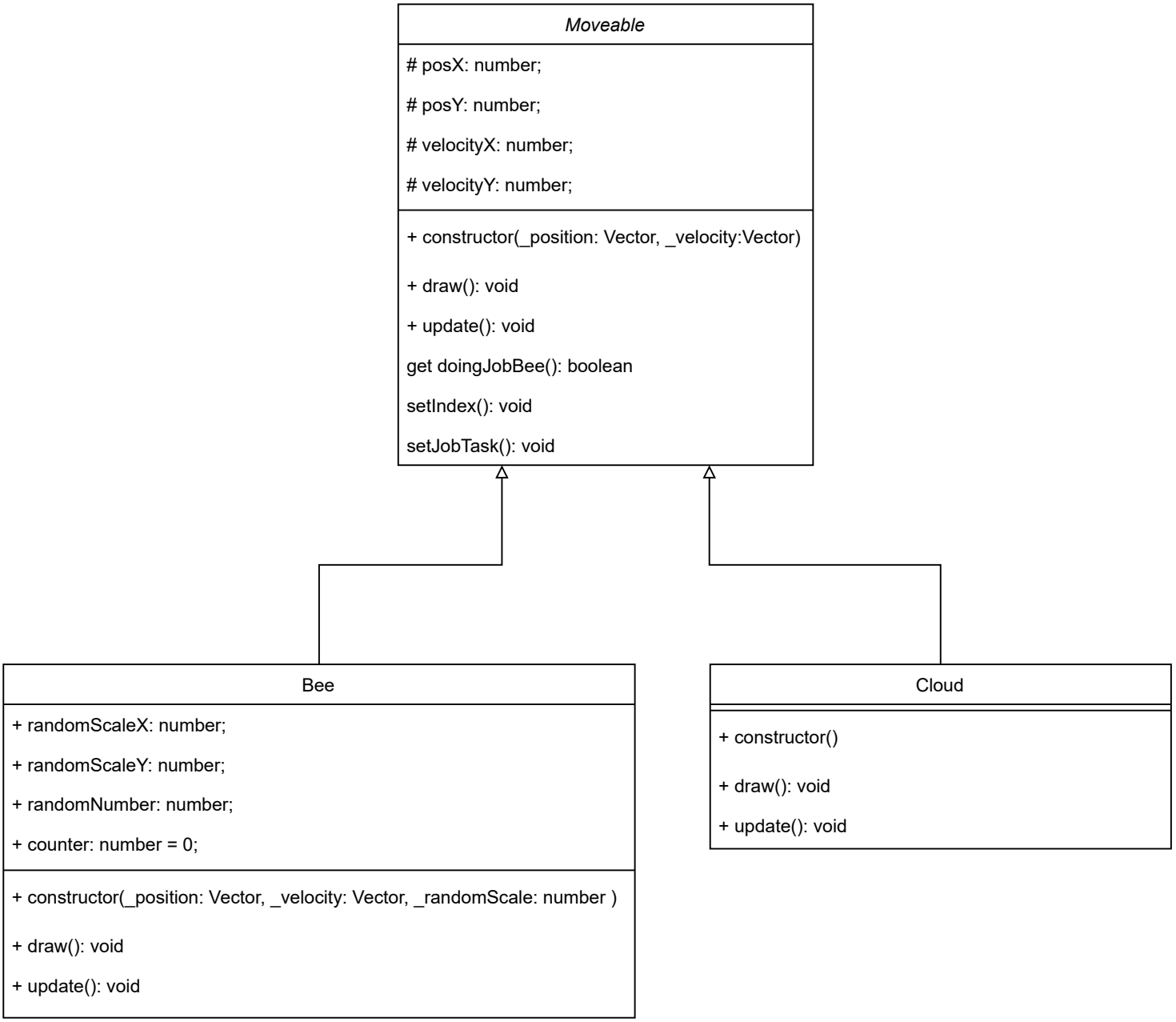
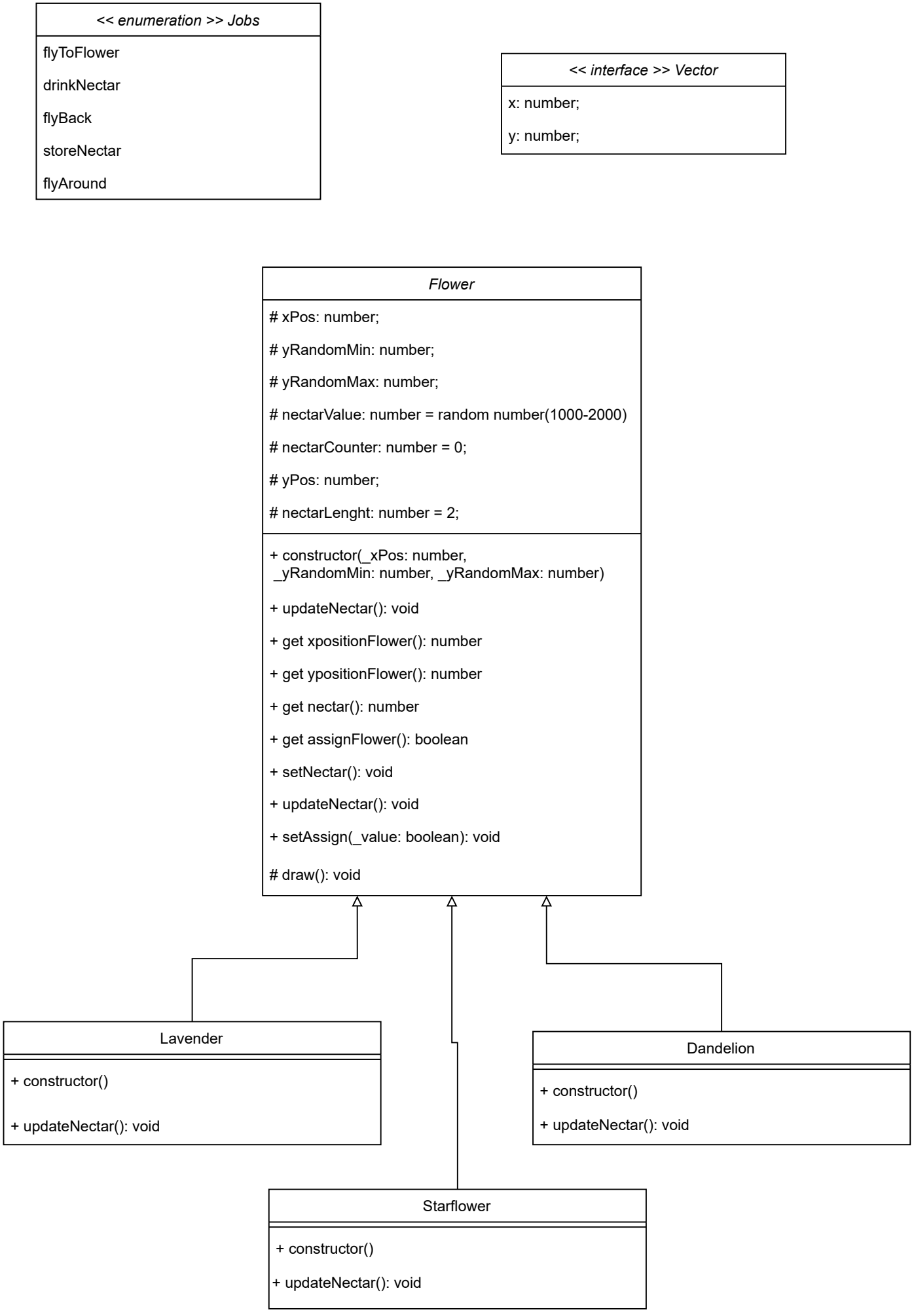
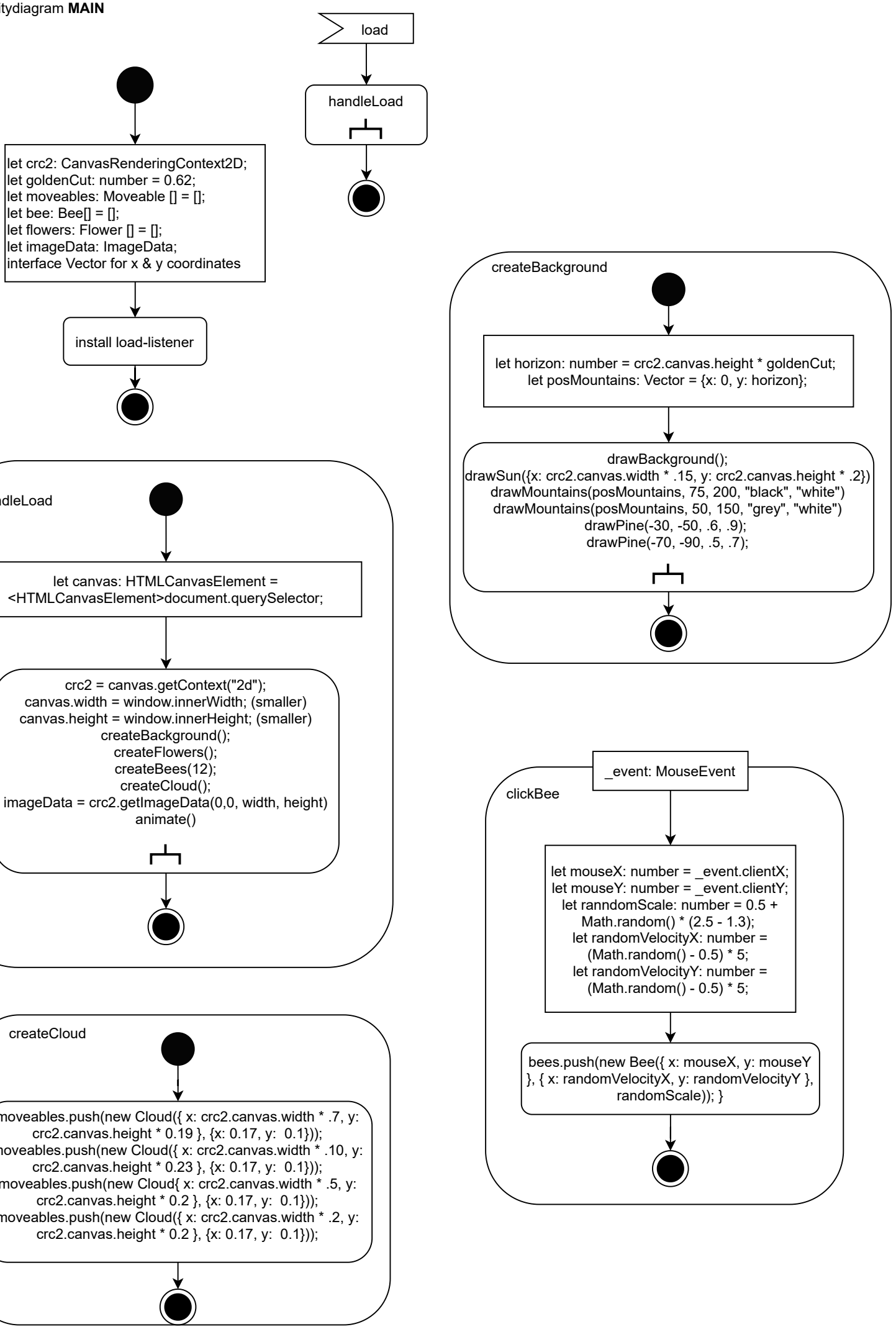
L11.2 Intelligent Meadow Advanced

