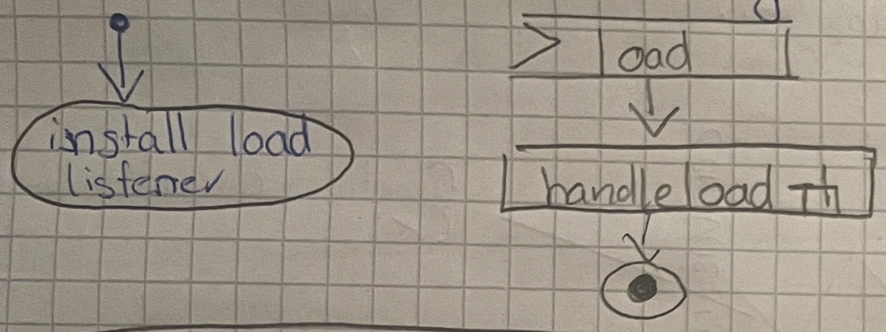
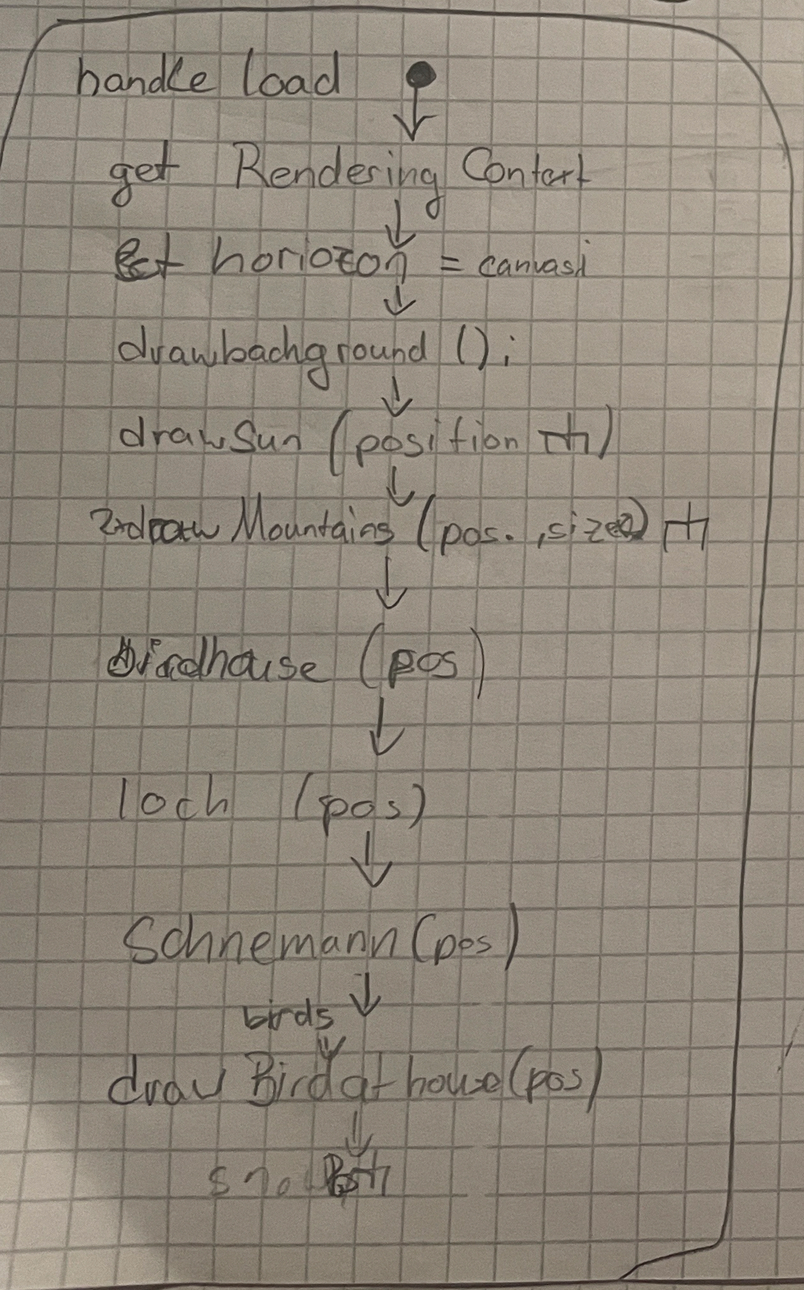

function drawSnow
canvas.innerwidth
function
function drawM
(x:
canvas.innerheight
let horizon =
ctx.canvasheight·0.6
max
min
function Schneemann

