L09.2 class diagram

Flower
type: string; color: string; position: Vector; size: number; velocity: Vector;

constructor(): void;
drawTulip(): void;
drawDaisy(): void;
drawRose(): void;
drawStarflower(): void;
move(_timeslice:number): void;

Daisy
type: string; color: string; position: Vector; size: number; velocity: Vector;

constructor(): void;
drawTulip(): void;
drawDaisy(): void;
drawRose(): void;
drawStarflower(): void;
move(_timeslice:number): void;

Rose
type: string; color: string; position: Vector; size: number; velocity: Vector;

constructor(): void;
drawTulip(): void;
drawDaisy(): void;
drawRose(): void;
drawStarflower(): void;
move(_timeslice:number): void;

StarFlower
type: string; color: string; position: Vector; size: number; velocity: Vector;

constructor(): void;
drawTulip(): void;
drawDaisy(): void;
drawRose(): void;
drawStarflower(): void;
move(_timeslice:number): void;

Cloud
position: Vector; velocity: Vector;

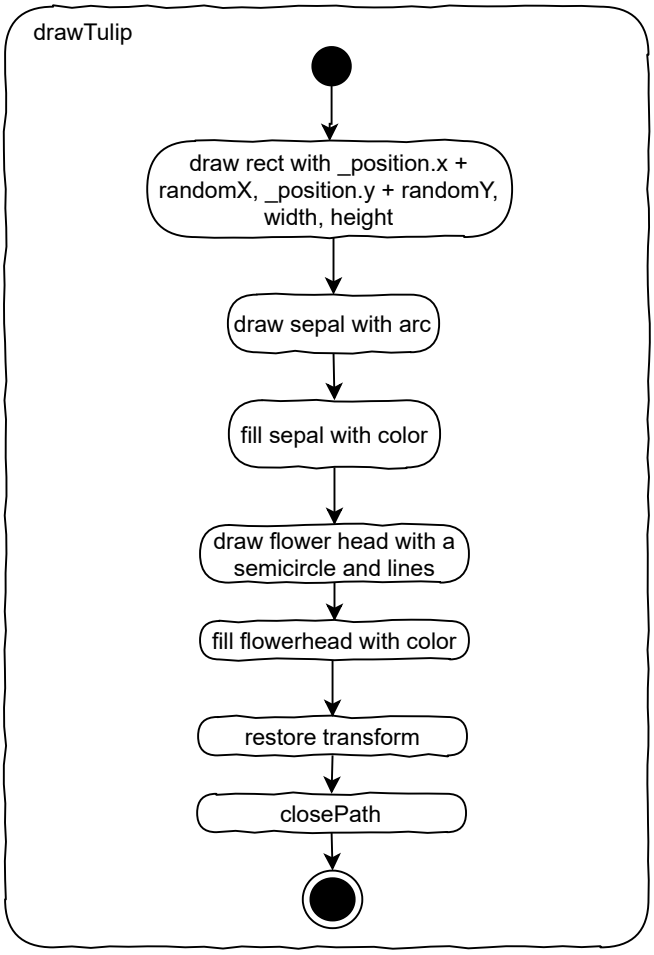
constructor(): void;
draw(): void;
moce(_timeslice:number): void;

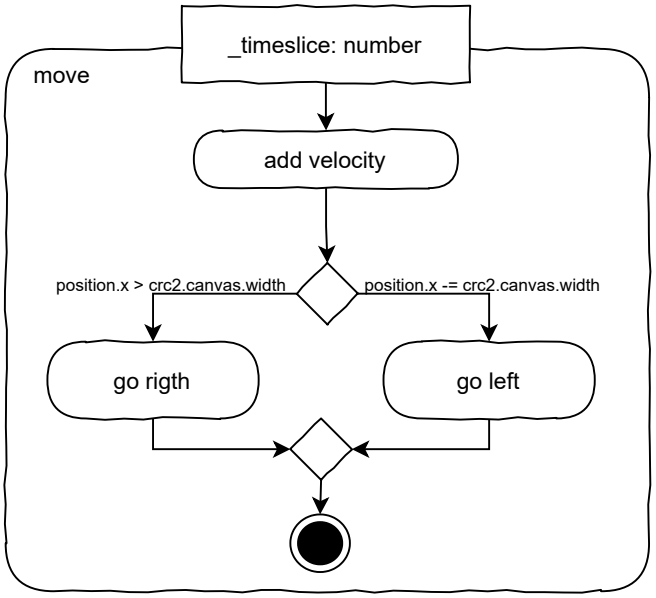
Bee
position: Vector; velocity: Vector;

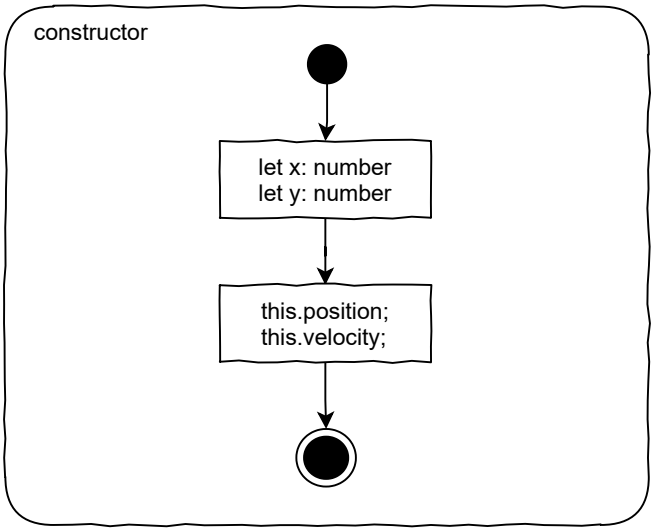
constructor();
draw(): void;
moce(_timeslice:number): void;

Vector
x: number; y: number;
constructor(); set() scale() add()

