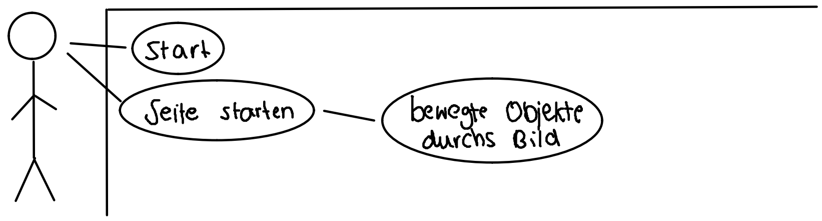
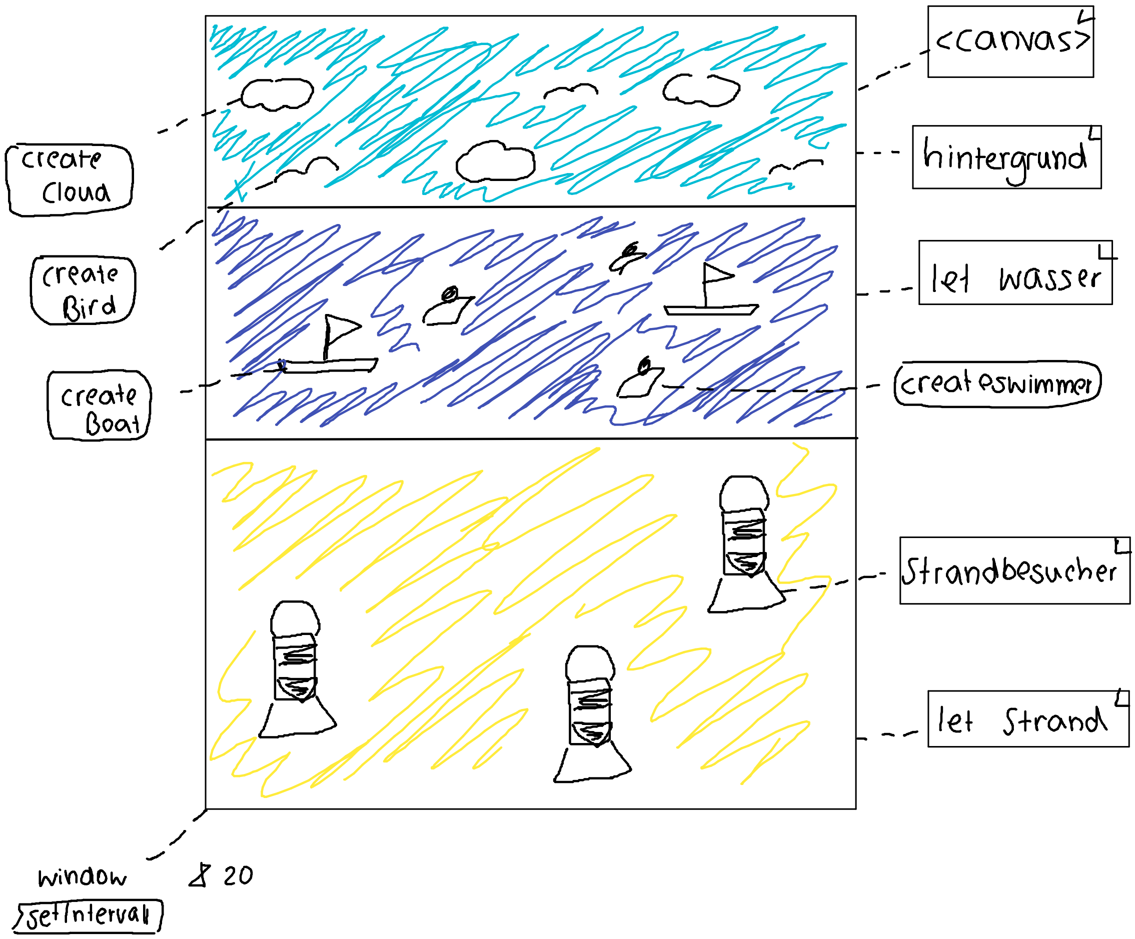
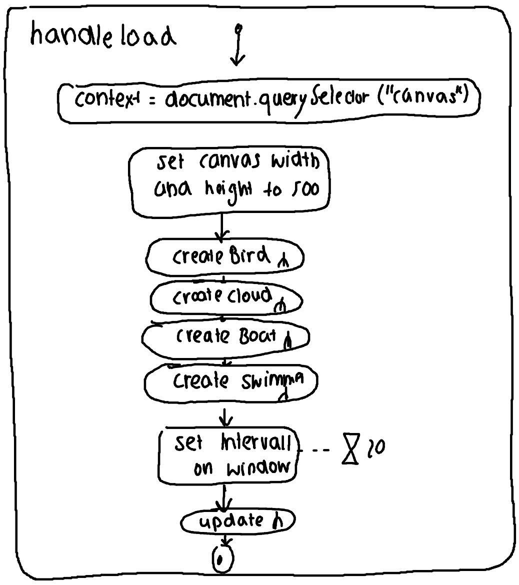
# In Zusammenarbeit mit Celine:

