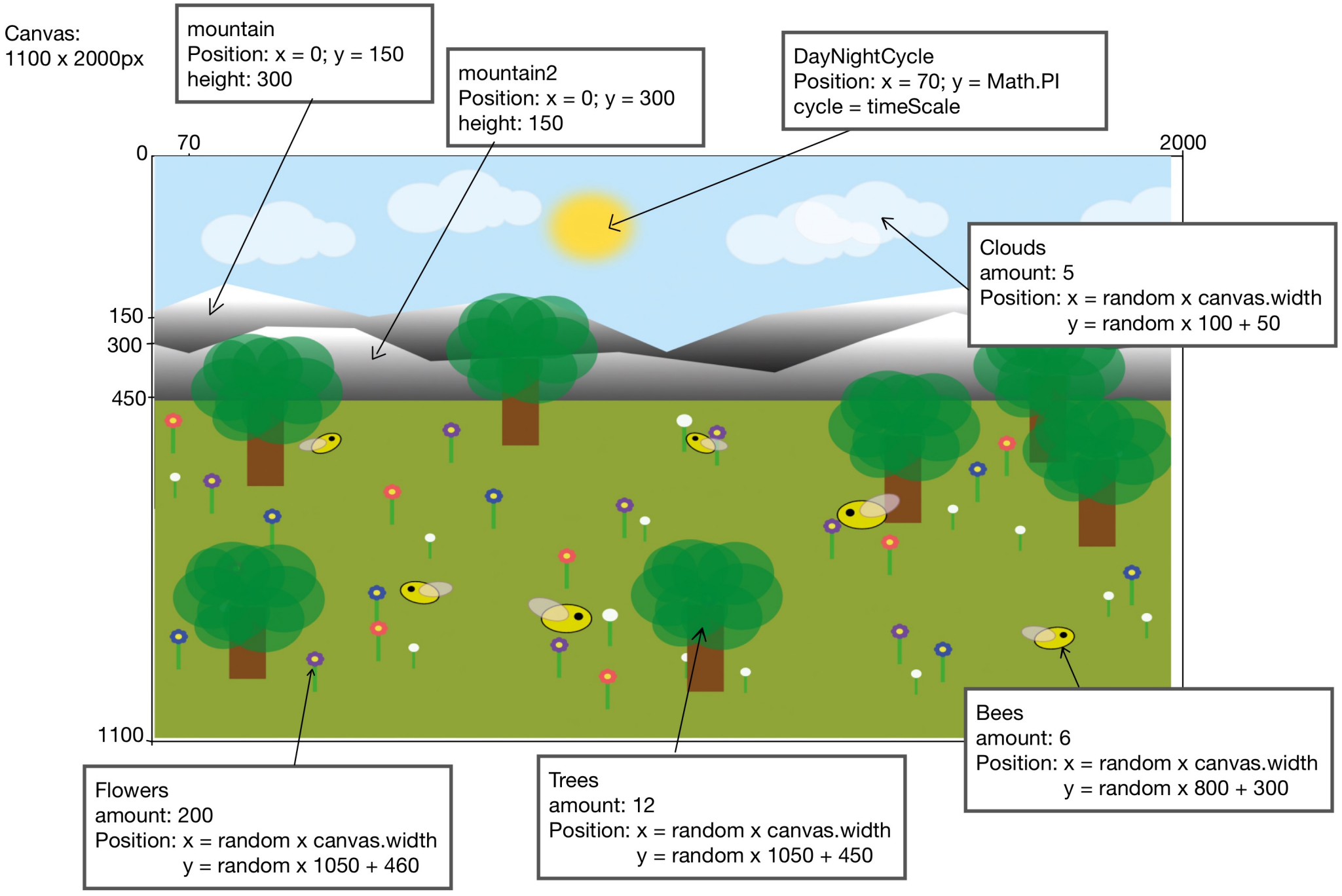
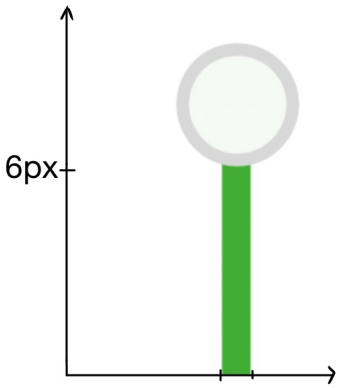
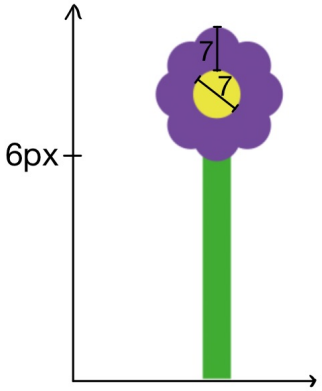
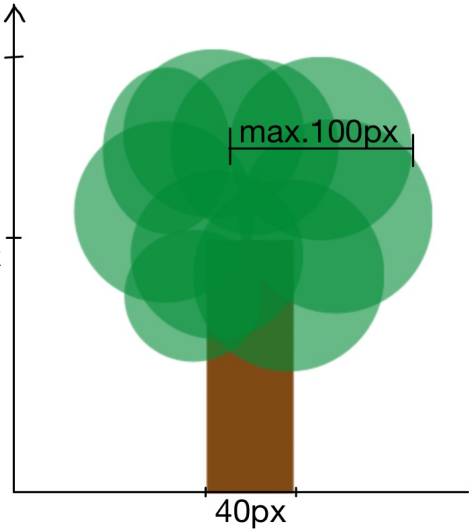
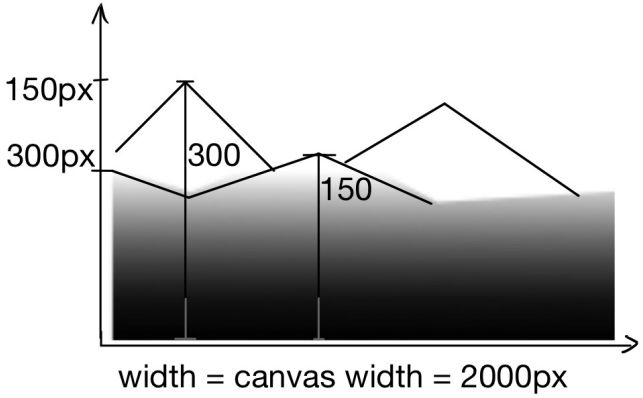

Canvas:
1100 x 2000px
mountain
Position: x = 0; y = 150
height: 300
mountain2
Position: x = 0; y = 300
height: 150
DayNightCycle
Position: x = 70; y = Math.PI
cycle = timeScale
0
70
2000
150
300
450
1100
Clouds
amount: 5
Position: x = random x canvas.width
y = random x 100 + 50
Flowers
amount: 200
Position: x = random x canvas.width
y = random x 1050 + 460
Trees
amount: 12
Position: x = random x canvas.width
y = random x 1050 + 450
Bees
amount: 6
Position: x = random x canvas.width
y = random x 800 + 300



