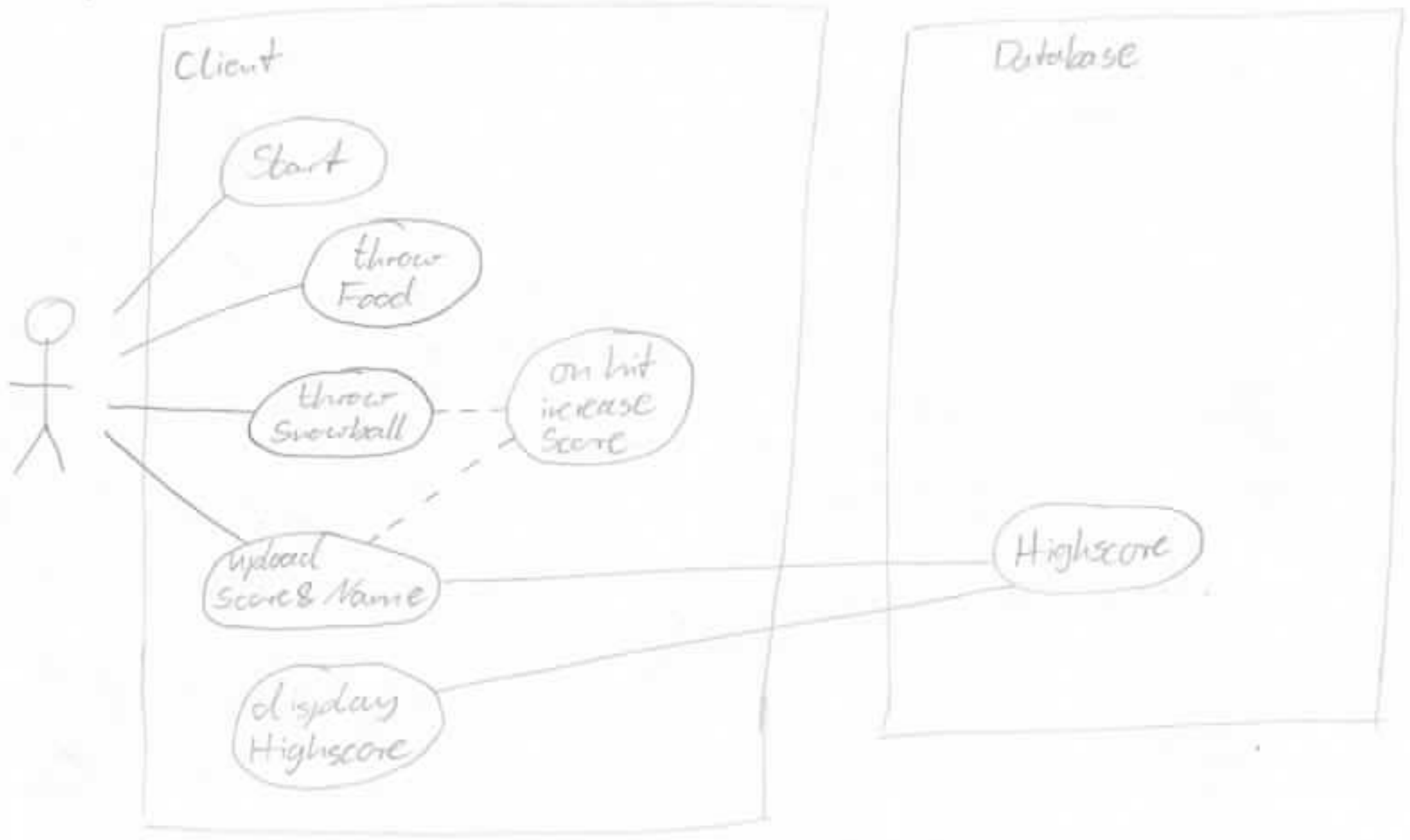
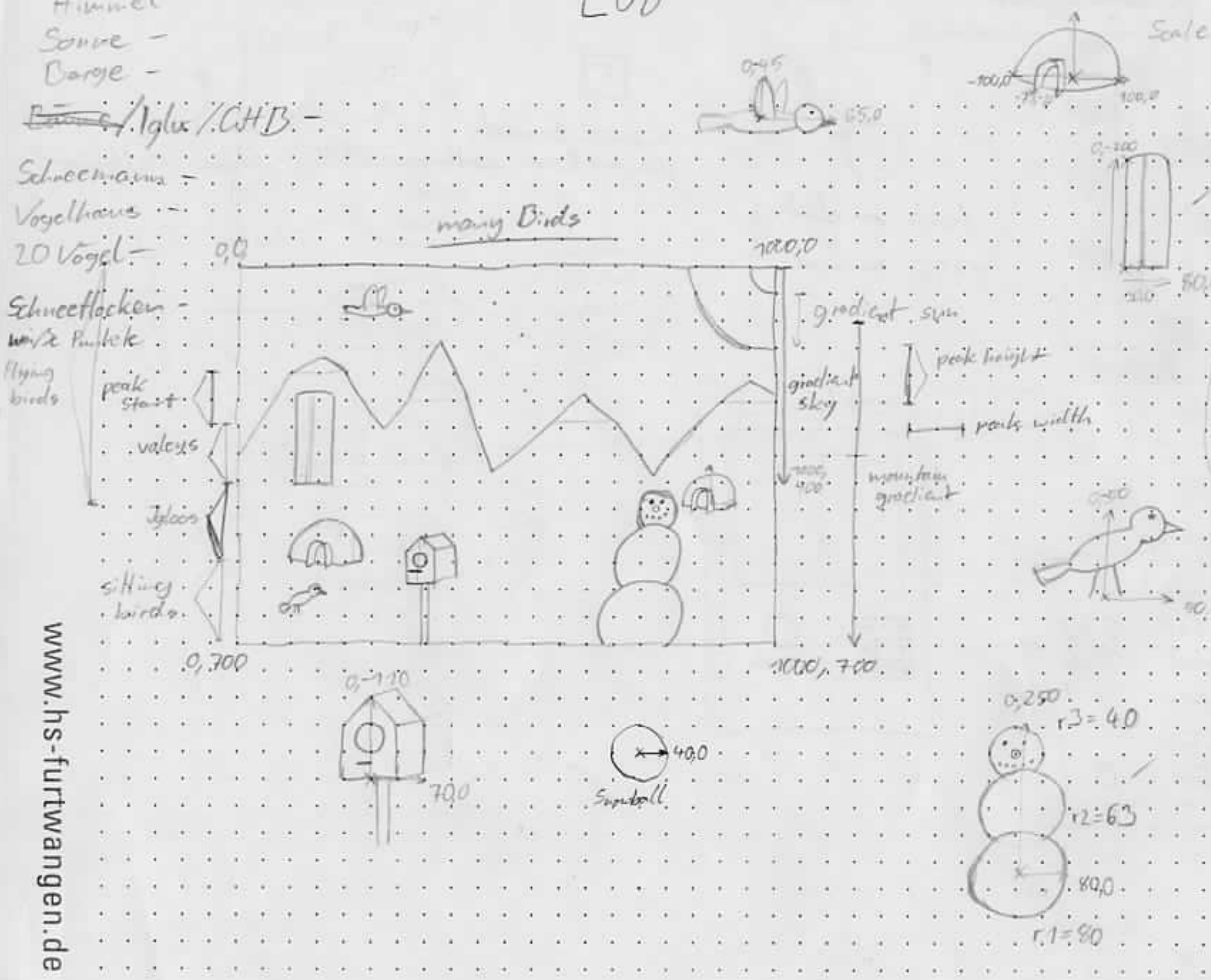
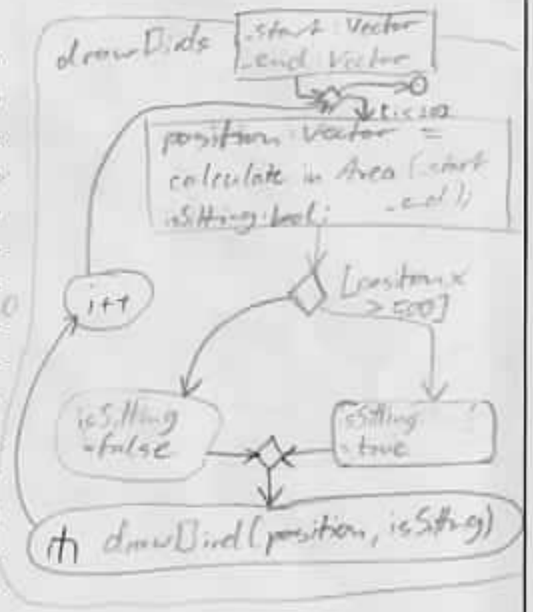