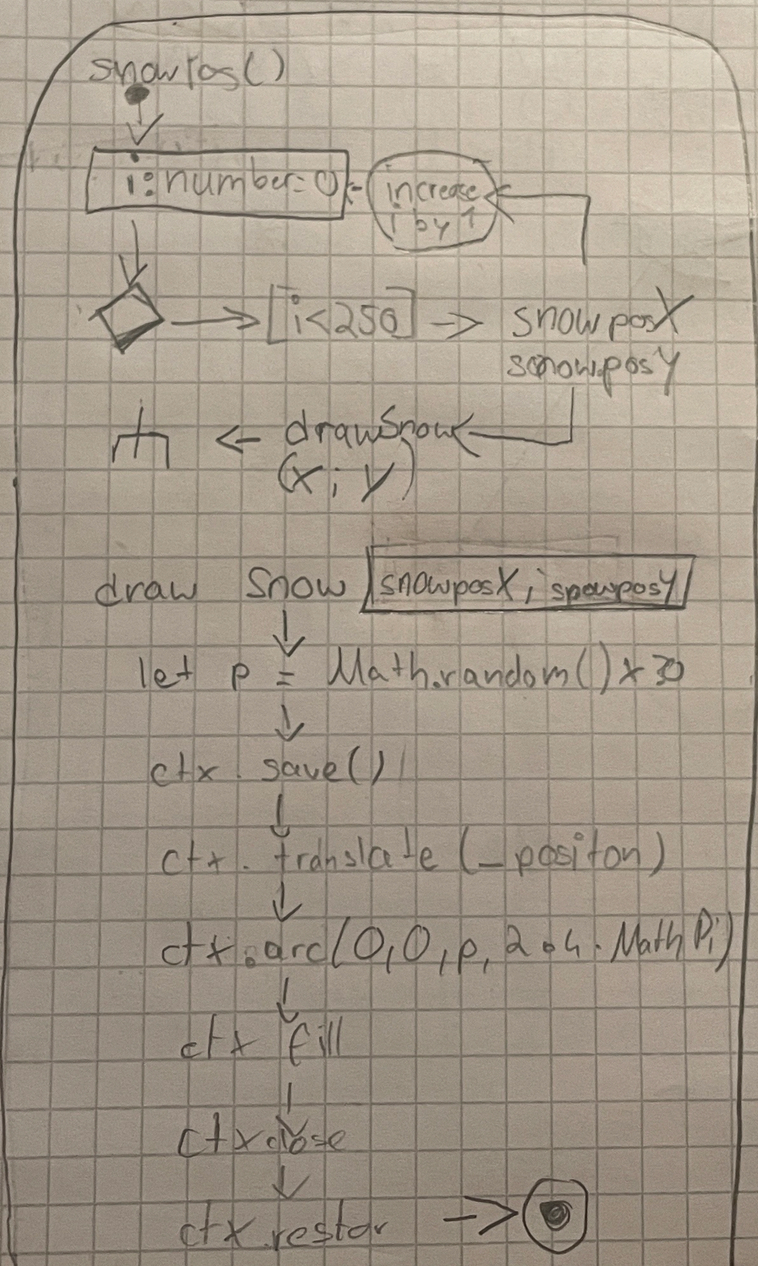
A08.2 - Aktivitätsdiagramm

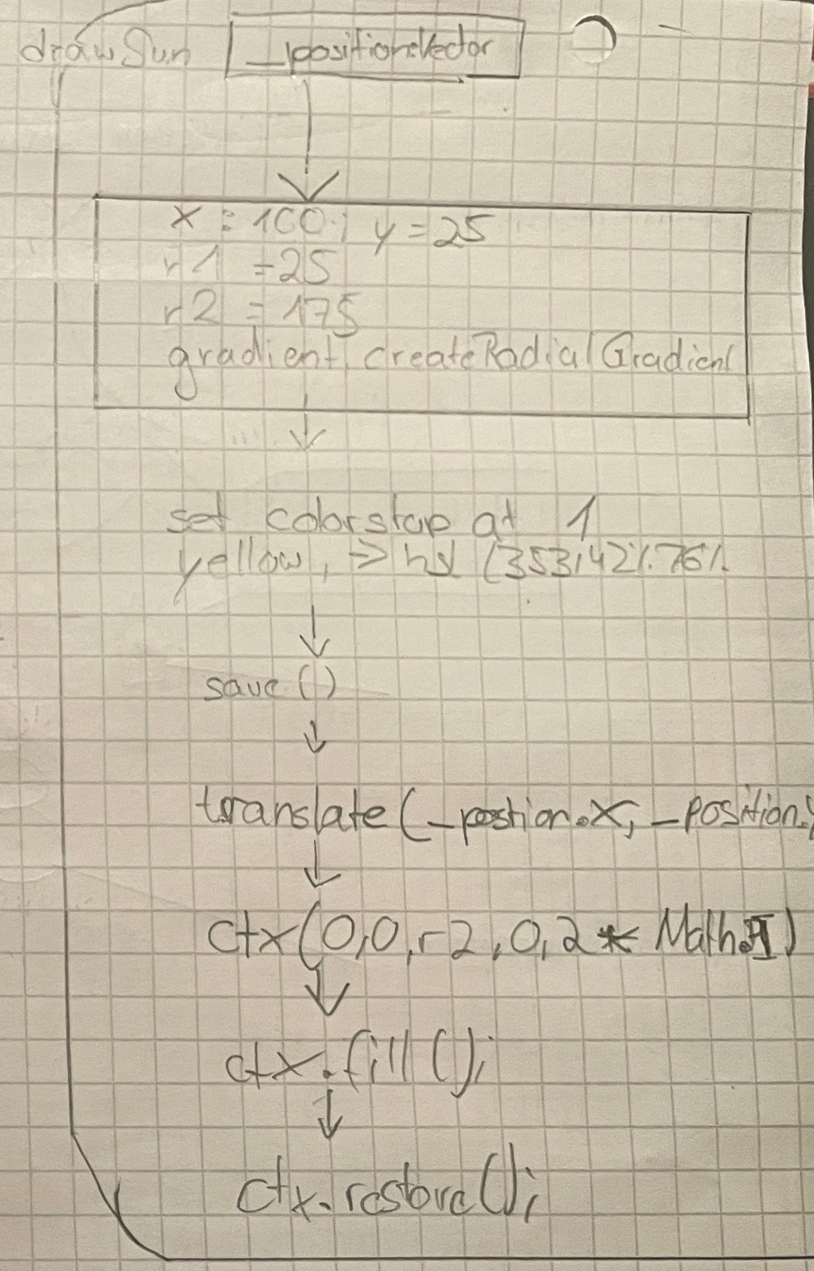


Vector

x: number
y: number







birdhouse

- positiv Vector
  x: 150 y: 2000

ctx. save

↓

translate (= pos x pos y)

↓

beginPath ()

↓

move to

↓

line to

↓

closePath ()

↓

fillStyle = "brown

↓

restor

loch {x: 150 y: 2000}

ctx. save()

↓

translate (pos x : pos y)

↓

ctx beginPath

↓

ctx. arc (0, -250, 75, 0, 2 * Math.PI)

ctx. fillStyle= "black";

↓

fill()