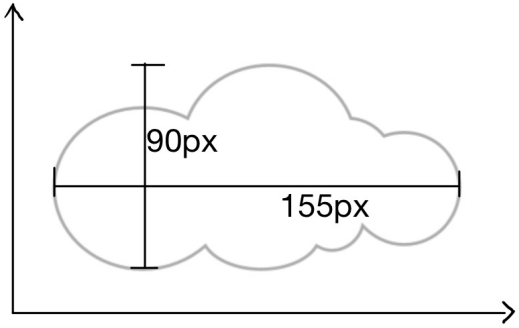


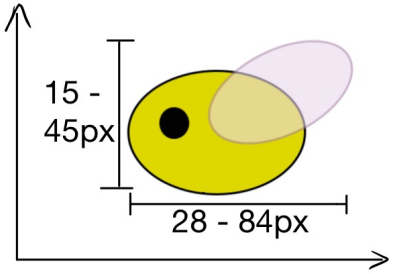
height trunk:
-150px - random x 100px

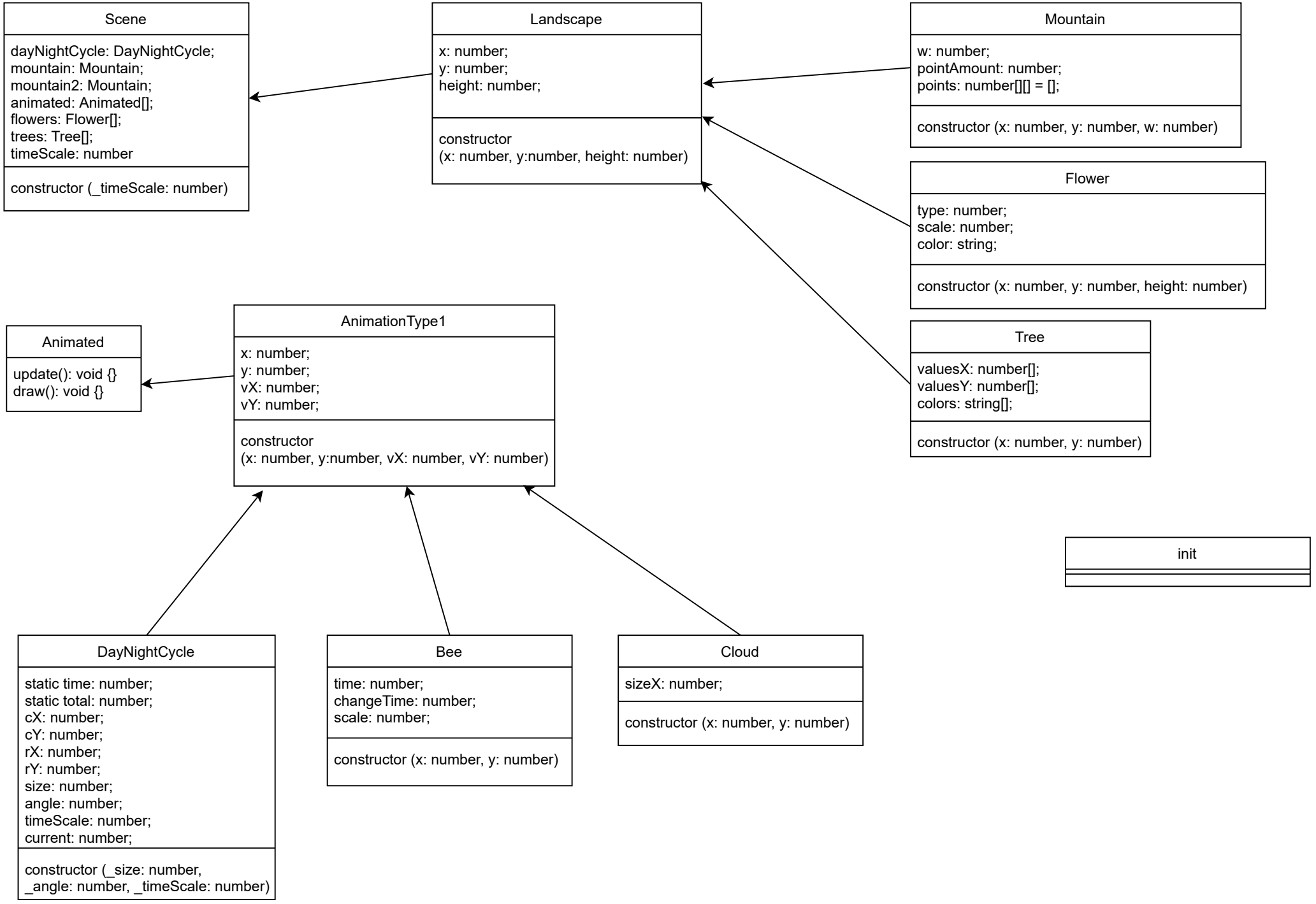




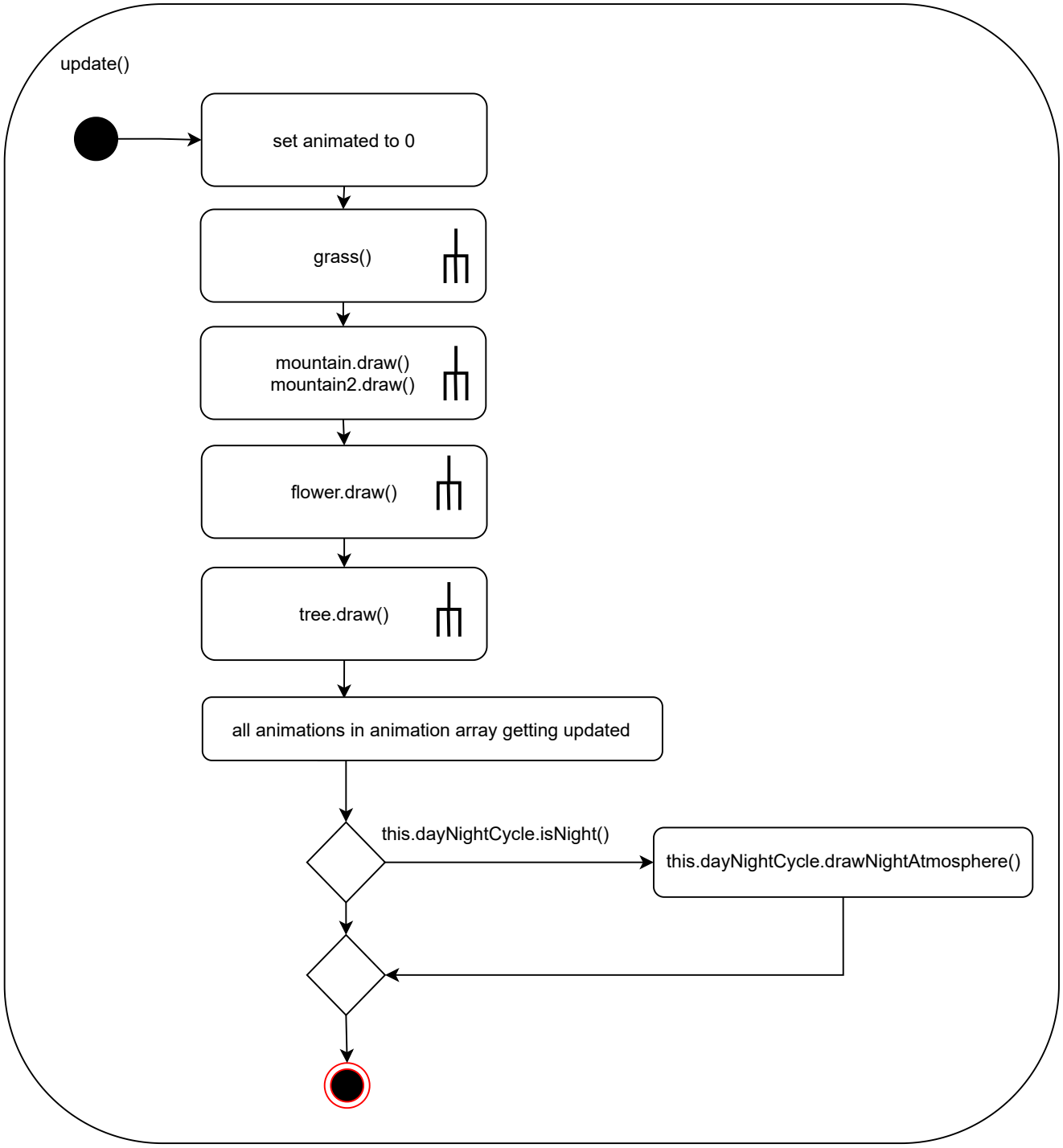
width = canvas width = 2000px

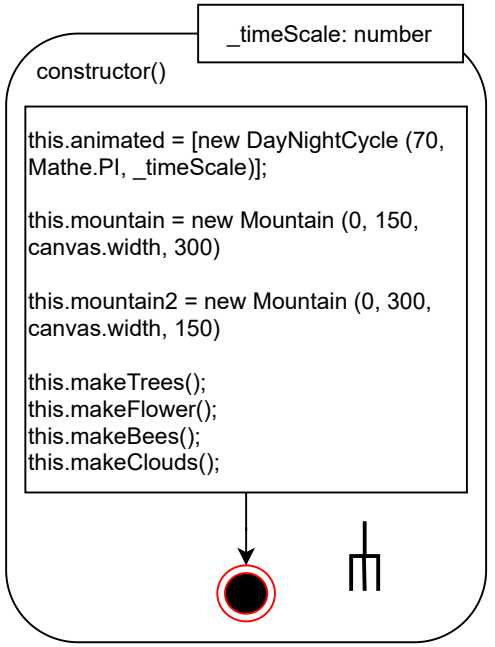


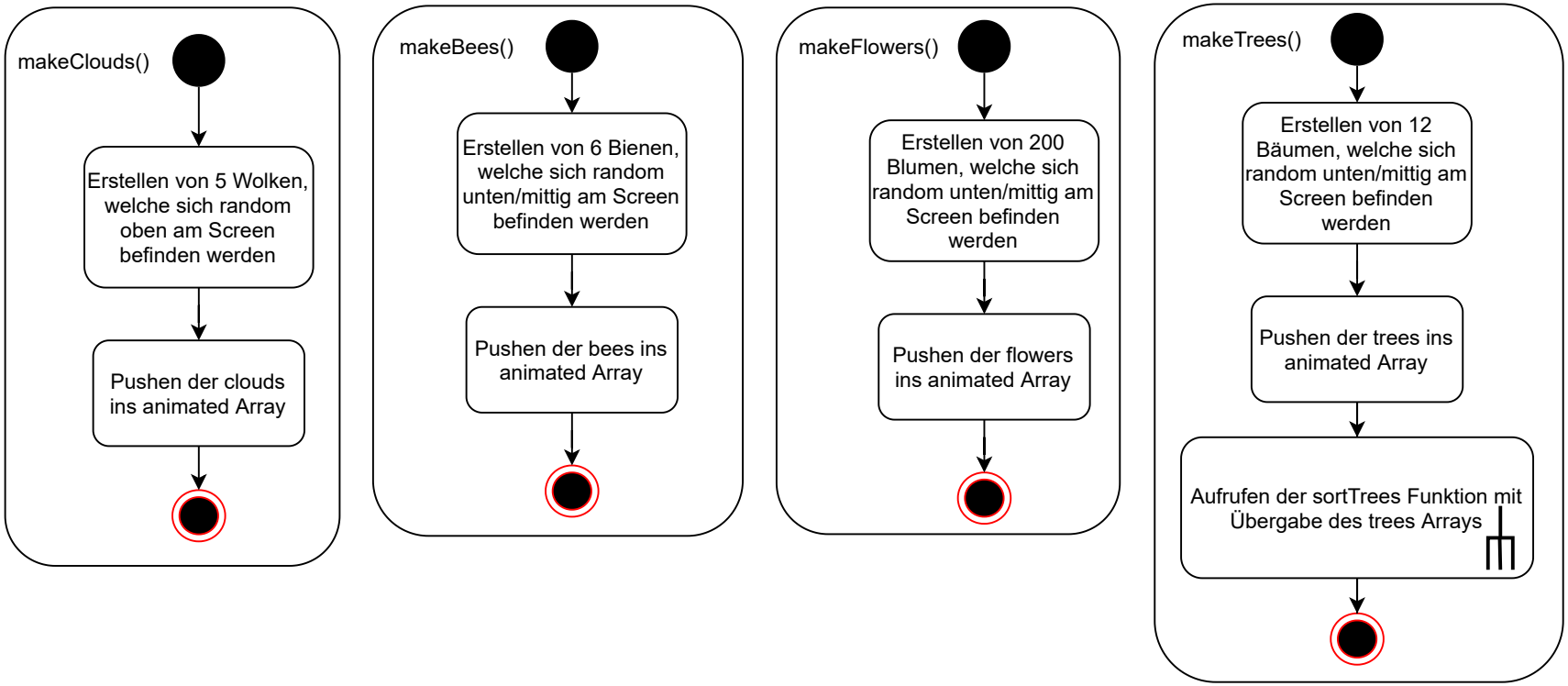


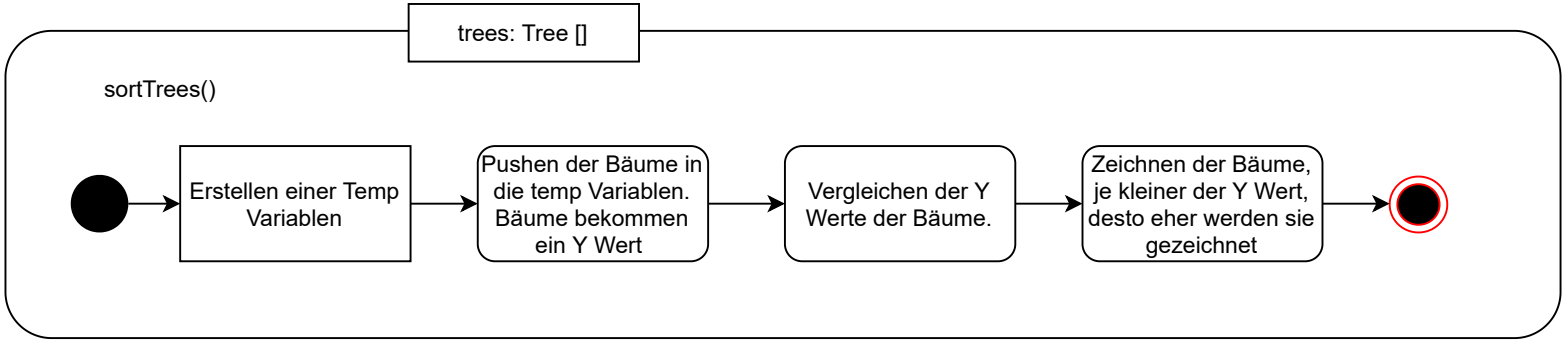

Scene
dayNightCycle: DayNightCycle;
mountain: Mountain;
mountain2: Mountain;
animated: Animated[];
flowers: Flower[];
trees: Tree[];
timeScale: number
constructor (_timeScale: number)
Landscape
x: number;
y: number;
height: number;
constructor
(x: number, y:number, height: number)
Mountain
w: number;
pointAmount: number;
points: number[][] = [];
constructor (x: number, y: number, w: number)
Flower
type: number;
scale: number;
color: string;
constructor (x: number, y: number, height: number)
Tree
valuesX: number[];
valuesY: number[];
colors: string[];
constructor (x: number, y: number)
Animated
update(): void {}
draw(): void {}
AnimationType1
x: number;
y: number;
vX: number;
vY: number;
constructor
(x: number, y:number, vX: number, vY: number)
init
DayNightCycle
static time: number;
static total: number;
cX: number;
cY: number;
rX: number;
rY: number;
size: number;
angle: number;
timeScale: number;
current: number;
constructor (_size: number,
_angle: number, _timeScale: number)
Bee