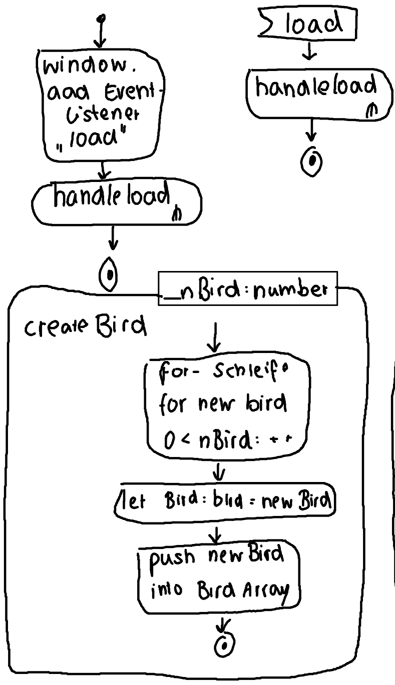


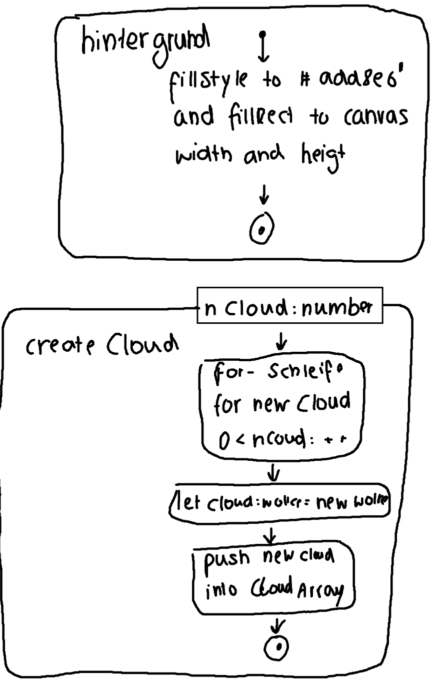


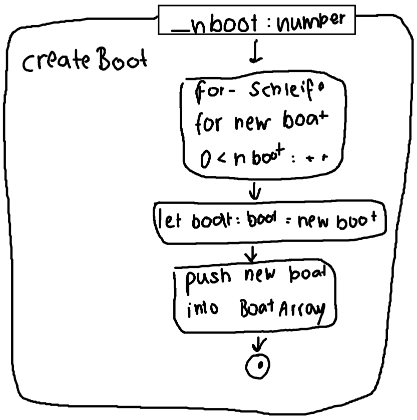
## Activity Diagramm

```
export let context: CanvasRenderingContext2
       let biras: bira[]=[]
       let clouds: wolke []=[]
       let boats: boot[]=[]
       let swimmer: schwimmer[]=[]
```

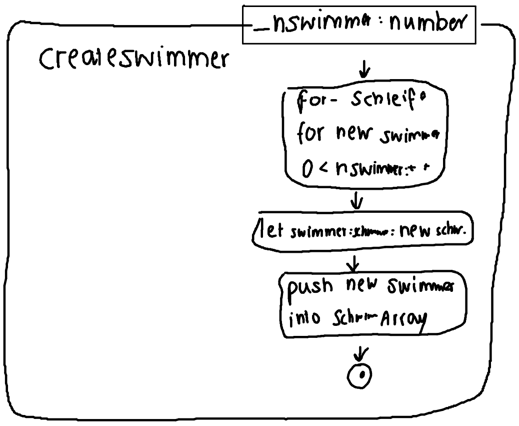



window. add Event-Listener „load"
handle load
load
handleload
nBird: number
create Bird
for- Schleife for new bird 0 < nBird: ++
let Bird: bird = new Bird
push new Bird into Bird Array