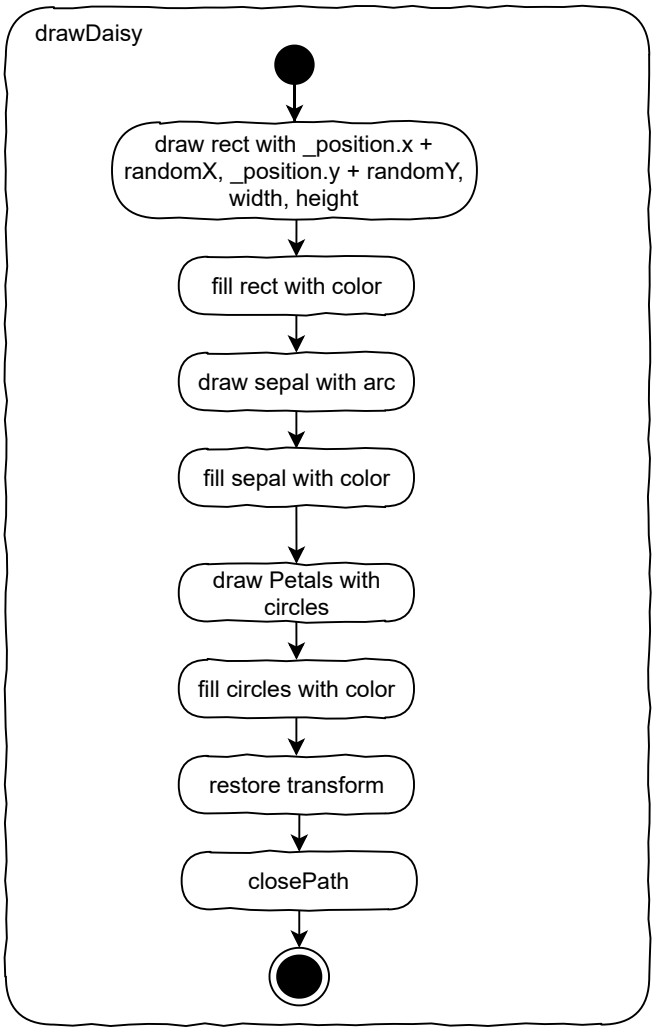
Activity Diagram Flower

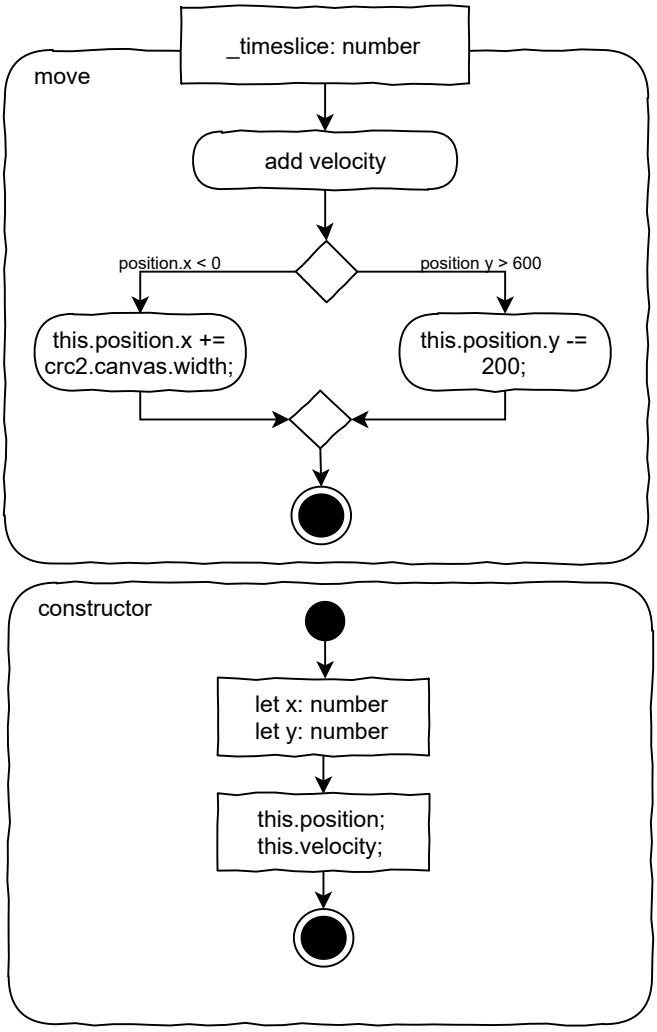




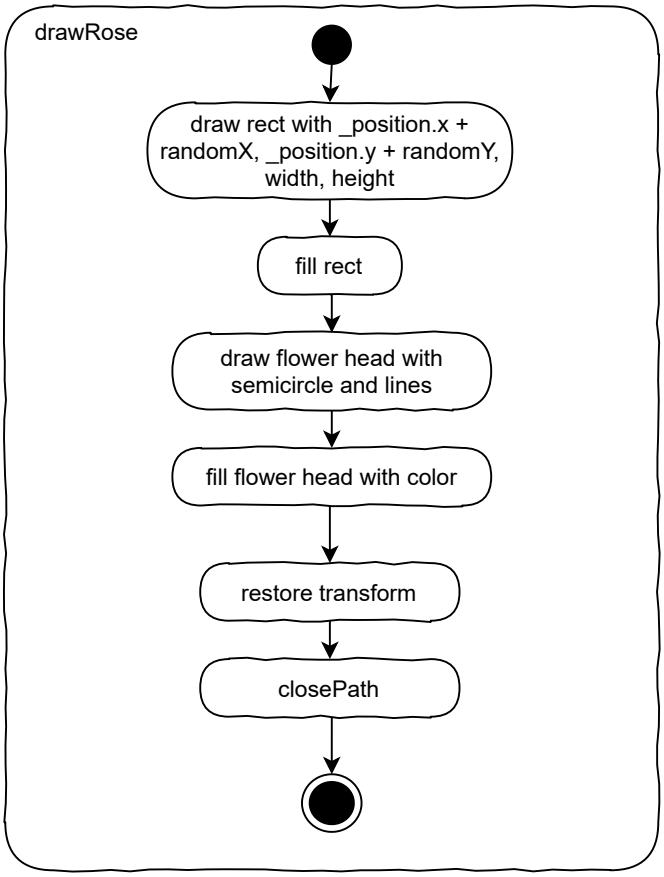


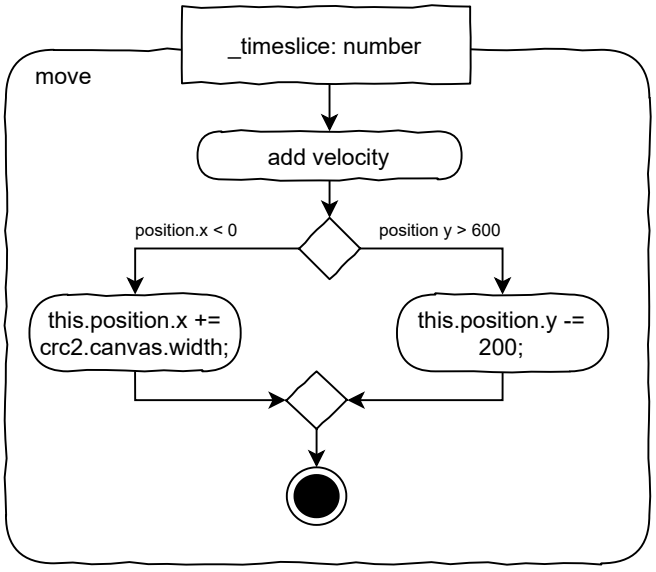
Activity Diagram Daisy

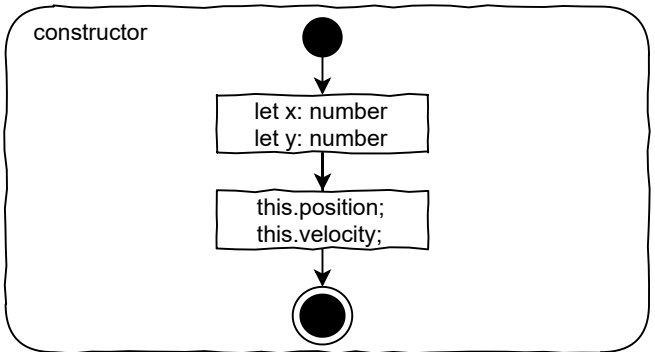