time: number;
changeTime: number;
scale: number;
constructor (x: number, y: number)
Cloud
sizeX: number;
constructor (x: number, y: number)

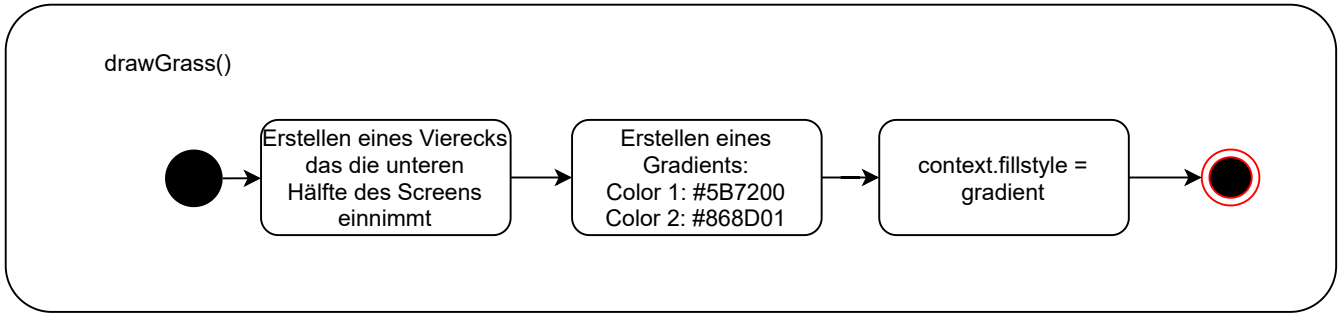
Aktivitätsdiagramm: Scene



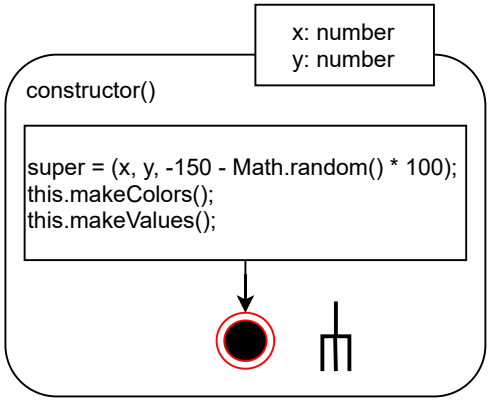


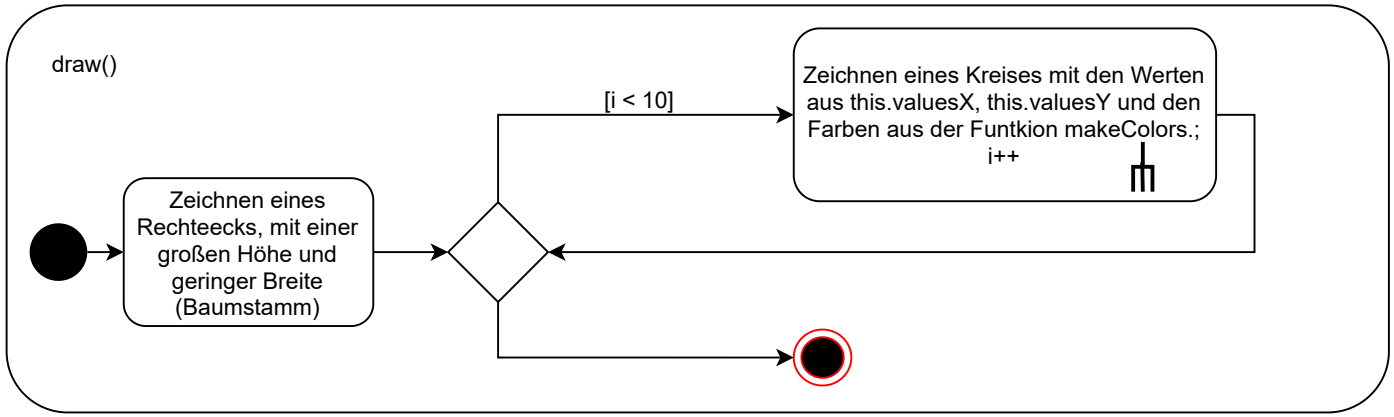

makeClouds()
Erstellen von 5 Wolken, welche sich random oben am Screen befinden werden
Pushen der clouds ins animated Array
makeBees()
Erstellen von 6 Bienen, welche sich random unten/mittig am Screen befinden werden
Pushen der bees ins animated Array
makeFlowers()
Erstellen von 200 Blumen, welche sich random unten/mittig am Screen befinden werden
Pushen der flowers ins animated Array
makeTrees()
Erstellen von 12 Bäumen, welche sich random unten/mittig am Screen befinden werden
Pushen der trees ins animated Array
Aufrufen der sortTrees Funktion mit Übergabe des trees Arrays

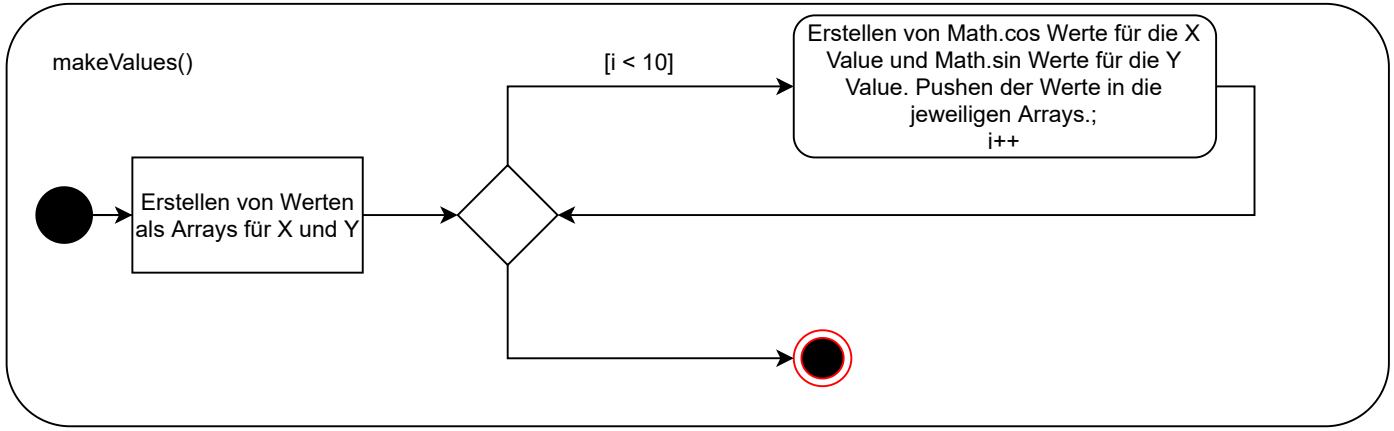

trees: Tree []
sortTrees()
Erstellen einer Temp Variablen
Pushen der Bäume in die temp Variablen. Bäume bekommen ein Y Wert
Vergleichen der Y Werte der Bäume.
Zeichnen der Bäume, je kleiner der Y Wert, desto eher werden sie gezeichnet

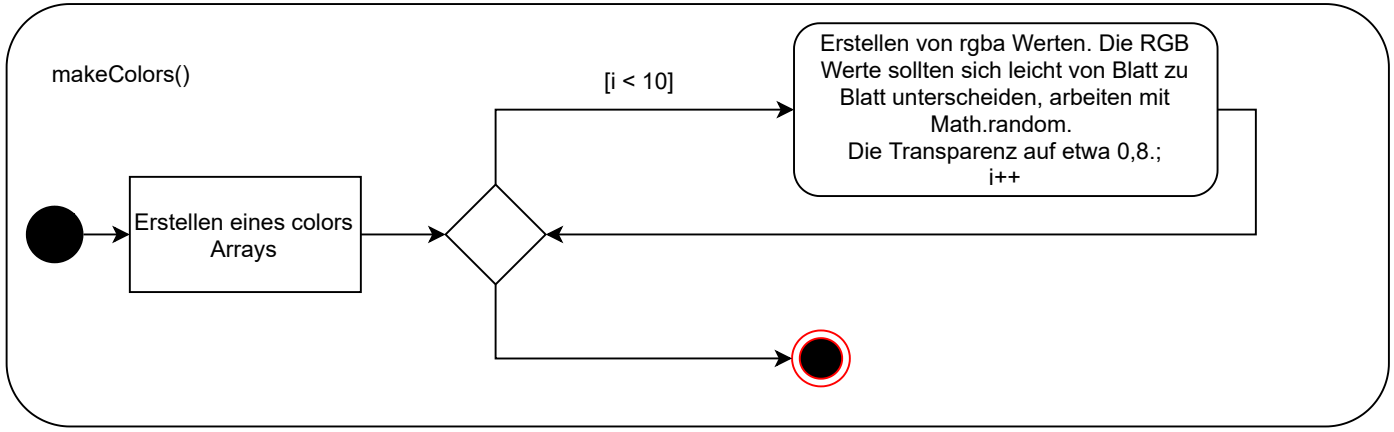

drawGrass()
Erstellen eines Vierecks das die unteren Hälfte des Screens einnimmt
Erstellen eines Gradients:
Color 1: #5B7200
Color 2: #868D01
context.fillstyle = gradient