Client
Start
throw Food
throw Snowball
on hit increase Score
Upload Score & Name
display Highscore
Database
Highscore

L08

Himmel -
Sonne -
Berge -
~~[illegible]~~/Iglu/CHB -
Schneemann -
Vogelhaus -
20 Vögel -
Schneeflocken -
weiße Punkte

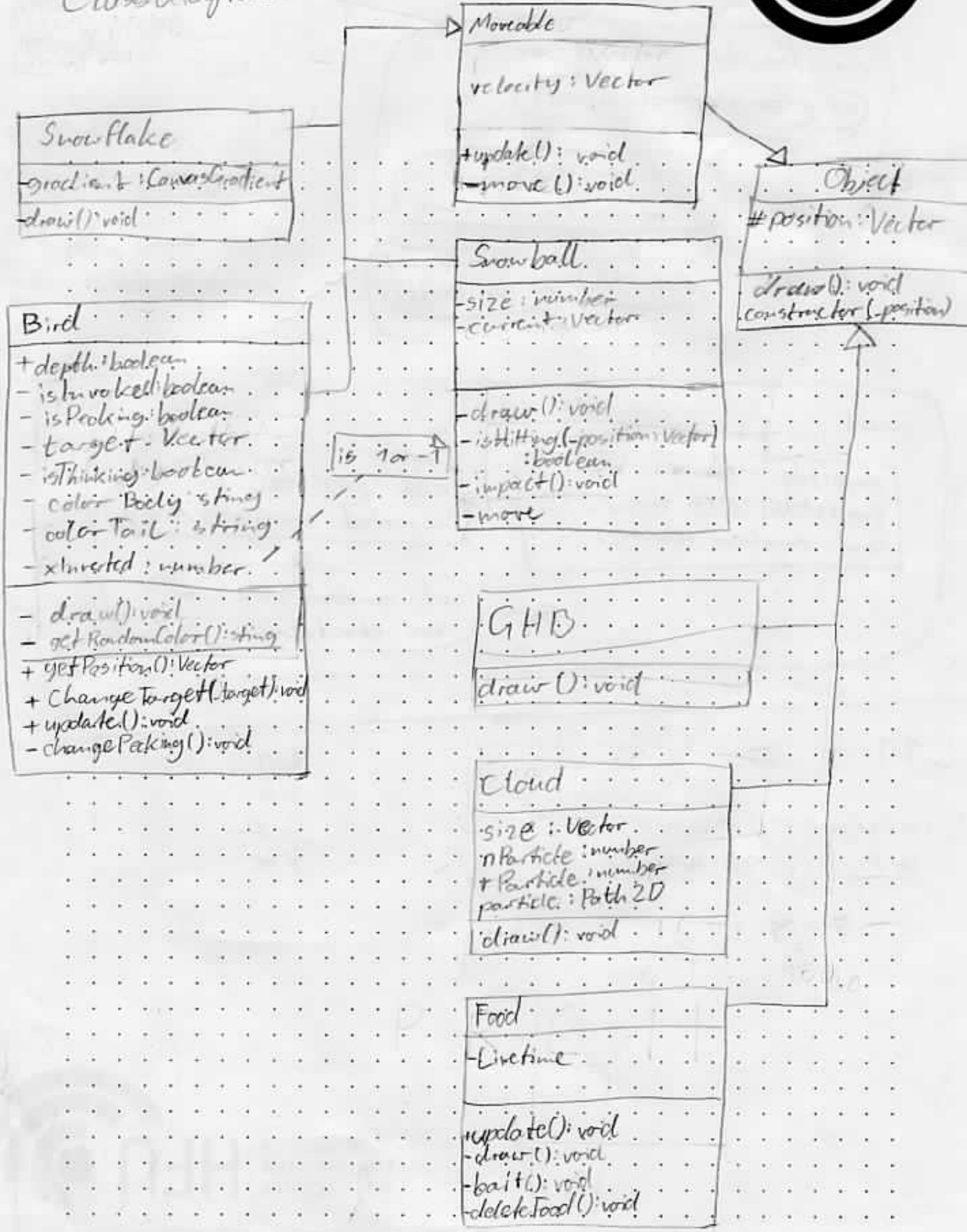

0,45
65,0
-100,0
100,0
0,-200