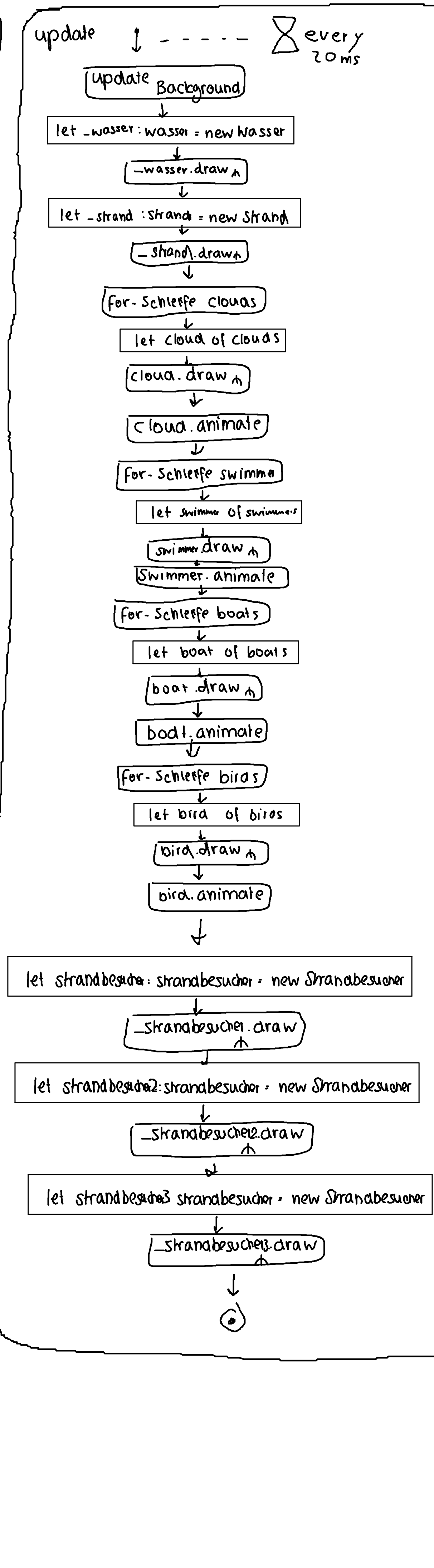

hintergrund
fillstyle to #add8e6' and fillRect to canvas width and heigt
n Cloud: number
create Cloud
for- Schleife for new Cloud 0 < nCloud: ++
let cloud: wolke = new wolke
push new cloud into Cloud Array



