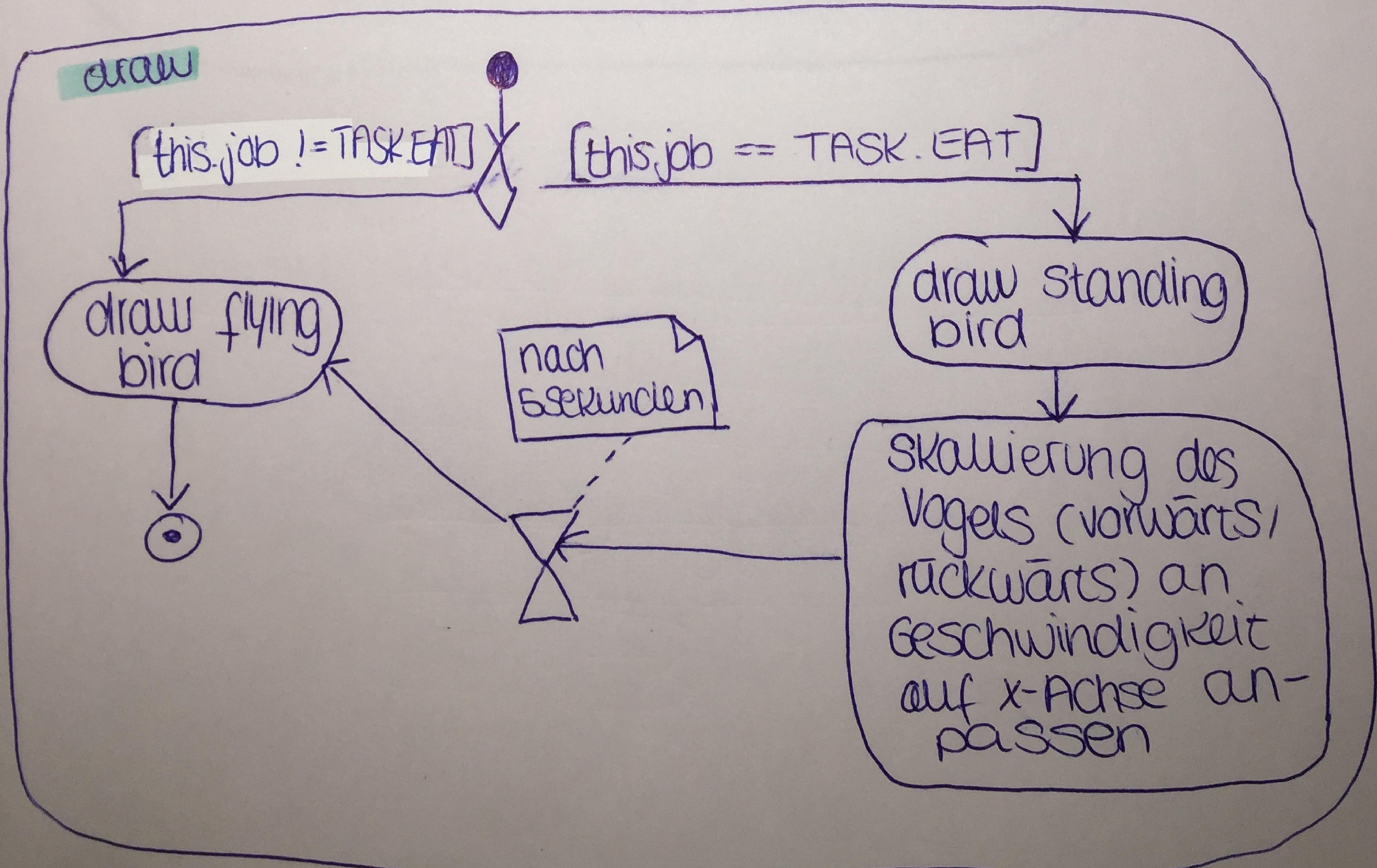
Return Entfernung zum Keks

main

Bird



draw



isPidaing

Bird

neuen vector erstellen
und Geschwindigkeit des