Aktivitätsdiagramm: Trees

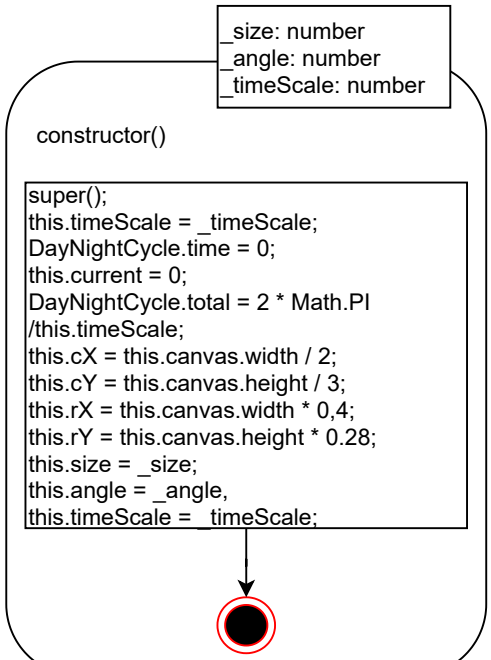


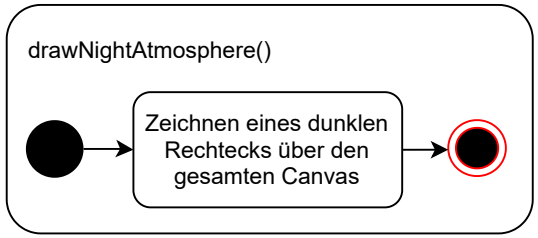


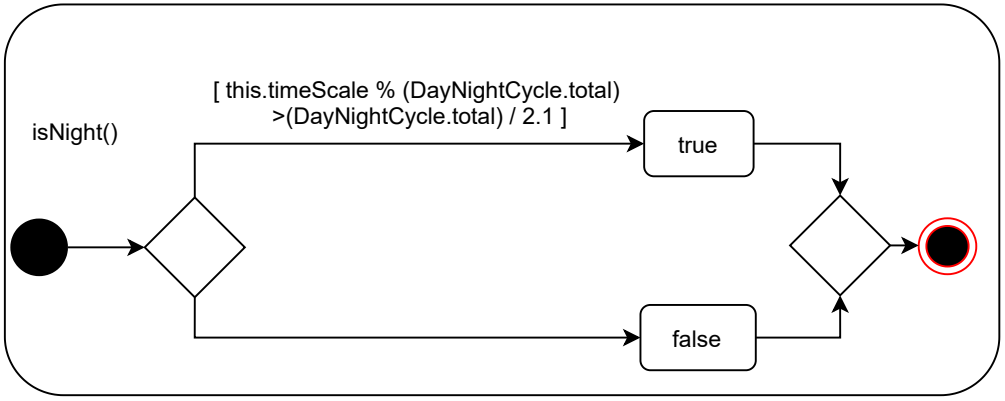


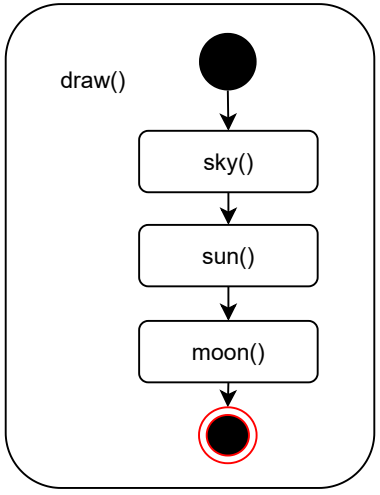


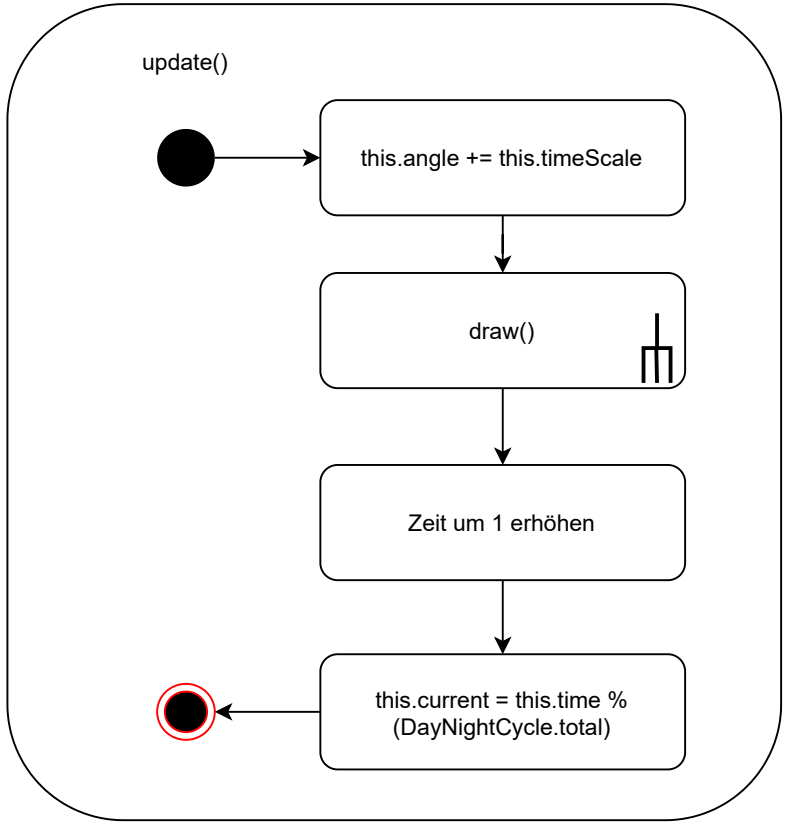
Aktivitätsdiagramm: DayNightCycle

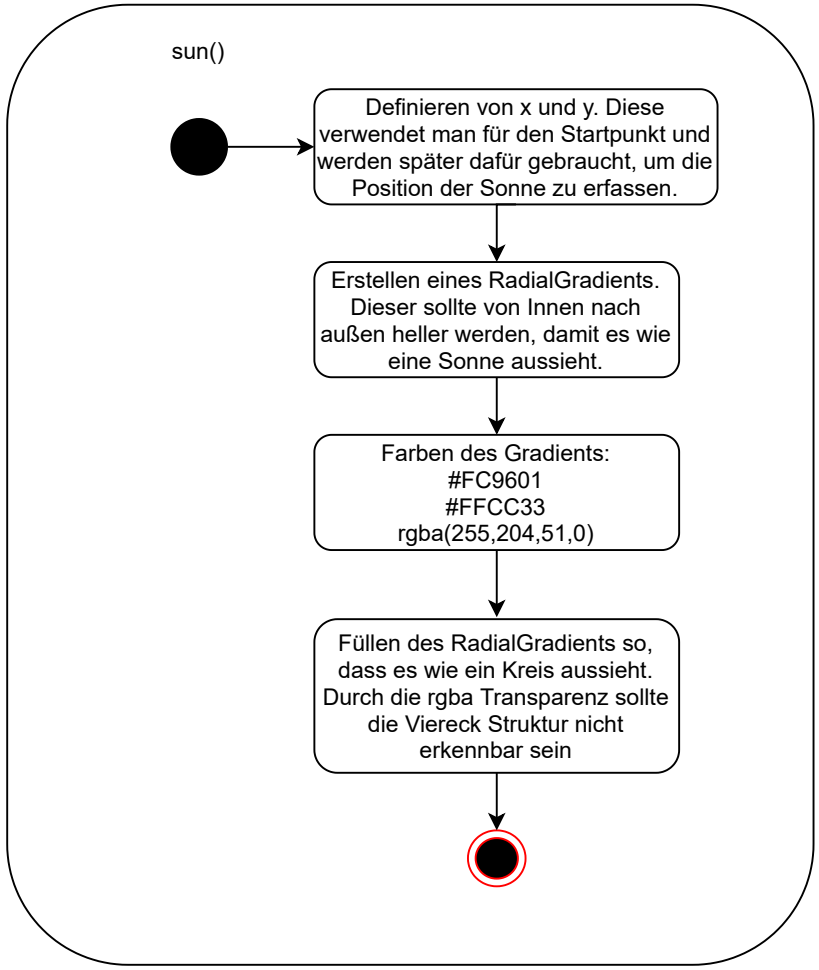


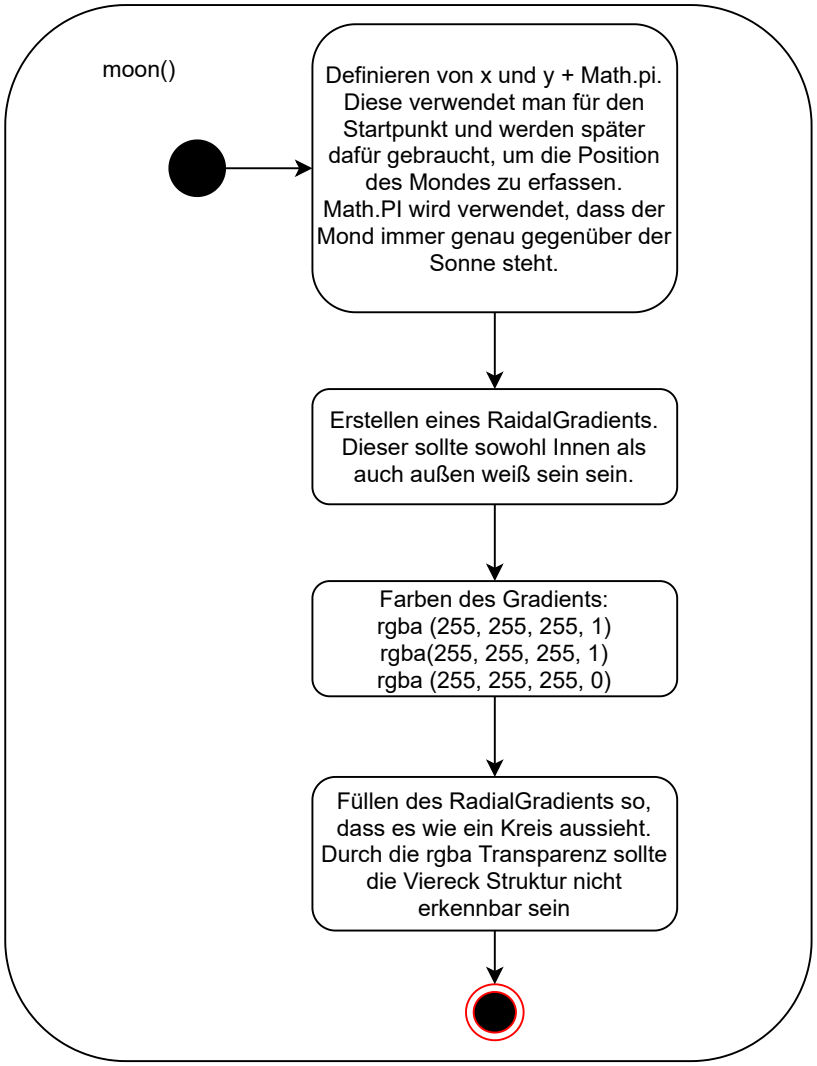


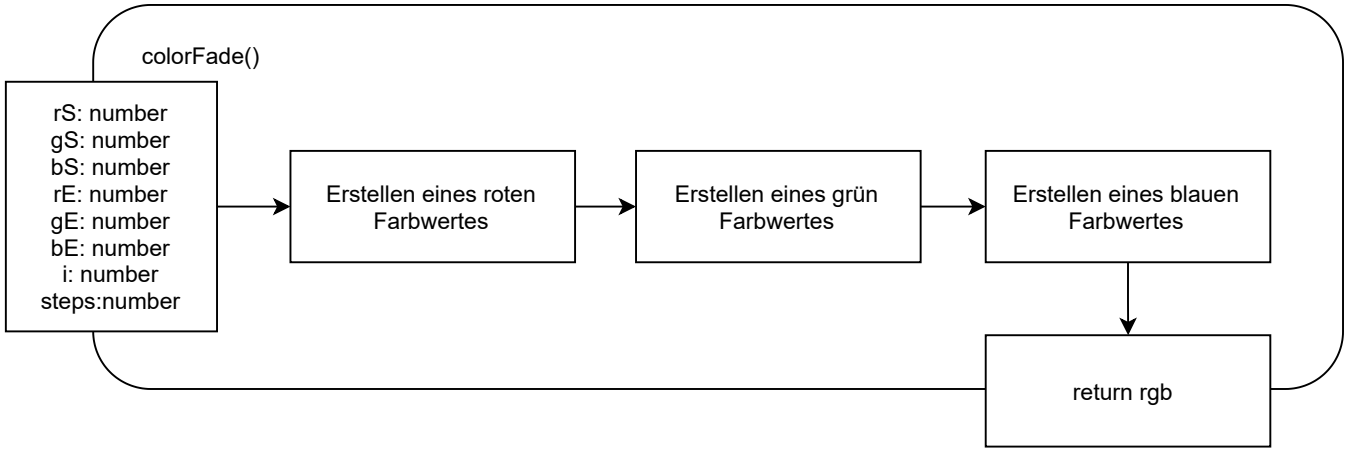


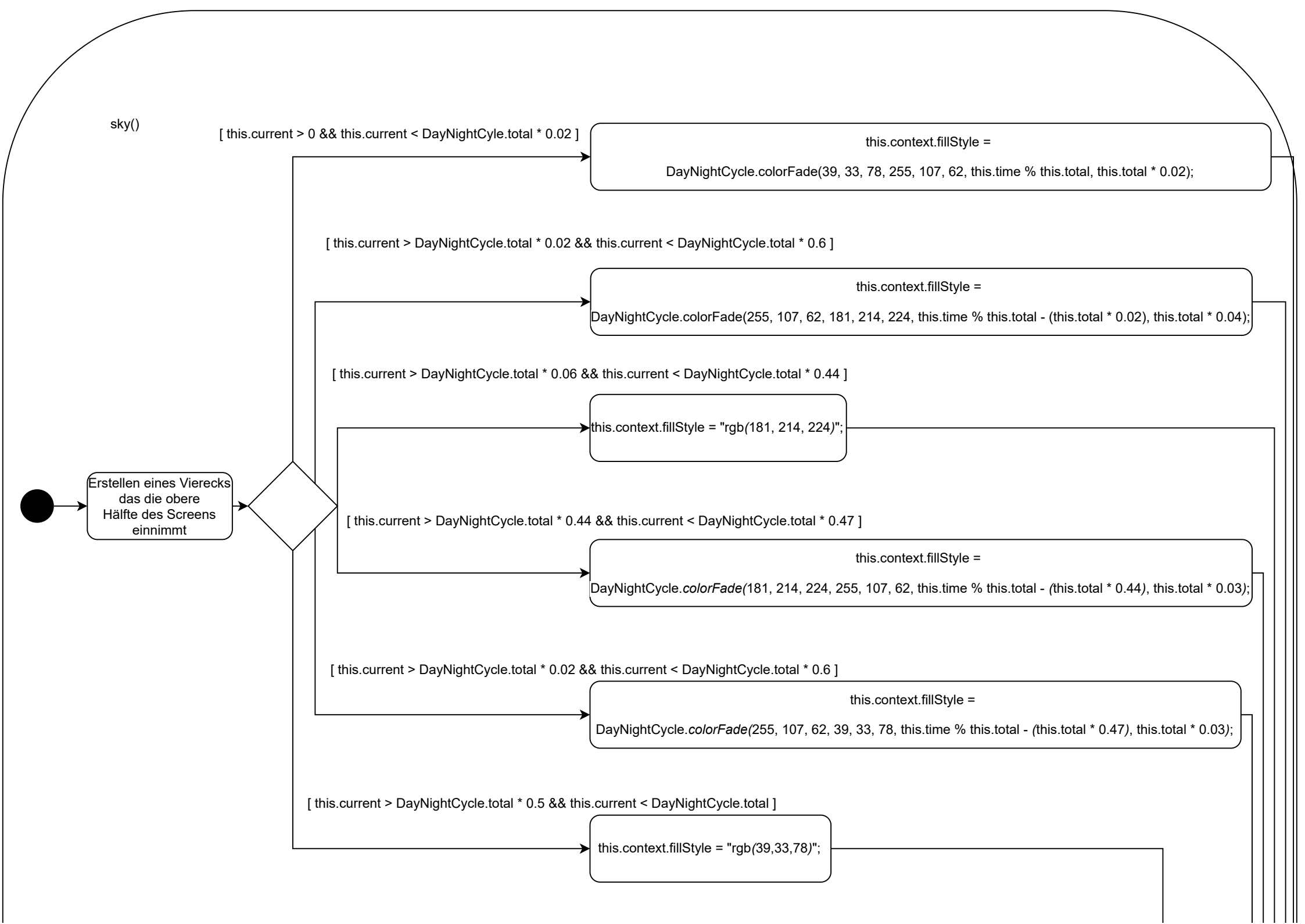


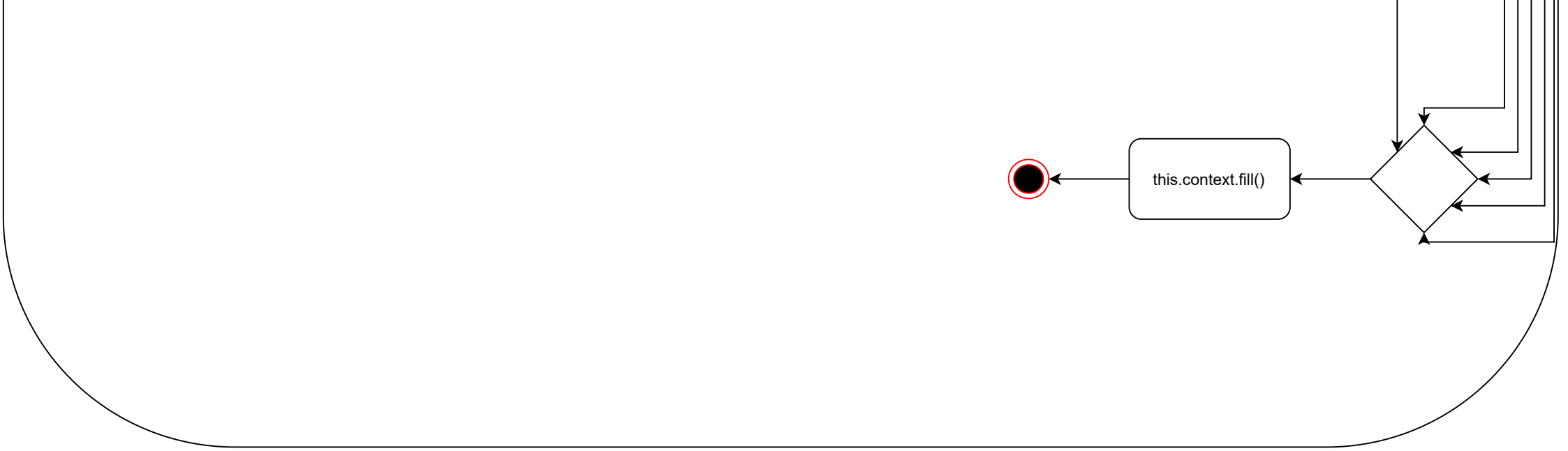




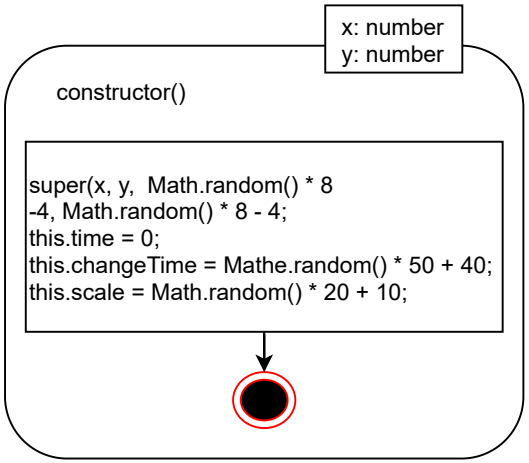

moon()
Definieren von x und y + Math.pi. Diese verwendet man für den Startpunkt und werden später dafür gebraucht, um die Position des Mondes zu erfassen. Math.PI wird verwendet, dass der Mond immer genau gegenüber der Sonne steht.
Erstellen eines RaidalGradients. Dieser sollte sowohl Innen als auch außen weiß sein sein.
Farben des Gradients:
rgba (255, 255, 255, 1)
rgba(255, 255, 255, 1)
rgba (255, 255, 255, 0)
Füllen des RadialGradients so, dass es wie ein Kreis aussieht. Durch die rgba Transparenz sollte die Viereck Struktur nicht erkennbar sein