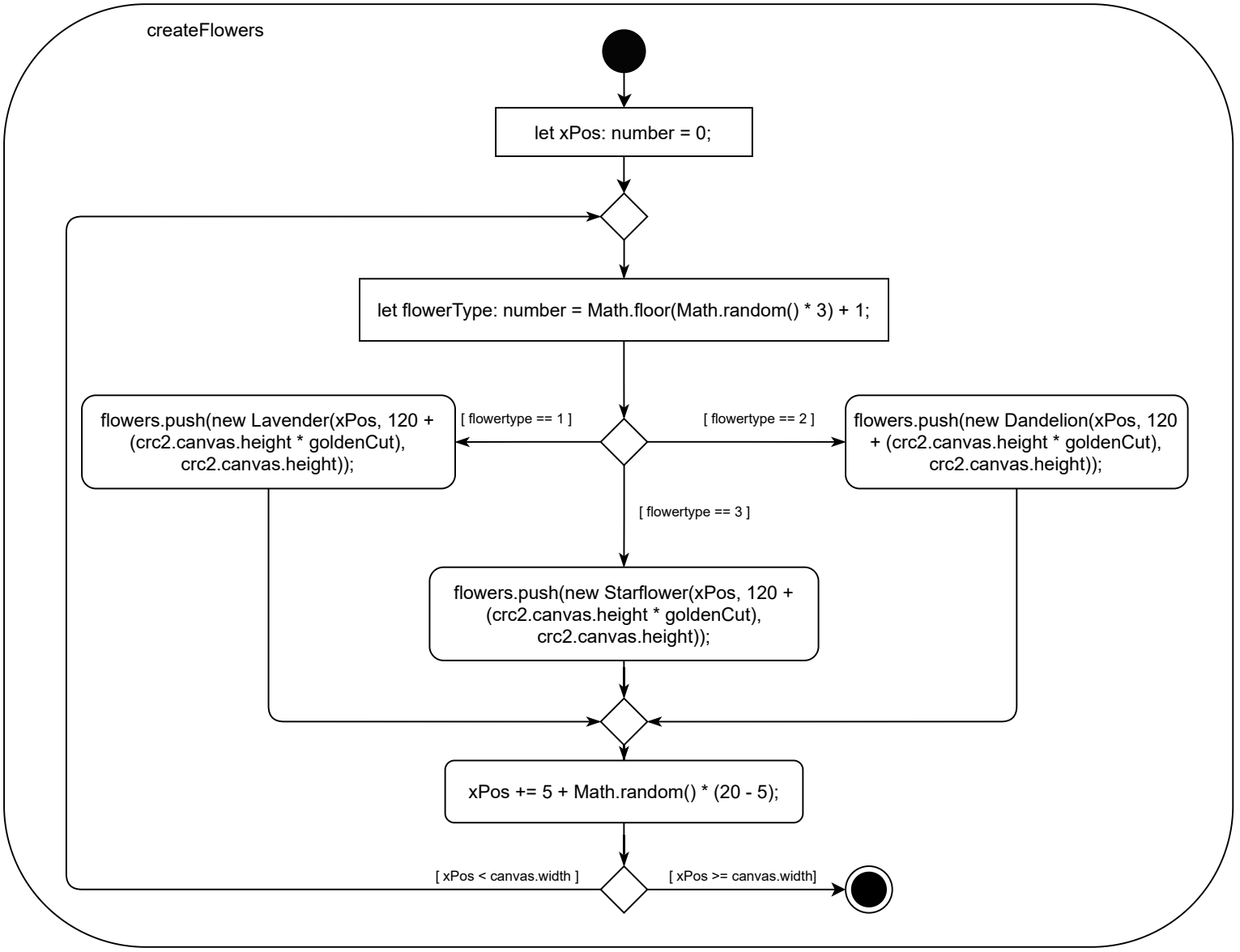
Classdiagrams

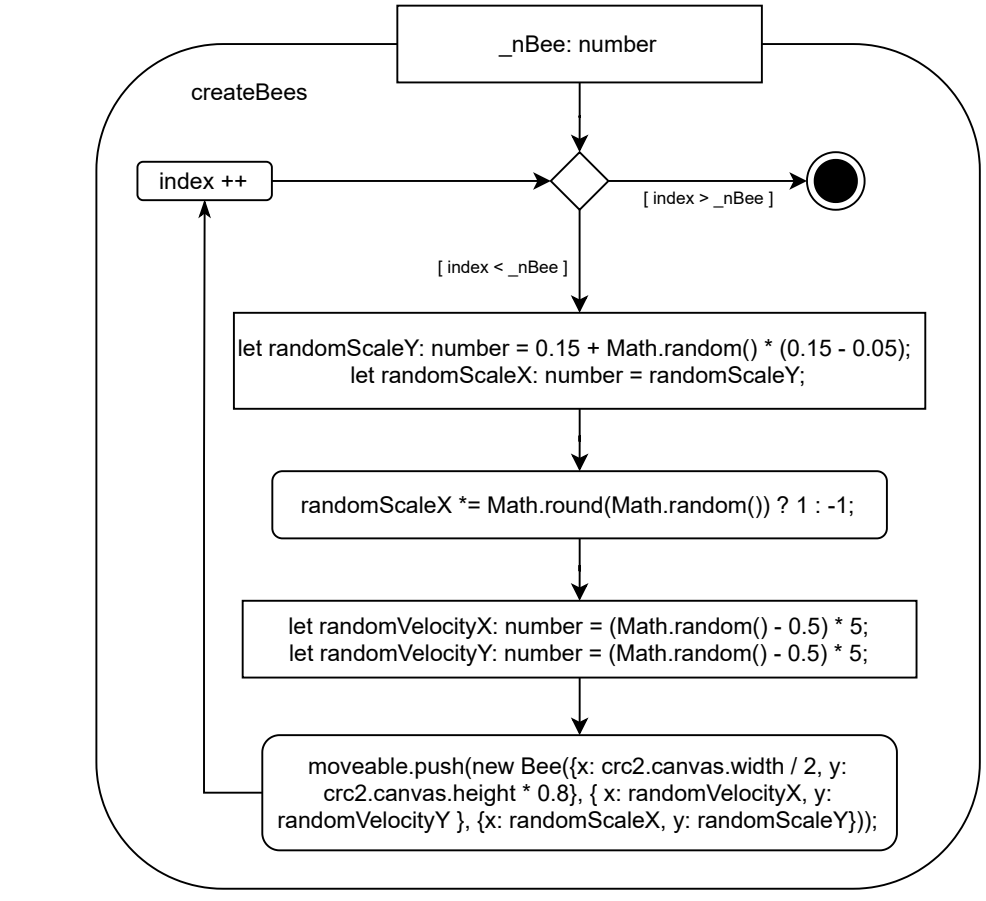


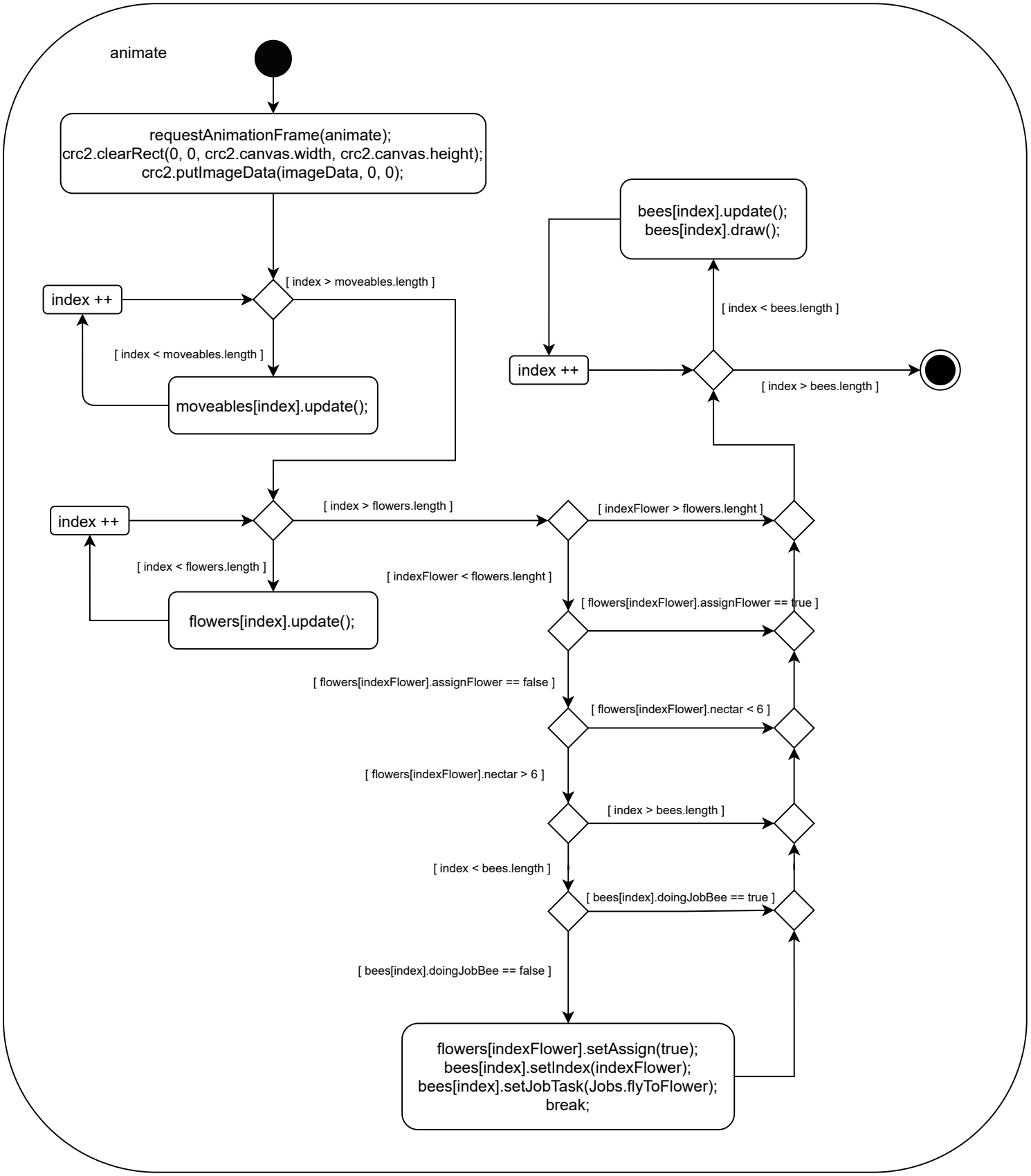

<< enumeration >> Jobs
flyToFlower
drinkNectar
flyBack
storeNectar
flyAround
<< interface >> Vector
x: number;
y: number;
Flower
xPos: number;
yRandomMin: number;
yRandomMax: number;
nectarValue: number = random number(1000-2000)
nectarCounter: number = 0;
yPos: number;
nectarLenght: number = 2;
+ constructor(_xPos: number, _yRandomMin: number, _yRandomMax: number)
+ updateNectar(): void
+ get xpositionFlower(): number
+ get ypositionFlower(): number
+ get nectar(): number
+ get assignFlower(): boolean
+ setNectar(): void
+ updateNectar(): void
+ setAssign(_value: boolean): void
# draw(): void
Lavender
+ constructor()
