

Eleonora Wolf – EIA2 – Endabgabe

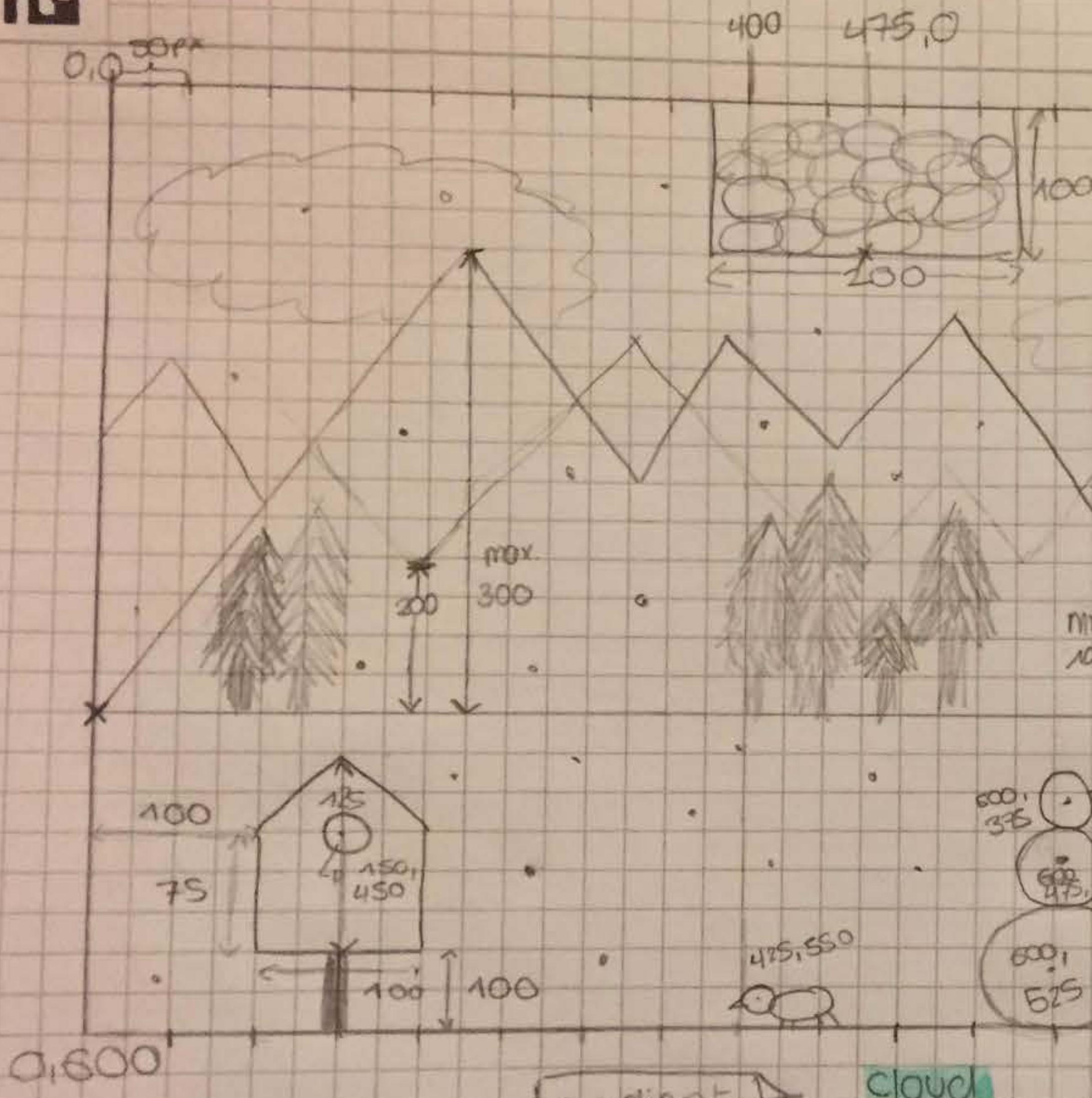
Vogeljagd

Anleitung zur Interaktion:

15 Vögel fliegen wild durch eine Winterlandschaft. Der Spieler hat die Aufgabe diese Vögel so schnell wie möglich abzuwerfen. Dazu kann er mit einem Klick auf die rechte Maustaste Schneebälle werfen, die mit etwas Geschick einen Vogel treffen. Mit dem Klick auf die rechte Maustaste kann der Spieler Futter verstreuen und die Vögel locken. Dazu muss er darauf achten, dass die Vögel in einer gewissen Reichweite sind. Die Vögel bleiben jedoch nur eine gewisse Zeit stehen, bevor sie wieder die gewohnte Position annehmen und weiterfliegen.

Kriterien der Plattform:

Meine Anleitung ist vorerst für den Desktop ausgelegt, da dieser eine große Fläche zur Darstellung der Landschaft bietet. Des weiteren sind Click-Events eingebaut, welche für den Spielverlauf benötigt werden.



sun

12

Gradient
- bright yellow
- orange

Particles
randomly placed
in area A

cloud

A hand-drawn diagram on grid paper. It features five interlocking circles representing the Olympic rings in the upper left. To the right of the rings, the word "particle" is written vertically. Below the word, there is a small circle containing a cross, and below that, a larger circle containing a smaller cross.

Gradient
 $a = 0,5$ zu 0

Mountains

random step
to next point

Bird

A simple line drawing of a dog standing on a grid. The dog is facing left, with its head turned slightly towards the viewer. Two arrows point from the word "orange" to the dog's head. One arrow points from the word "strove" to the dog's front right paw.