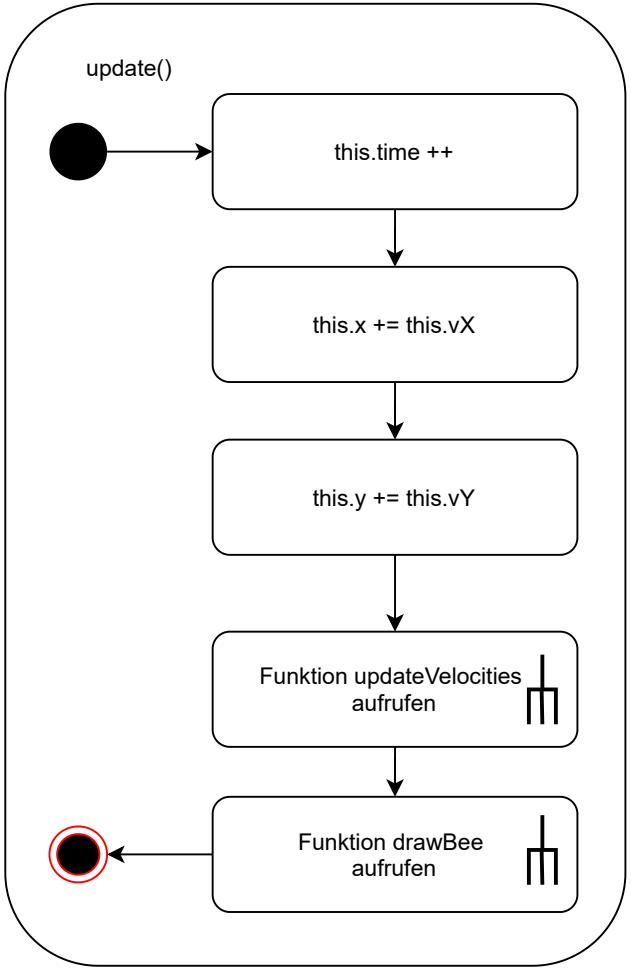

colorFade()
rS: number
gS: number
bS: number
rE: number
gE: number
bE: number
i: number
steps:number
Erstellen eines roten Farbwertes
Erstellen eines grün Farbwertes
Erstellen eines blauen Farbwertes
return rgb

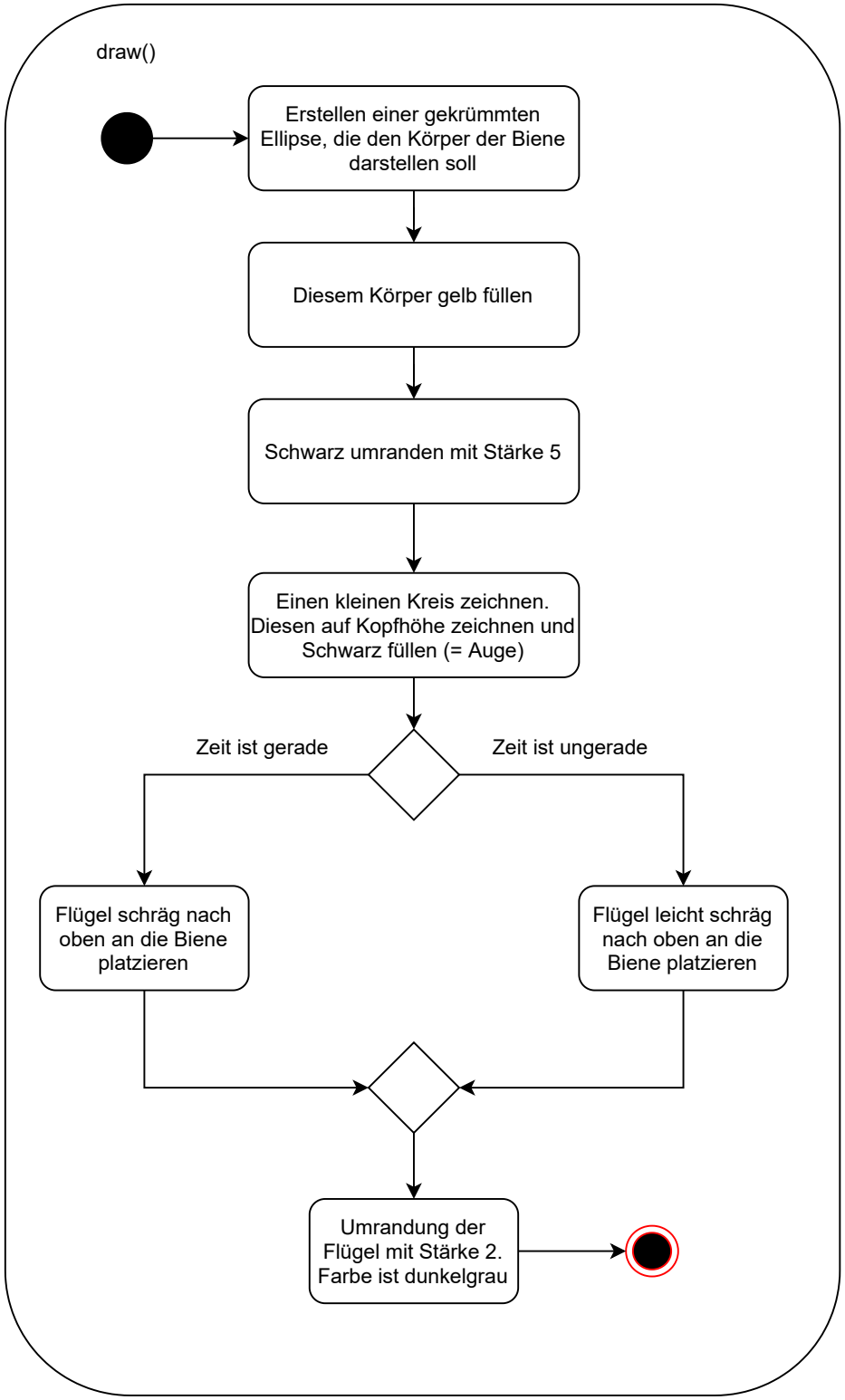

sky()
Erstellen eines Vierecks das die obere Hälfte des Screens einnimmt
[this.current > 0 && this.current < DayNightCyle.total * 0.02]
this.context.fillStyle =
DayNightCycle.colorFade(39, 33, 78, 255, 107, 62, this.time % this.total, this.total * 0.02);
[this.current > DayNightCycle.total * 0.02 && this.current < DayNightCycle.total * 0.6]
this.context.fillStyle =
DayNightCycle.colorFade(255, 107, 62, 181, 214, 224, this.time % this.total - (this.total * 0.02), this.total * 0.04);
[this.current > DayNightCycle.total * 0.06 && this.current < DayNightCycle.total * 0.44]
this.context.fillStyle = "rgb(181, 214, 224)";
[this.current > DayNightCycle.total * 0.44 && this.current < DayNightCycle.total * 0.47]
this.context.fillStyle =
DayNightCycle.colorFade(181, 214, 224, 255, 107, 62, this.time % this.total - (this.total * 0.44), this.total * 0.03);
[this.current > DayNightCycle.total * 0.02 && this.current < DayNightCycle.total * 0.6]
this.context.fillStyle =
DayNightCycle.colorFade(255, 107, 62, 39, 33, 78, this.time % this.total - (this.total * 0.47), this.total * 0.03);
[this.current > DayNightCycle.total * 0.5 && this.current < DayNightCycle.total]
this.context.fillStyle = "rgb(39,33,78)";

