L09.2_BlumenwieseClasses

//Komilitonen mit denen ich zusammengearbeitet habe: Huu Thien Phan Ngoc, Mona Kabelka

Klassendiagramm-Flowers

Flowers
+ x: number; + flowerType: number; + yRandomMin: number; + yRandomMax: number;
+ constructor(_flowerType: number, _xPos: number, _yRandomMin: number, _yRandomMax: number) + draw(): void

Klassendiagramm-Cloud

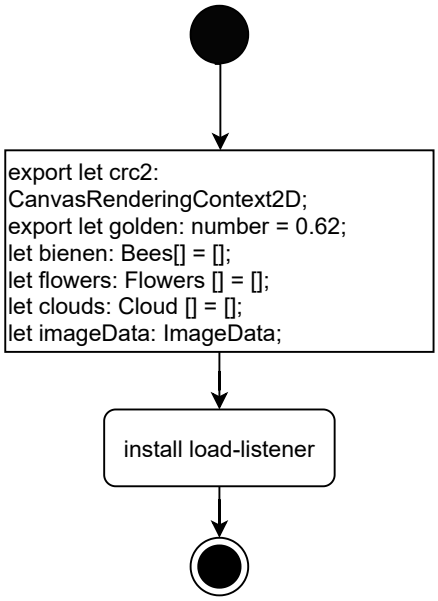
Cloud
+ posX: number; + posY: number; + velocityX: number = 0.5; + velocityY: number = 0.1;
+ constructor(_position: Vector) + draw(): void + update(): void

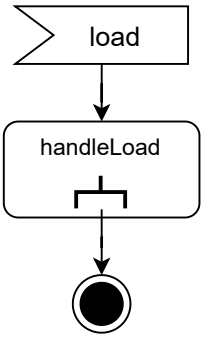
<< *interface* >> *Vector*
x: number; y: number;

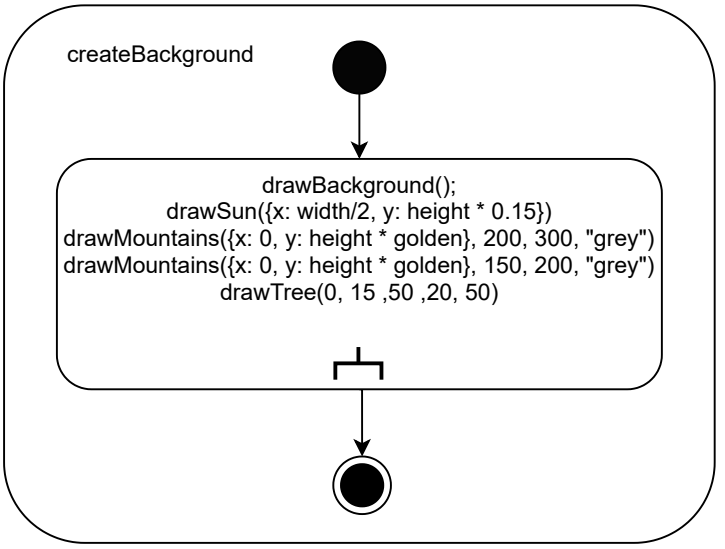
Klassendiagramm-Bee

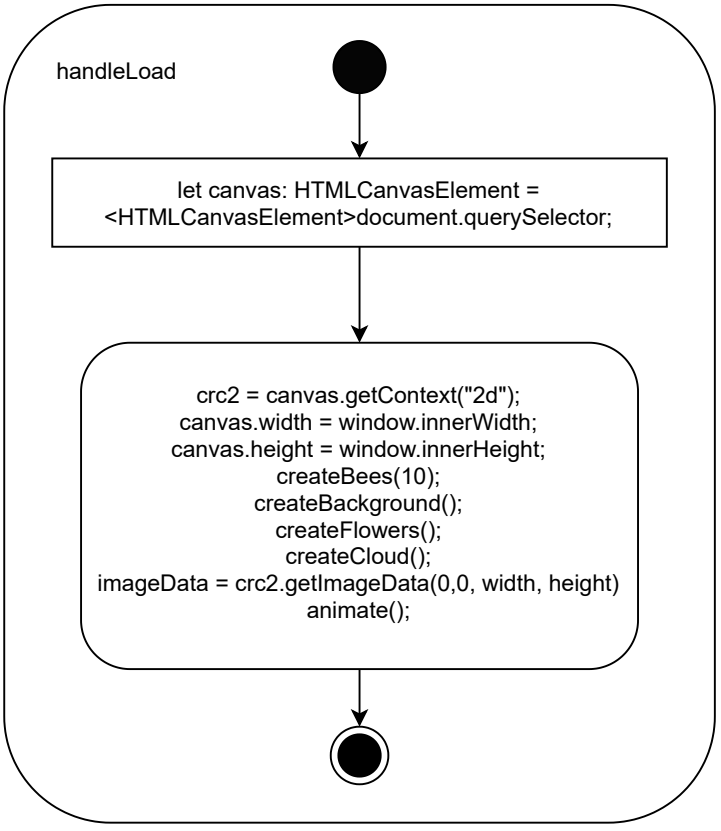
Bees
+ posX: number; + posY: number; + velocityX: number; + velocityY: number; + randomScale: number; + randomNumber: number = (Math.floor(Math.random() * 2000) +1000); + counter: number = 0;
+ constructor(_position: Vector, _velocity: Vector, _randomScale: number + draw(): void + update(): void