Bird house

- Gradient
- light brown

Gradient
-brown

Gradient darkbrown

Gradient
- darkgreen

Gradient
downward

TNEC

$X \rightarrow$ random
 $y = 400px$
place

snowflake

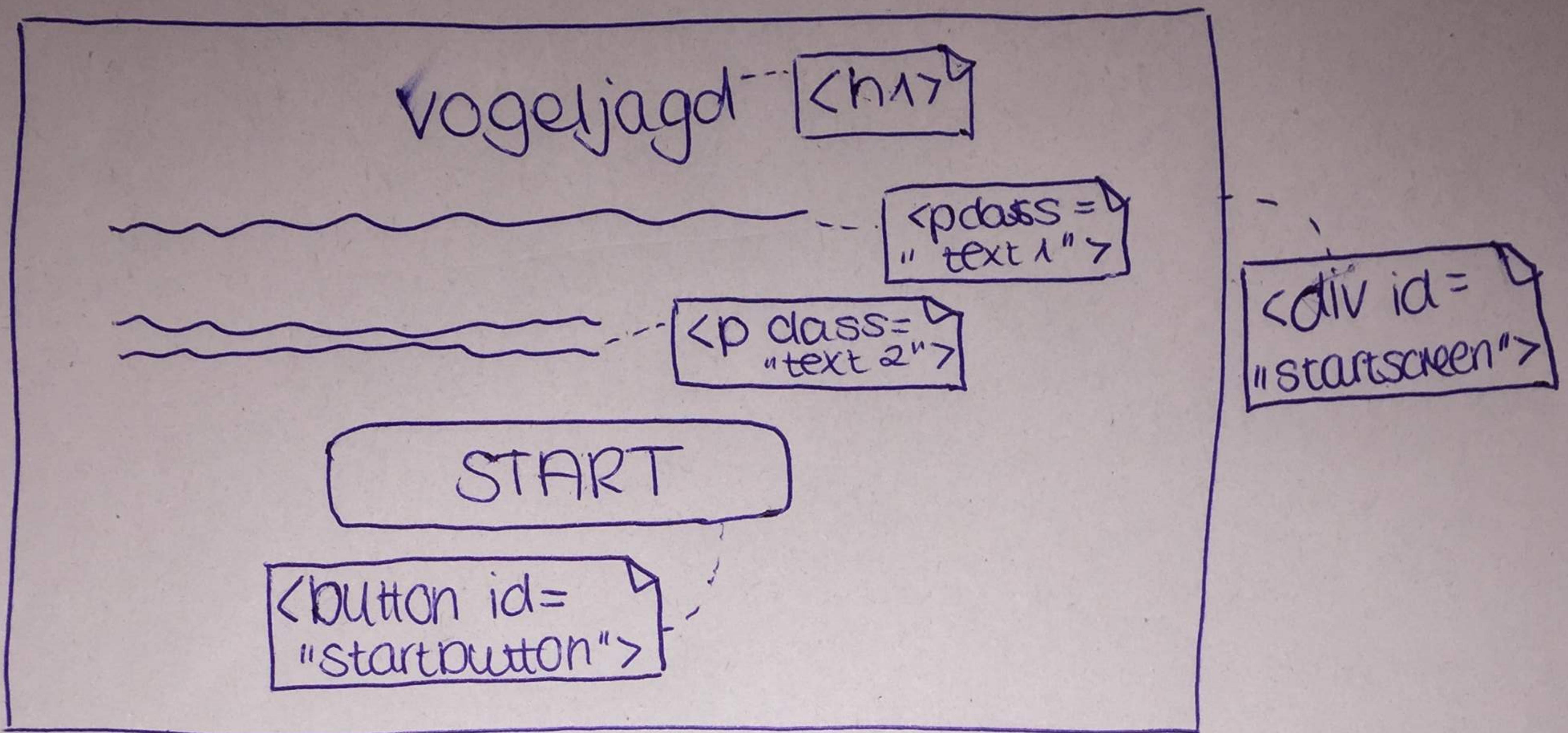
snowman

100

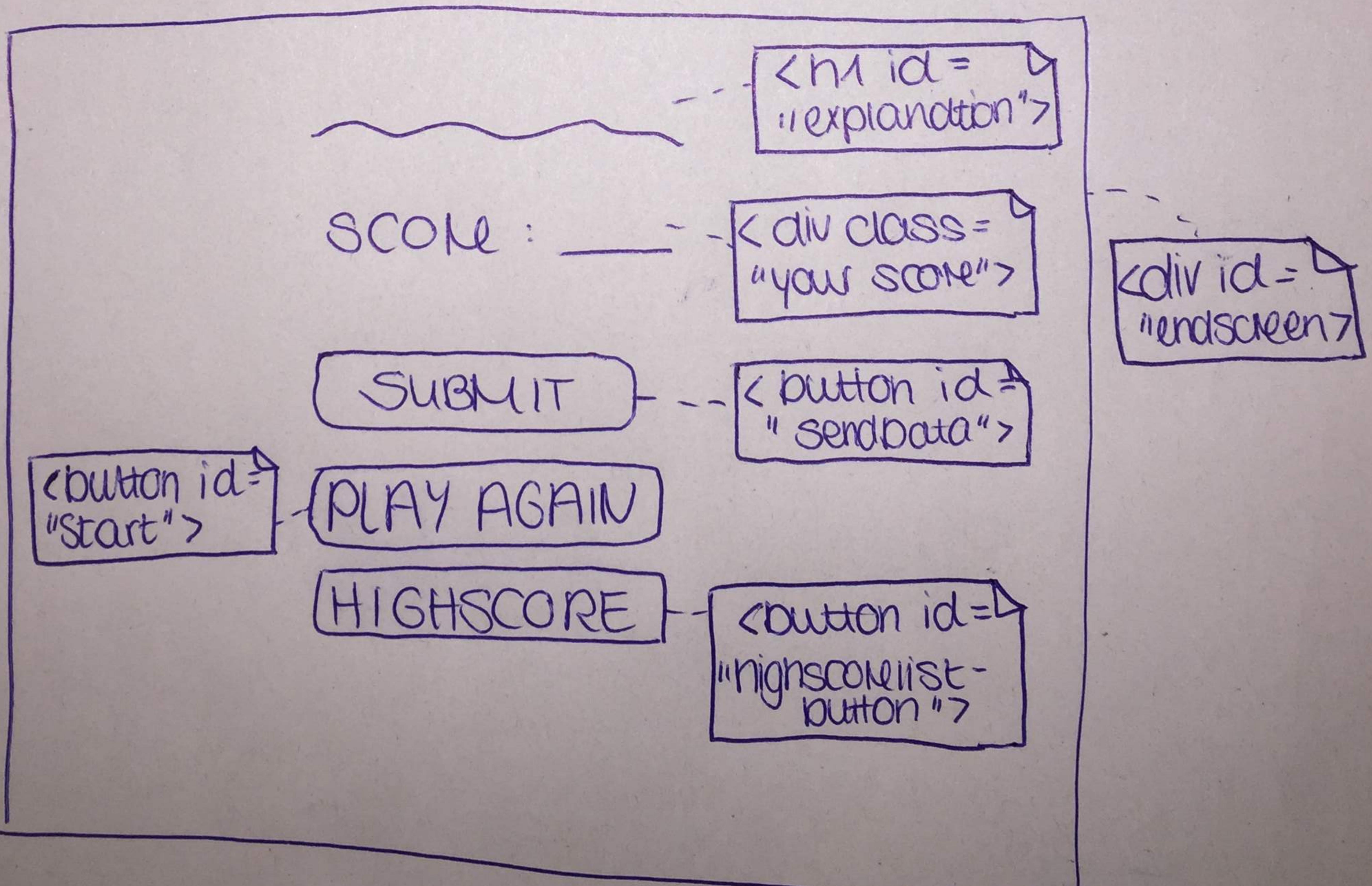
particles
pieces random
in area

Godent
- white

Scribble Start



Endscreen





Benutzer

start game

throw food

throw snowball

get feedback +
scoresee Endscreen +
scoreenter name
to save score

don't save score

see RESULT

client

distribute food
on position

birds pos. on food

check if hit bird

show End
screen + Highscore
and savecollect
informationsend name +
score as requestReceive + display
response