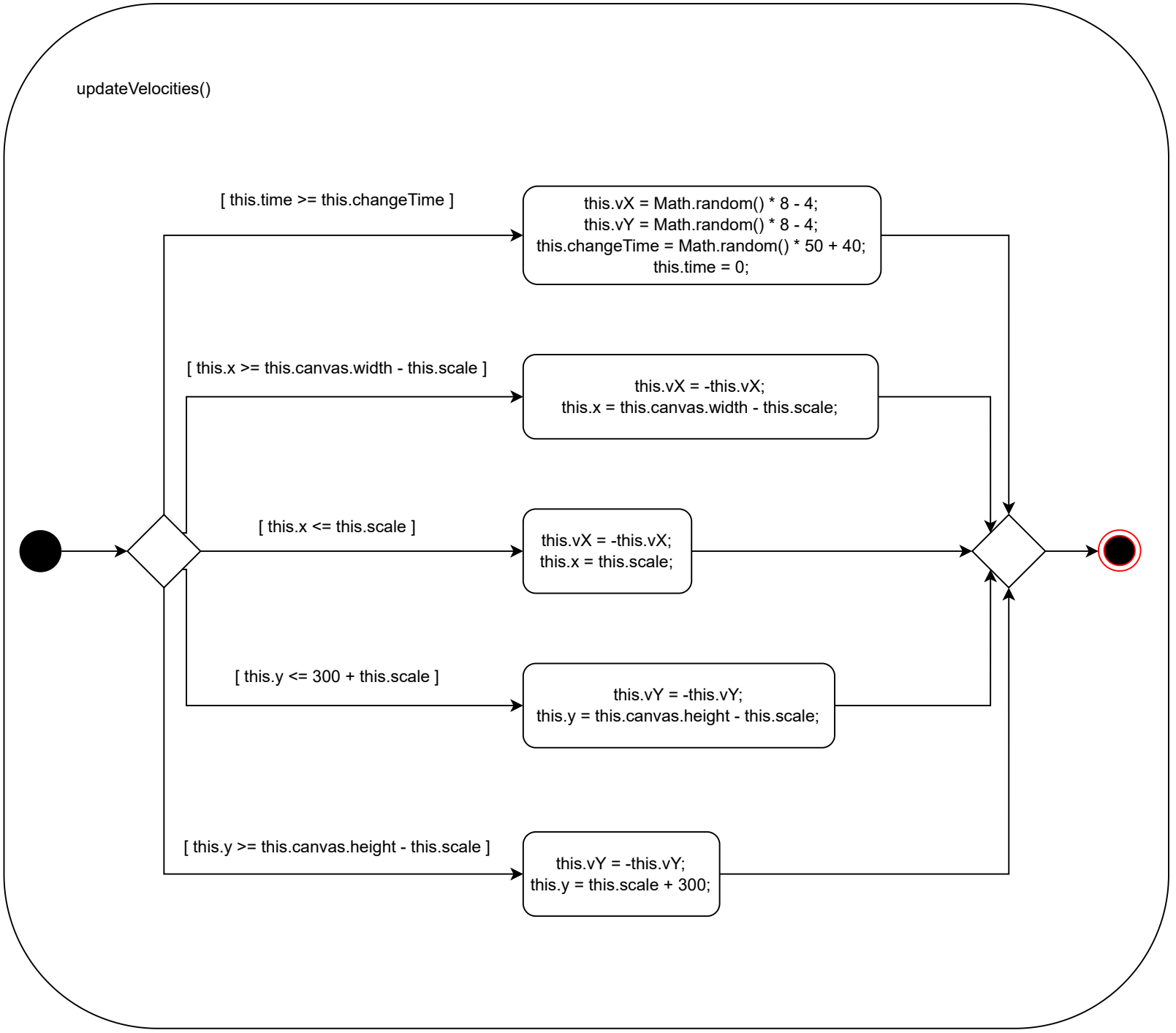

this.context.fill()

Aktivitätsdiagramm: Bee

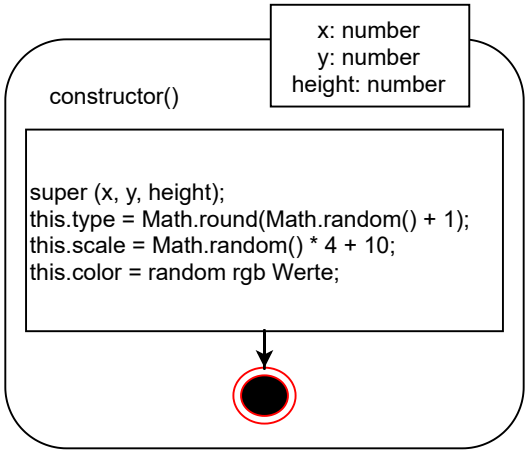


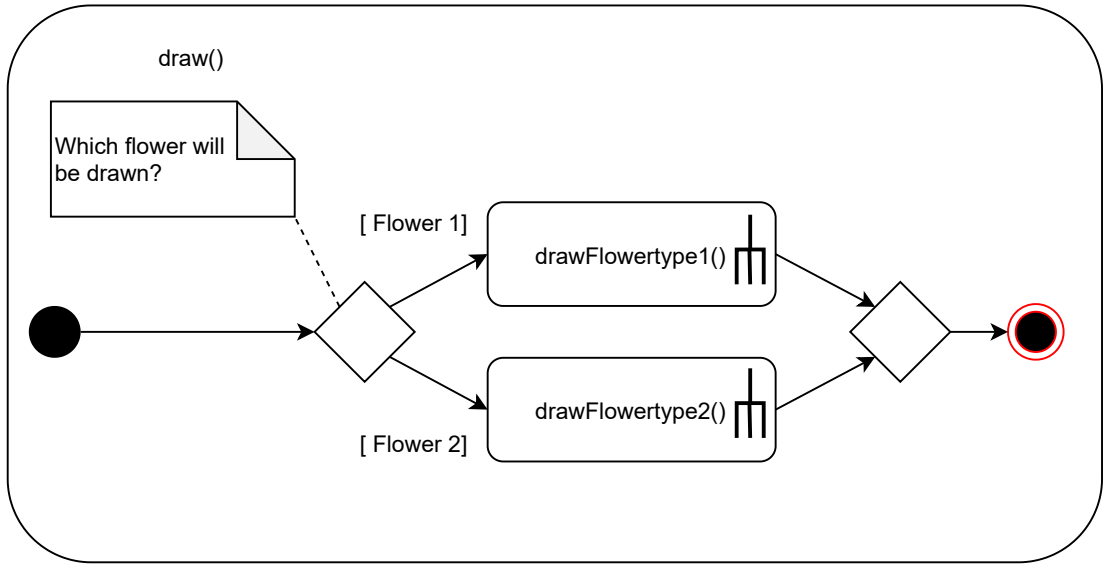


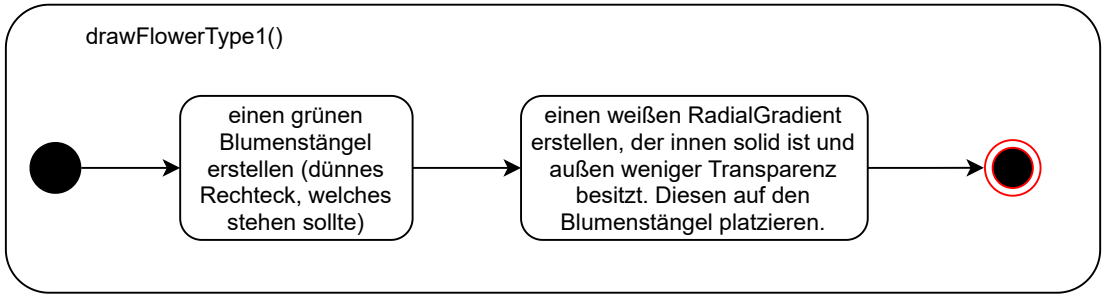


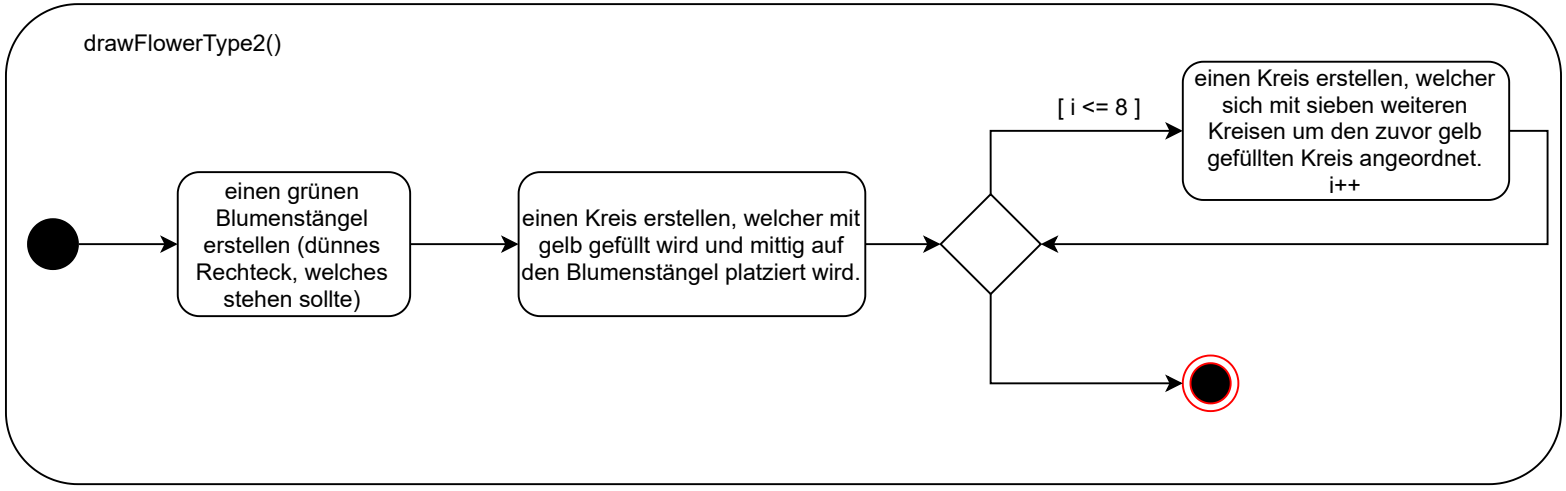

updateVelocities()
[this.time >= this.changeTime]
this.vX = Math.random() * 8 - 4;
this.vY = Math.random() * 8 - 4;
this.changeTime = Math.random() * 50 + 40;
this.time = 0;
[this.x >= this.canvas.width - this.scale]
this.vX = -this.vX;
this.x = this.canvas.width - this.scale;
[this.x <= this.scale]
this.vX = -this.vX;
this.x = this.scale;
[this.y <= 300 + this.scale]
this.vY = -this.vY;
this.y = this.canvas.height - this.scale;
[this.y >= this.canvas.height - this.scale]
this.vY = -this.vY;
this.y = this.scale + 300;

