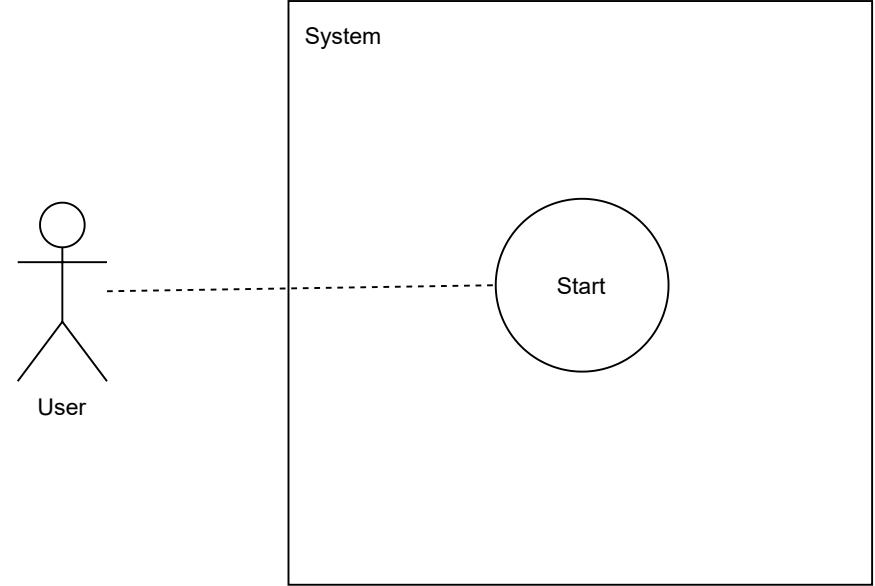
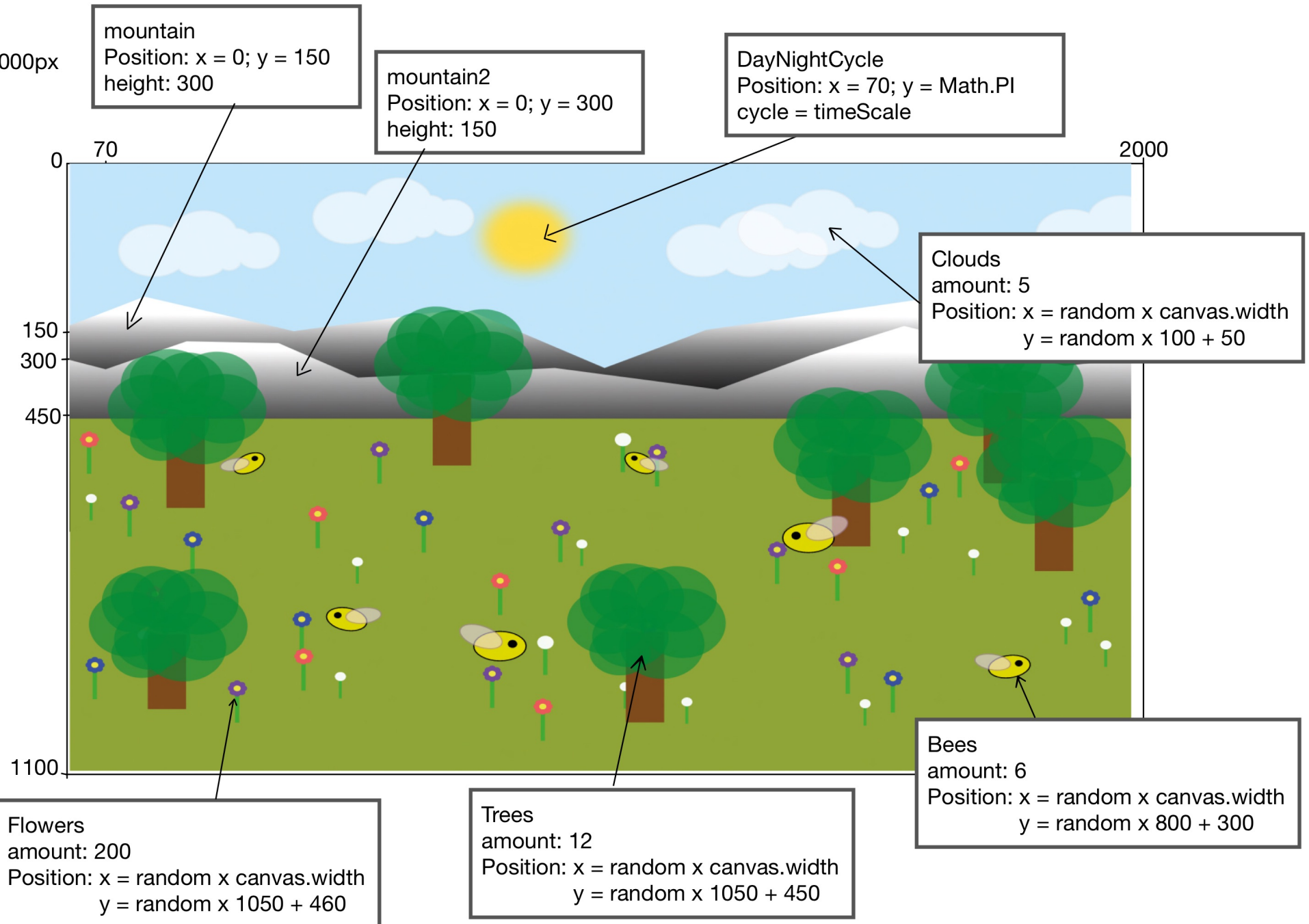