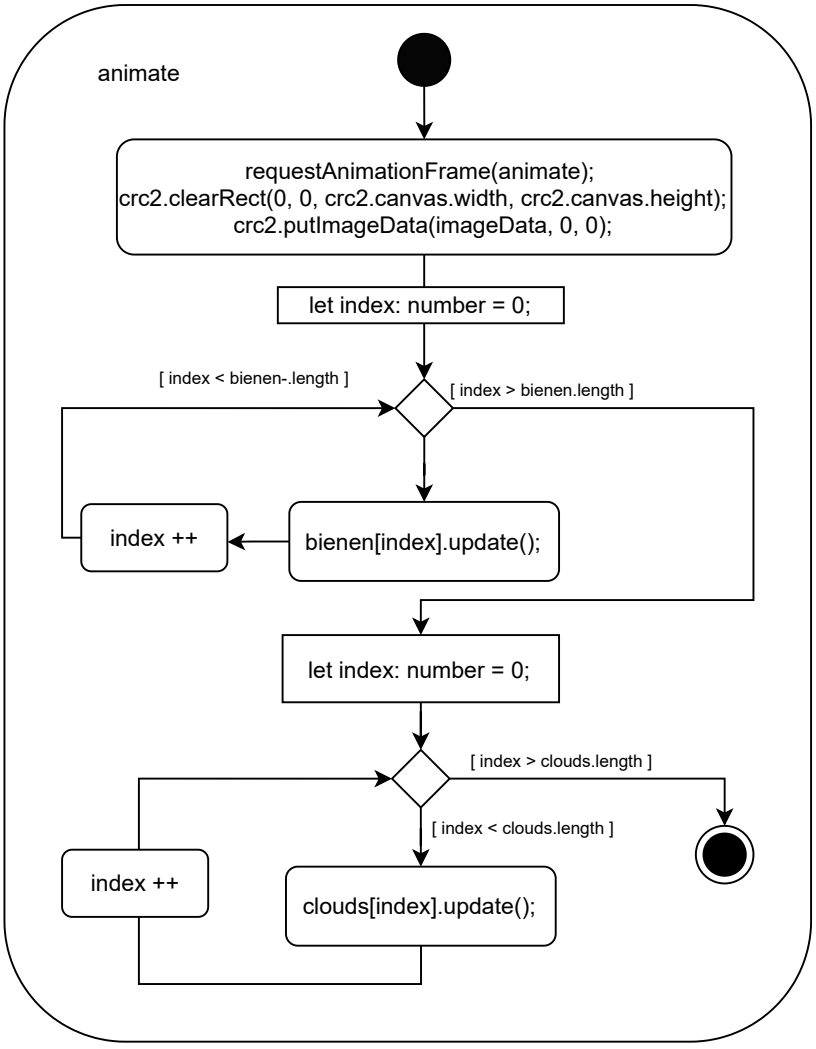
Aktivitätdiagramm - Main

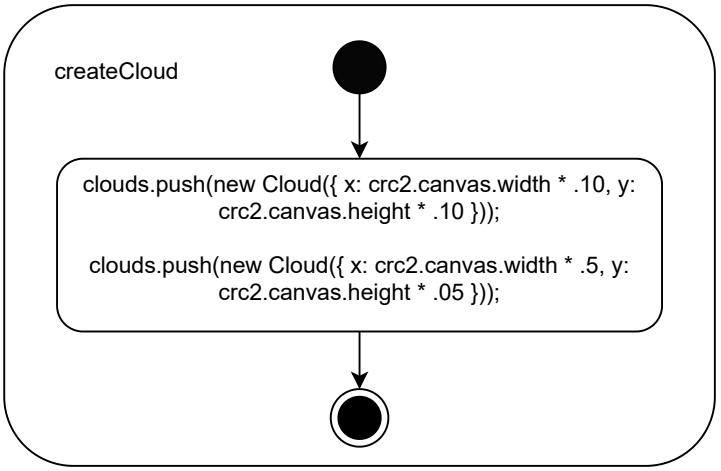


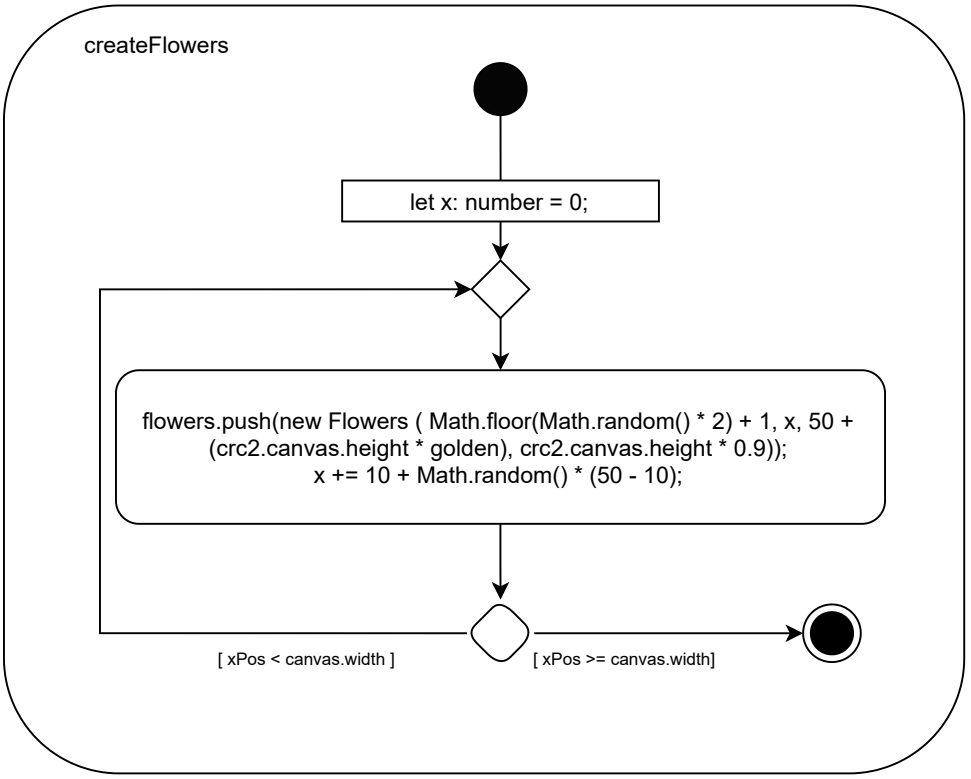


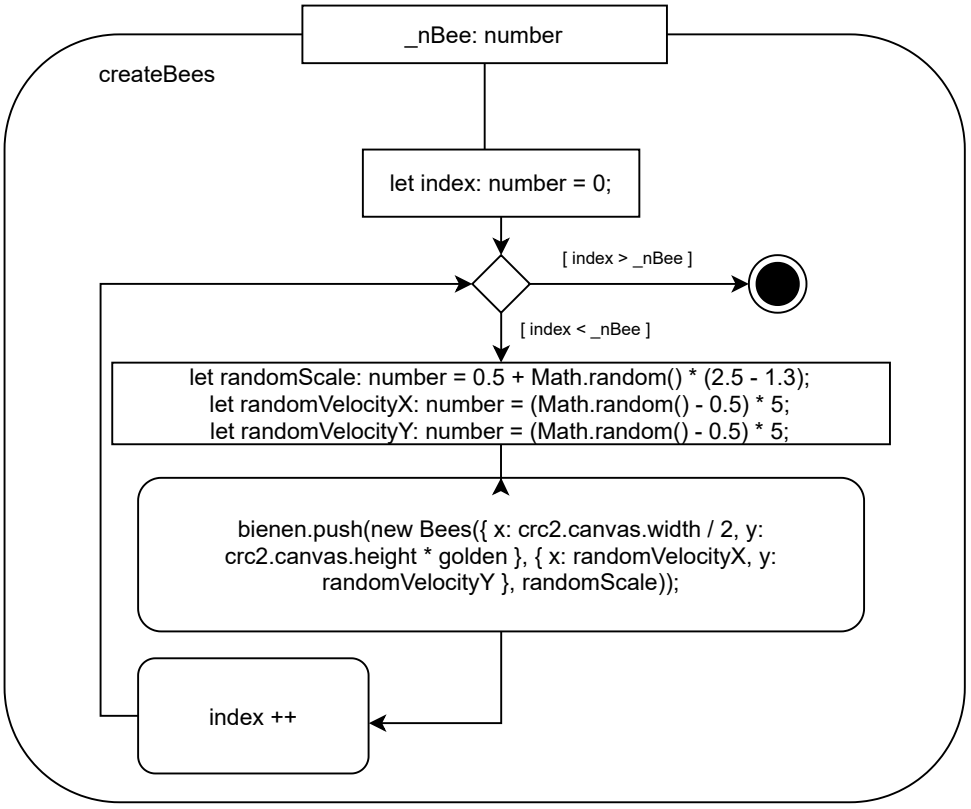




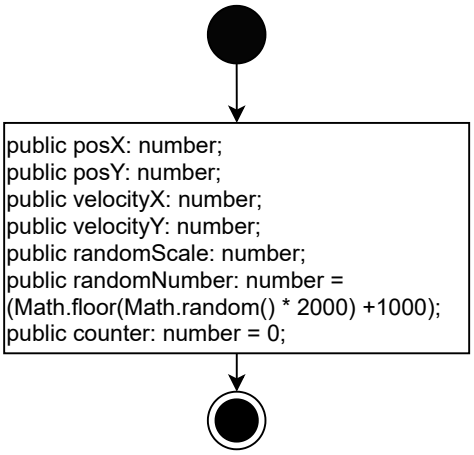


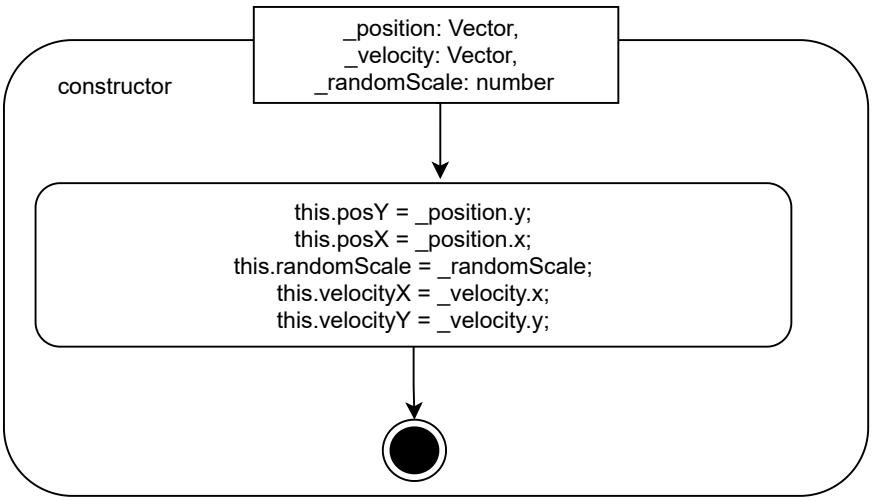


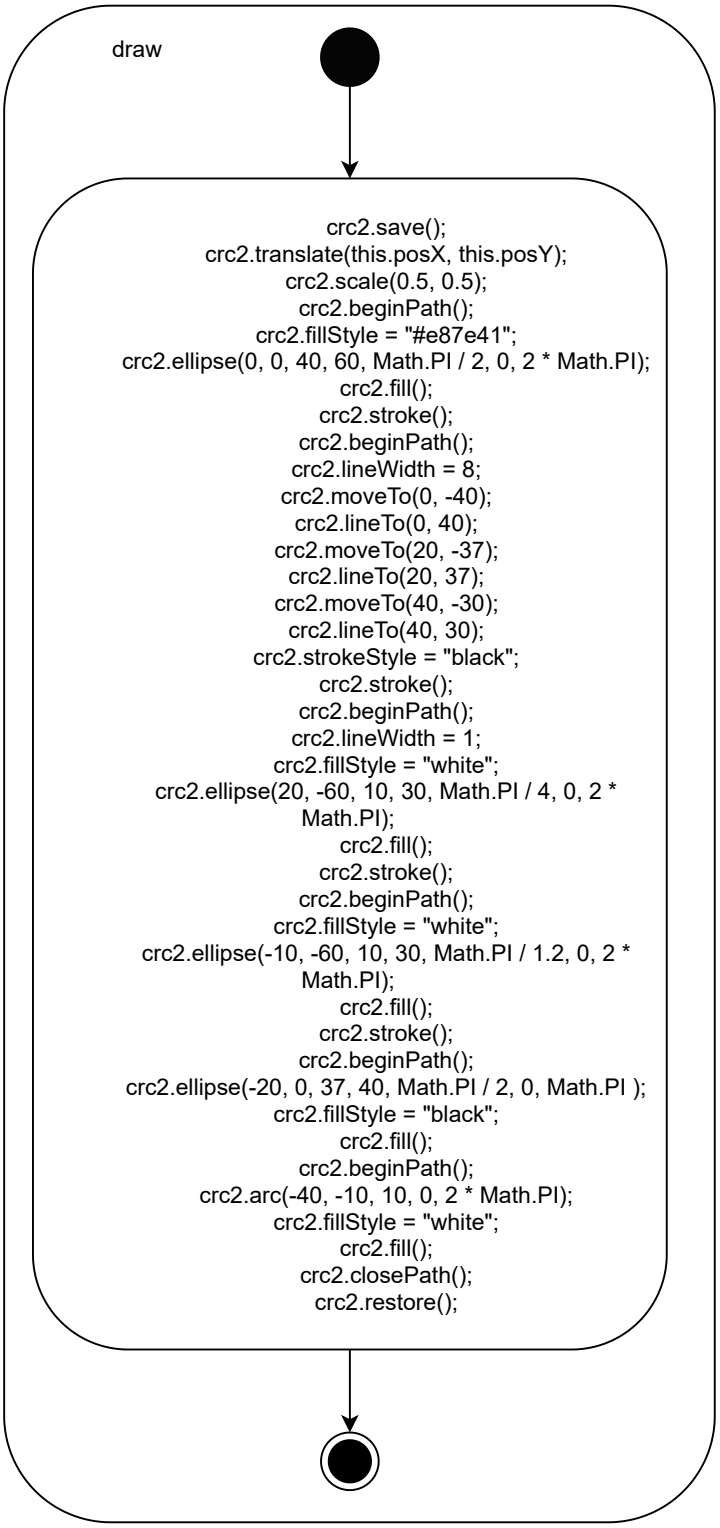

createFlowers