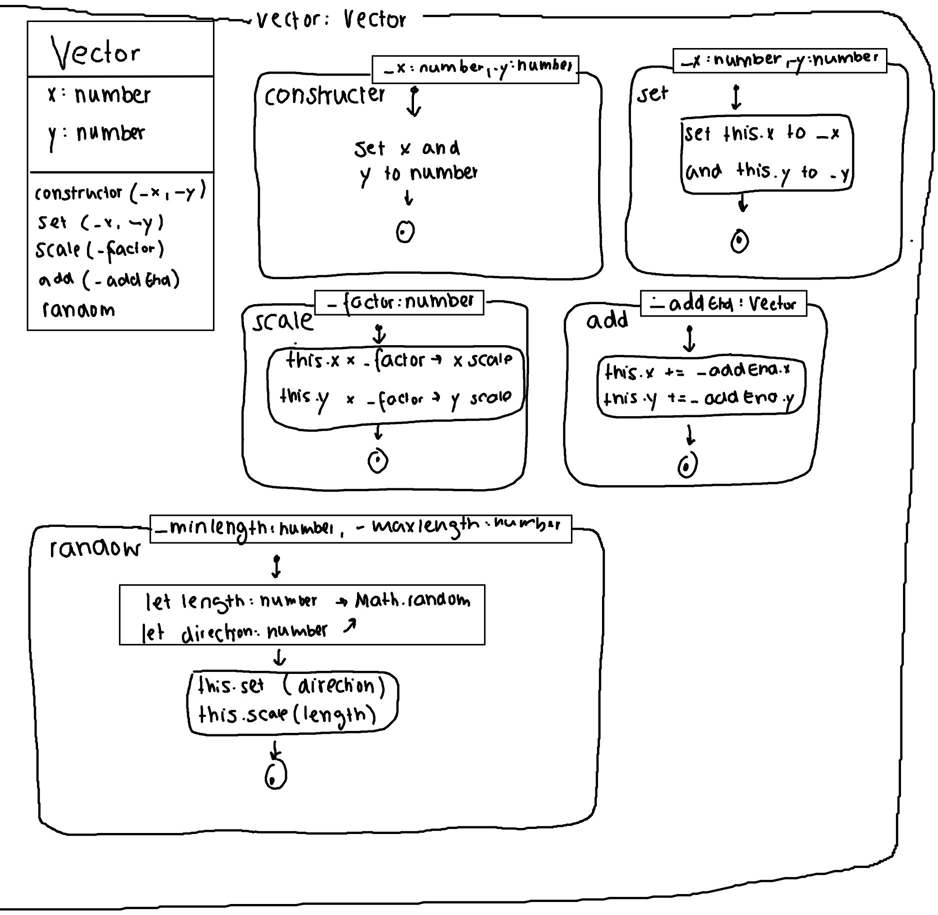
_nboot : number
create Boot
for- Schleife
for new boat
0 < nboot : ++
let boat : boot = new boot
push new boat
into Boat Array



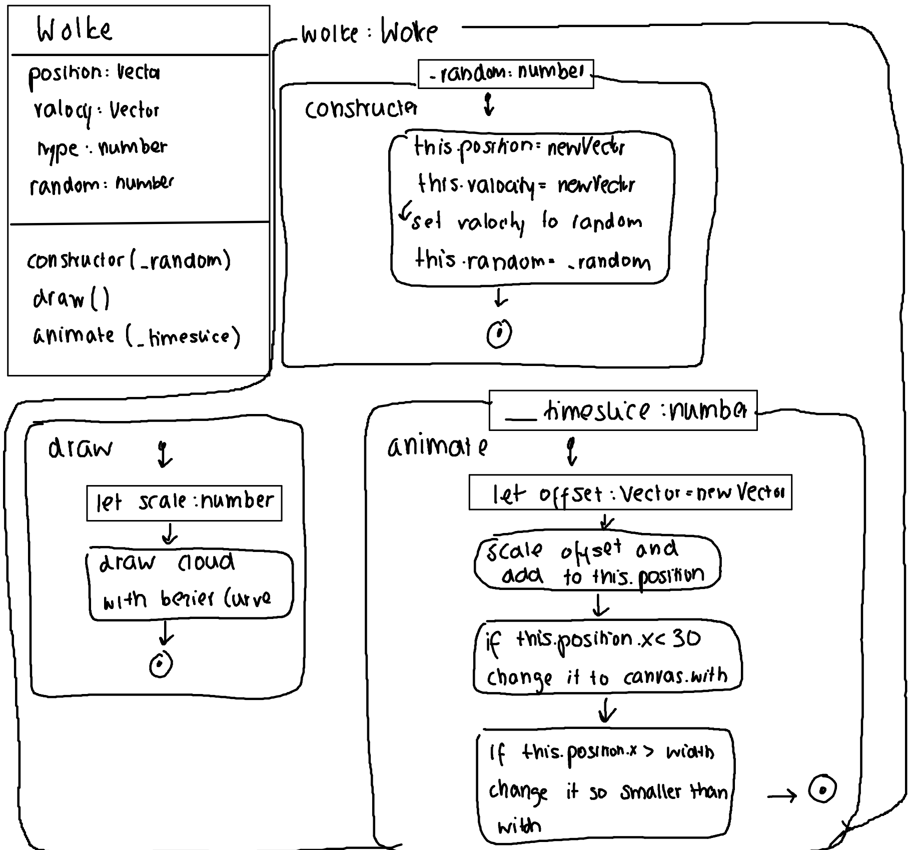
_nswimmer : number
createswimmer
for- Schleife
for new swimmer
0 < nswimmer : ++
let swimmer : schwimmer : new schw.
push new Swimmer
into Schwimmer Array