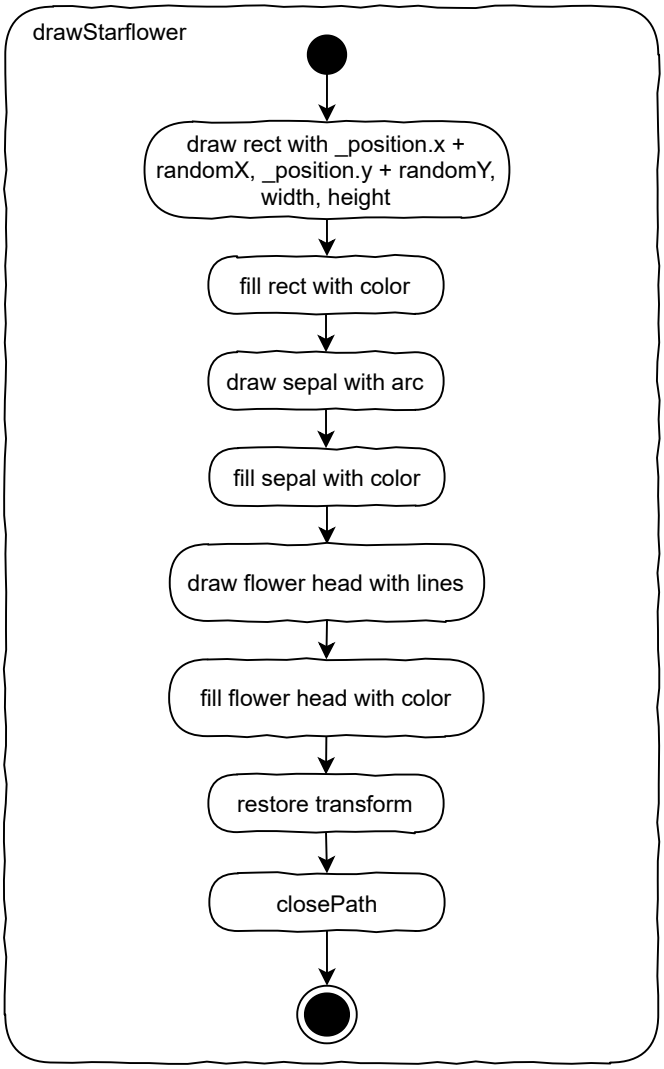
Activity Diagram Rose

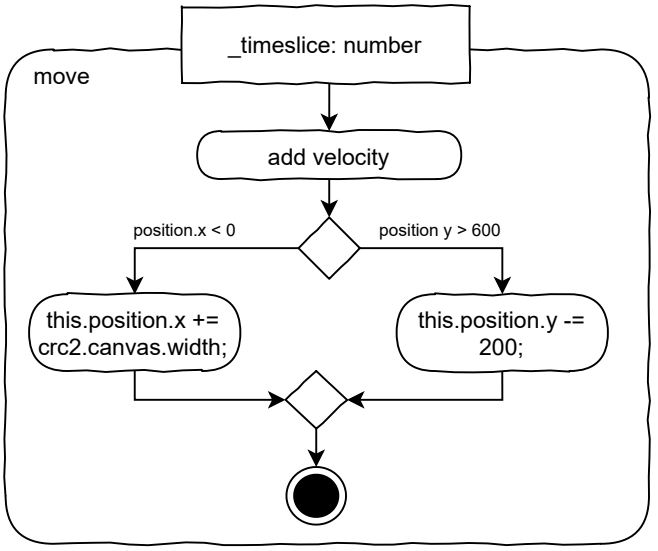


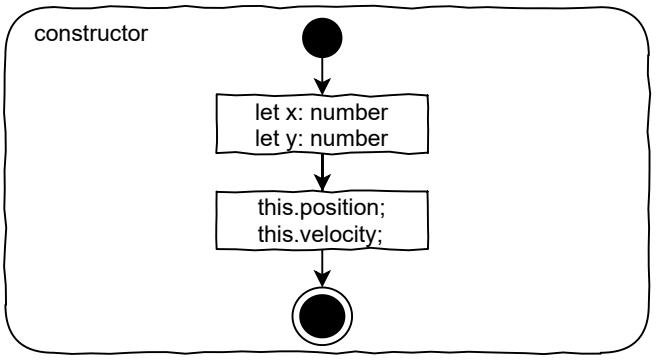




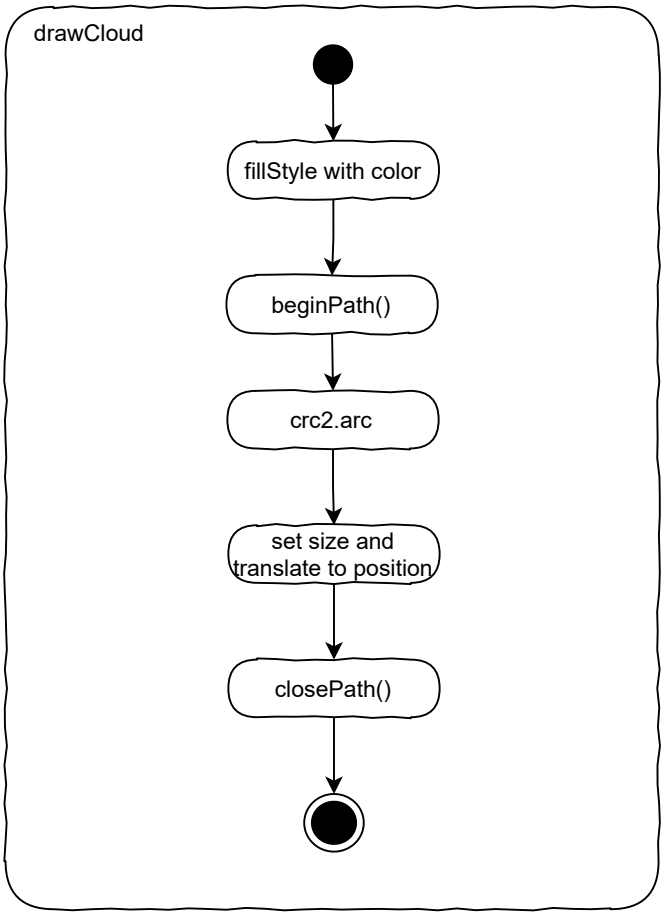
Activity Diagram StarFlower

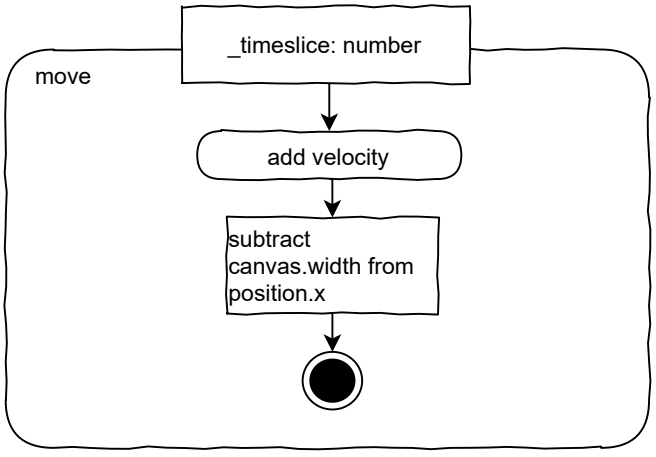


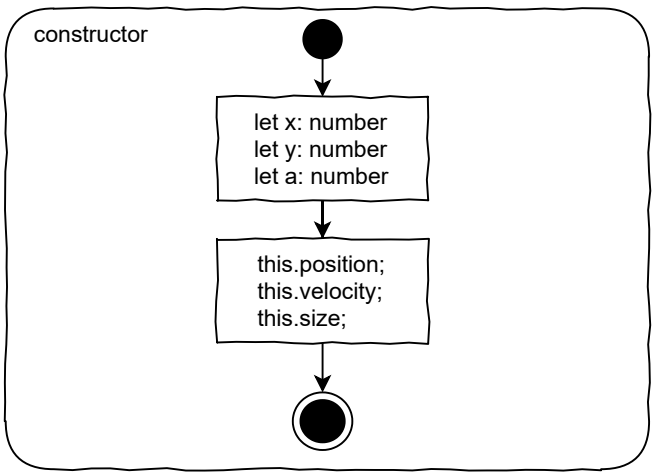