80,0
many Birds
0,0
1000,0
gradient sun
gradient sky
1000,400
mountain gradient
peak height
peak width
flying birds
peak start
valeys
Igloos
sitting birds
0,700
1000,700
0,-80
90,0
0,-110
70,0
40,0
Snowball
0,250
r3 = 40
r2 = 63
80,0
r1 = 80

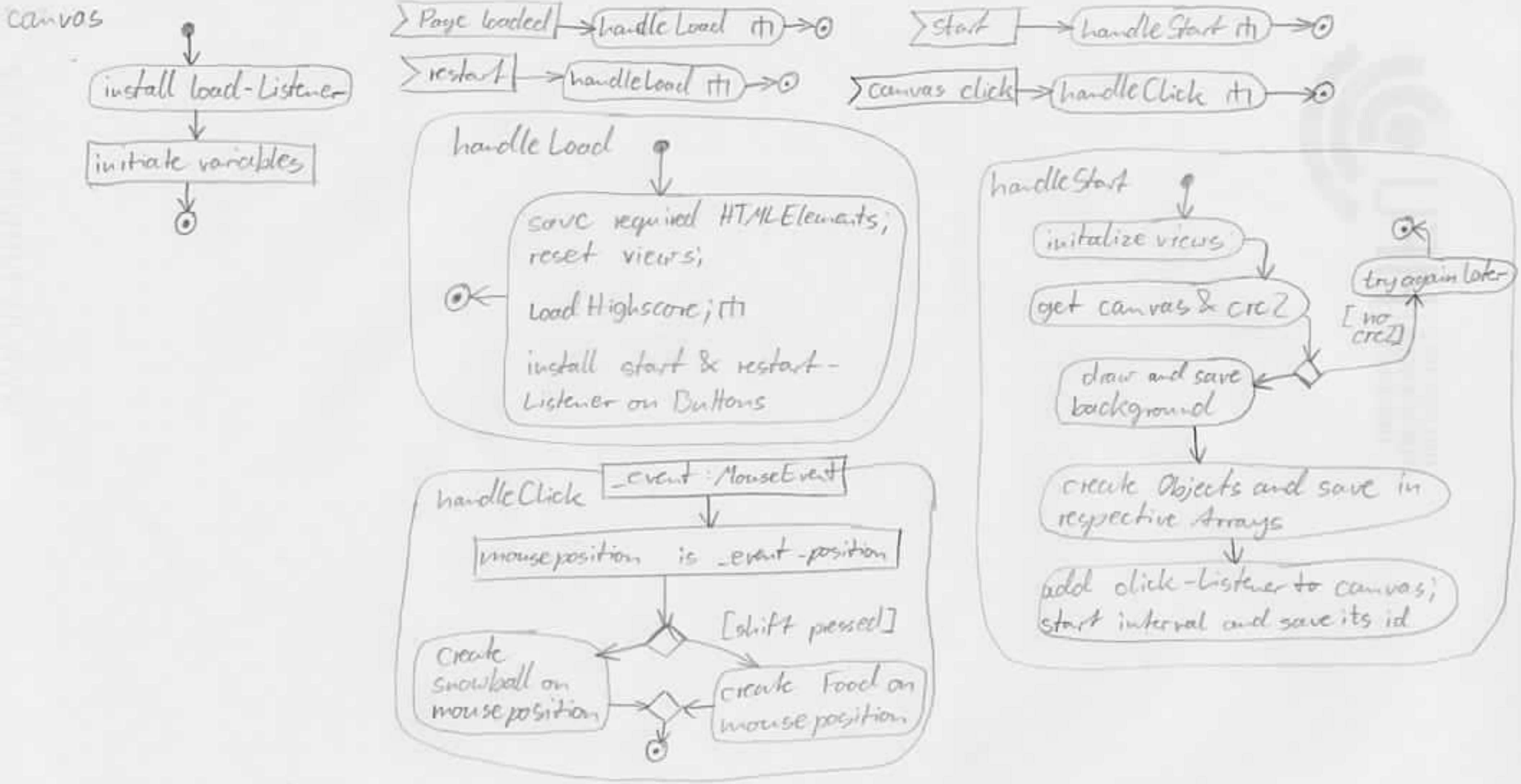
Scale with y (distance)