User-Interface-Diagramm

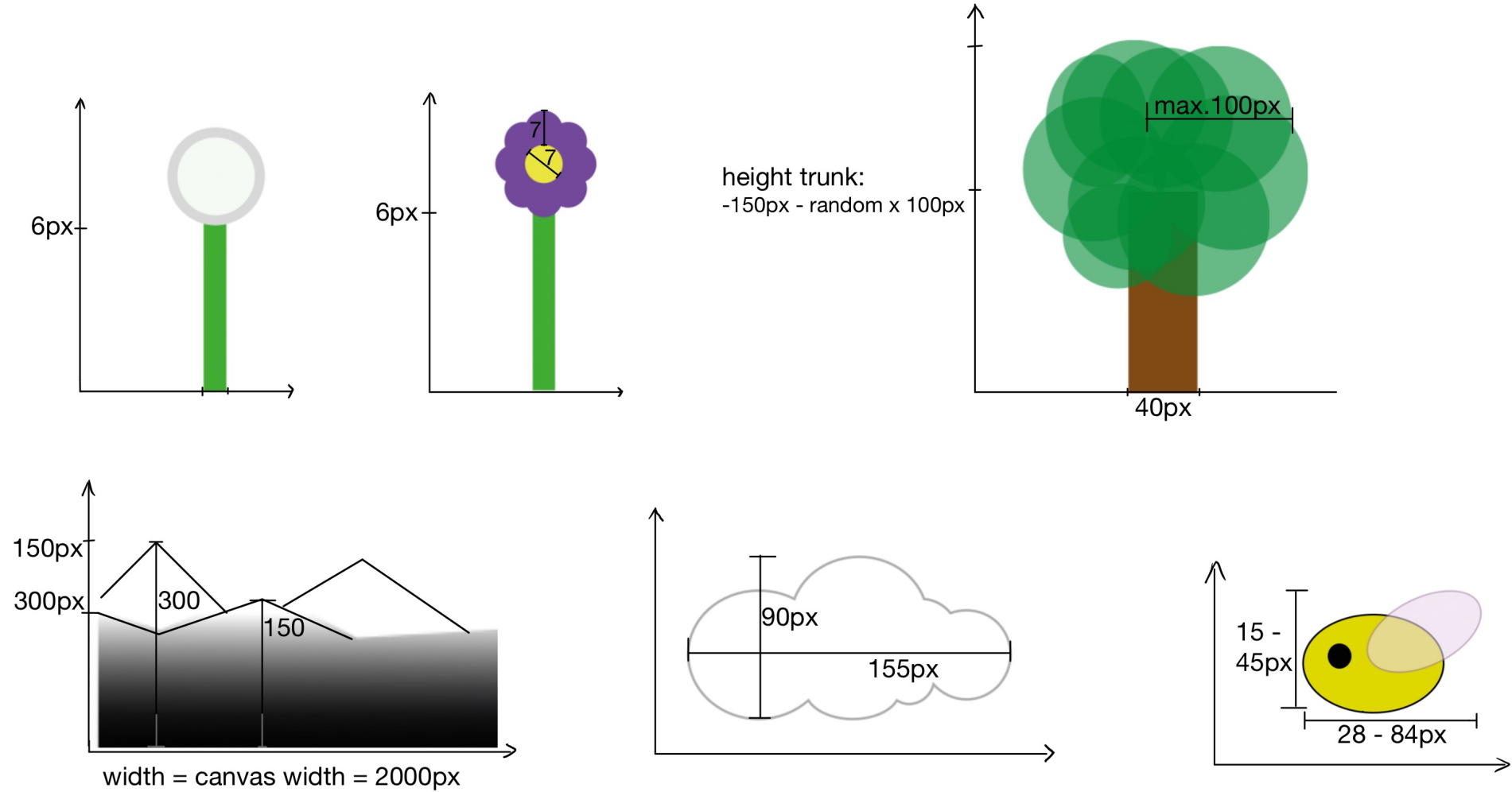


UI Scribble: Canvas

Canvas:
1100 x 2000px



UI Scribble: Assets



Blumenwiese: Classes: Klassendiagramm

CanvasRenderingContext

Scene
dayNightCycle: DayNightCycle; mountain: Mountain; mountain2: Mountain; flowers: Flower[]; trees: Tree[]; bees: Bee[]; clouds: Cloud[];
constructor (_ canvas: HTMLCanvasElement) (_ timeScale: number)

Tree
x: number; y: number; height: number; valuesX: number[]; valuesY: number[]; colors: string[];
constructor (_ canvas: HTMLCanvasElement) (x: number) (y: number)

DayNightCycle
cX: number; cY: number; rX: number; rY: number; size: number; angle: number; timeScale: number; time: number; total: number; current: number;
constructor (_ canvas: HTMLCanvasElement) (_ size: number) (_ angle: number) (_ timeScale: number)

Bee
x: number; y: number; vX: number; vY: number; time: number; changeTime: number; scale: number;
constructor (_ canvas: HTMLCanvasElement) (x: number) (y: number)

Flower
x: number; y: number; height: number; type: number; scale: number; color: string;
constructor (_ canvas: HTMLCanvasElement) (x: number) (y: number) (height: number)

Mountain
x: number; y: number; w: number; h: number; pointAmount: number; points: number[][] = [];
constructor (_ canvas: HTMLCanvasElement) (x: number) (y: number) (w: number) (h: number)

Cloud
x: number; y: number; vX: number; sizeX: number;
constructor (_ canvas: HTMLCanvasElement) (x: number) (y: number)

Aktivitätsdiagramm: Scene

update()



update dayNightCycle

funktion drawGrass aufrufen



Jede Wolke im Cloud Array wird
geupdated

draw mountain1 & mountain2

Jede Blume im Blumen Array wird
gezeichnet

Jeder Baum im Baumen Array wird
gezeichnet

Jede Biene im Bienen Array wird
geupdated

this.dayNightCycle.isNight()

this.dayNightCycle.drawNightAtmosphere()