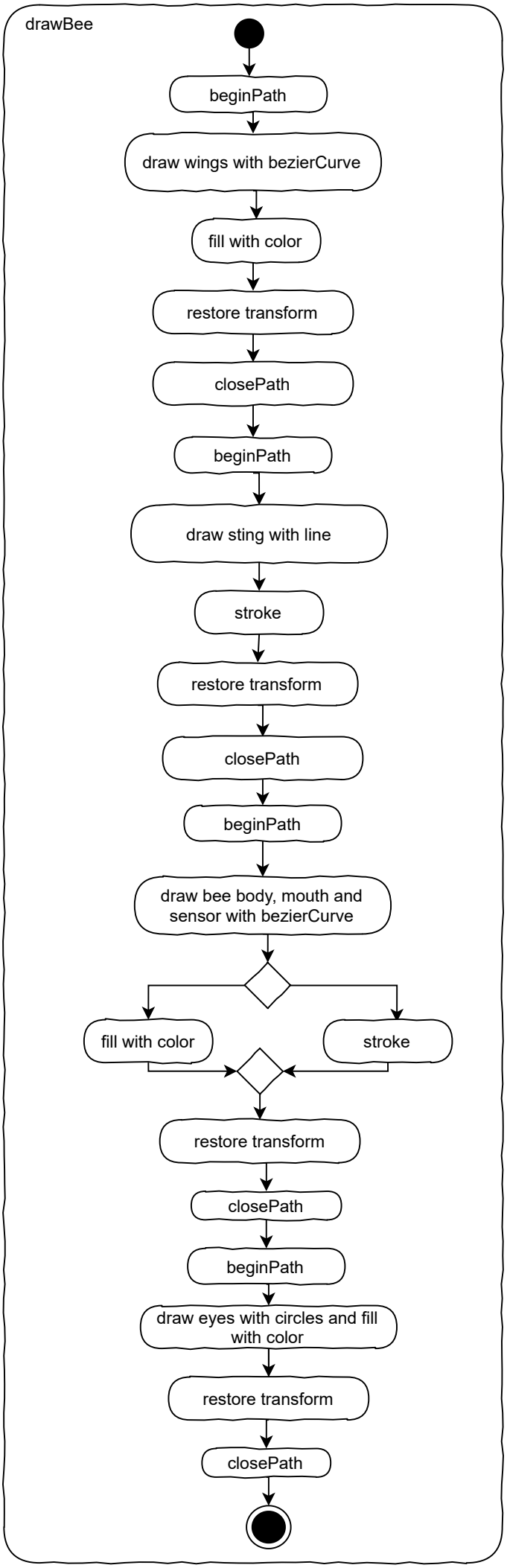
Activity Diagram Cloud

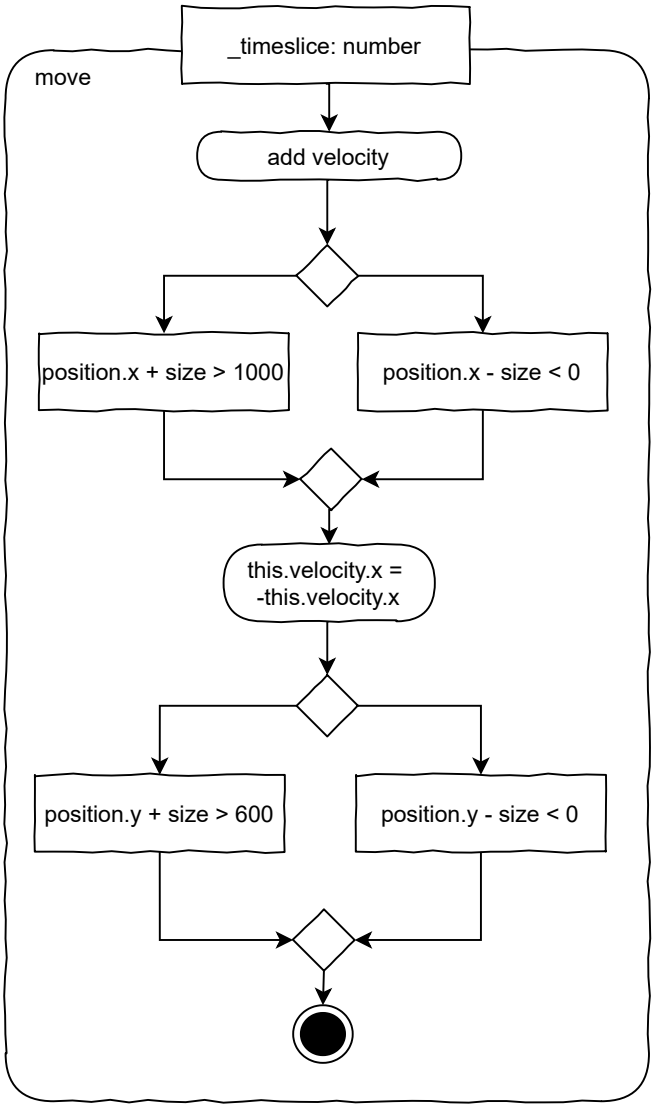


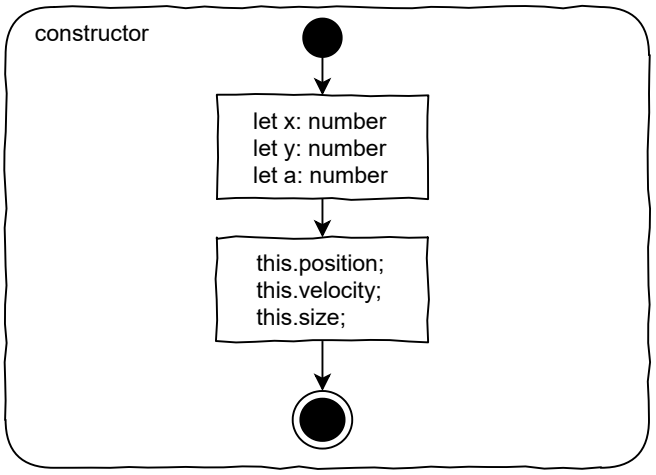




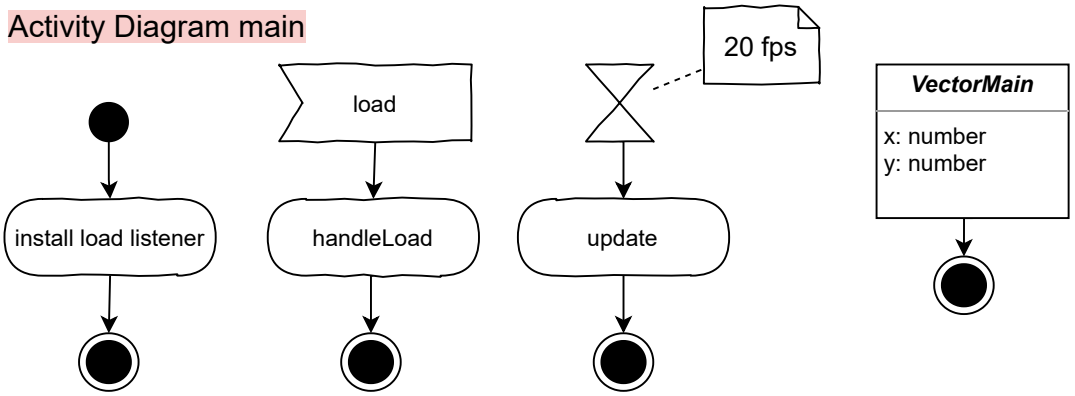
Activity Diagram Bee

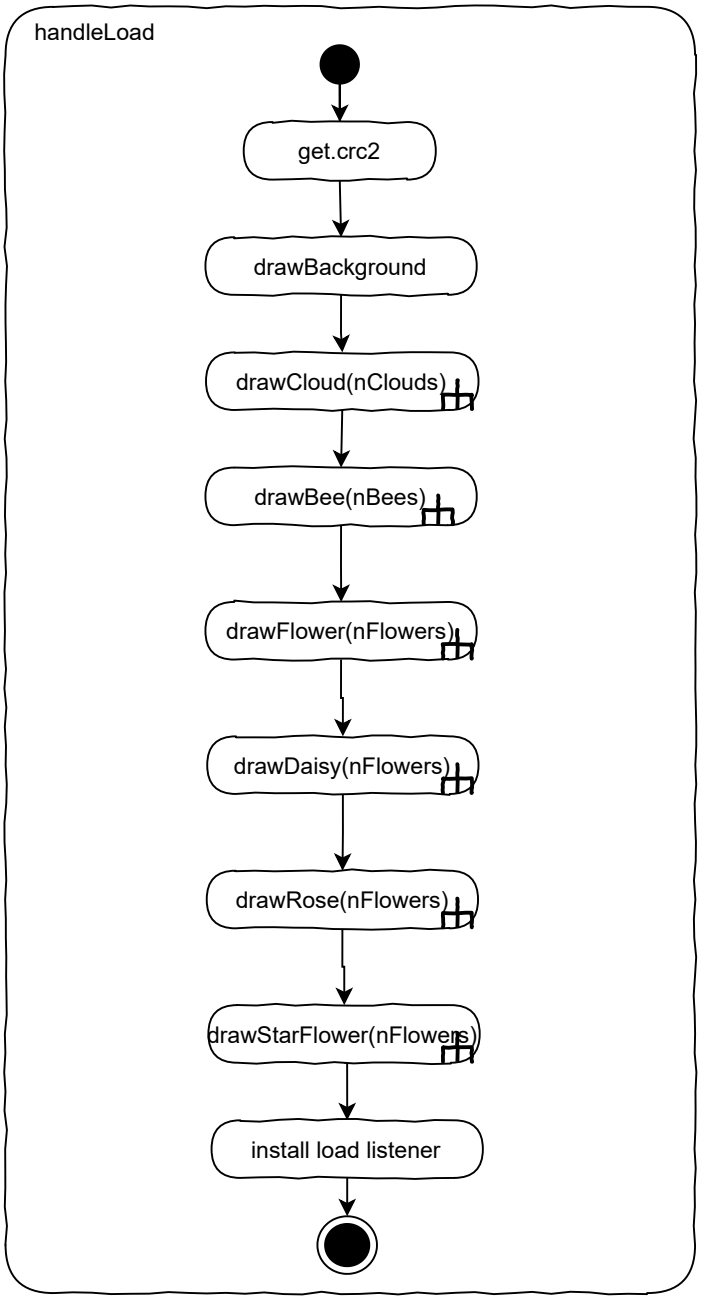