server

DB


CREATE REQUEST

PROCESS QUERY

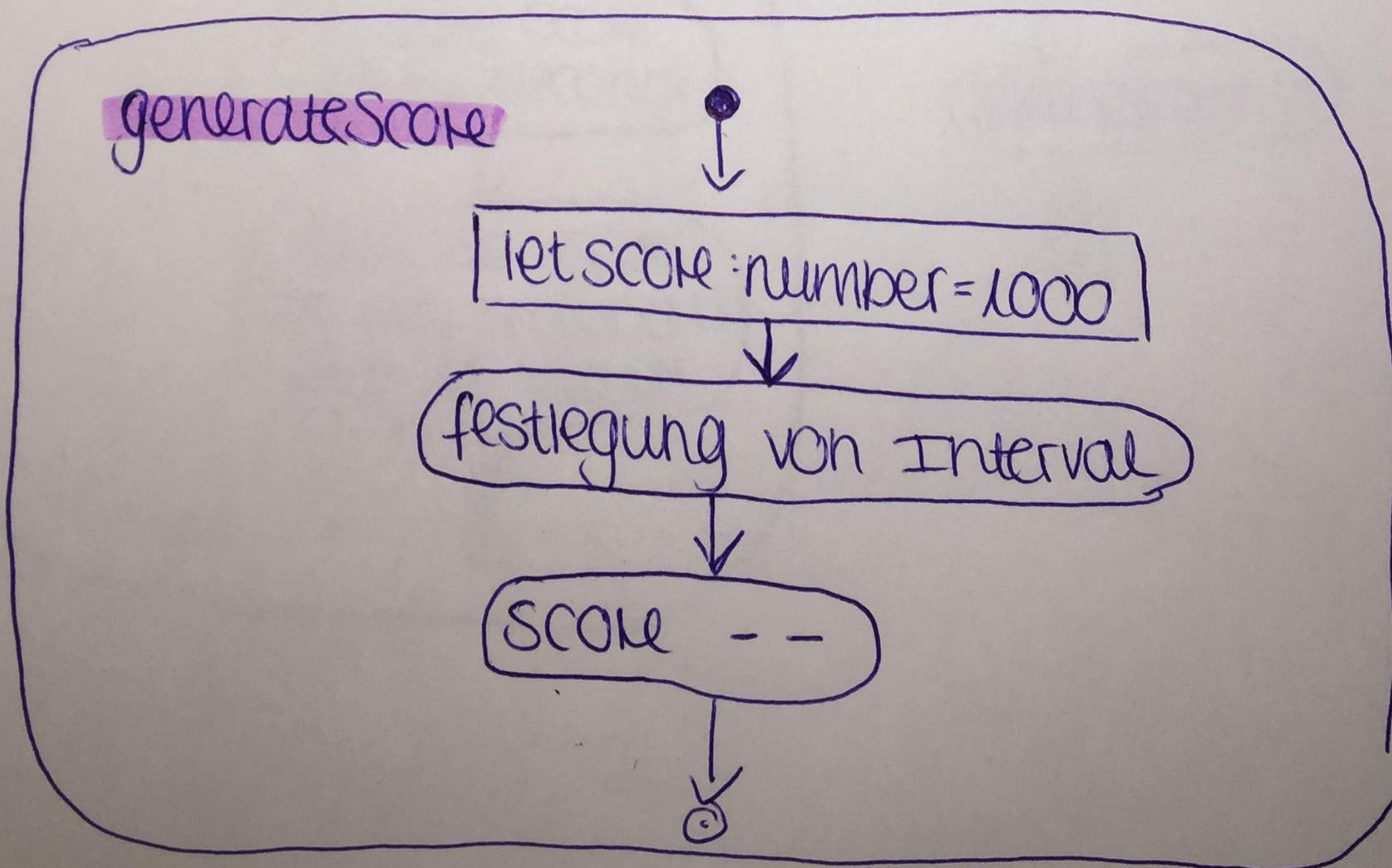
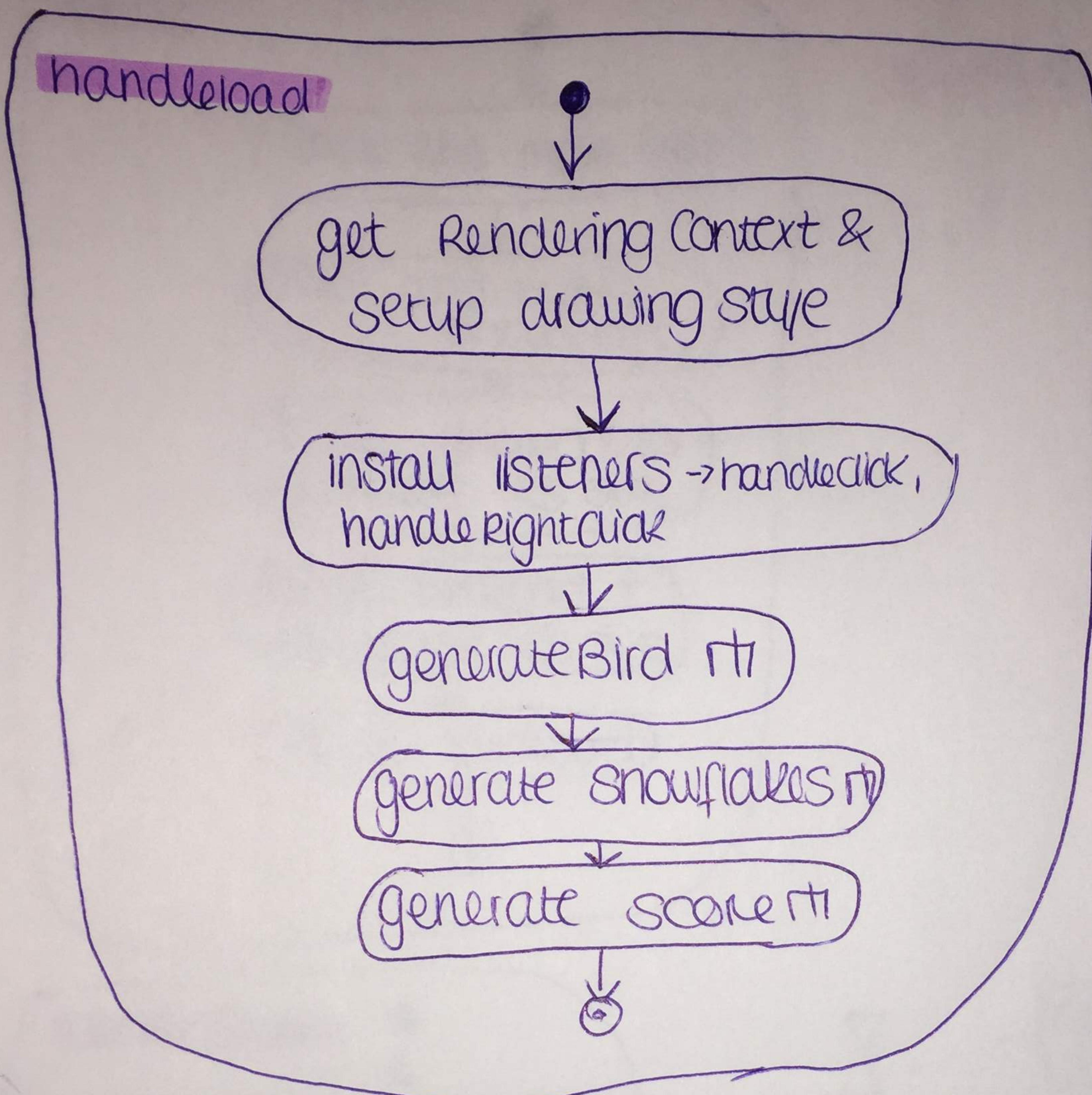
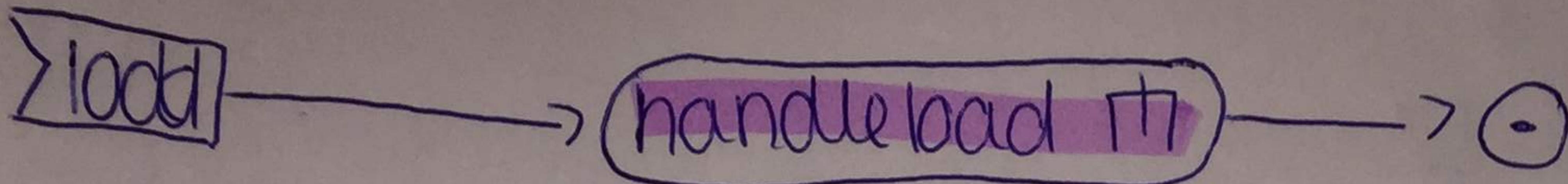
CREATE RESPONSE