drawBirds
start: Vector
end: Vector
i<10
position: Vector = calculate in Area (start, end);
isSitting: bool;
[position.x > 500]
isSitting = false
isSitting = true
drawBird (position, isSitting)
i++

Dat is Leuk

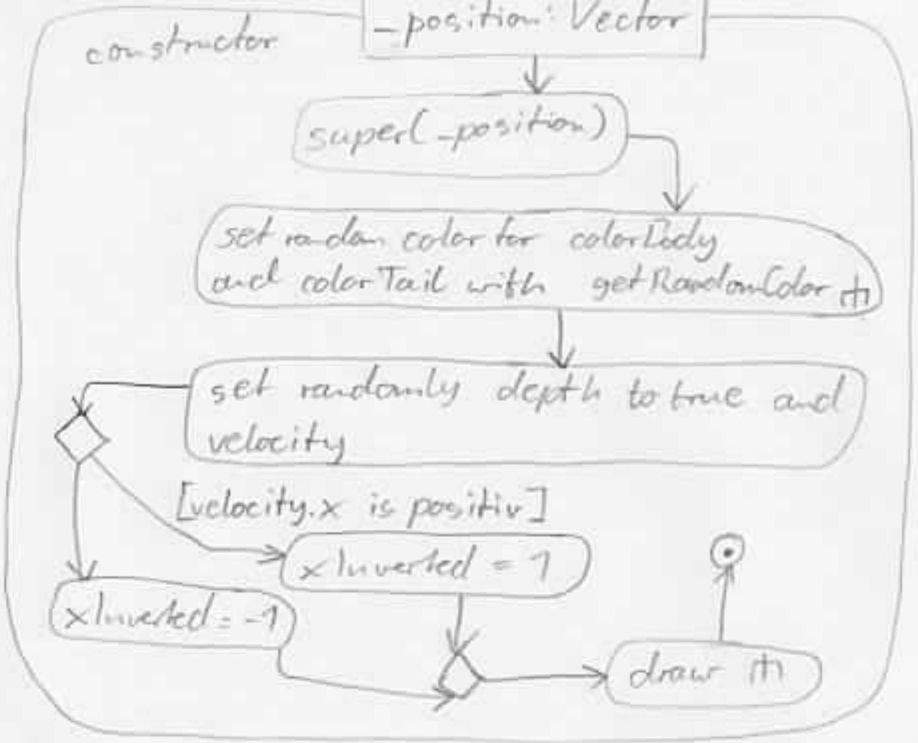
Classdiagram

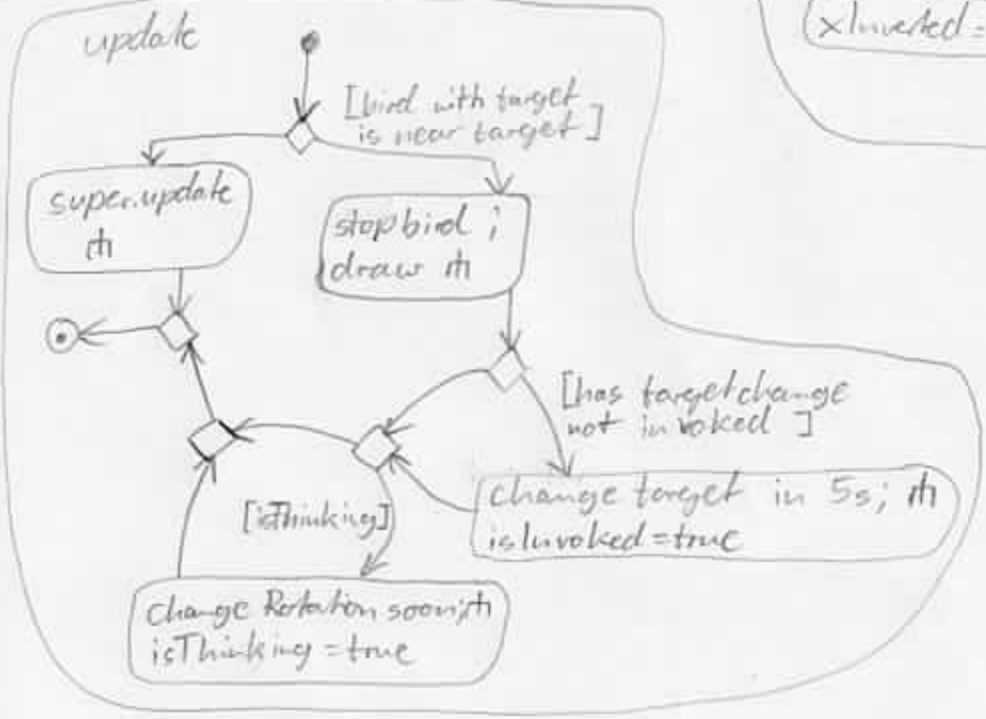

Moveable
velocity : Vector
+update(): void
-move (): void
Snowflake
-gradient : CanvasGradient
-draw(): void
Object
#position: Vector
draw(): void
constructor (_position)
Snowball
-size : number
-current : Vector
-draw(): void
-isHitting(_position: Vector)
: boolean
-impact(): void
-move
is 1 or -1
Bird
+depth: boolean
-isInvoked: boolean
-isPecking: boolean
-target: Vector
-isThinking: boolean
-colorBody: string
-colorTail: string
-xInverted : number
-draw(): void
-getRandomColor(): string
+getPosition(): Vector
+ChangeTarget(_target): void
+update(): void
-changePecking(): void
GHB
draw (): void
Cloud
size : Vector
nParticle : number
rParticle : number
particle : Path2D
draw(): void
Food
-Livetime
update(): void
-draw(): void
-bait(): void
-deleteFood(): void


canvas
install load-Listener
initiate variables
Page loaded
handleLoad
restart
handleLoad
start
handleStart
canvas click
handleClick
handleLoad
save required HTMLElements;
reset views;
Load Highscore;
install start & restart-
Listener on Buttons
handleClick
_event : MouseEvent
mouseposition is _event-position
[shift pressed]
Create snowball on mouseposition
create Food on mouse position
handleStart
initalize views
get canvas & crc2
[no crc2]
try again later
draw and save background
create Objects and save in respective Arrays
add click-Listener to canvas;
start interval and save its id