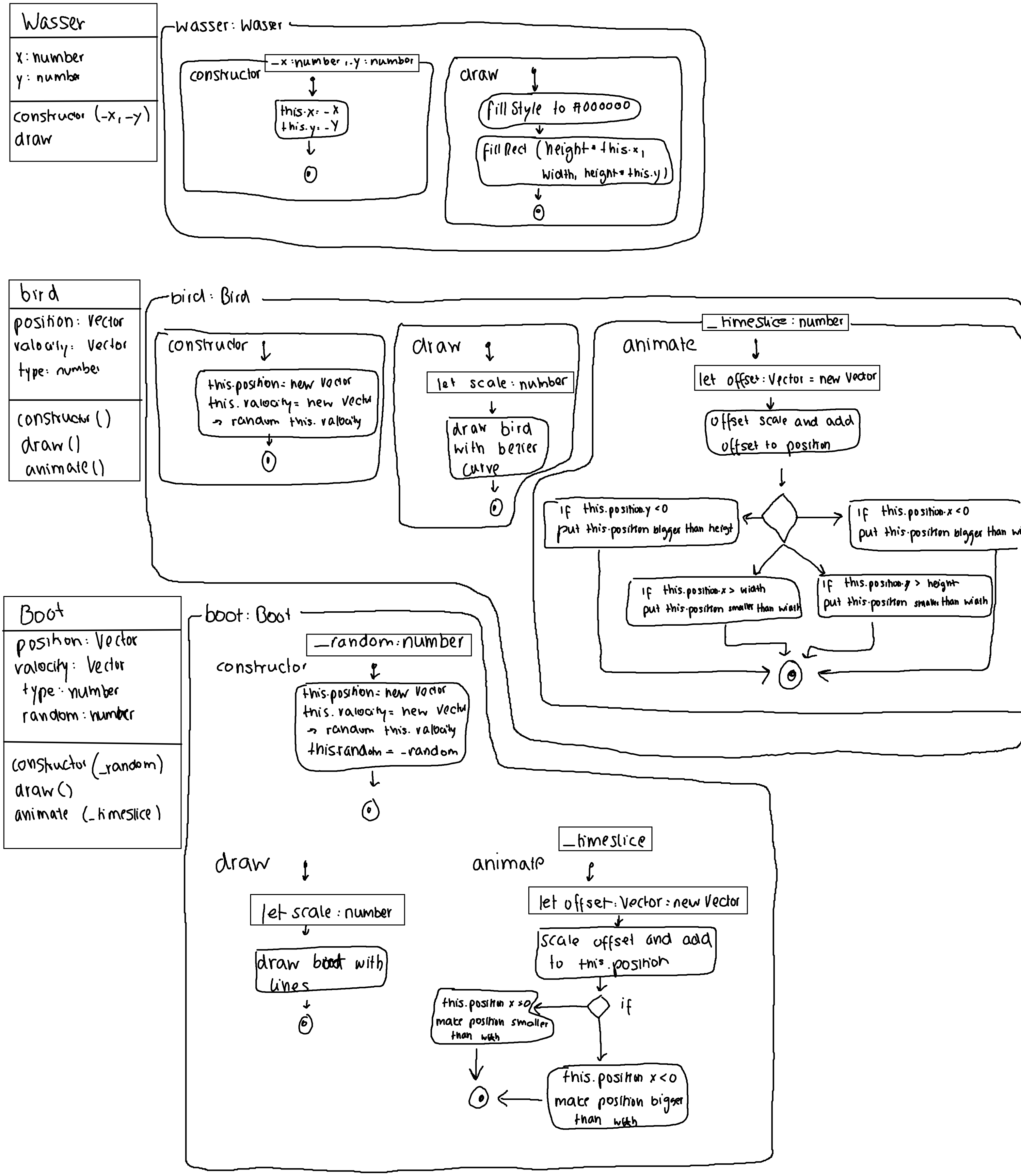
update
every 20 ms
update Background
let _wasser : wasser = new Wasser
_wasser.draw
let _strand : Strand = new Strand
_strand.draw
for- Schleife clouds
let cloud of clouds
cloud.draw
cloud.animate
for- Schleife swimmer
let swimmer of swimmers
swimmer.draw
Swimmer.animate
for- Schleife boats
let boat of boats
boat.draw
boat.animate
for- Schleife birds
let bird of birds
bird.draw
bird.animate
let strandbesucher : Strandbesucher = new Strandbesucher
_strandbesucher.draw
let strandbesucher2 : Strandbesucher = new Strandbesucher
_strandbesucher2.draw
let strandbesucher3 Strandbesucher = new Strandbesucher
_strandbesucher3.draw