+ updateNectar(): void
Dandelion
+ constructor()
+ updateNectar(): void
Starflower
+ constructor()
+ updateNectar(): void

Activitydiagram **MAIN**

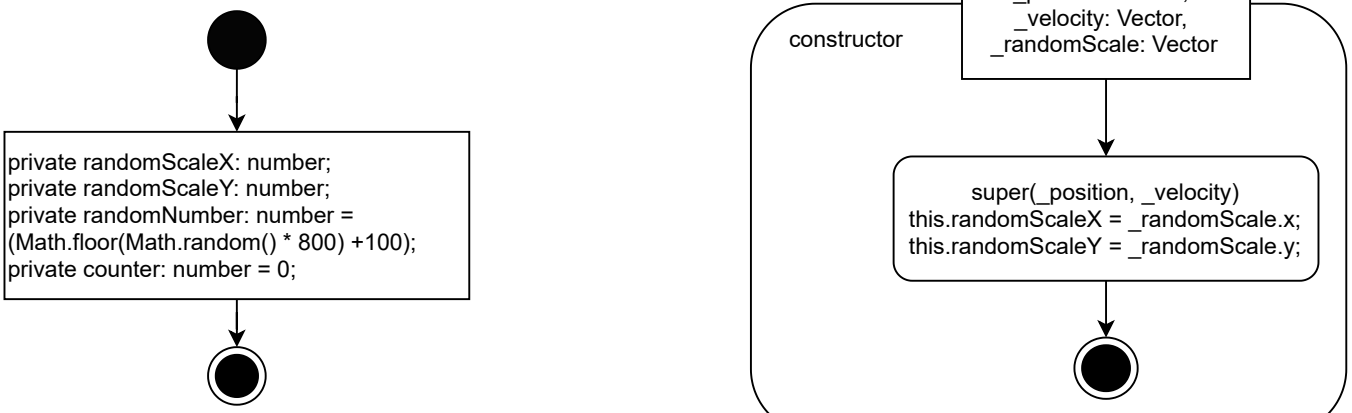


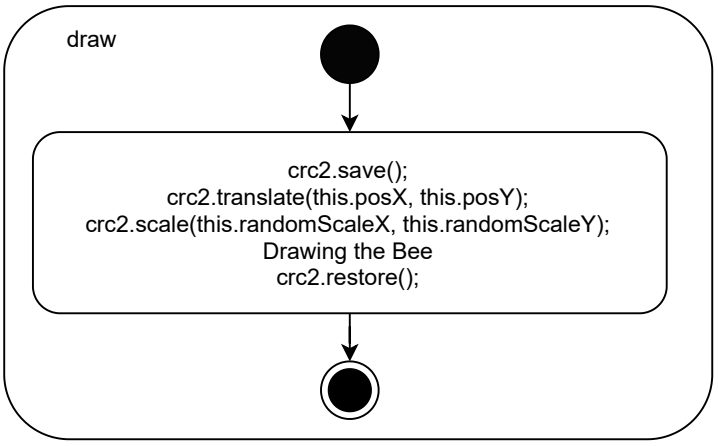

createFlowers
let xPos: number = 0;
let flowerType: number = Math.floor(Math.random() * 3) + 1;
[flowertype == 1]
flowers.push(new Lavender(xPos, 120 + (crc2.canvas.height * goldenCut), crc2.canvas.height));
[flowertype == 2]
flowers.push(new Dandelion(xPos, 120 + (crc2.canvas.height * goldenCut), crc2.canvas.height));
[flowertype == 3]
flowers.push(new Starflower(xPos, 120 + (crc2.canvas.height * goldenCut), crc2.canvas.height));
xPos += 5 + Math.random() * (20 - 5);
[xPos < canvas.width]
[xPos >= canvas.width]