SEND RESPONSE

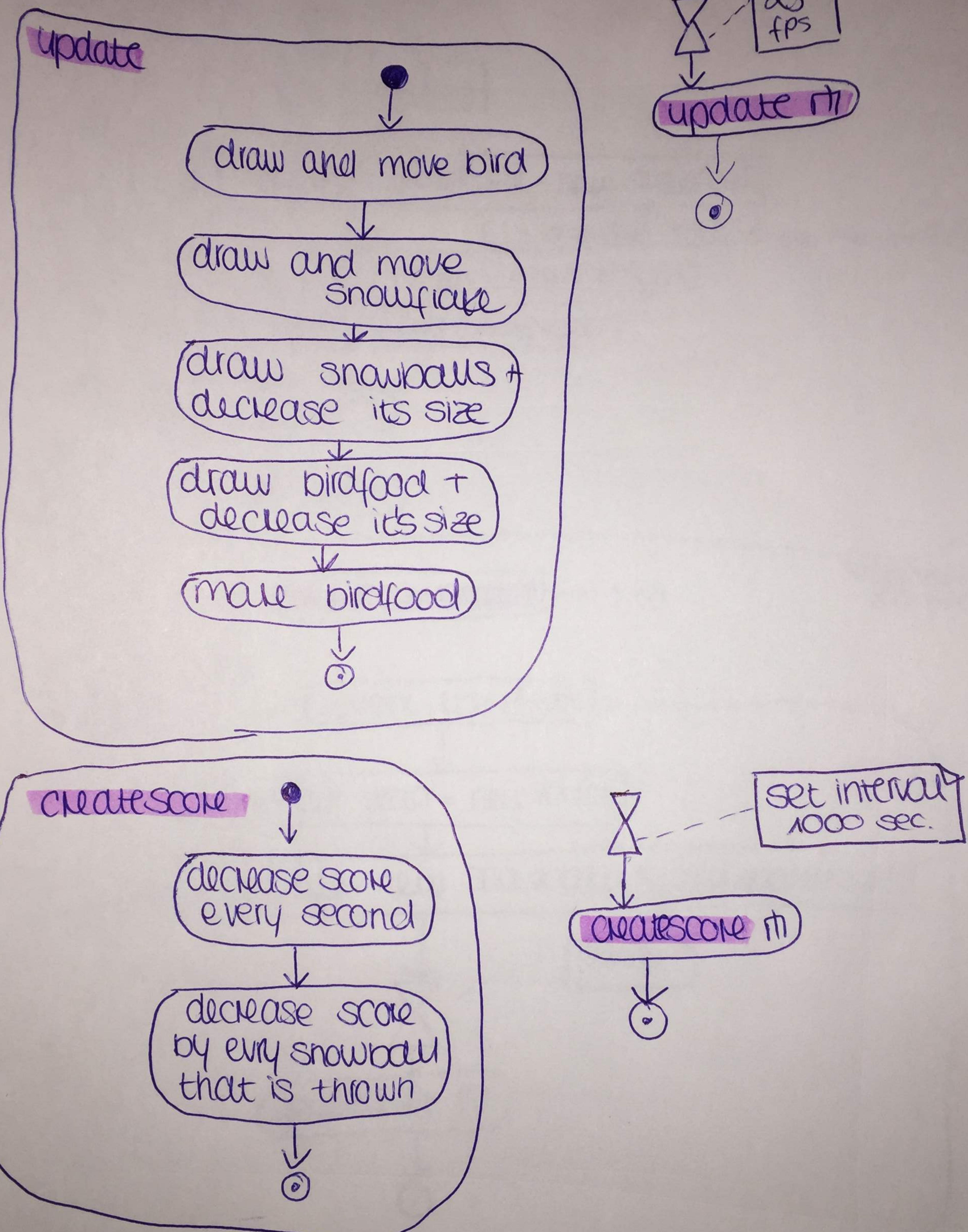
SAVE NAME +
SCORE



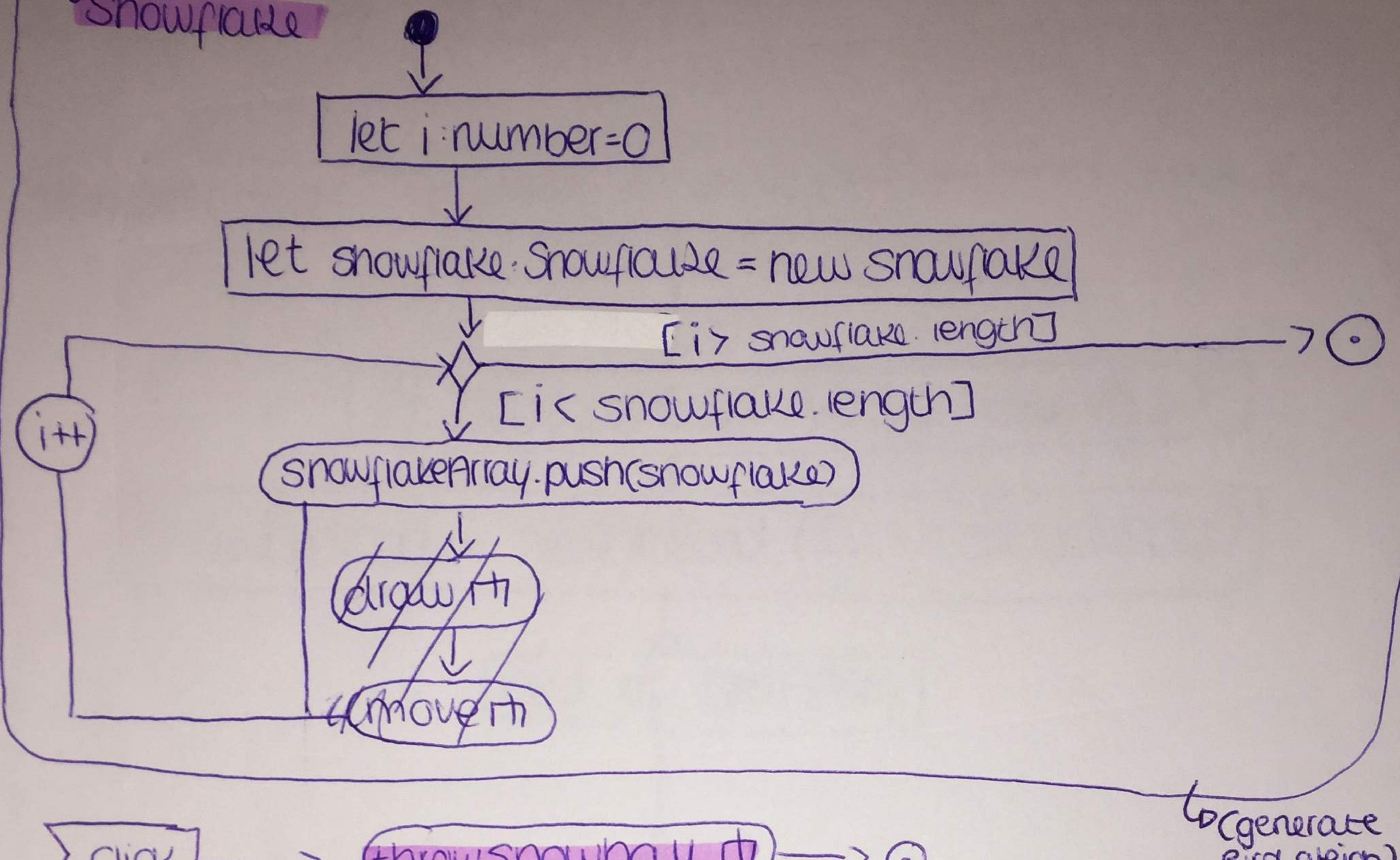
main



main



generate Snowflake

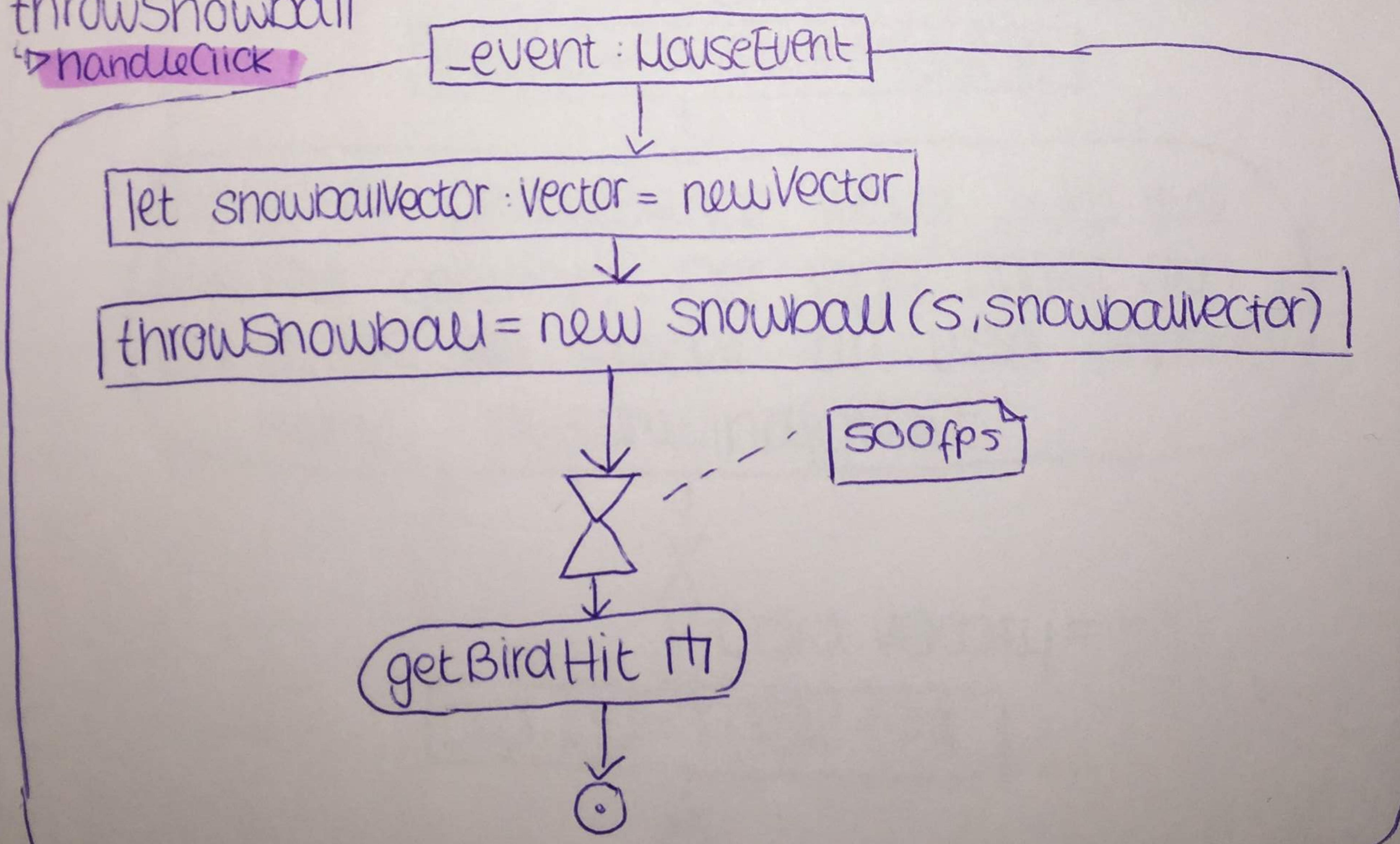


click → throwSnowball

