Class Diagram: Blumenwiese

CanvasRenderingContext

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

constructor();
drawFlower(): void;
move(_timeslice: number): void;
getRandomColor();
getRandomYellowColor();

Cloud
position: Vector; velocity: Vector; size: number; color: string

constructor();
drawCloud(): void;
move(_timeslice: number): void;

Bee
position: Vector; velocity: Vector; color: string colorwings: string size: number;

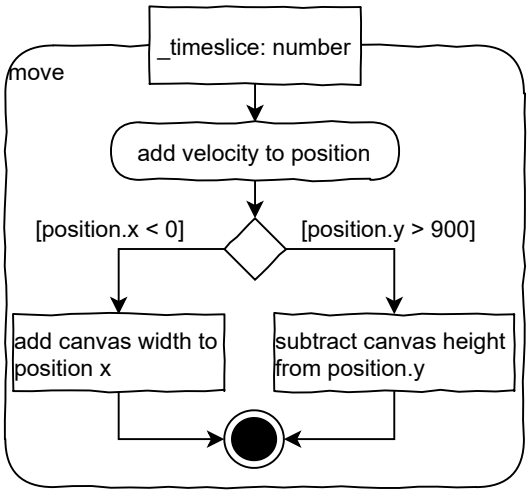
constructor(;
drawBee(): void;
move(_timeslice: number): void;

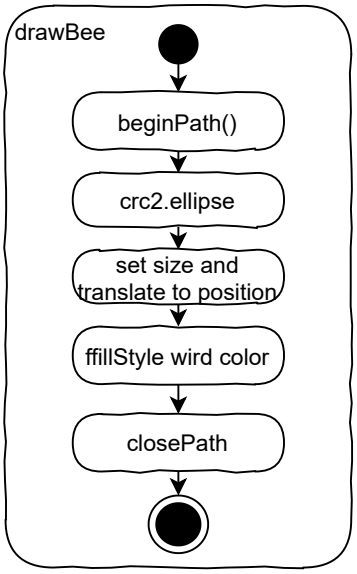
Sunflower
color: string; position: Vector; size: number; velocity: Vector;

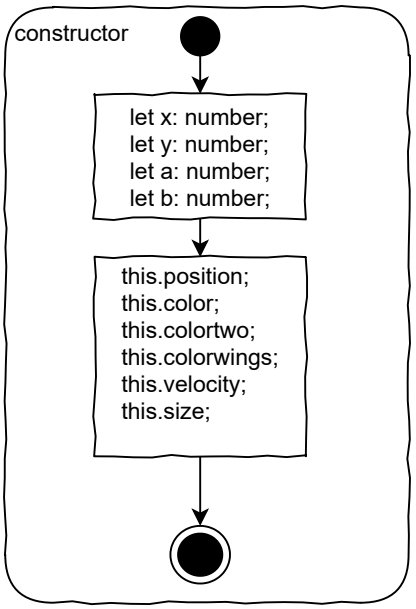
constructor();
drawSunflower(): void;
move(_timeslice: number): void;

Vector
x: number; y: number;
constructor(_x: number, _y: number set(_x: number, _y: number) scale(_factor: number) add(_addend: Vector)