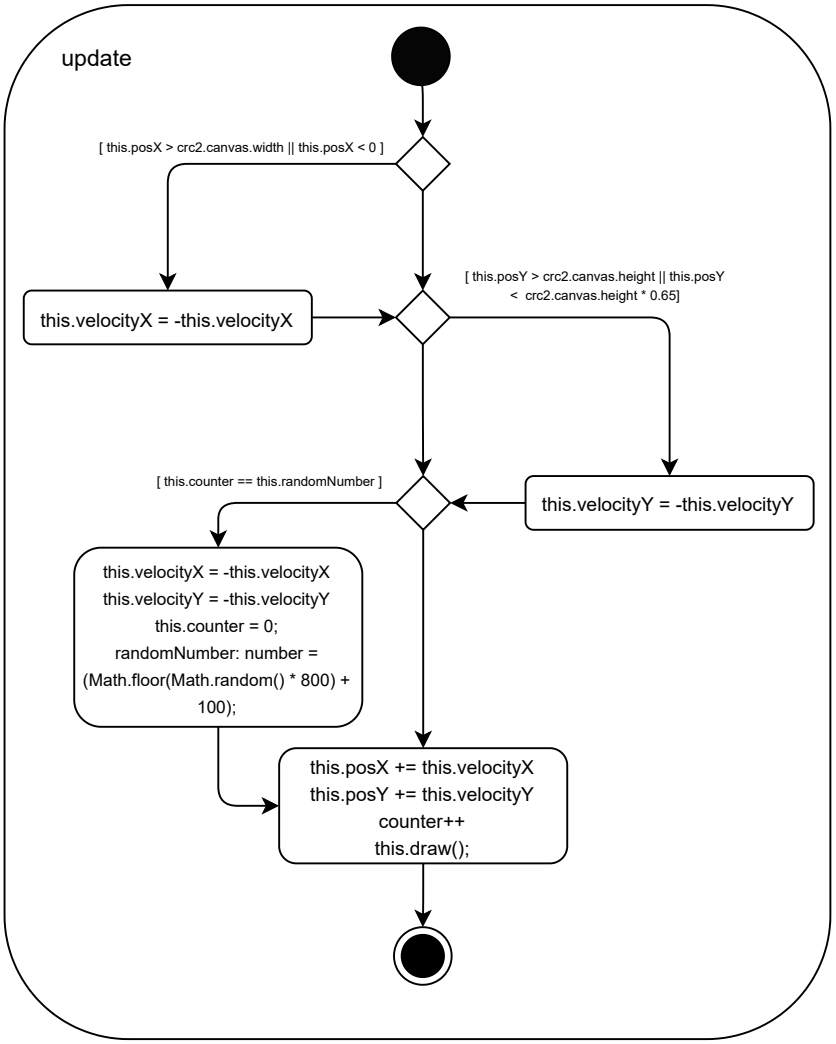

_nBee: number
createBees
index ++
[index > _nBee]
[index < _nBee]
let randomScaleY: number = 0.15 + Math.random() * (0.15 - 0.05);
let randomScaleX: number = randomScaleY;
randomScaleX *= Math.round(Math.random()) ? 1 : -1;
let randomVelocityX: number = (Math.random() - 0.5) * 5;
let randomVelocityY: number = (Math.random() - 0.5) * 5;
moveable.push(new Bee({x: crc2.canvas.width / 2, y: crc2.canvas.height * 0.8}, { x: randomVelocityX, y: randomVelocityY }, {x: randomScaleX, y: randomScaleY}));

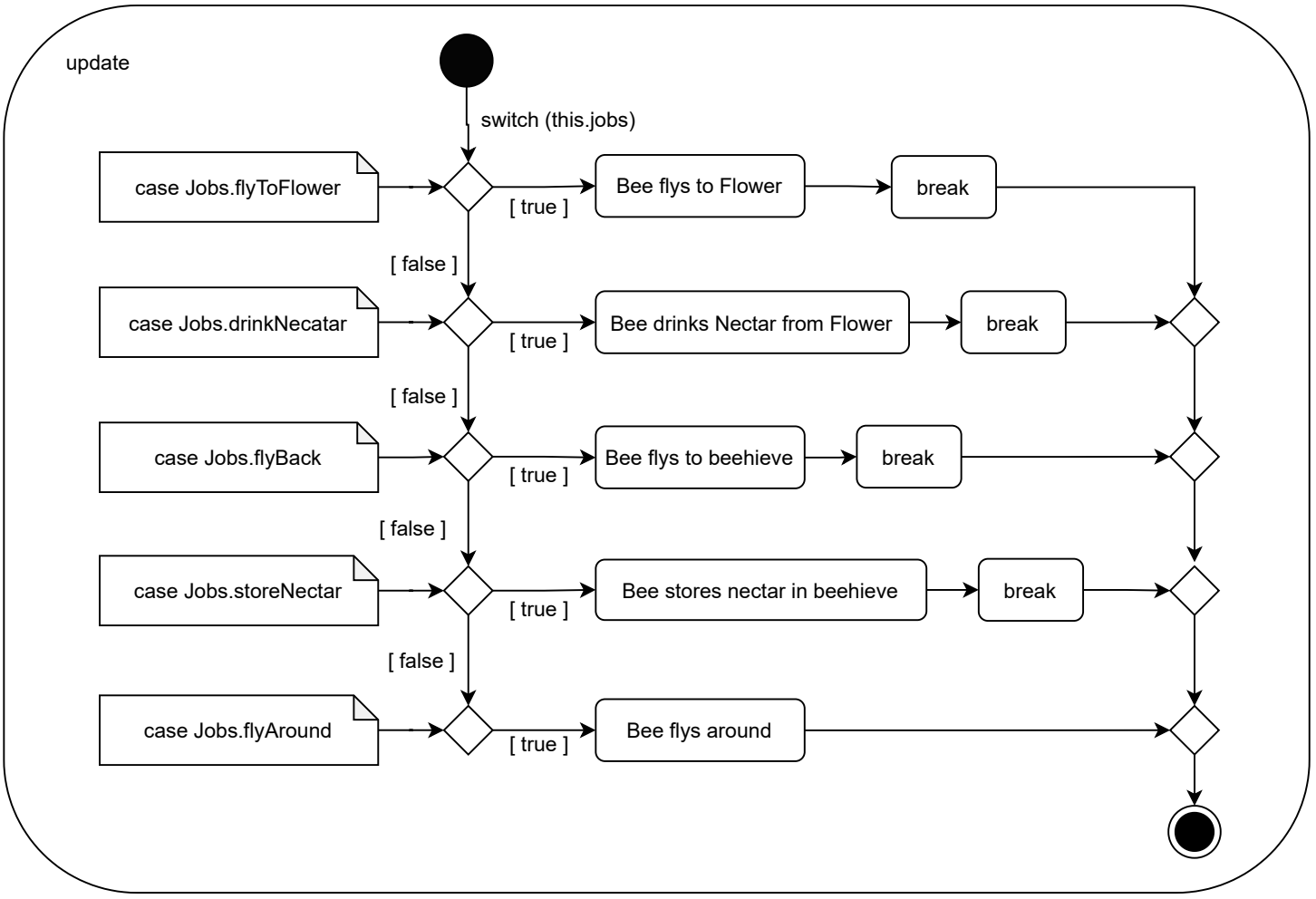

animate
requestAnimationFrame(animate);
crc2.clearRect(0, 0, crc2.canvas.width, crc2.canvas.height);
crc2.putImageData(imageData, 0, 0);
[index > moveables.length]
index ++
[index < moveables.length]
moveables[index].update();
[index > flowers.length]
index ++
[index < flowers.length]
flowers[index].update();
[indexFlower > flowers.lenght]
[indexFlower < flowers.lenght]
[flowers[indexFlower].assignFlower == true]
[flowers[indexFlower].assignFlower == false]
[flowers[indexFlower].nectar < 6]
[flowers[indexFlower].nectar > 6]
[index > bees.length]
[index < bees.length]
[bees[index].doingJobBee == true]
[bees[index].doingJobBee == false]
flowers[indexFlower].setAssign(true);
bees[index].setIndex(indexFlower);
bees[index].setJobTask(Jobs.flyToFlower);
break;
bees[index].update();
bees[index].draw();
[index < bees.length]
index ++
[index > bees.length]