↳ (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

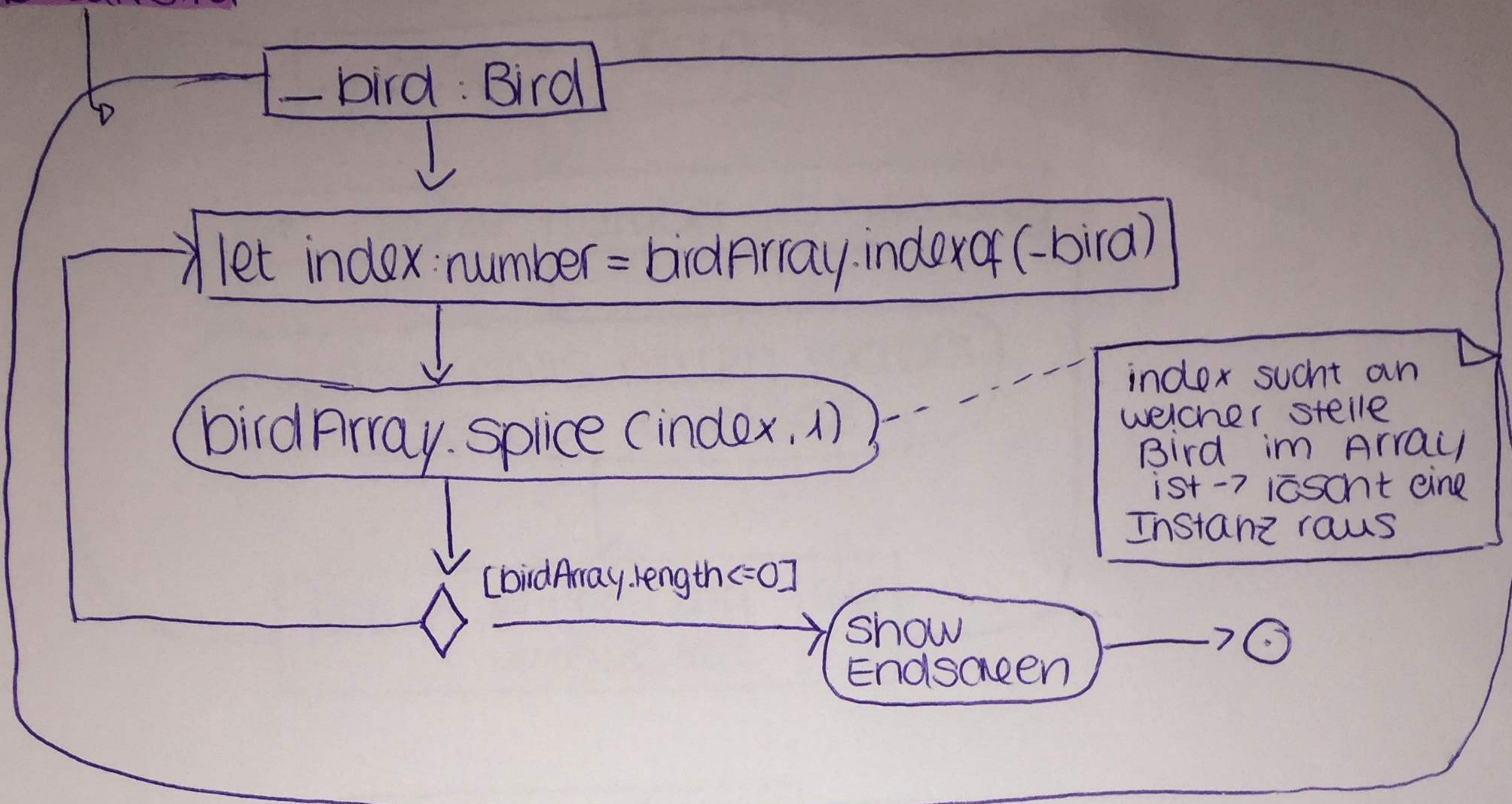
aus dem getdifference vector wird eine Strecke generiert. Der Vogel passt sich an diese ~~Strecke~~ Strecke an und ändert seine Geschwindigkeit

[bird.velocity = 0]