let x: number = 0;
flowers.push(new Flowers (Math.floor(Math.random() * 2) + 1, x, 50 +
(crc2.canvas.height * golden), crc2.canvas.height * 0.9));
x += 10 + Math.random() * (50 - 10);
[xPos < canvas.width]
[xPos >= canvas.width]

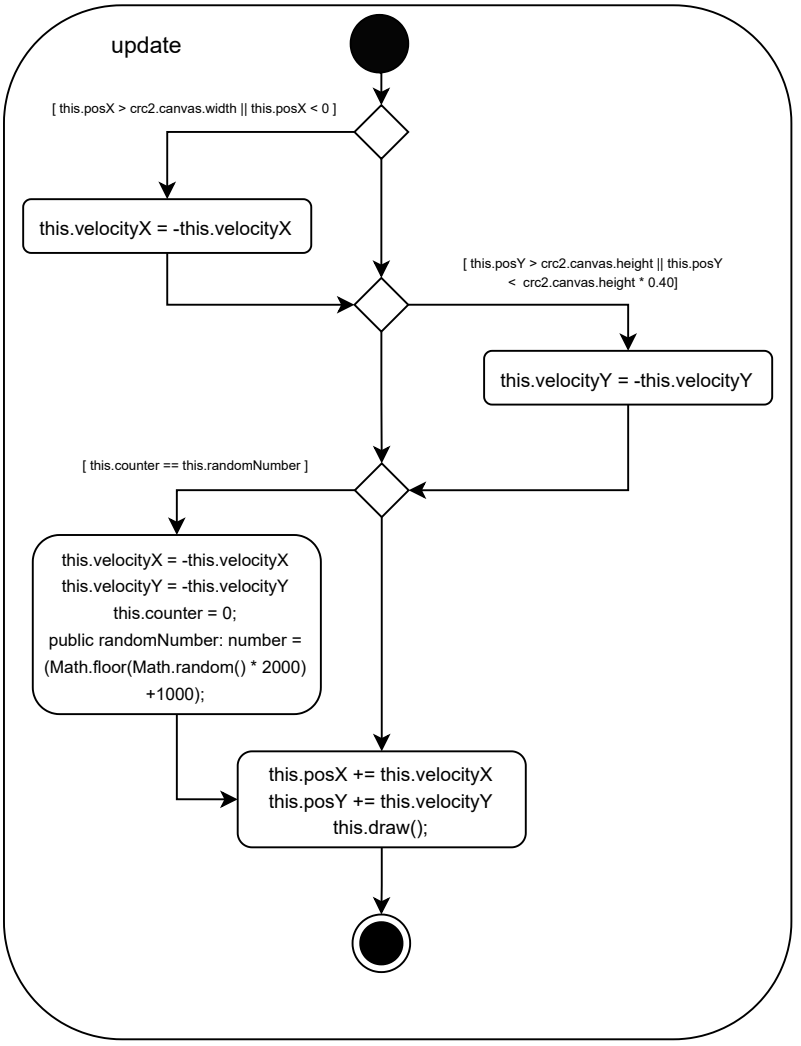

_nBee: number
createBees
let index: number = 0;
[index > _nBee]
[index < _nBee]
let randomScale: number = 0.5 + Math.random() * (2.5 - 1.3);
let randomVelocityX: number = (Math.random() - 0.5) * 5;
let randomVelocityY: number = (Math.random() - 0.5) * 5;
bienen.push(new Bees({ x: crc2.canvas.width / 2, y:
crc2.canvas.height * golden }, { x: randomVelocityX, y:
randomVelocityY }, randomScale));
index ++