Activitydiagram **BEE**

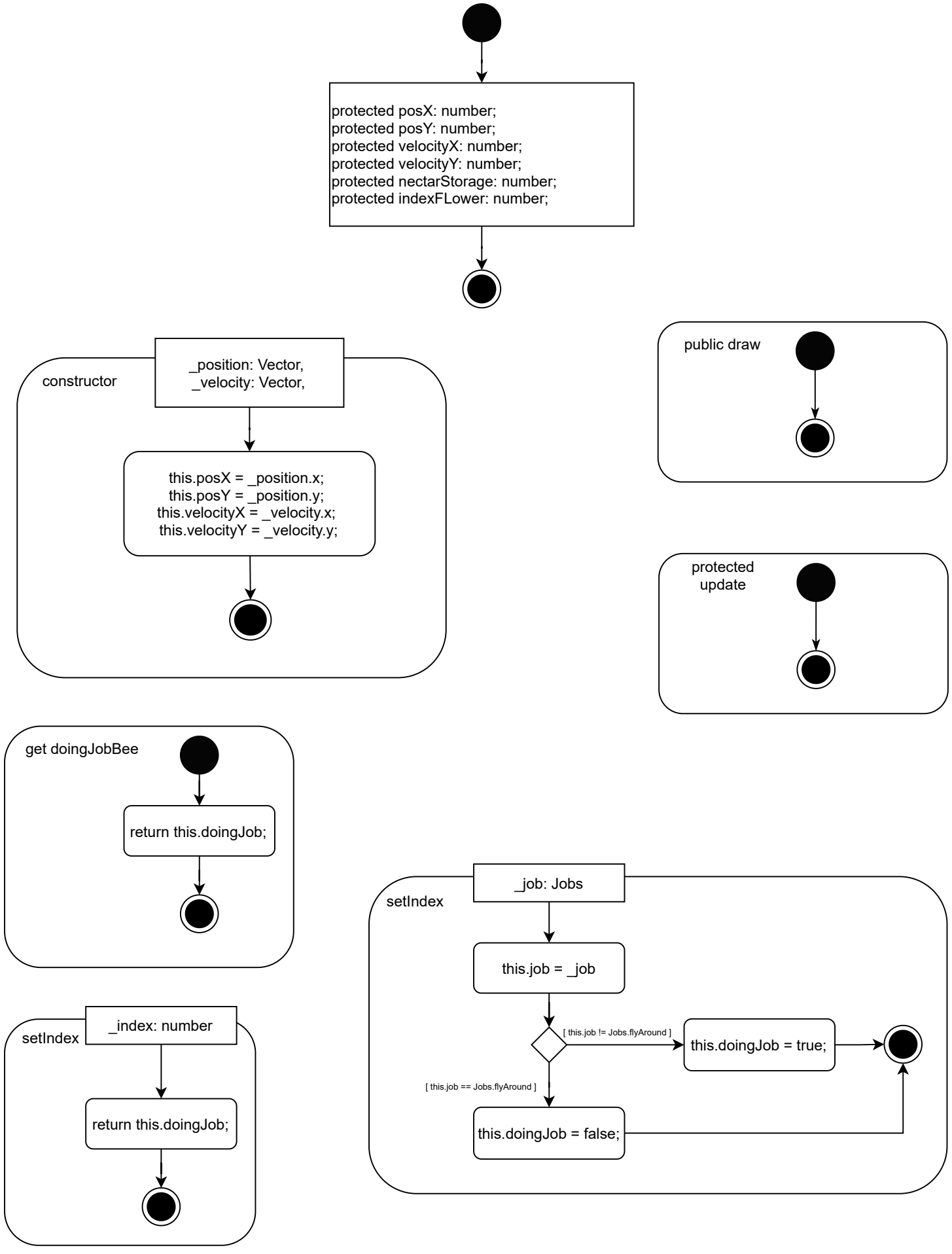




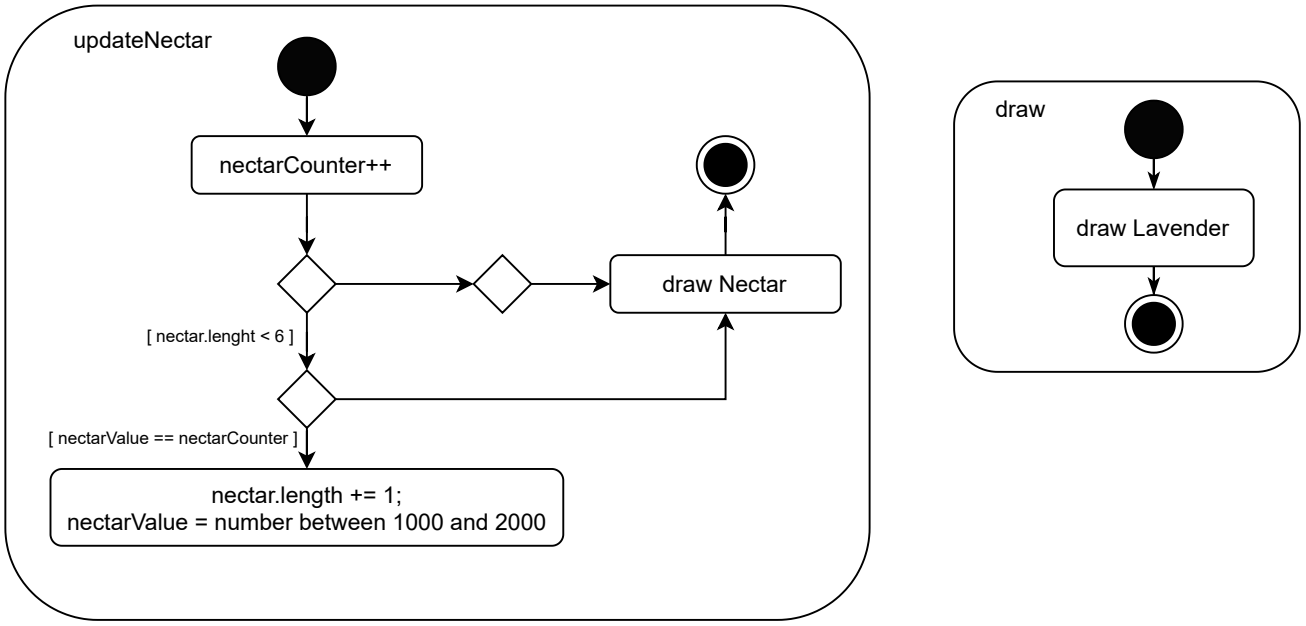



update
switch (this.jobs)
case Jobs.flyToFlower
[true]
Bee flys to Flower
break
[false]
case Jobs.drinkNecatar
[true]
Bee drinks Nectar from Flower
break
[false]
case Jobs.flyBack
[true]
Bee flys to beehieve
break
[false]
case Jobs.storeNectar
[true]
Bee stores nectar in beehieve
break
[false]
case Jobs.flyAround
[true]
Bee flys around