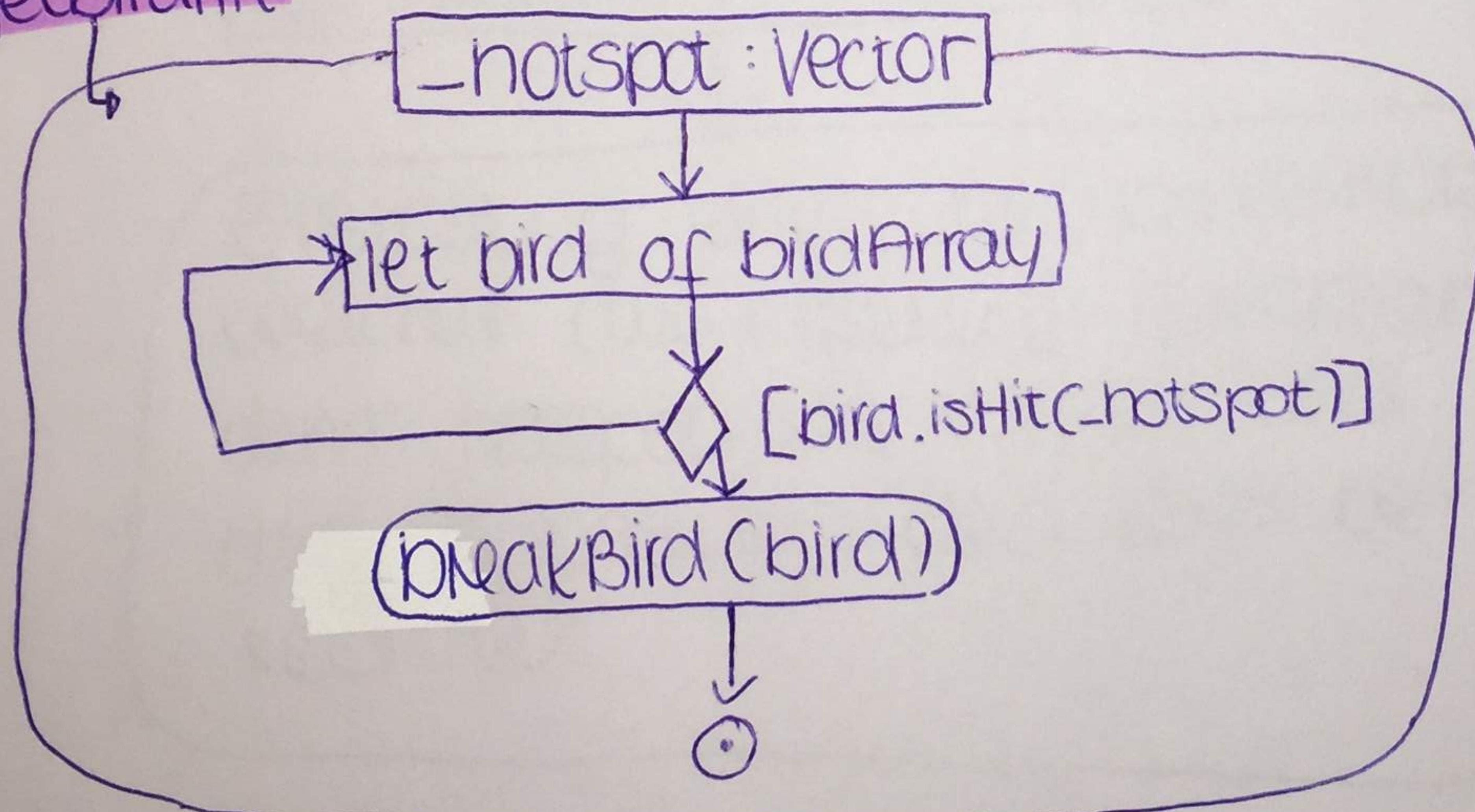
bird.job = TASK.EAT

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

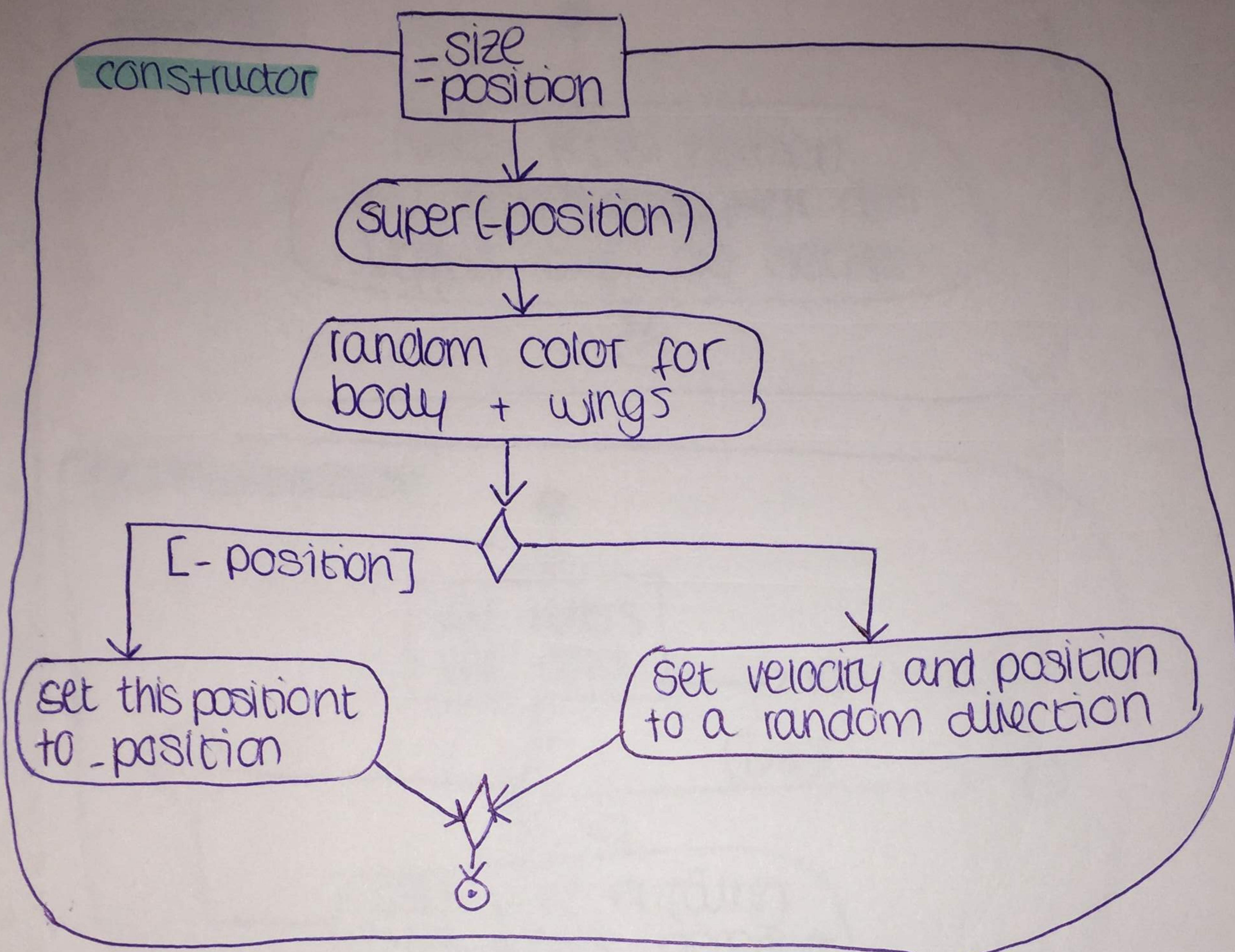
let nearsize: number = 100;