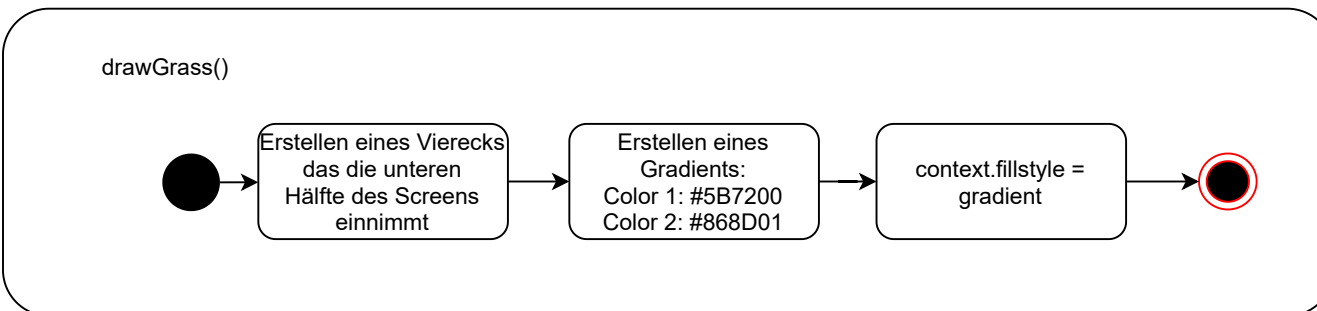
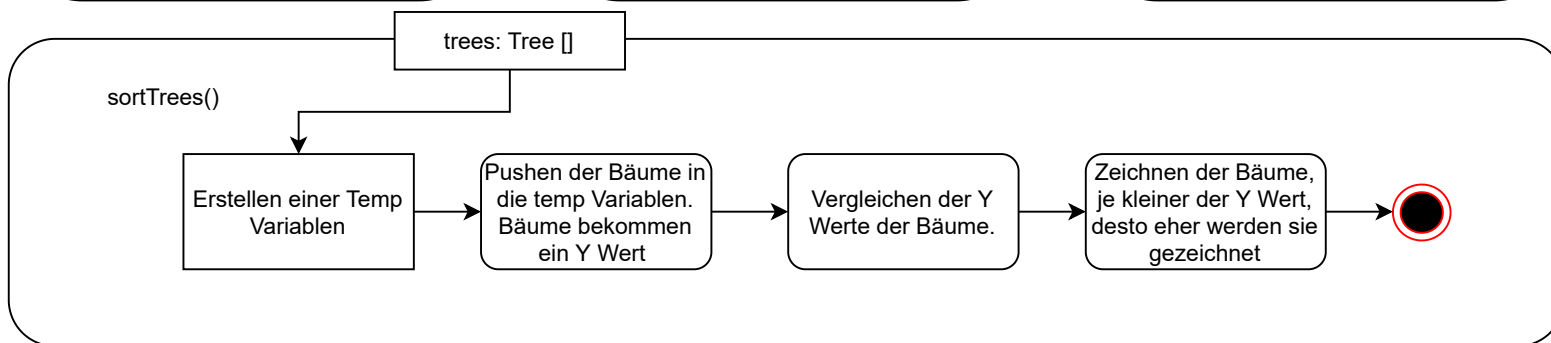