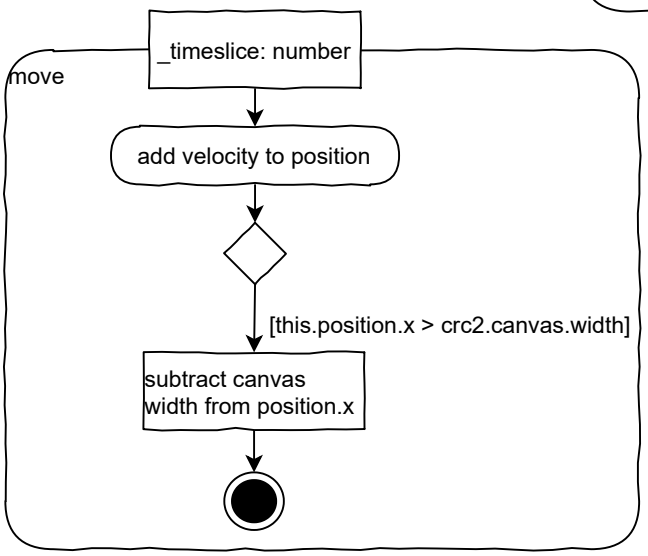
Activity Diagram Bee

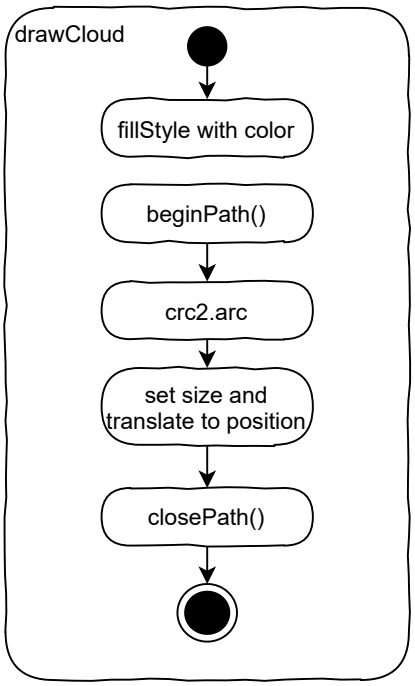


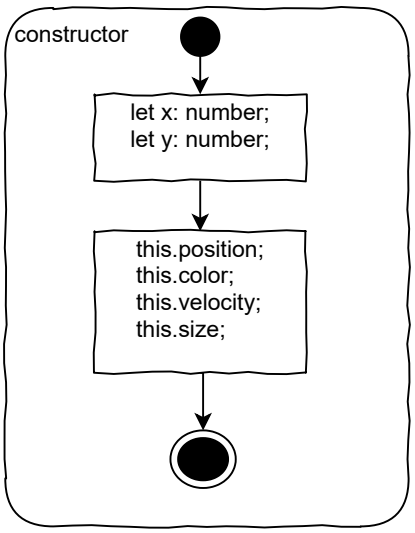




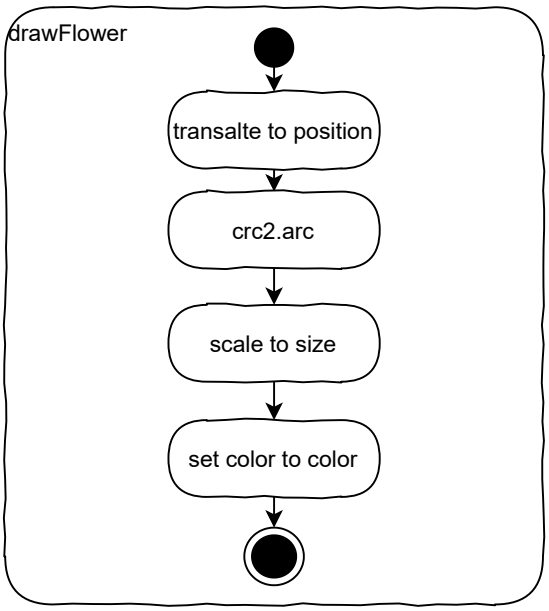
Activity Diagram Cloud

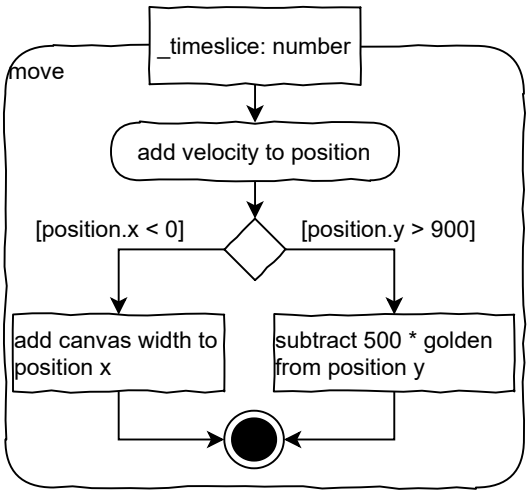


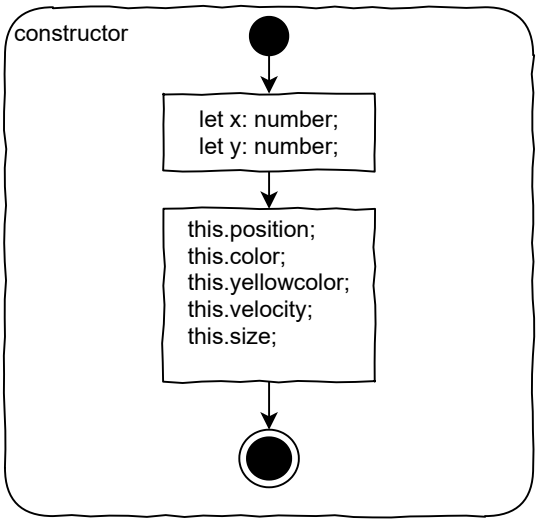


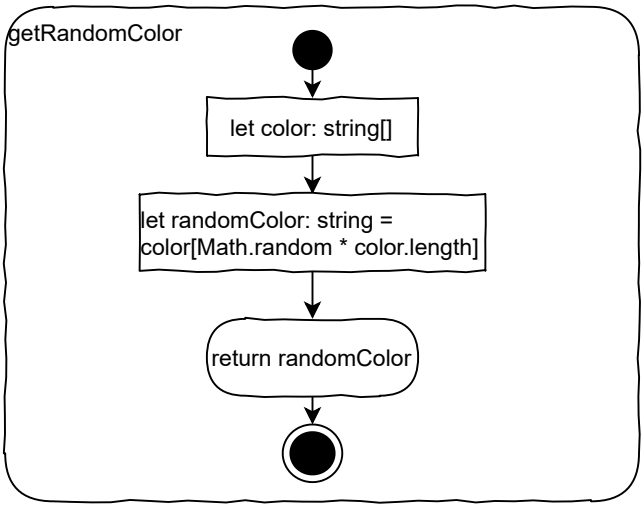


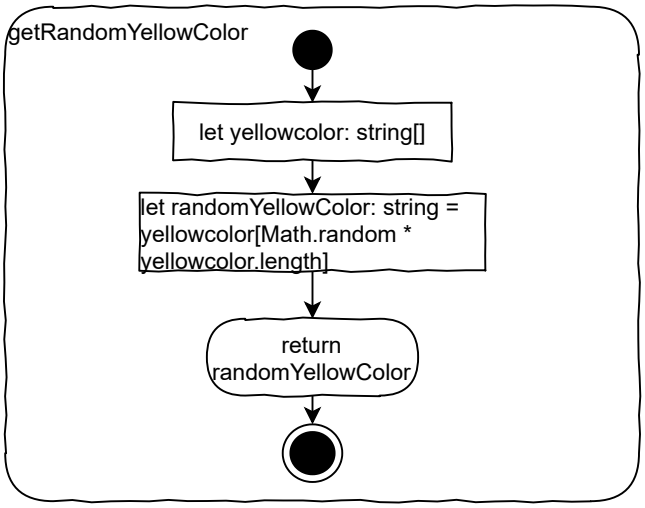
Activity Diagram Flower

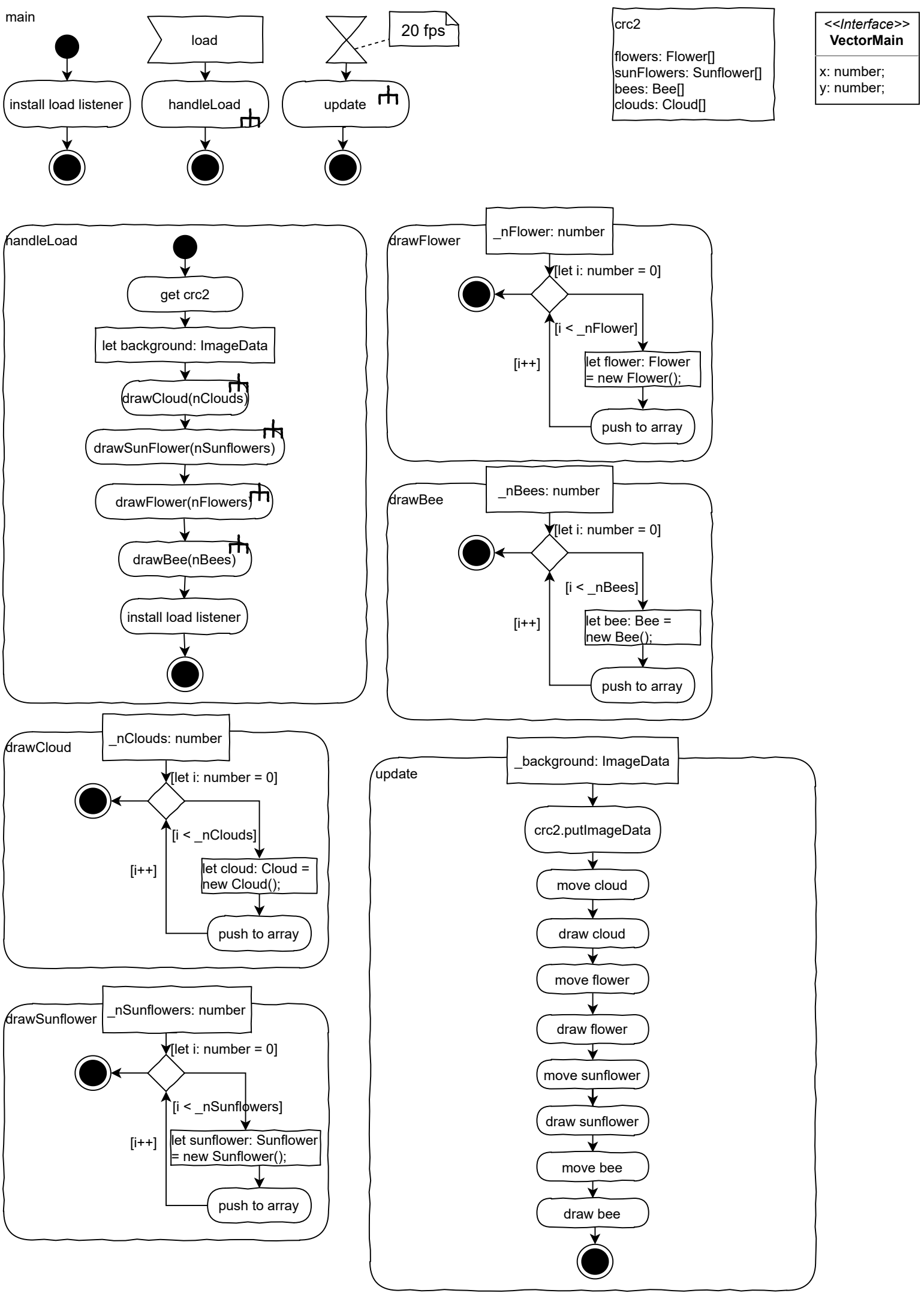











main
load
20 fps
install load listener
handleLoad
update
crc2
flowers: Flower[]
sunFlowers: Sunflower[]
bees: Bee[]
clouds: Cloud[]
<<Interface>>
VectorMain
x: number;
y: number;
handleLoad
get crc2
let background: ImageData
drawCloud(nClouds)
drawSunFlower(nSunflowers)
drawFlower(nFlowers)
drawBee(nBees)
install load listener
drawFlower
_nFlower: number
[let i: number = 0]
[i < _nFlower]
[i++]
let flower: Flower
= new Flower();
push to array
drawBee
_nBees: number
[let i: number = 0]
[i < _nBees]
[i++]
let bee: Bee =
new Bee();
push to array
drawCloud
_nClouds: number
[let i: number = 0]
[i < _nClouds]
[i++]
let cloud: Cloud =
new Cloud();
push to array
drawSunflower
_nSunflowers: number
[let i: number = 0]
[i < _nSunflowers]
[i++]
let sunflower: Sunflower
= new Sunflower();
push to array
update
_background: ImageData
crc2.putImageData
move cloud
draw cloud
move flower
draw flower
move sunflower
draw sunflower
move bee
draw bee