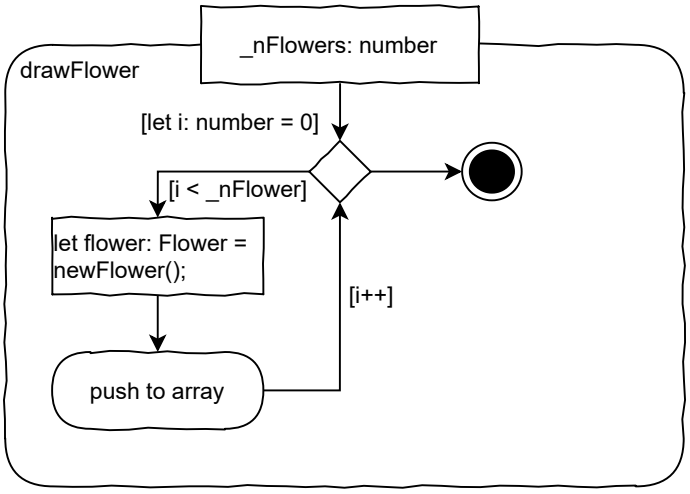


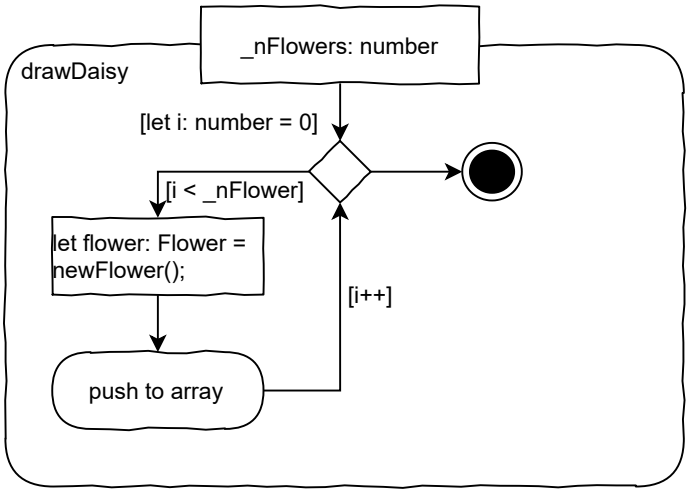


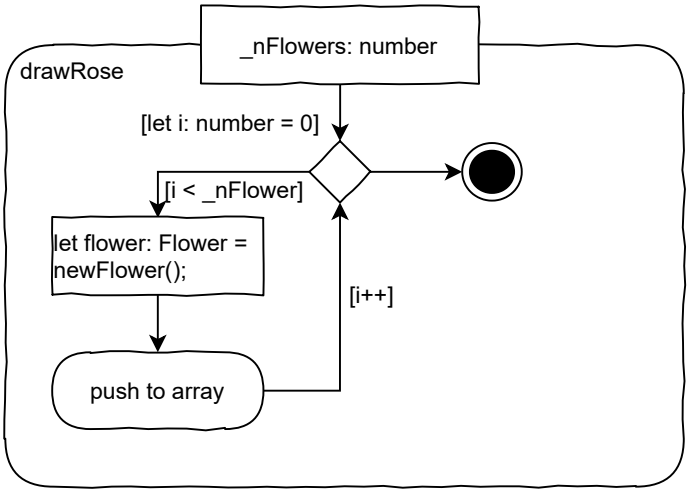
Activity Diagram main

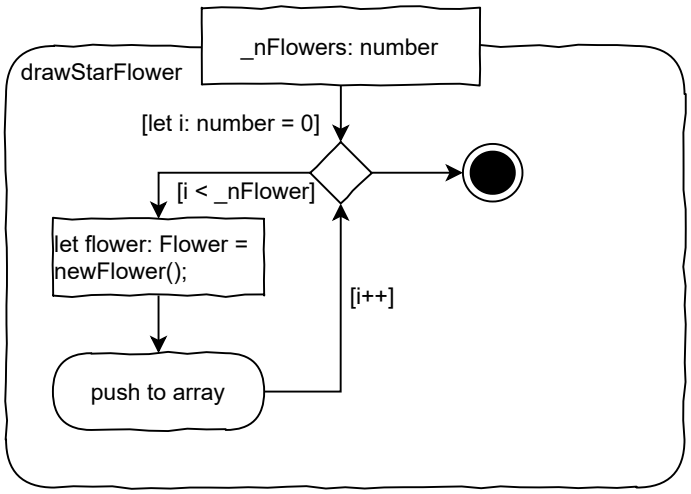


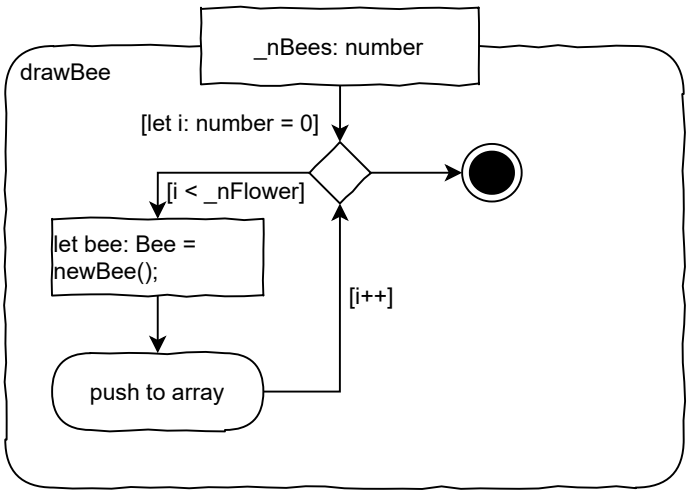


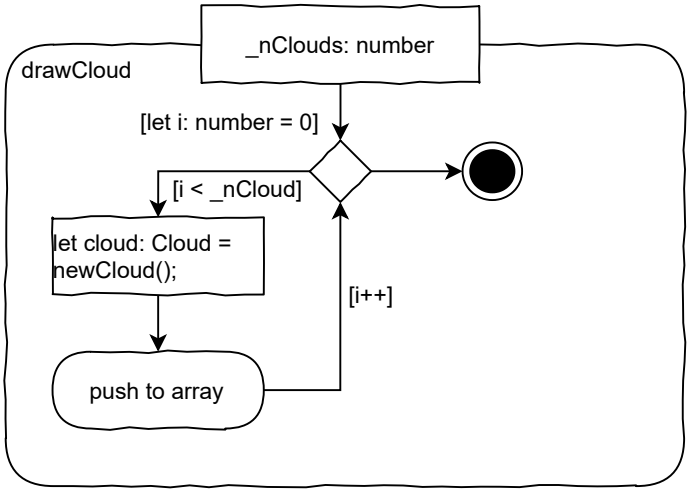