Bird

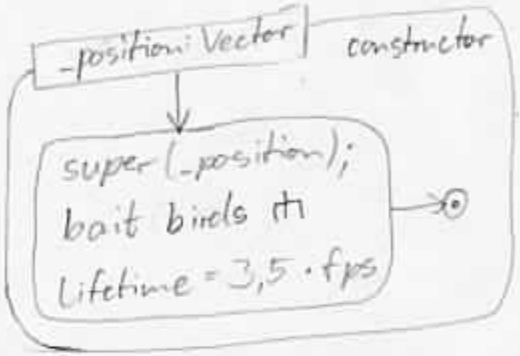
colorBody : string
colorTail : string
xInverted : number
depth : boolean
target : Vector
isTargeting : boolean
isThinking : boolean
isPecking : boolean
isInvoked : boolean = true


constructor
_position : Vector
super(_position)
set randon color for colorBody and colorTail with getRandomColor ⋔
set randomly depth to true and velocity
[velocity.x is positiv]
xInverted = 1
xInverted = -1
draw ⋔

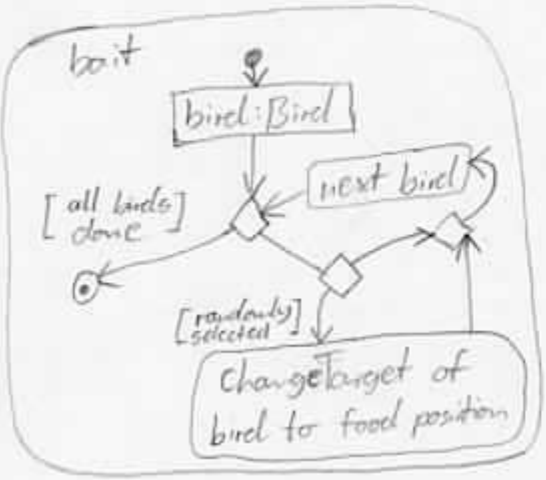

update
[bird with target is near target]
super.update ⋔
stop bird ; draw ⋔
[has targetchange not invoked]
change target in 5s; ⋔ isInvoked = true
[isThinking]
change Rotation soon ⋔ isThinking = true

Food

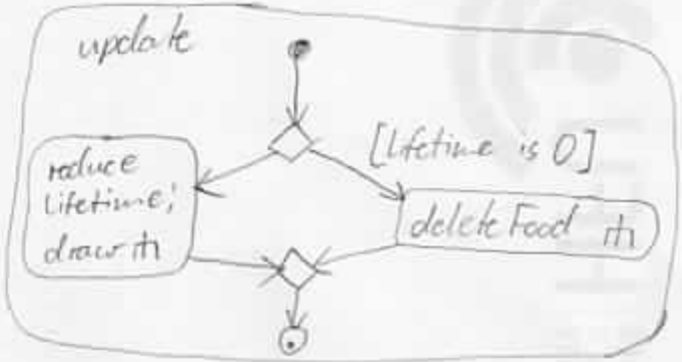
Lifetime: number



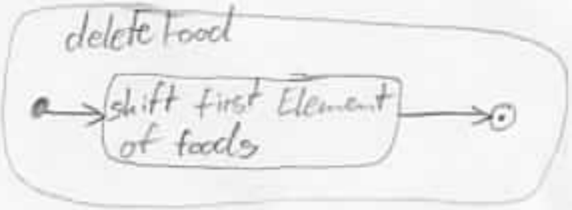
_position: Vector
constructor
super(_position);
bait birds ⋔
Lifetime = 3,5 · fps