Aktivitätsdiagramm: Flower

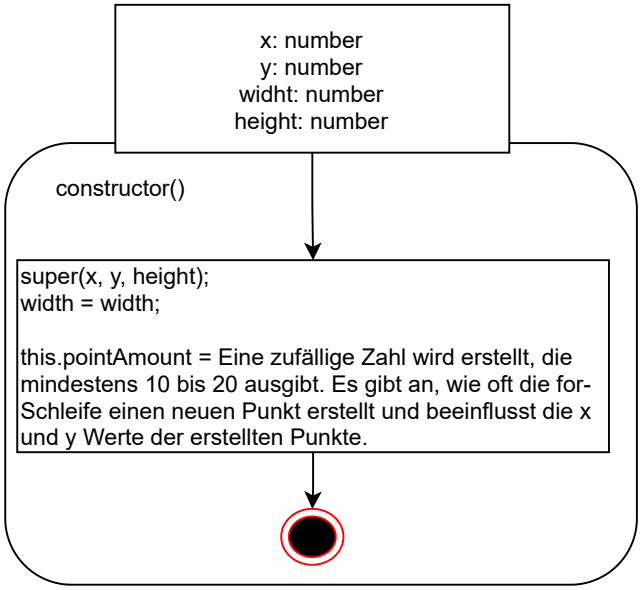


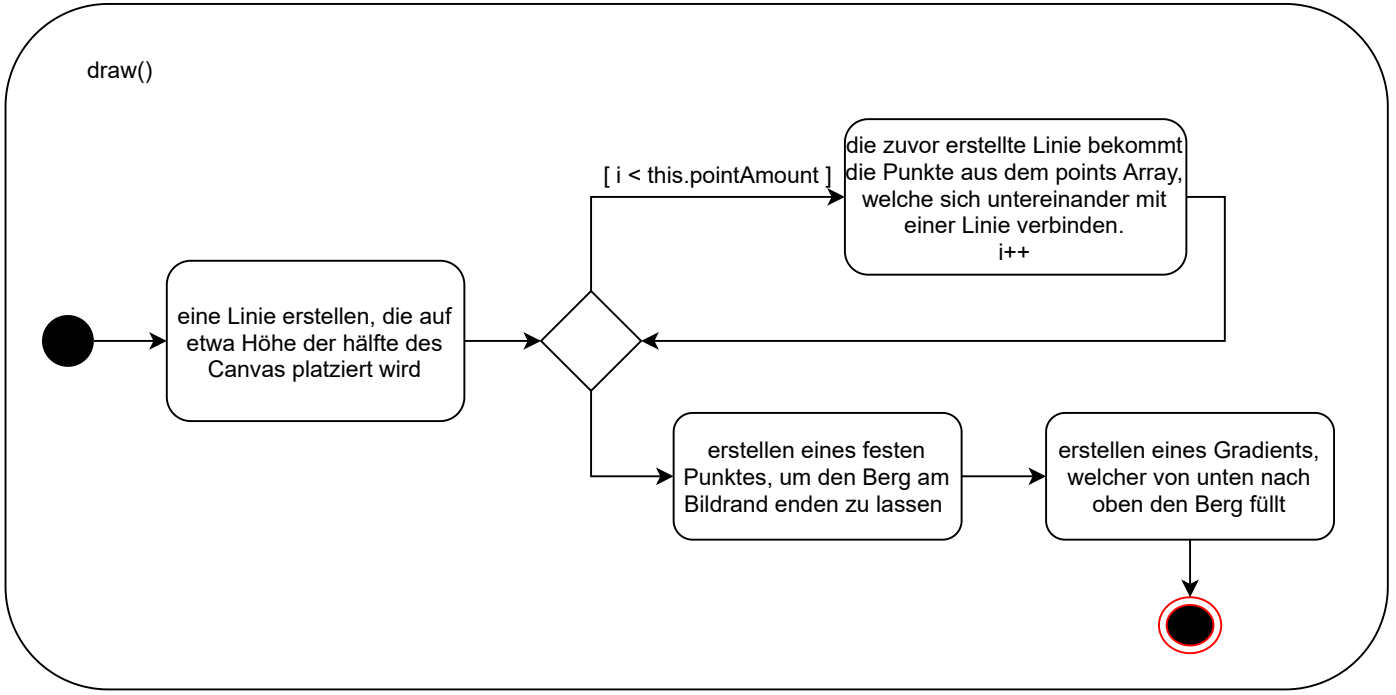




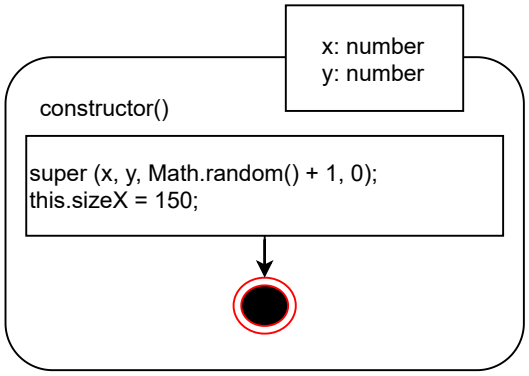


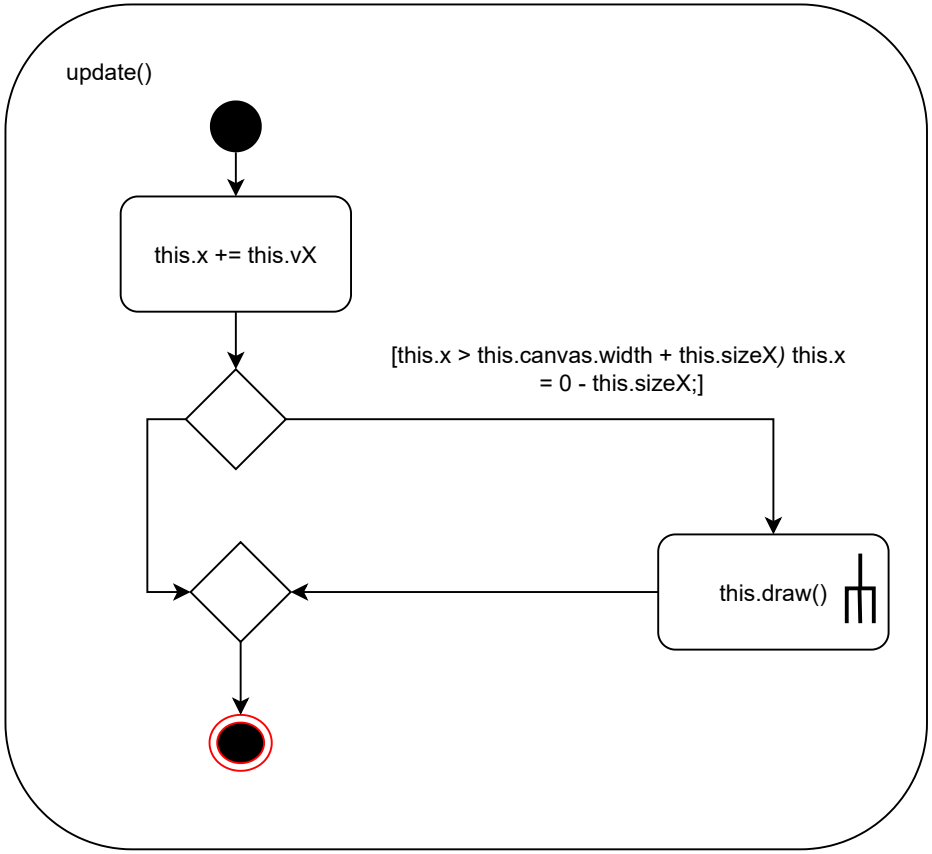
Aktivitätsdiagramm: Mountain

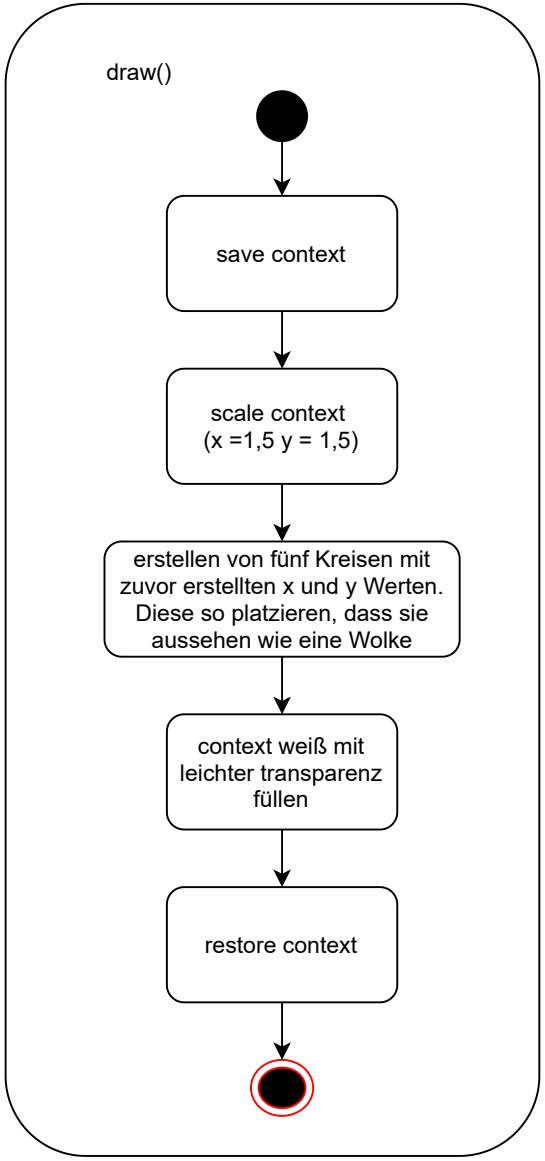




Aktivitätsdiagramm: Cloud



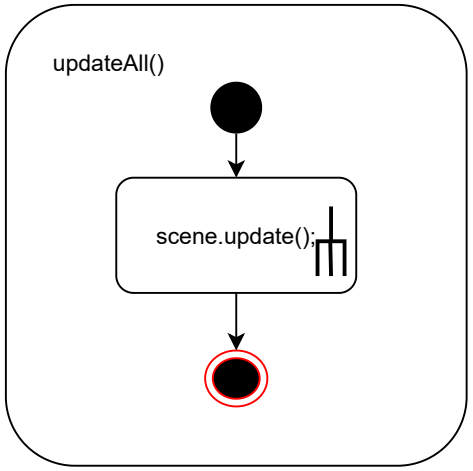


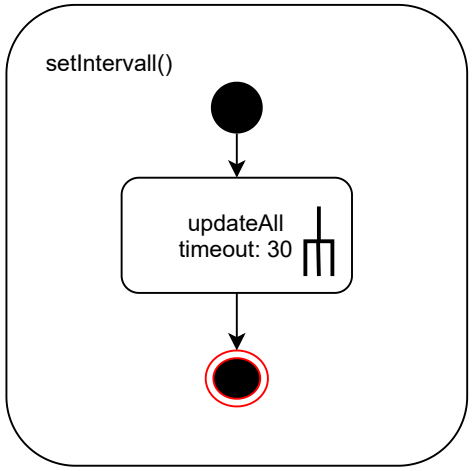


Aktivitätsdiagramm: Script

timeScale: number = 0.005;
(kann angepasst werden, beeinflusst die Geschwindigkeit,
in der der Tag zur Nacht wird und umgekehrt)
scene: Scene = new Scene(timeScale);

setInterval (updateAll, 30)





Aktivitätsdiagramm: Init

```
export let canvas: HTMLCanvasElement = <HTMLCanvasElement> document.getElementById("canvas");
export let c: CanvasRenderingContext2D = canvas.getContext("2d");
```