Aktivitätdiagramm - bees

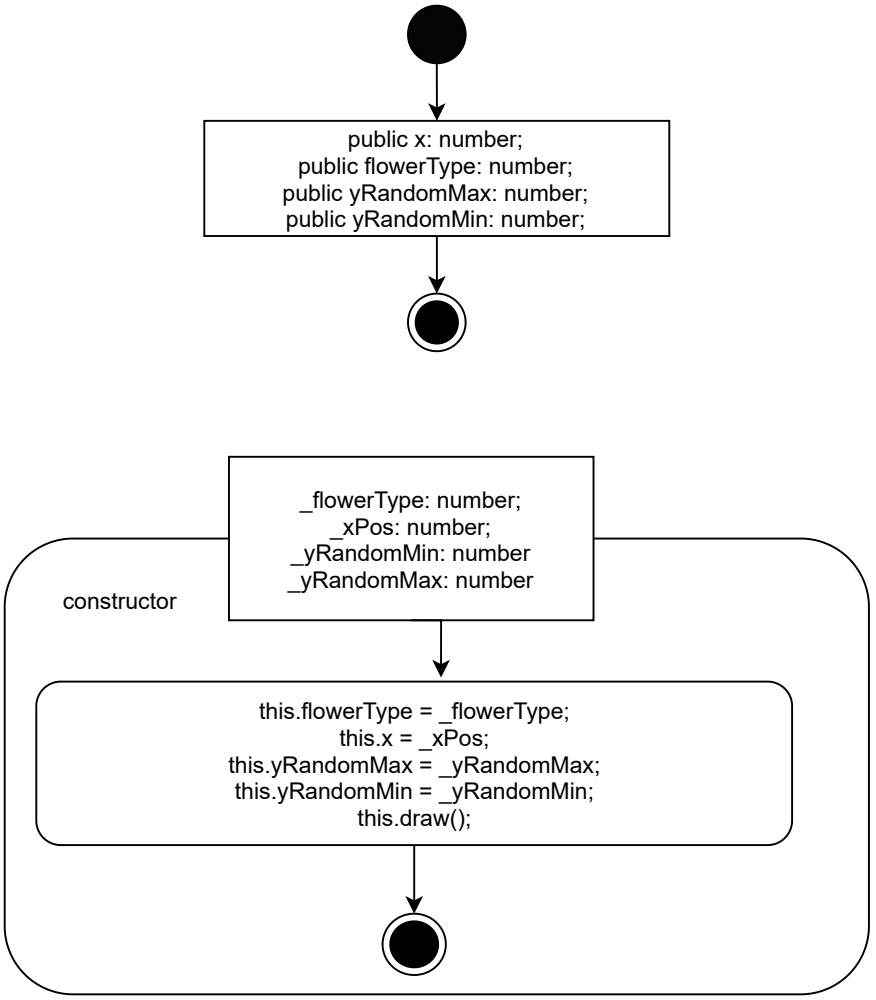




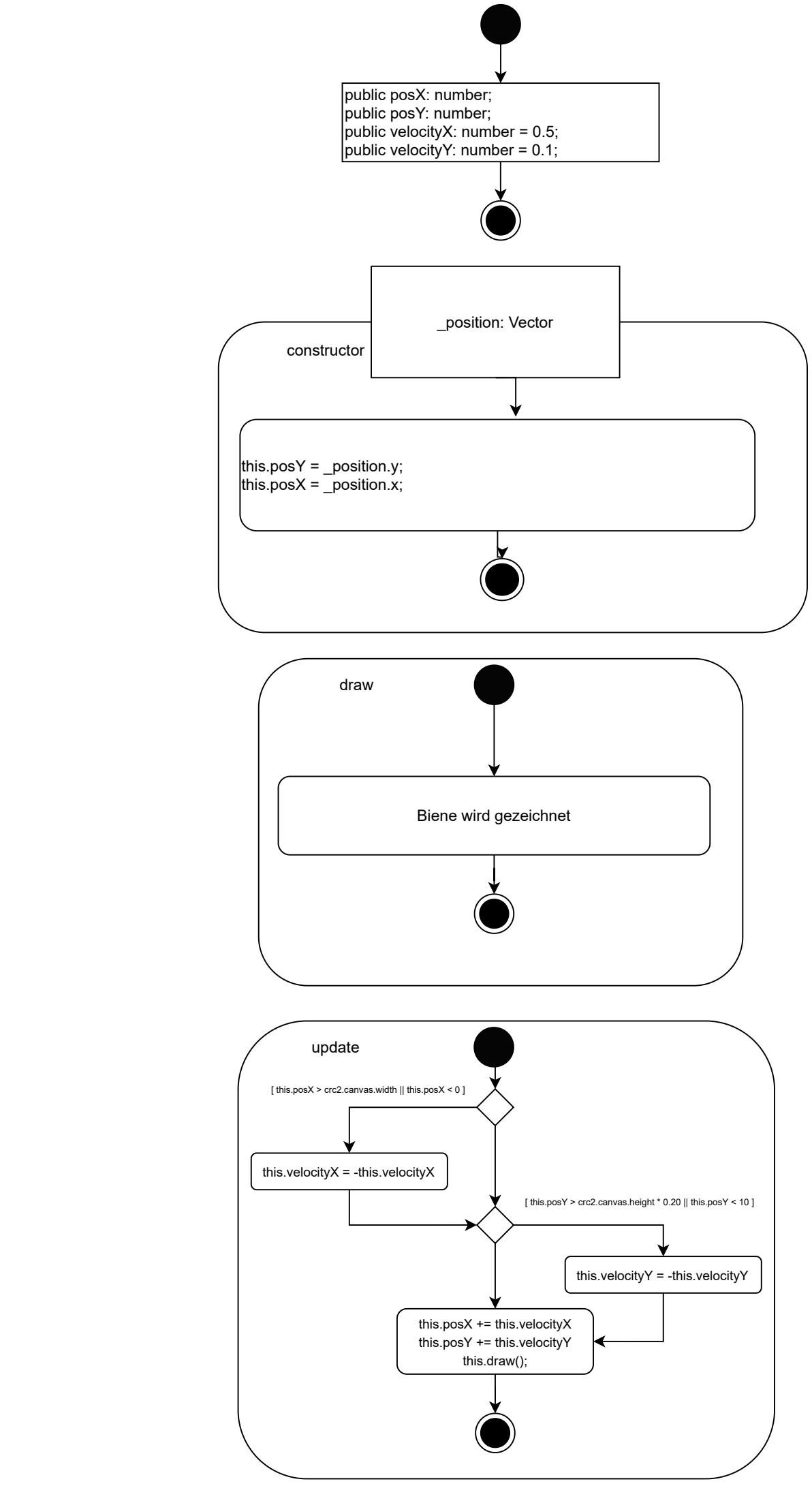




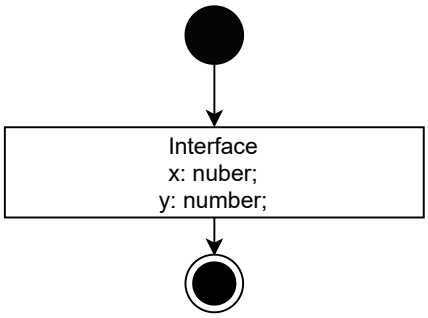
Aktivitätdiagramm - flower

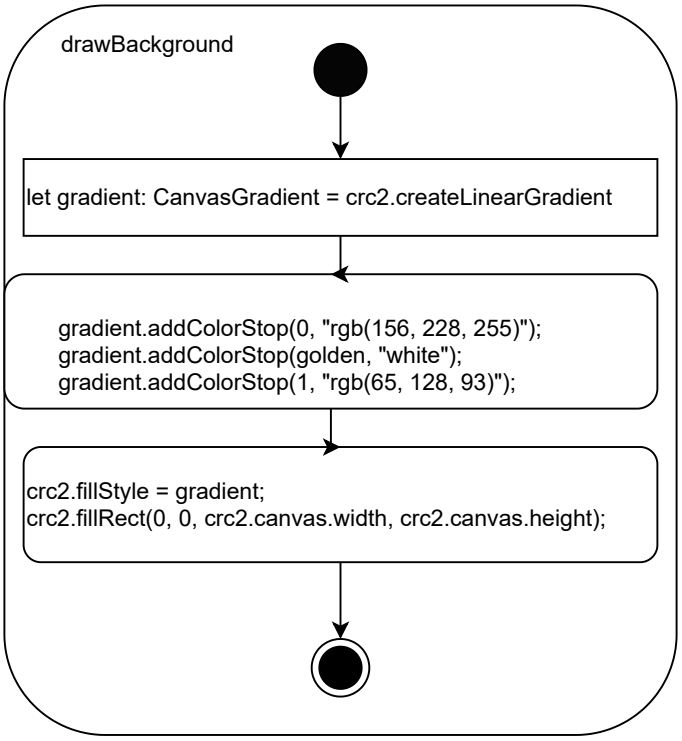


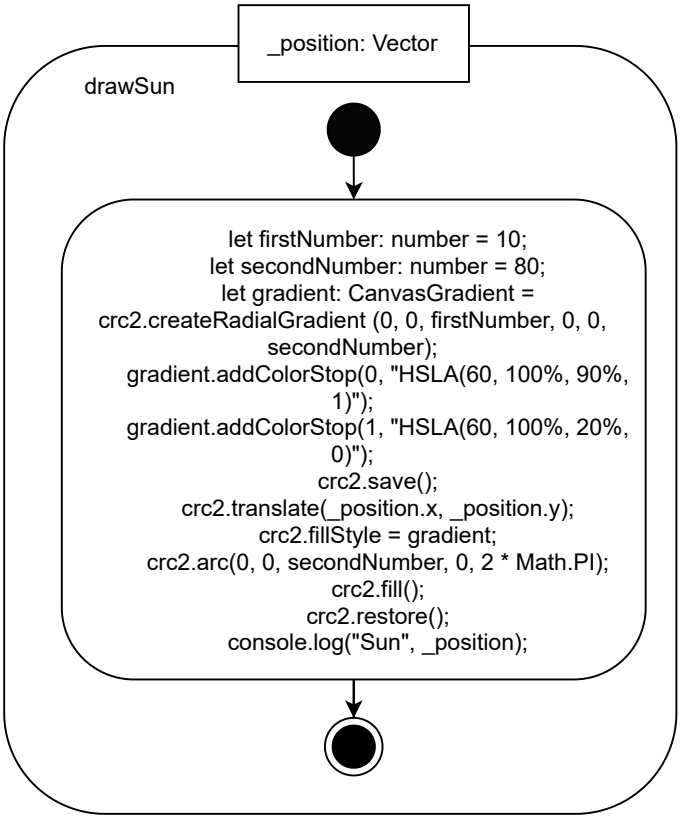
Aktivitätdiagramm - clouds

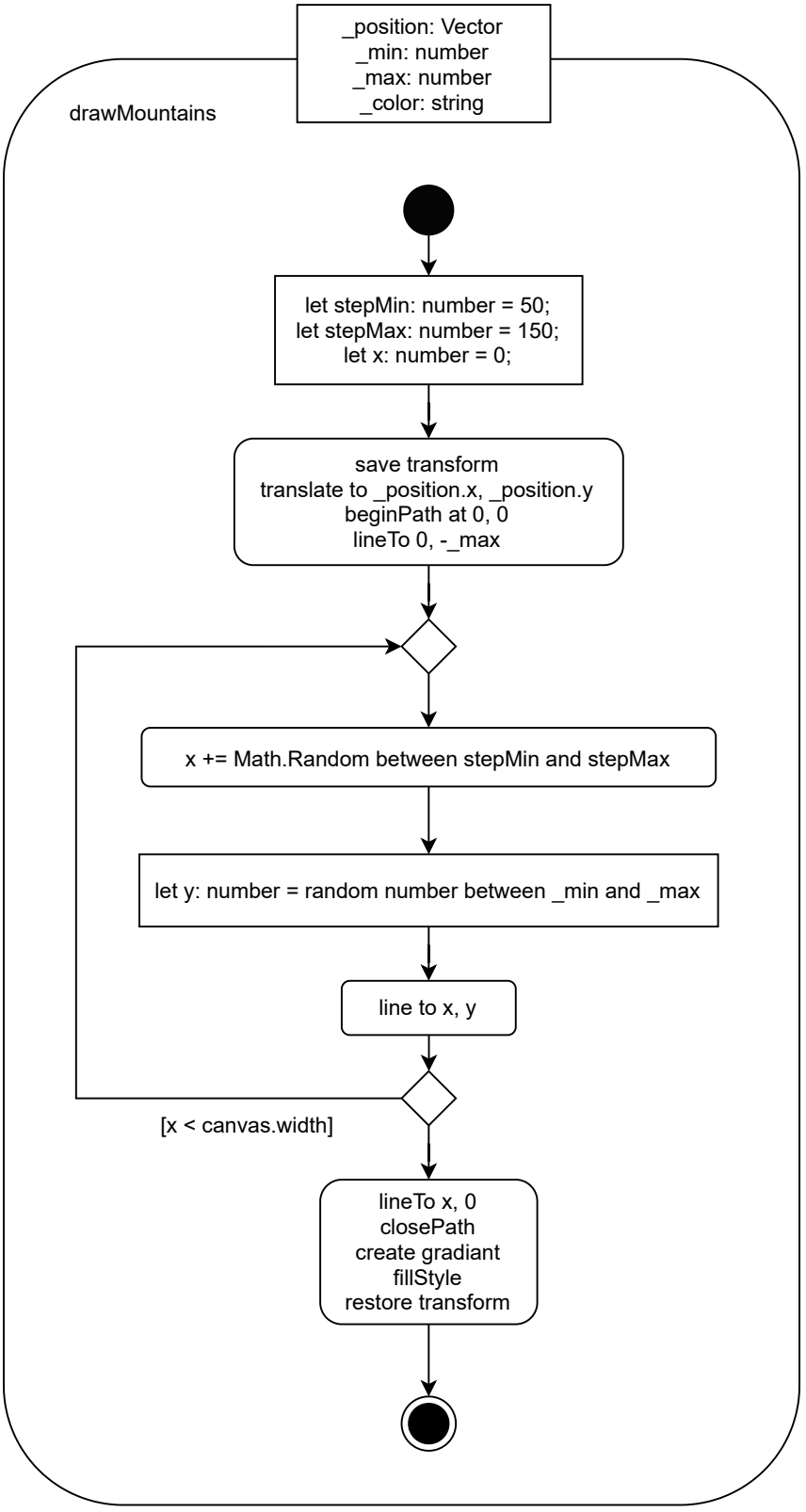


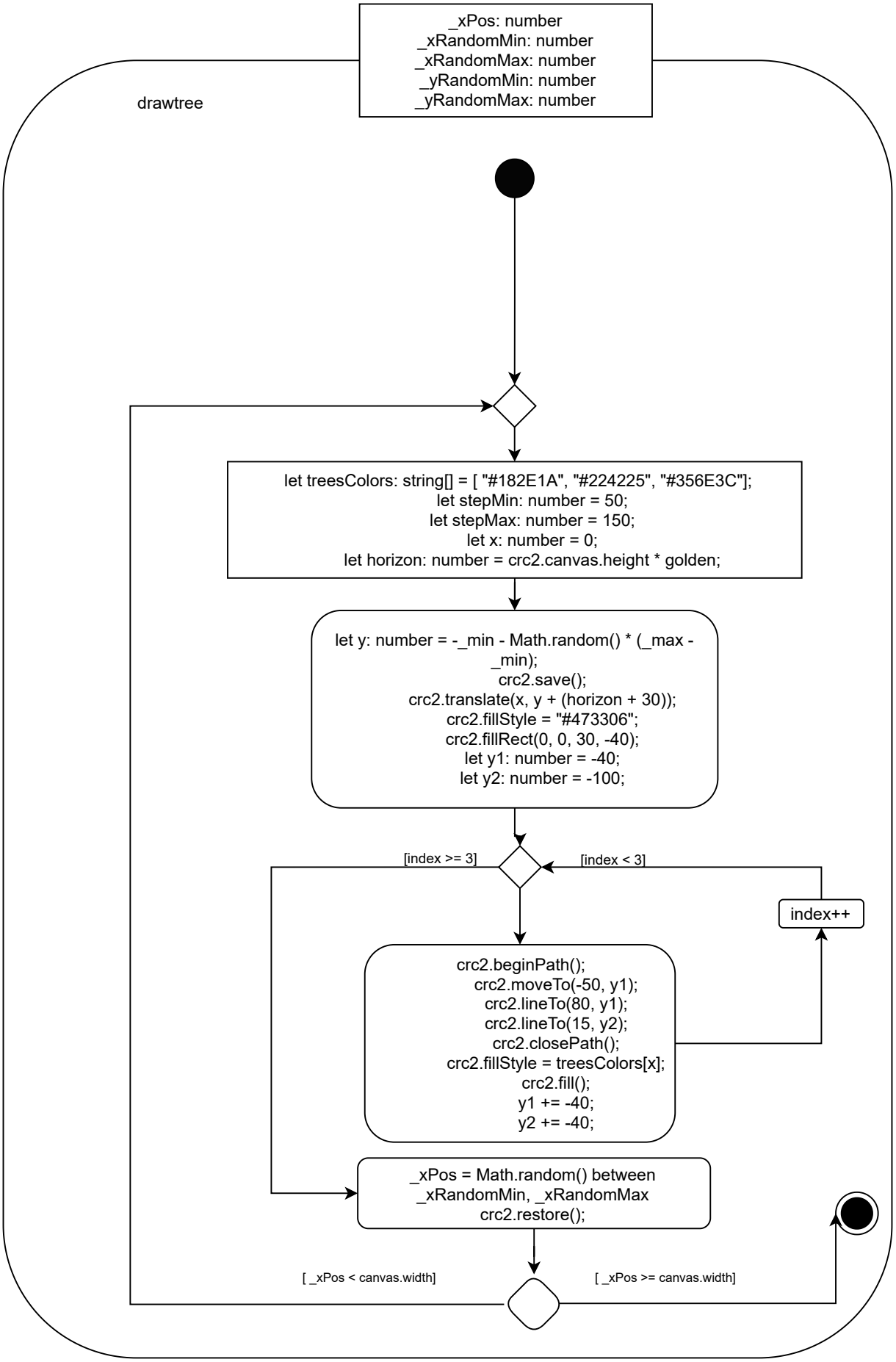
Aktivitätdiagramm - background

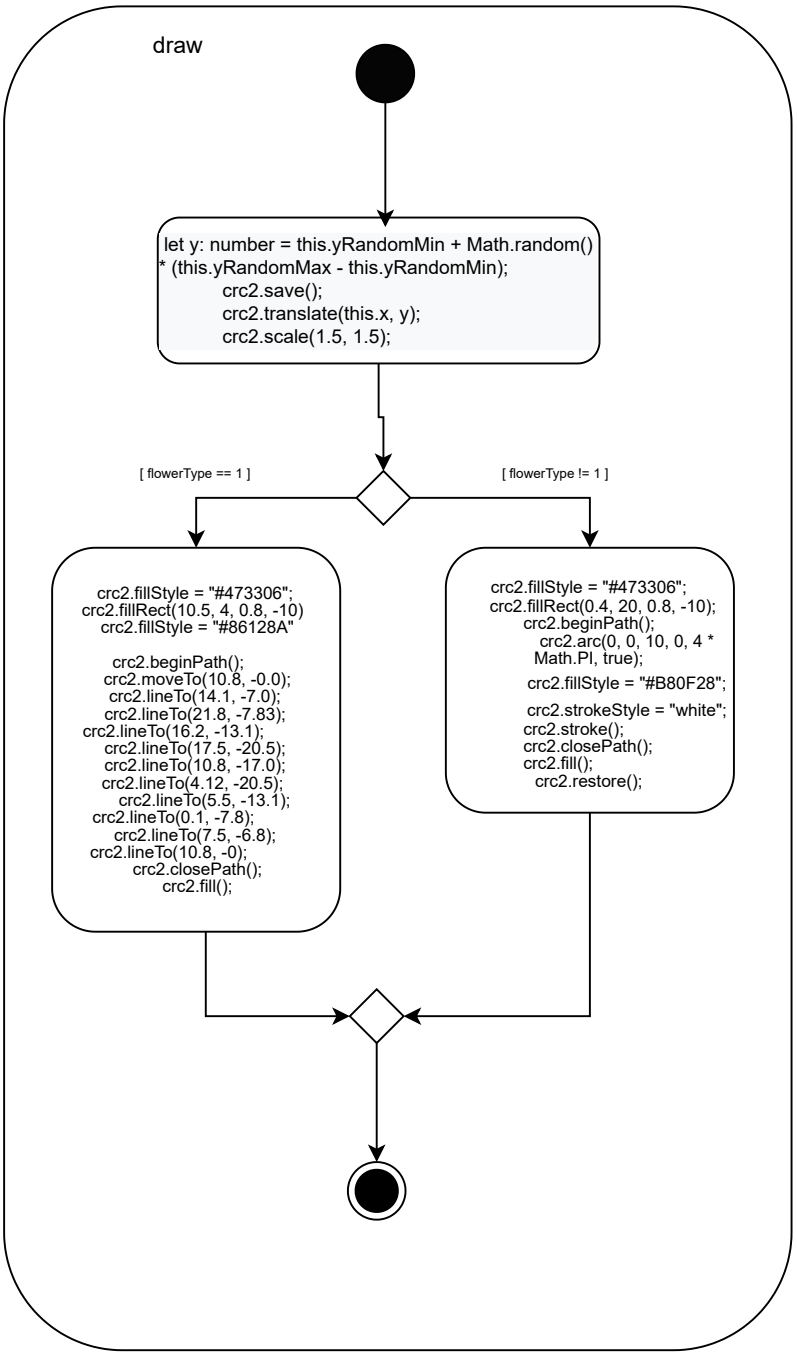









_xPos: number
_xRandomMin: number
_xRandomMax: number
_yRandomMin: number