Vogels auf 0.0 setzen

getRandomColor

Var letters
Var color

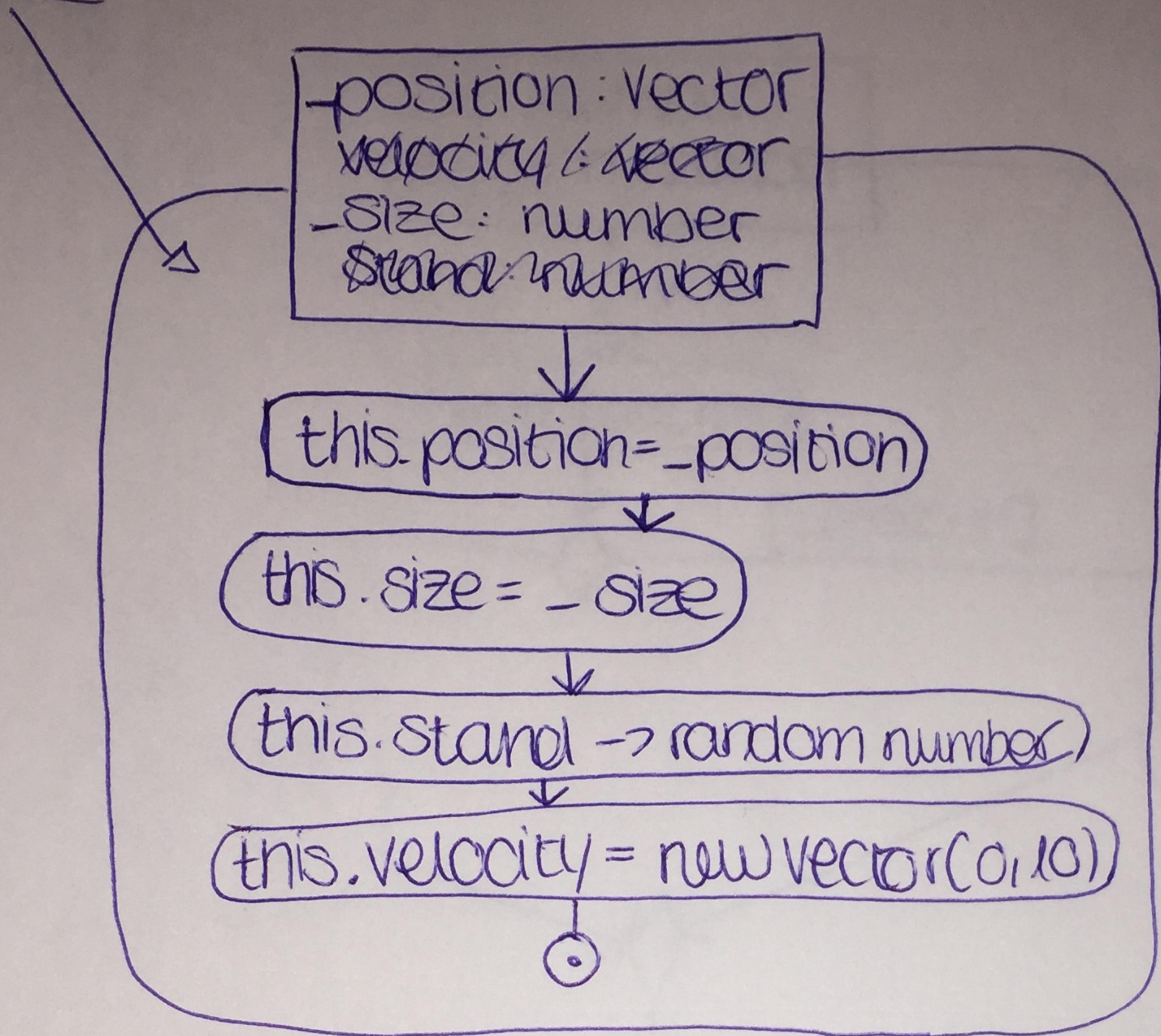
var i=0

i++