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

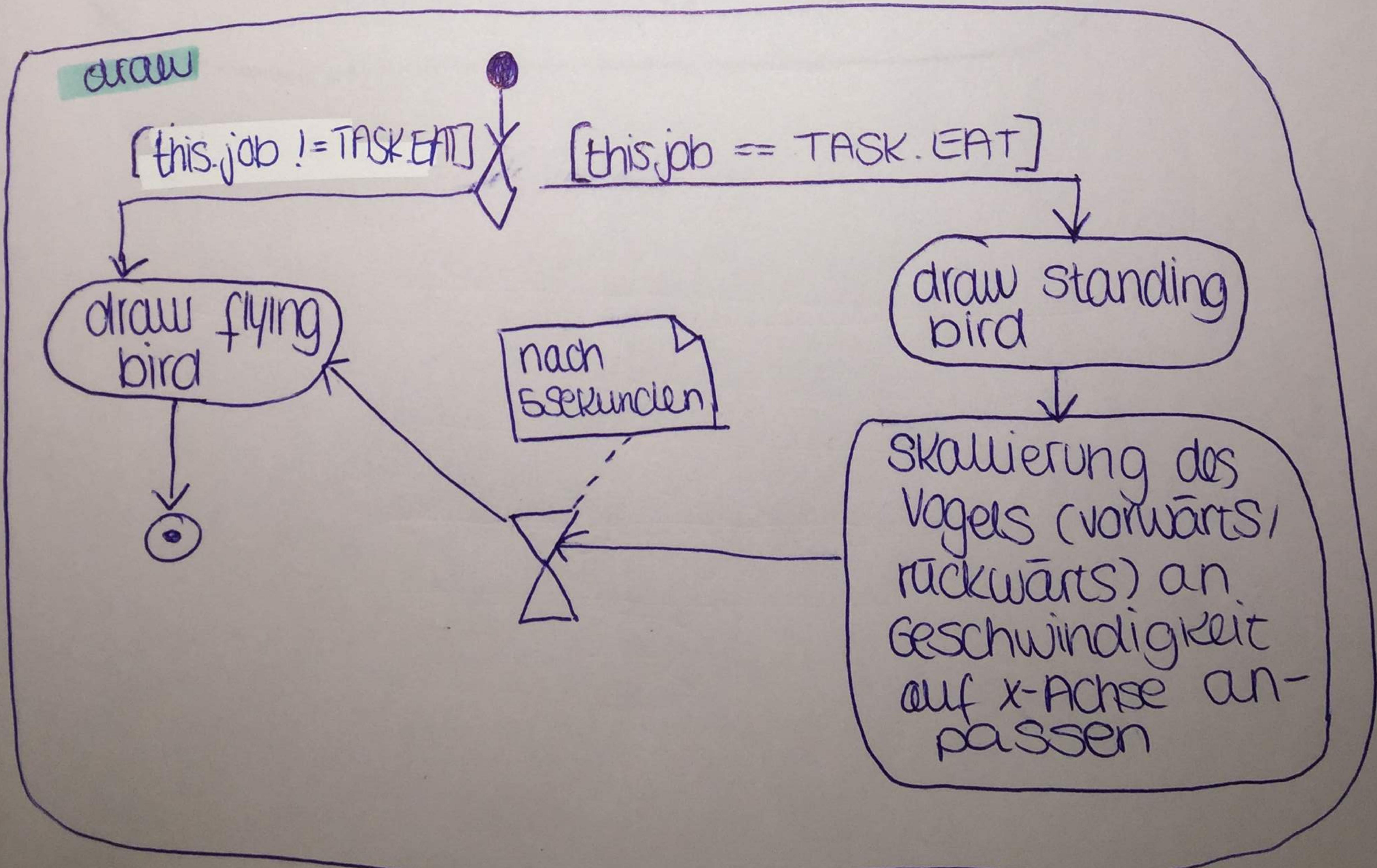
Return Entfernung zum Keks

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++