_yRandomMax: number
drawtree
let treesColors: string[] = ["#182E1A", "#224225", "#356E3C"];
let stepMin: number = 50;
let stepMax: number = 150;
let x: number = 0;
let horizon: number = crc2.canvas.height * golden;
let y: number = -_min - Math.random() * (_max -
_min);
crc2.save();
crc2.translate(x, y + (horizon + 30));
crc2.fillStyle = "#473306";
crc2.fillRect(0, 0, 30, -40);
let y1: number = -40;
let y2: number = -100;
[index >= 3]
[index < 3]
index++
crc2.beginPath();
crc2.moveTo(-50, y1);
crc2.lineTo(80, y1);
crc2.lineTo(15, y2);
crc2.closePath();
crc2.fillStyle = treesColors[x];
crc2.fill();
y1 += -40;
y2 += -40;
_xPos = Math.random() between
_xRandomMin, _xRandomMax
crc2.restore();
[_xPos < canvas.width]
[_xPos >= canvas.width]


draw
let y: number = this.yRandomMin + Math.random()
* (this.yRandomMax - this.yRandomMin);
crc2.save();
crc2.translate(this.x, y);
crc2.scale(1.5, 1.5);
[flowerType == 1]
[flowerType != 1]
crc2.fillStyle = "#473306";
crc2.fillRect(10.5, 4, 0.8, -10)
crc2.fillStyle = "#86128A"
crc2.beginPath();
crc2.moveTo(10.8, -0.0);
crc2.lineTo(14.1, -7.0);
crc2.lineTo(21.8, -7.83);
crc2.lineTo(16.2, -13.1);
crc2.lineTo(17.5, -20.5);
crc2.lineTo(10.8, -17.0);
crc2.lineTo(4.12, -20.5);
crc2.lineTo(5.5, -13.1);
crc2.lineTo(0.1, -7.8);
crc2.lineTo(7.5, -6.8);
crc2.lineTo(10.8, -0);
crc2.closePath();
crc2.fill();
crc2.fillStyle = "#473306";
crc2.fillRect(0.4, 20, 0.8, -10);
crc2.beginPath();
crc2.arc(0, 0, 10, 0, 4 *
Math.PI, true);
crc2.fillStyle = "#B80F28";
crc2.strokeStyle = "white";
crc2.stroke();
crc2.closePath();
crc2.fill();
crc2.restore();