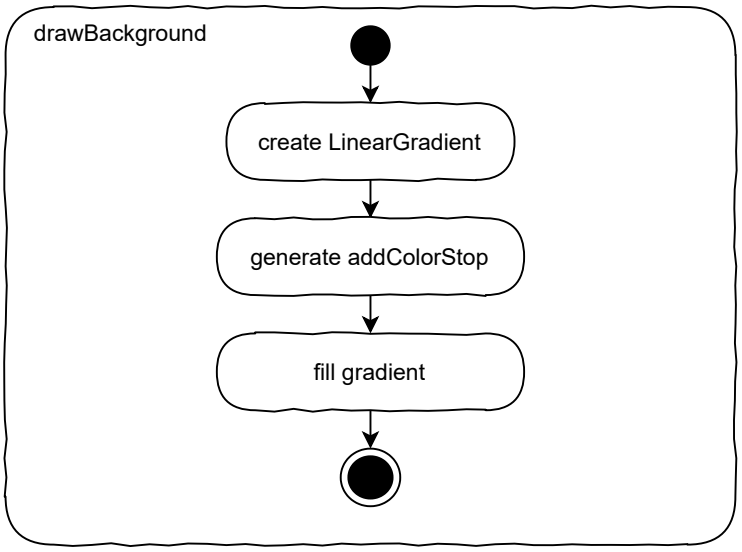


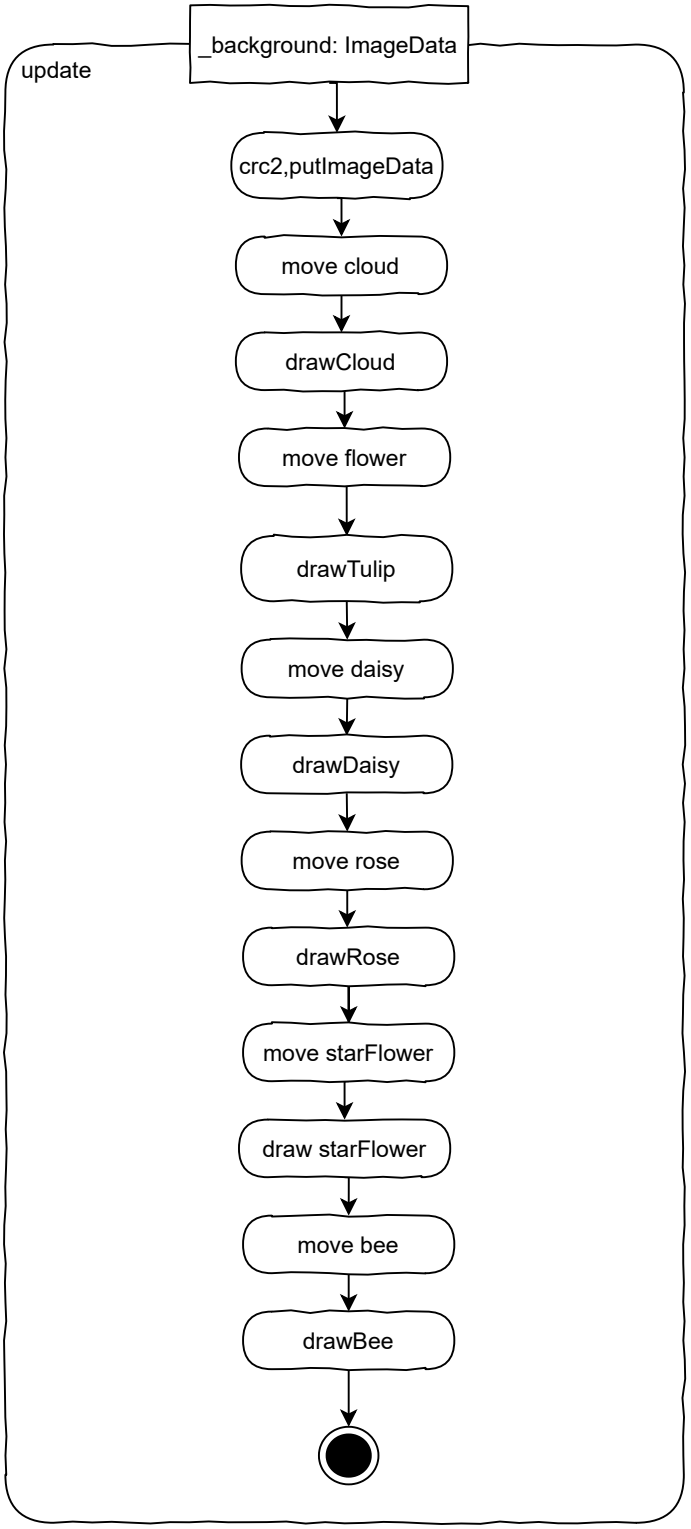


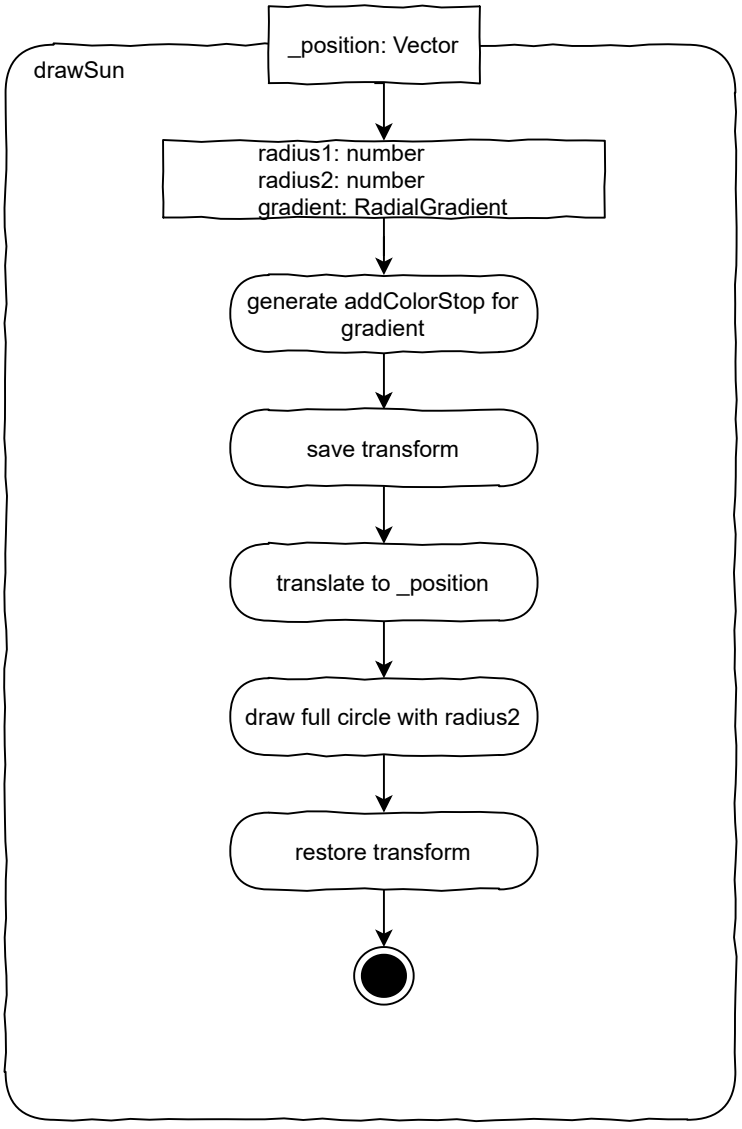

drawStarFlower
_nFlowers: number
[let i: number = 0]
[i < _nFlower]
let flower: Flower =
newFlower();
push to array
[i++]

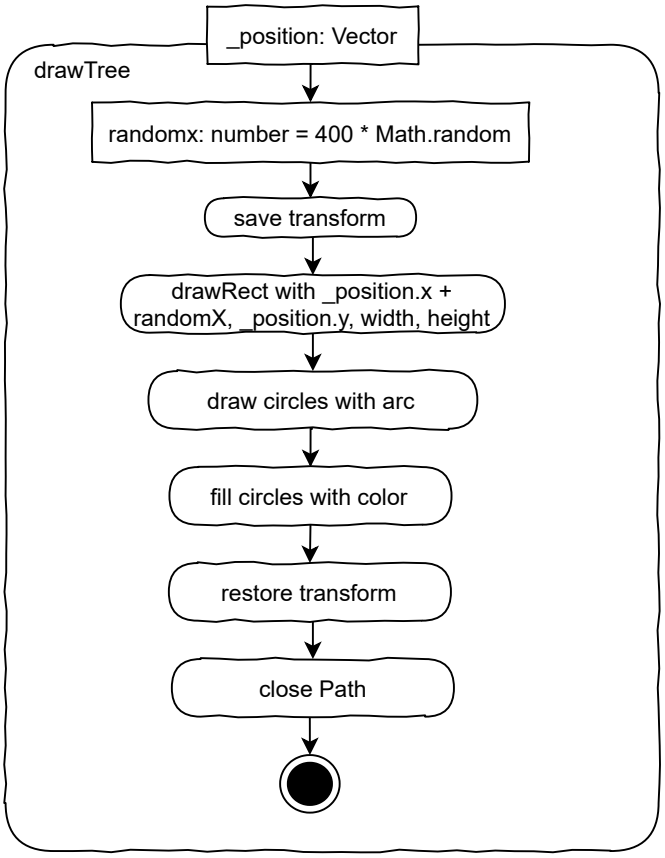