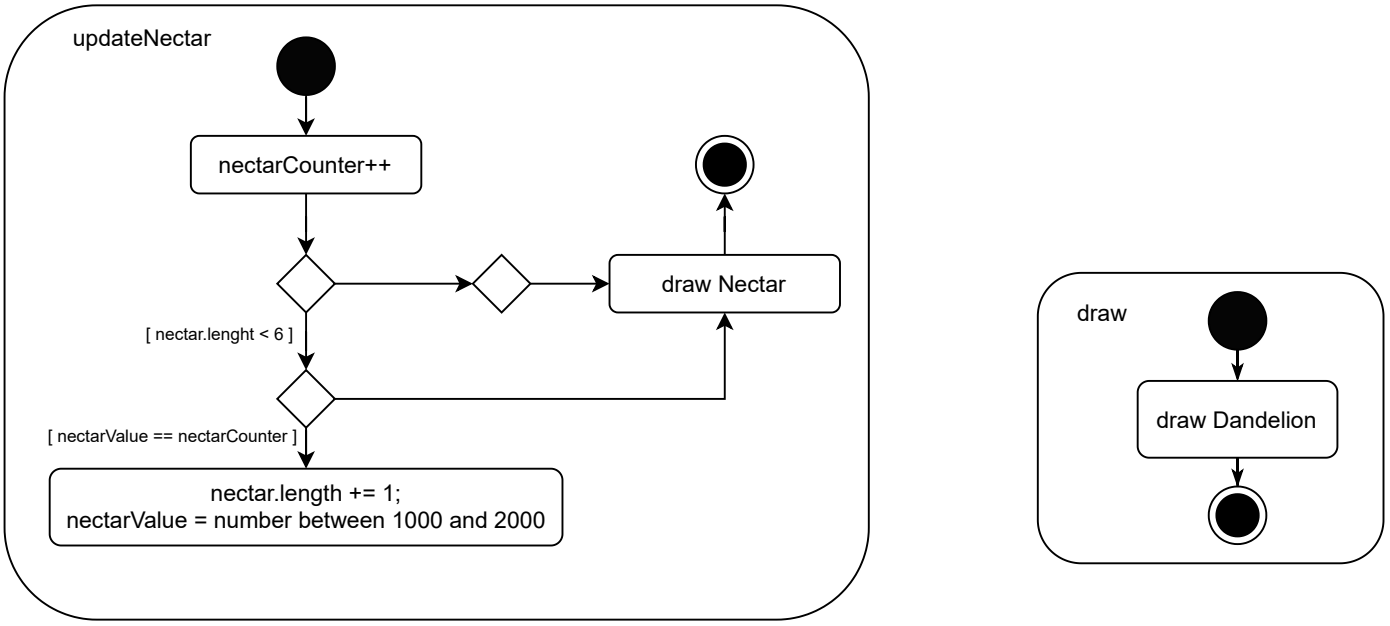
Activitydiagram **MOVEABLE**



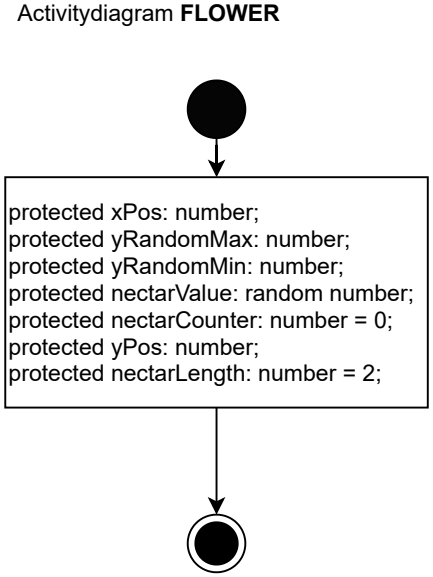
Activitydiagram **LAVENDER**

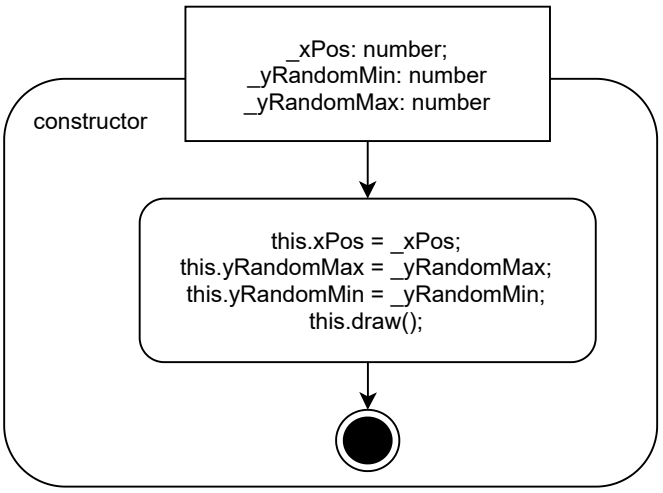


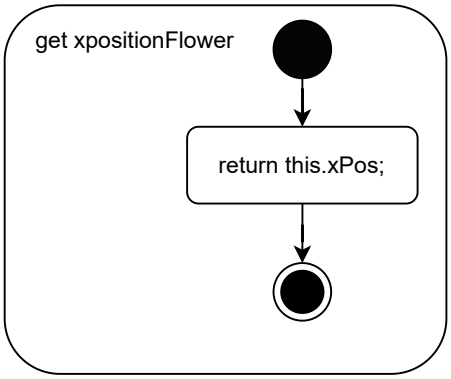
Activitydiagram **DANDELION**

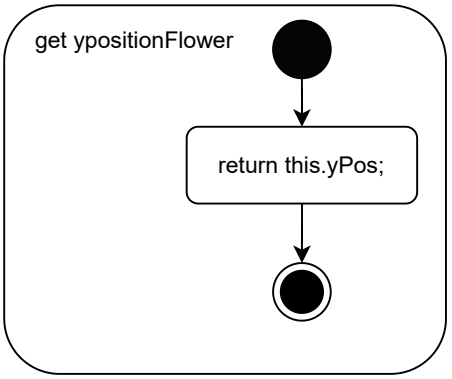


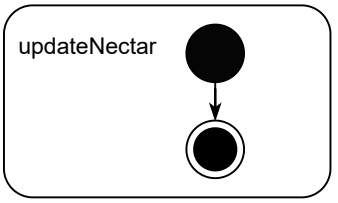
Activitydiagram **FLOWER**

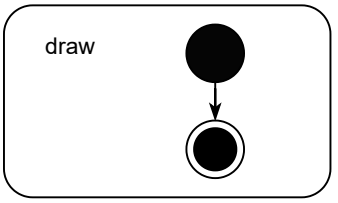


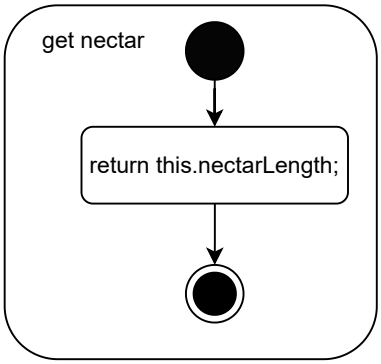


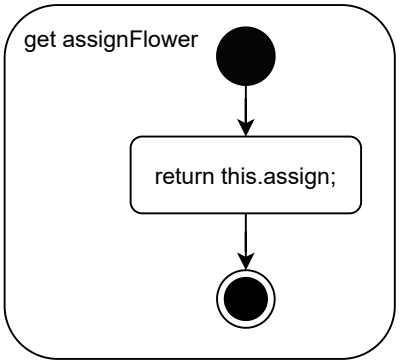


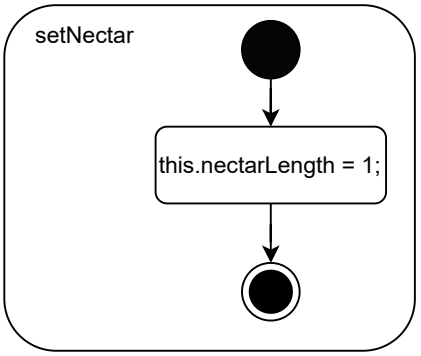


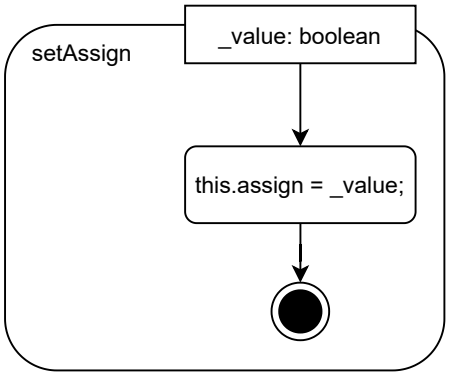




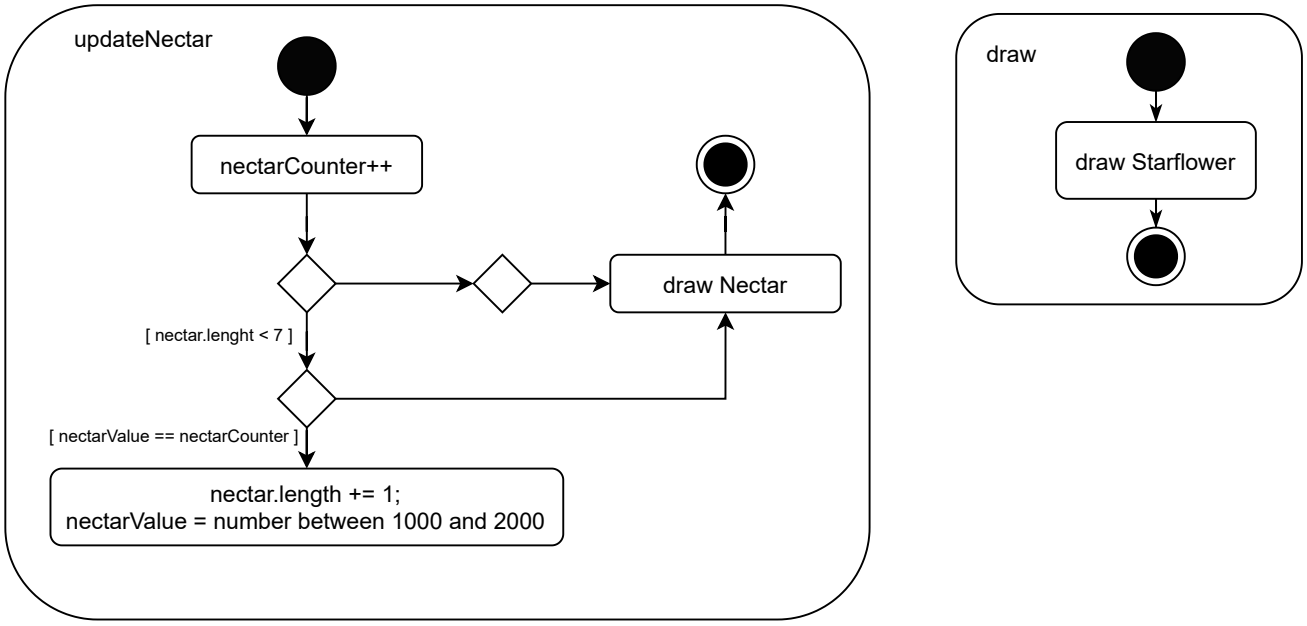