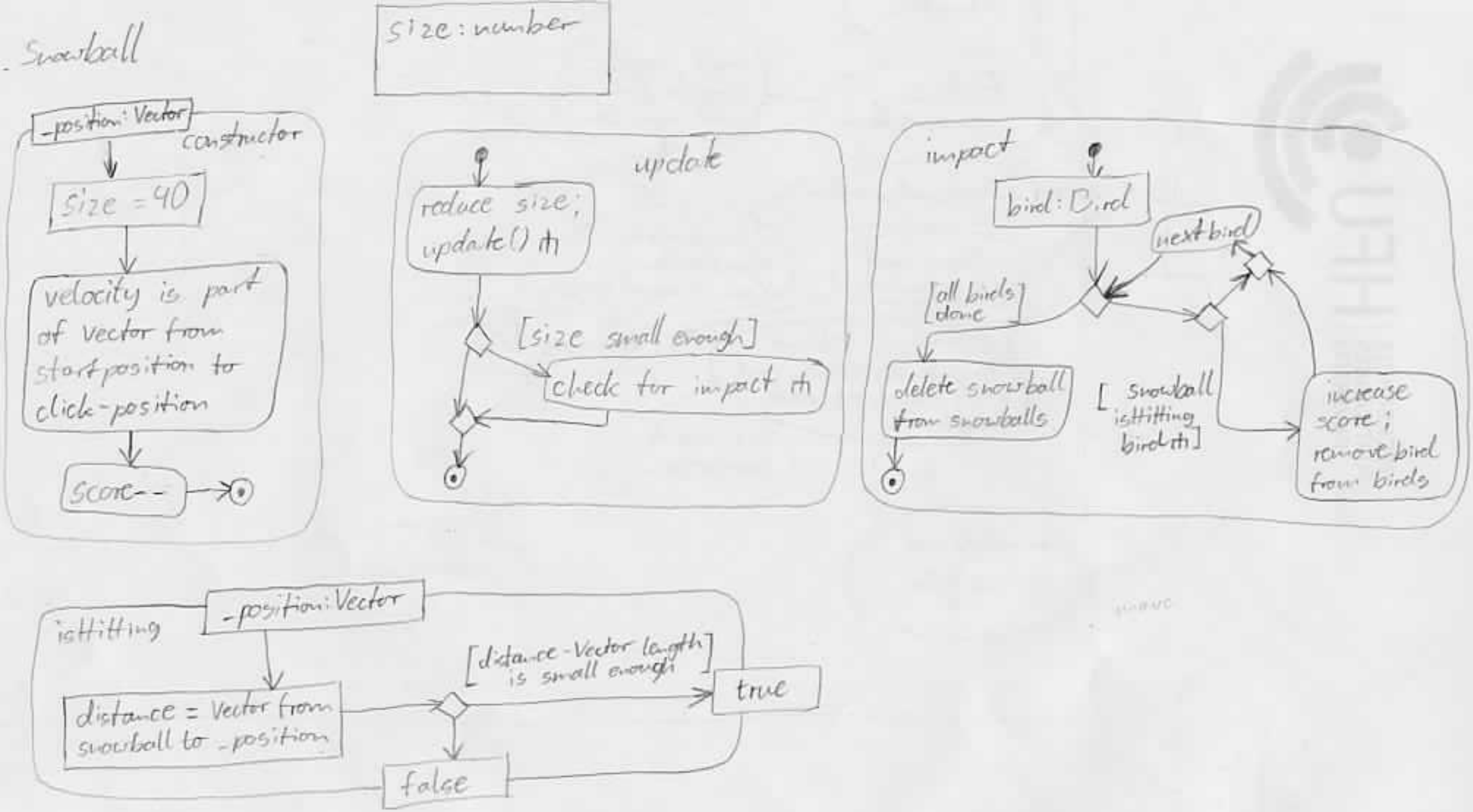
bait
bird: Bird
next bird
[all birds done]
[randomly selected]
ChangeTarget of bird to food position



update
[Lifetime is 0]
reduce Lifetime; draw ⋔
deleteFood ⋔



deleteFood
shift first Element of foods


Snowball
size: number
constructor
_position: Vector
size = 40
velocity is part of vector from startposition to click-position
score--
update
reduce size; update() ⋔
[size small enough]
check for impact ⋔
impact
bird: Bird
next bird
[all birds done]
delete snowball from snowballs
[snowball isHitting bird ⋔]
increase score; remove bird from birds
isHitting
_position: Vector
distance = Vector from snowball to _position
[distance-Vector length is small enough]
true
false