drawBee
_nBees: number
[let i: number = 0]
[i < _nFlower]
let bee: Bee =
newBee();
push to array
[i++]

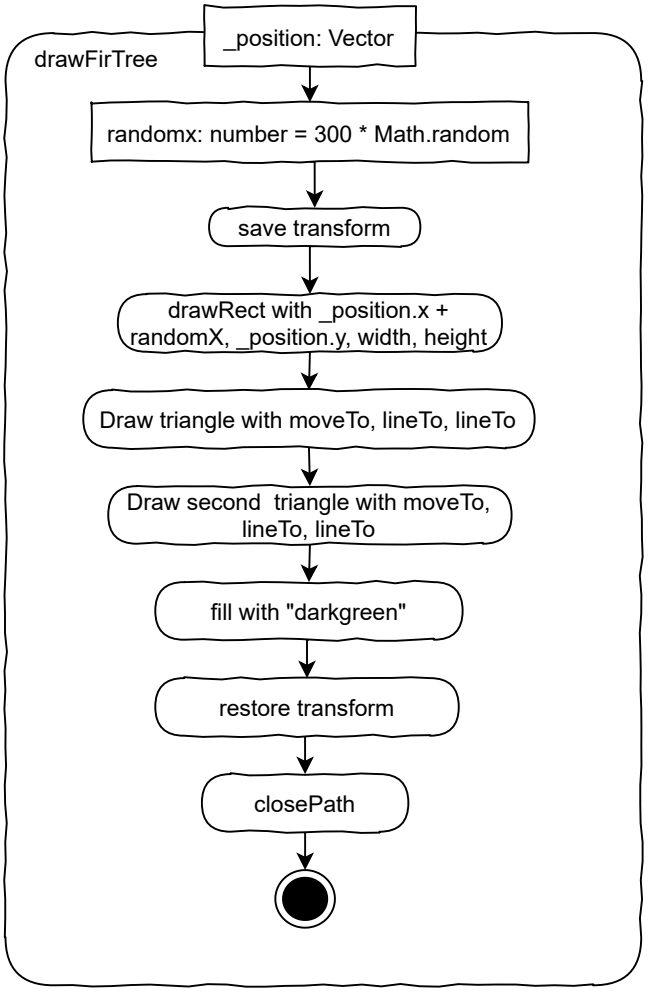

drawCloud
_nClouds: number
[let i: number = 0]
[i < _nCloud]
let cloud: Cloud =
newCloud();
push to array
[i++]