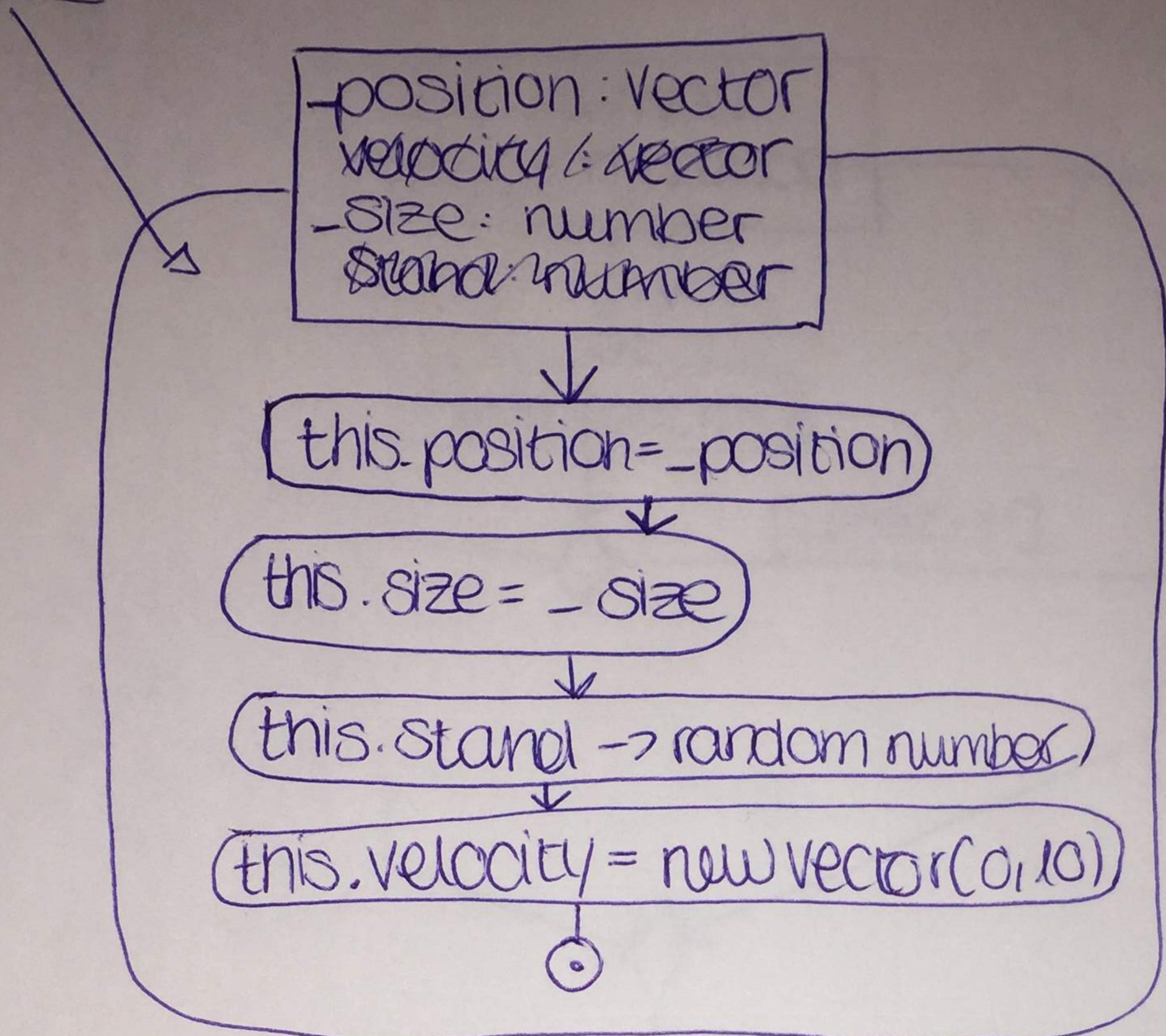
[i>6]

→

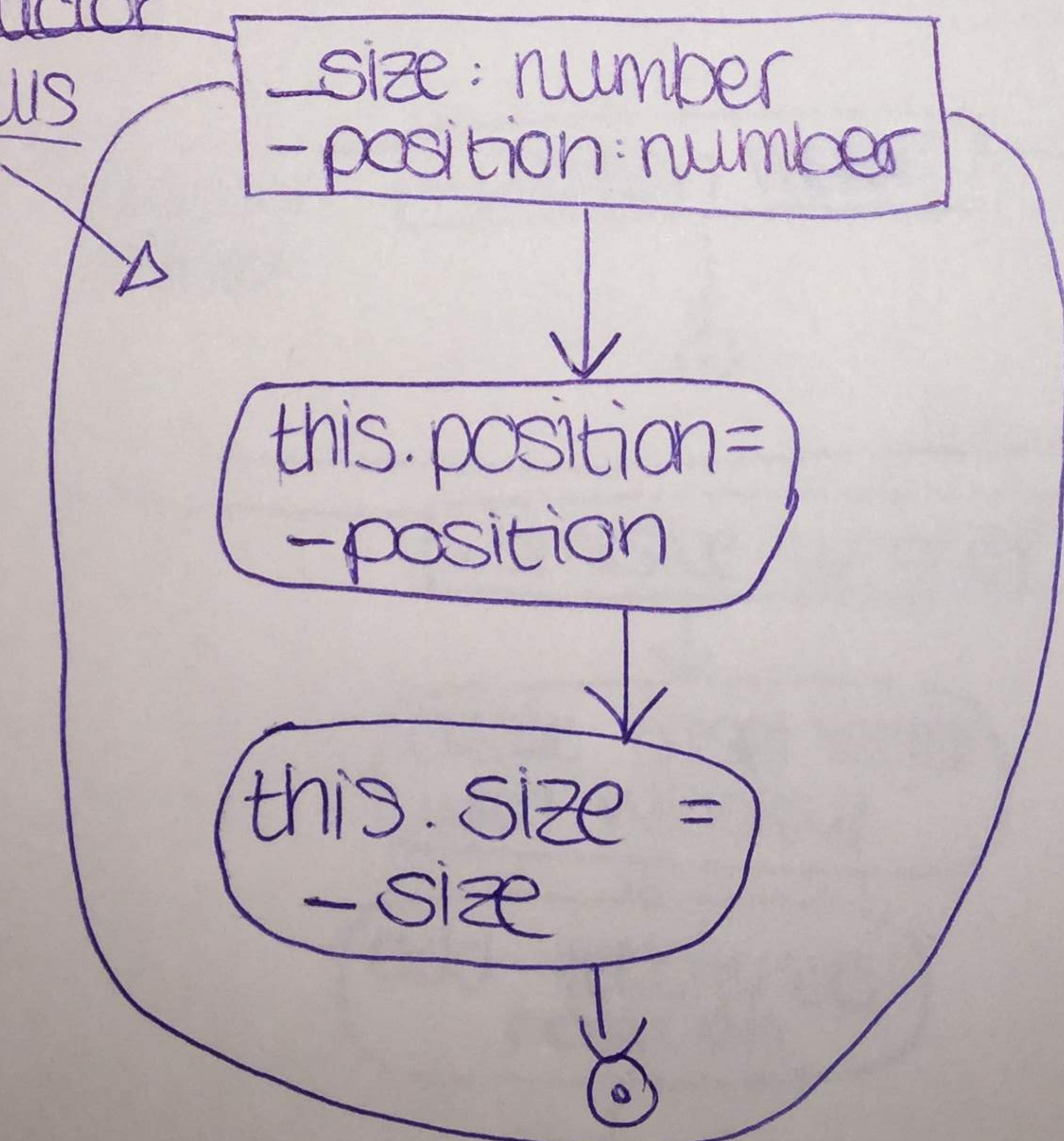
[i<6]

color wird zufällig
aus letters generiert

Constructor Birdfood



constructor snowballs



snowflakes