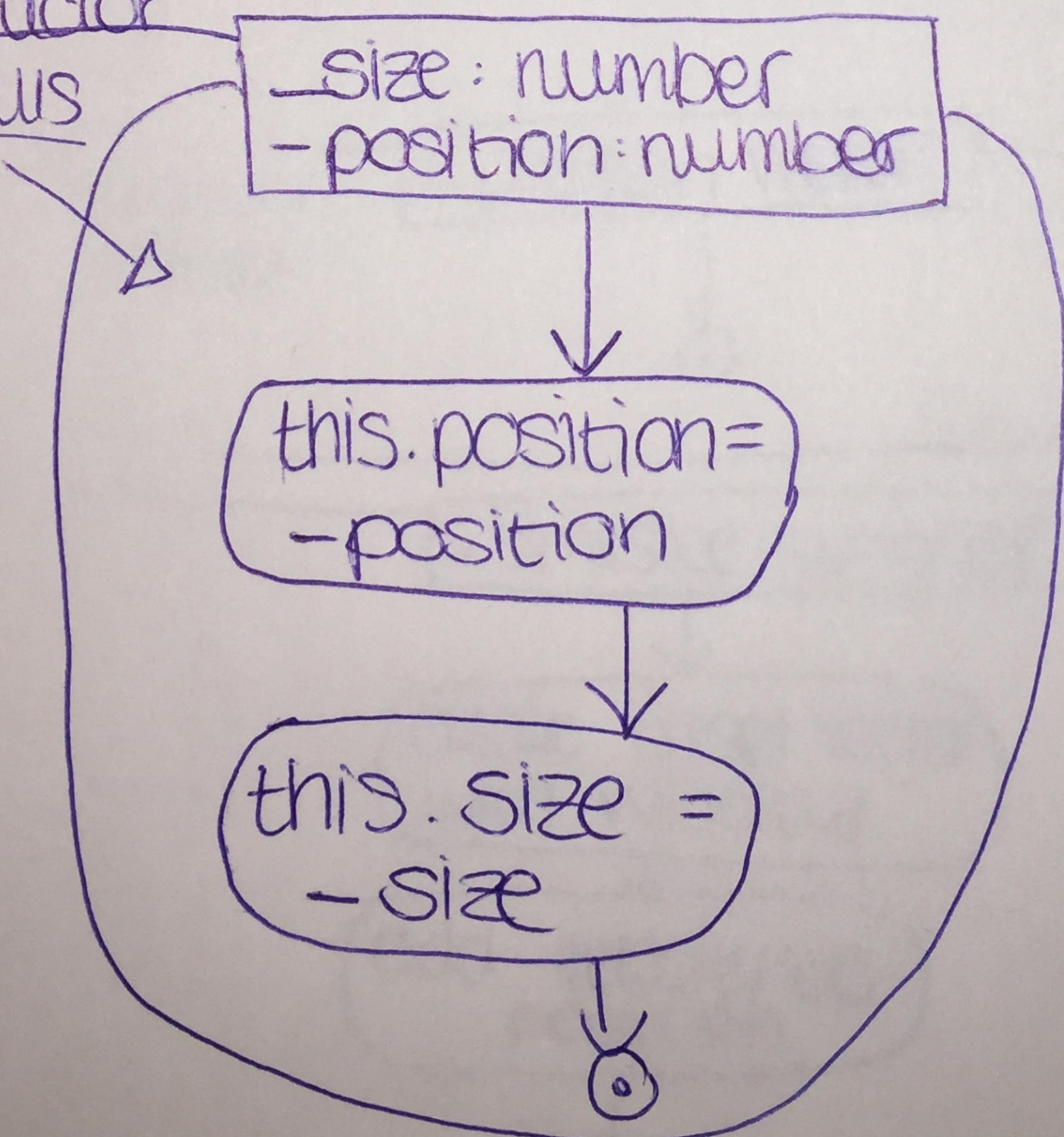
[i>6]

[i<6]
color wird zufällig
aus letters generiert

Constructor Birdfood



constructor snowballs



snowflakes