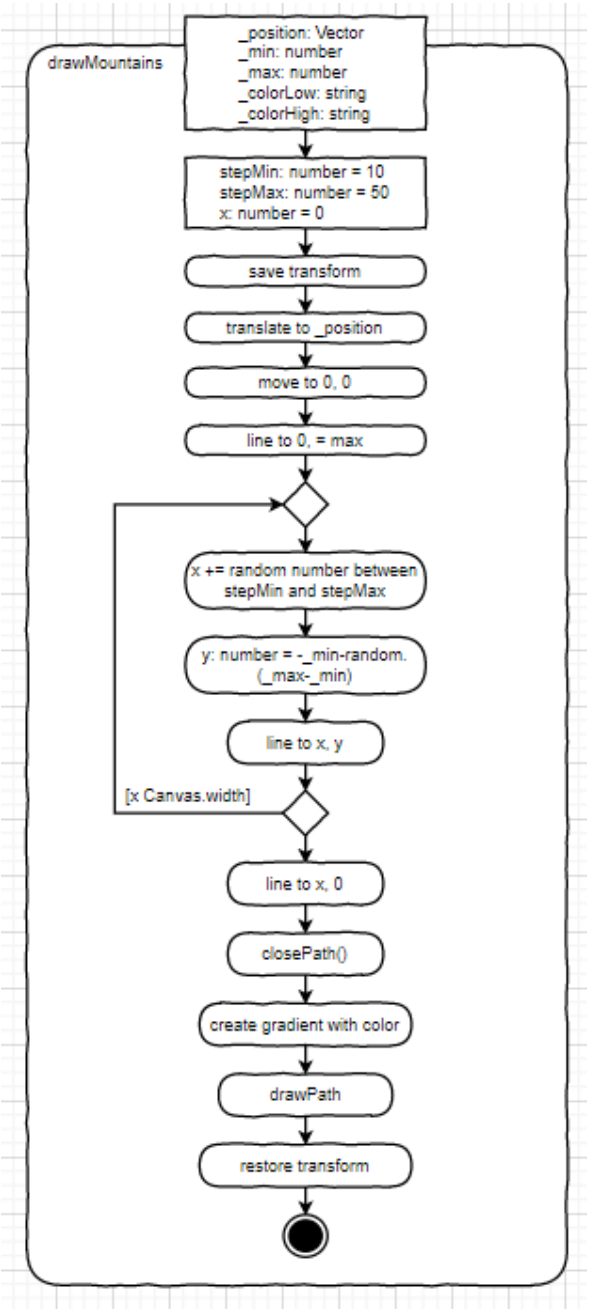

drawBackground
create LinearGradient
generate addColorStop
fill gradient


update
_background: ImageData
crc2,putImageData
move cloud
drawCloud
move flower
drawTulip
move daisy
drawDaisy
move rose
drawRose
move starFlower
draw starFlower
move bee
drawBee

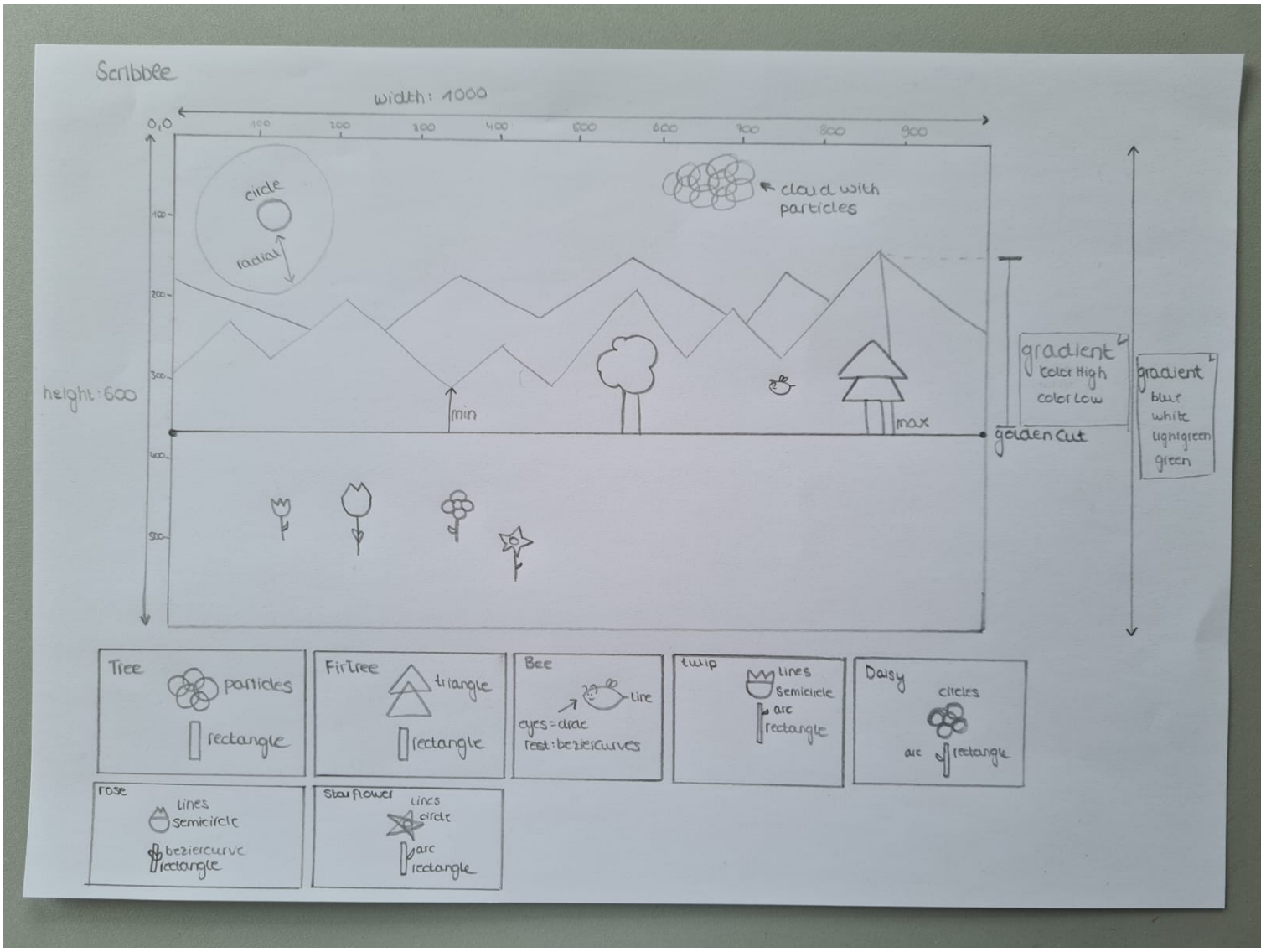

drawSun
_position: Vector
radius1: number
radius2: number
gradient: RadialGradient
generate addColorStop for gradient
save transform
translate to _position
draw full circle with radius2
restore transform


drawTree
_position: Vector
randomx: number = 400 * Math.random
save transform
drawRect with _position.x + randomX, _position.y, width, height
draw circles with arc
fill circles with color
restore transform
close Path


drawFirTree
_position: Vector
randomx: number = 300 * Math.random
save transform
drawRect with _position.x + randomX, _position.y, width, height
Draw triangle with moveTo, lineTo, lineTo
Draw second triangle with moveTo, lineTo, lineTo
fill with "darkgreen"
restore transform
closePath


drawMountains
_position: Vector
_min: number
_max: number
_colorLow: string
_colorHigh: string
stepMin: number = 10
stepMax: number = 50
x: number = 0
save transform
translate to _position
move to 0, 0
line to 0, = max
x += random number between stepMin and stepMax
y: number = -_min-random.(_max-_min)
line to x, y
[x Canvas.width]
line to x, 0
closePath()
create gradient with color
drawPath
restore transform

UI-Scribble


Scribbee
width: 1000
0,0
100
200
300
400
500
600
700
800
900
height: 600
circle
radius
cloud with particles
min
max
golden cut
gradient
color High
color low
gradient
blue
white
lightgreen
green
Tree
particles
rectangle
FirTree
triangle
rectangle
Bee
line
eyes = circle
rest: beziercurves
tulip
lines
semicircle
arc
rectangle
Daisy
circles
arc
rectangle
rose
lines
semicircle
beziercurve
rectangle
Starflower
lines
circle
arc
rectangle