Vector
x : number
y : number
constructor (_x, _y)
set (_x, _y)
scale (_factor)
add (_addEnd)
random
vector : Vector
constructer
_x : number, _y : number
set x and
y to number
set
_x : number, _y : number
set this.x to _x
and this.y to _y
scale
_factor : number
this.x × _factor → x scale
this.y × _factor → y scale
add
_addEnd : Vector
this.x += _addEnd.x
this.y += _addEnd.y
random
_minlength : number, _maxlength : number
let length : number → Math.random
let direction : number ↗
this.set (direction)
this.scale (length)



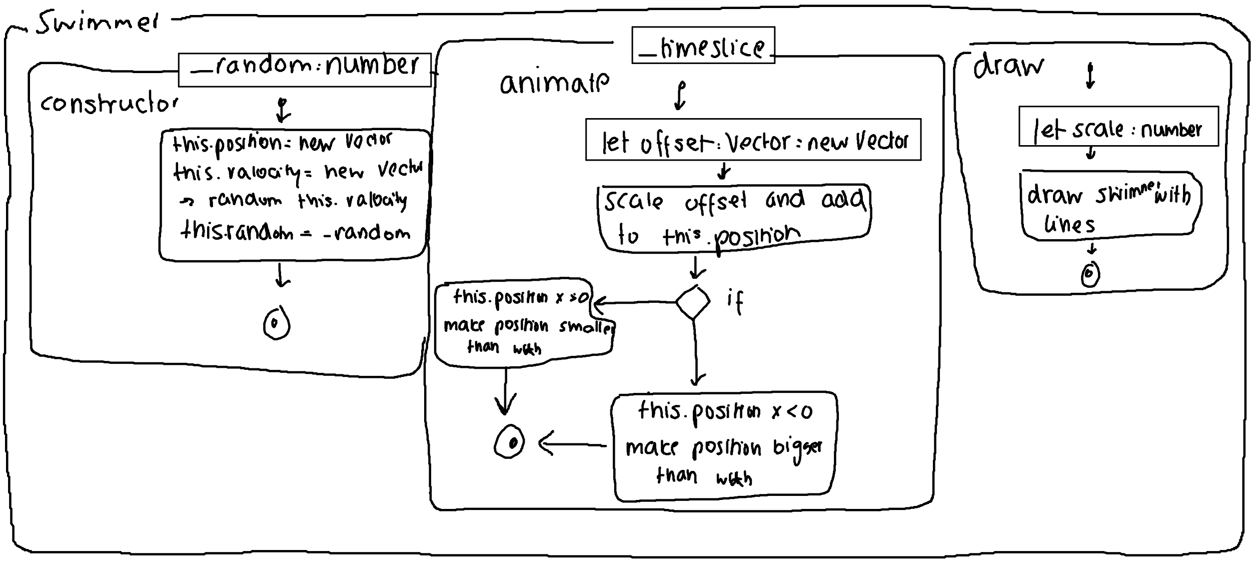
Wolke
position : Vector
valocity : Vector
type : number
random : number
constructor (_random)
draw ()
animate (_timeslice)
wolke : Wolke
_random : number
constructor
this.position = newVector
this.valocity = newVector
set valocity to random
this.random = _random
draw
let scale : number
draw cloud
with bezier Curve
animate
_timeslice : number
let offset : Vector = new Vector
scale offset and
add to this.position
if this.position.x < 30
change it to canvas.with
if this.position.x > width
change it so smaller than
with