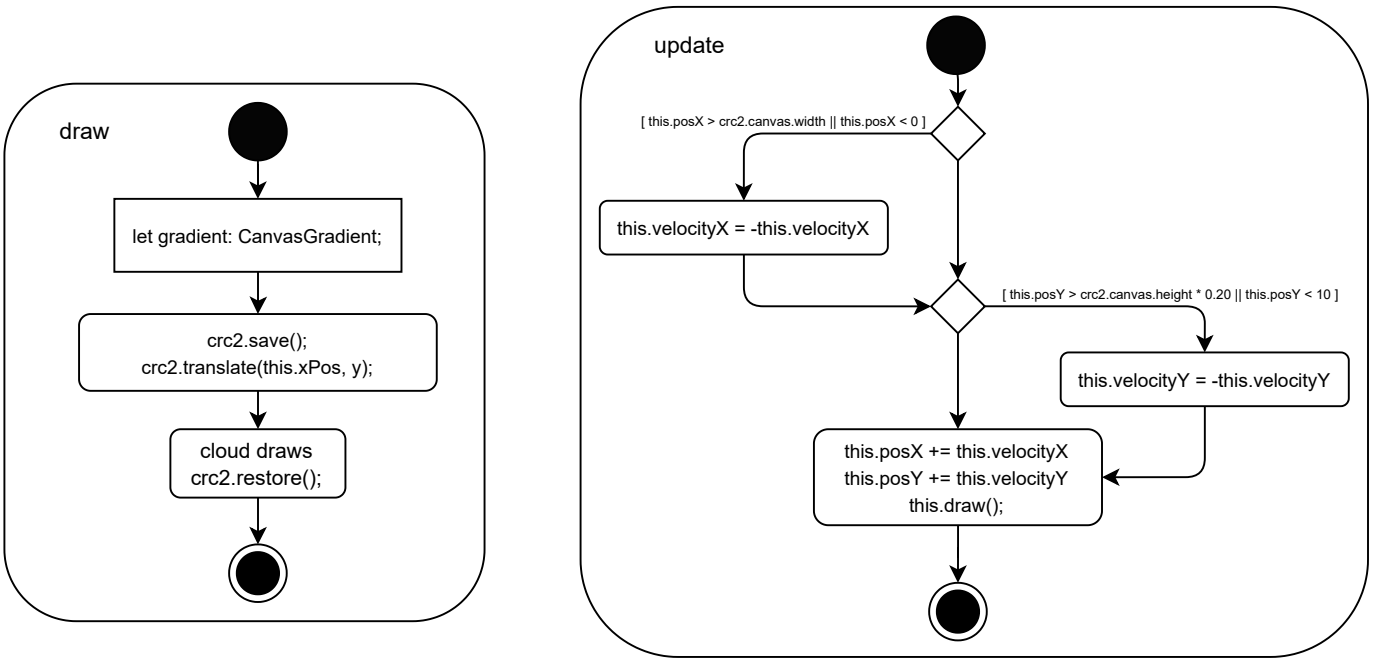




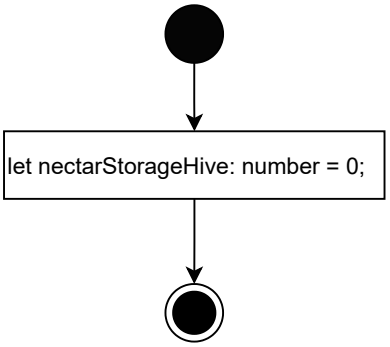
Activitydiagram **STARFLOWER**

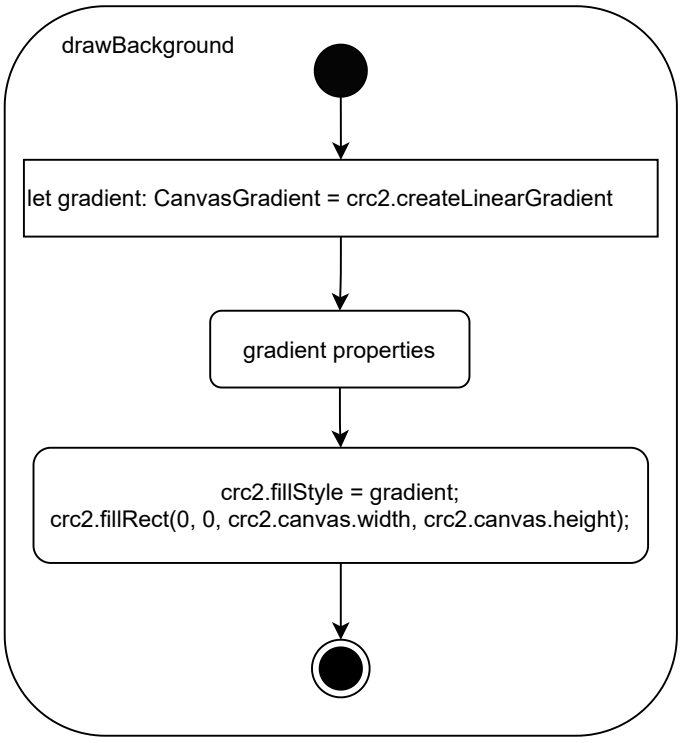


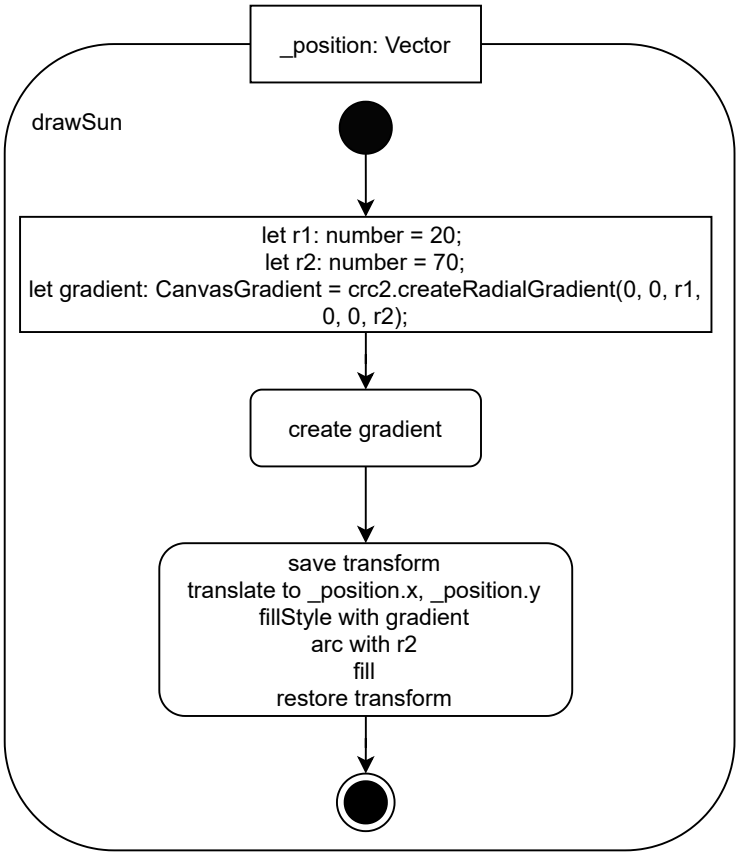
Activitydiagram **CLOUDS**

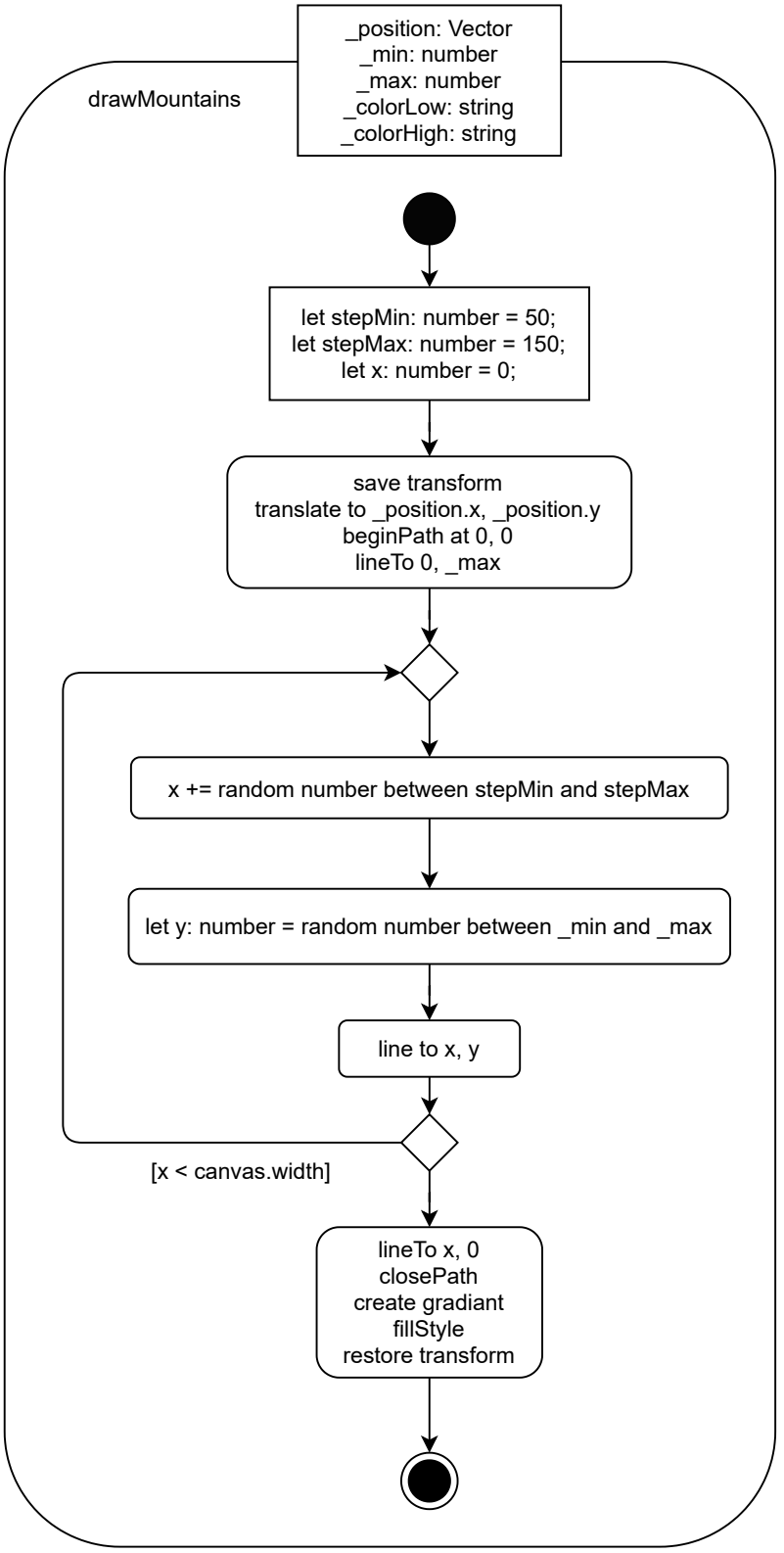


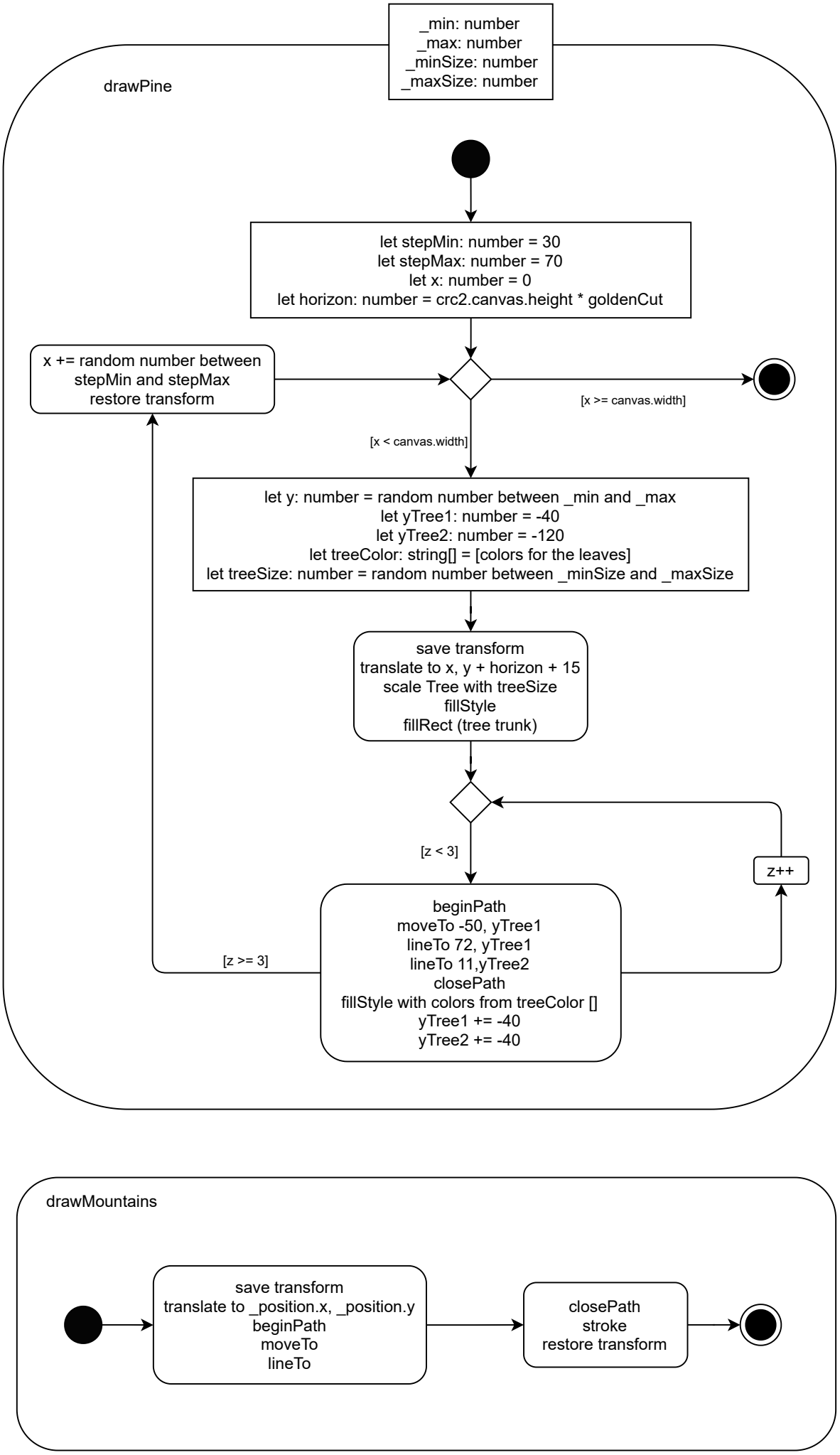
Activitydiagram **BACKGROUND**










_min: number
_max: number
_minSize: number
_maxSize: number
drawPine
let stepMin: number = 30
let stepMax: number = 70
let x: number = 0
let horizon: number = crc2.canvas.height * goldenCut
x += random number between
stepMin and stepMax
restore transform
[x >= canvas.width]
[x < canvas.width]
let y: number = random number between _min and _max
let yTree1: number = -40
let yTree2: number = -120
let treeColor: string[] = [colors for the leaves]
let treeSize: number = random number between _minSize and _maxSize
save transform
translate to x, y + horizon + 15
scale Tree with treeSize
fillStyle
fillRect (tree trunk)
[z < 3]
z++
beginPath
moveTo -50, yTree1
lineTo 72, yTree1
lineTo 11,yTree2
closePath
fillStyle with colors from treeColor []
yTree1 += -40
yTree2 += -40
[z >= 3]

drawMountains
save transform
translate to _position.x, _position.y
beginPath
moveTo
lineTo
closePath
stroke
restore transform