↓

restore

drawmountains

_posifion: Vector
_min: number
_max: number
_colorlow: string
_colorHigh: string

stepMin: 60;
stepMax: 150;
let x: number = 0

save transform
translate(_postion.x, _posion.y
begin path()
ctx. move (0,0)
ctx. LineTo(0, -max)

x += stepMin + Math.random(stepMax - stepMin);
let y: number = _min - Mathrandom()*(_max-_min)

lineTo(x,0)

line to x,0 → close path → execute gradient with give color

restore transform ← draw with path

increase i by 1

i: number → [i < 10] → let birdposX, let birdposY, drawBirds()

drawBirds ( )

ctx. save

ctx. translate(_position.x, _position.y)

ctx. moveto; ctx.line (. . .)

ctx.fillStyle= "hsl(" + Math.random()*18

ctx.fill ()

ctx.beginPath()

ctx.lineTo(. . .)

ctx.fillStyle= "white"

ctx.stroke → ctx.restore

Schneemann [_position : Vector]

ctx.save

↓

ctx.translate(_position.x, _position.y)

↓

ctx.beginPath();

↓

3x ctx.arc(0, -(r1 = 80 ; r2 = 100 ; r = 150)

↓

ctx.close

↓

ctx

ctx.fillStyle =

↓

ctx.beginPath();

↓

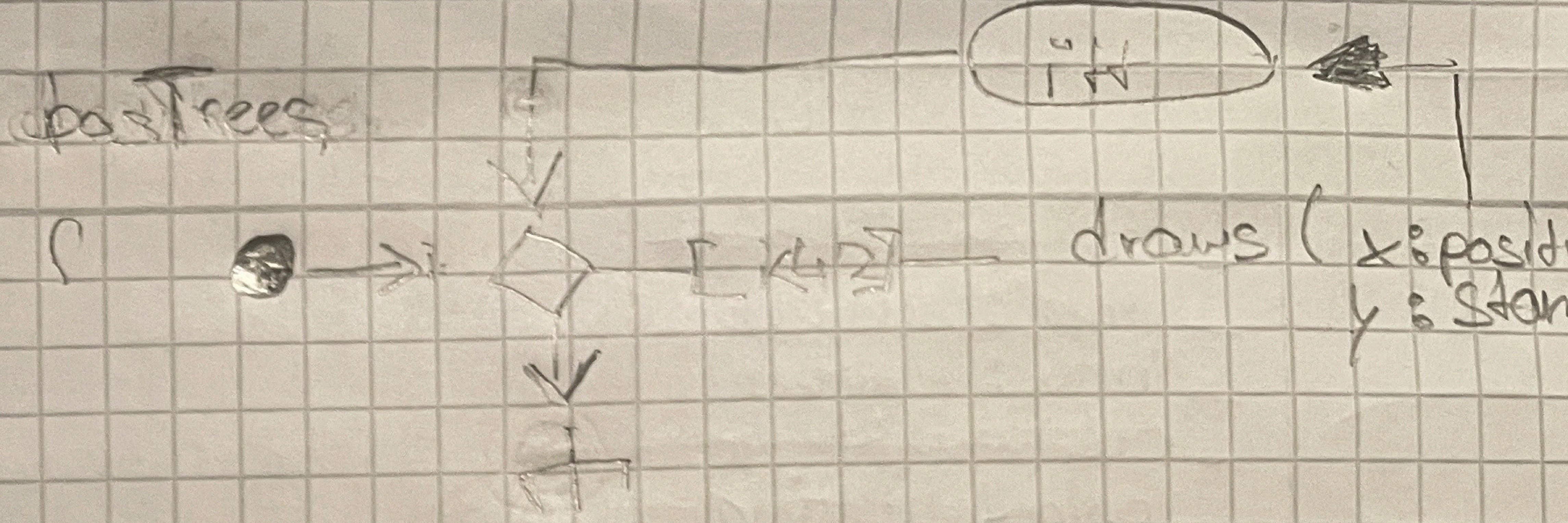
ctx.arc(25, -450, 20, 0, 2* Math.PI)

↓

ctx.fill();

↓

ctx.stroke();



drawTrees (x: positionbaum, y: stammhöhe)

↓

ctx.save();

↓

ctx.beginPath();

↓

shadow

↓

ctx.move

↓

ctx.lineTo

↓

ctx.stroke

↓

ctx.restore

↓

ctx.beginPath();