Wasser
x: number
y: number
constructor (-x, -y)
draw
wasser: Wasser
constructor
_x: number, _y: number
this.x: _x
this.y: _y
draw
fill Style to #000000
fillRect (height * this.x, width, height * this.y)
bird
position: Vector
valocity: Vector
type: number
constructor ()
draw ()
animate ()
bird: Bird
constructor
this.position = new Vector
this.valocity = new Vector
→ random this.valocity
draw
let scale: number
draw bird with bezier curve
animate
_timeslice: number
let offset: Vector = new Vector
offset scale and add offset to position
If this.position.y < 0 put this.position bigger than heigt
If this.position.x < 0 put this.position bigger than wi
If this.position.x > width put this.position smaller than width
If this.position.y > height put this.position smaller than width
Boot
position: Vector
valocity: Vector
type: number
random: number
constructor (_random)
draw ()
animate (_timeslice)
boot: Boot
_random: number
constructor
this.position = new Vector
this.valocity = new Vector
→ random this.valocity
this.random = _random
draw
let scale: number
draw boot with lines
animate
_timeslice
let offset: Vector = new Vector
scale offset and add to this.position
if
this.position x > 0 make position smaller than with
this.position x < 0 make position bigger than with

Swimmer

position: Vector
valocity: Vector
type: number
random: number

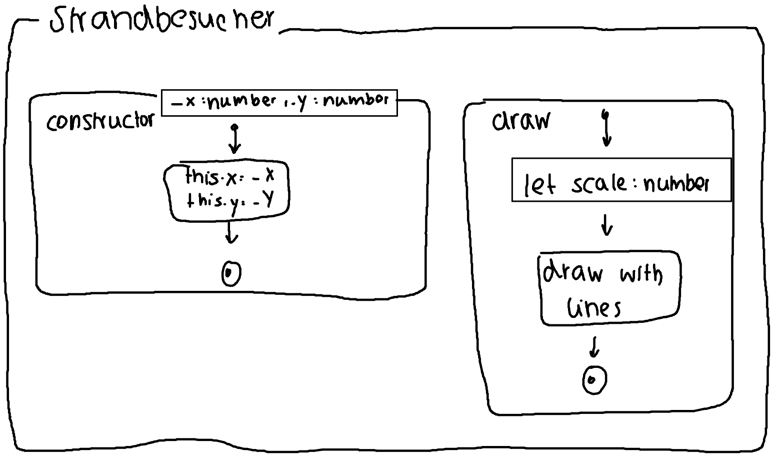
constructor (_random)
draw()
animate (_timeslice)


Swimmer
_random: number
constructor
this.position = new Vector
this. valocity = new Vector
→ randaum this. valocity
this.random = _random
_timeslice
animate
let offset: Vector: new Vector
scale offset and add to this.position
if
this.position x > 0
make position smaller than width
this.position x < 0
make position bigger than width
draw
let scale : number
draw swimmer with lines

Strandbesucher

x: number
y: number

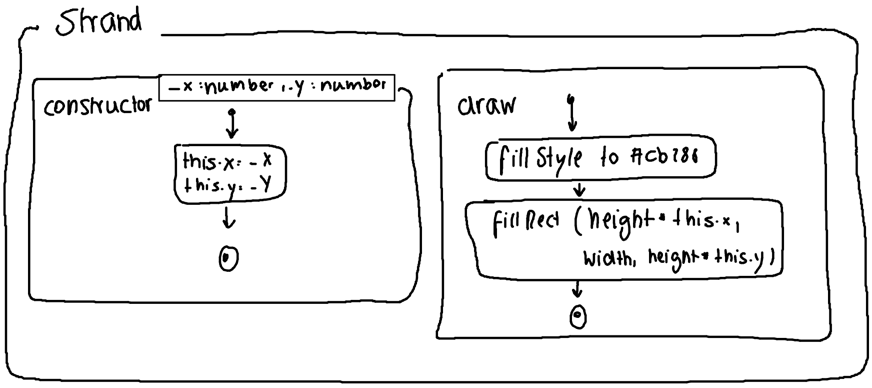
constructor (_x, _y)
draw



Strand

x: number
y: number

constructor (_x, _y)
draw


Strand
constructor
_x: number, _y: number
this.x = _x
this.y = _y
draw
fill Style to #cb786
fill Rect (height * this.x, width, height * this.y)