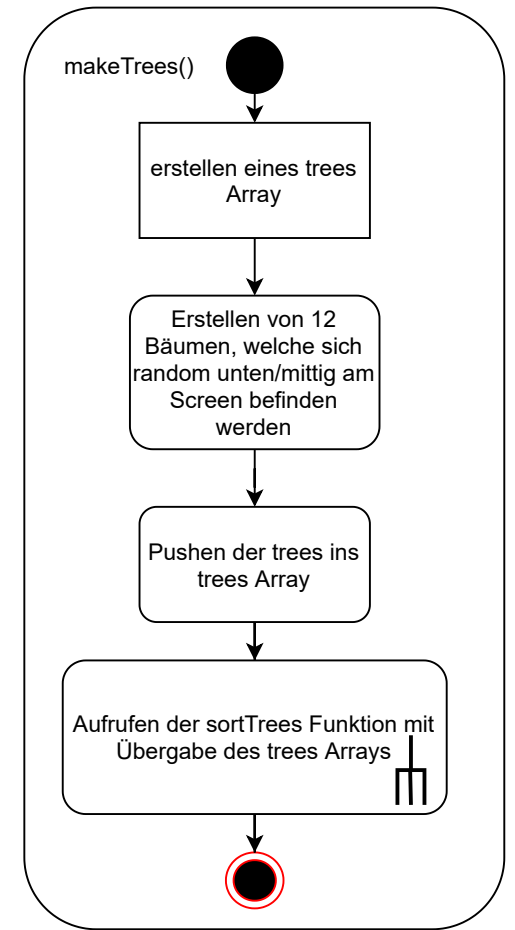
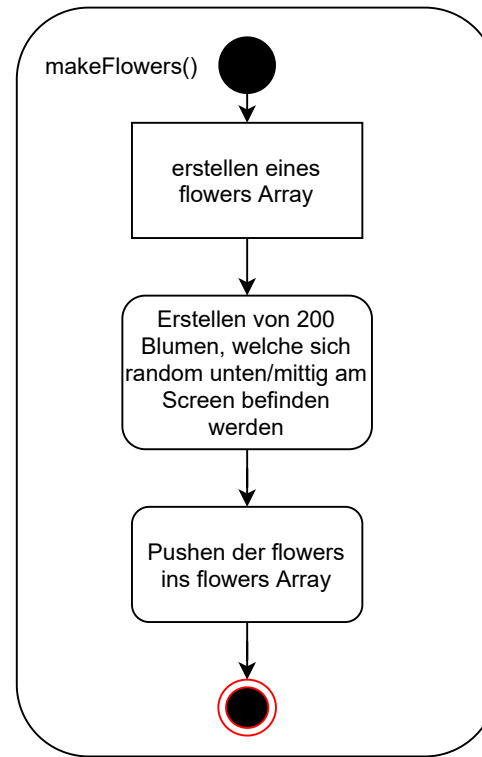
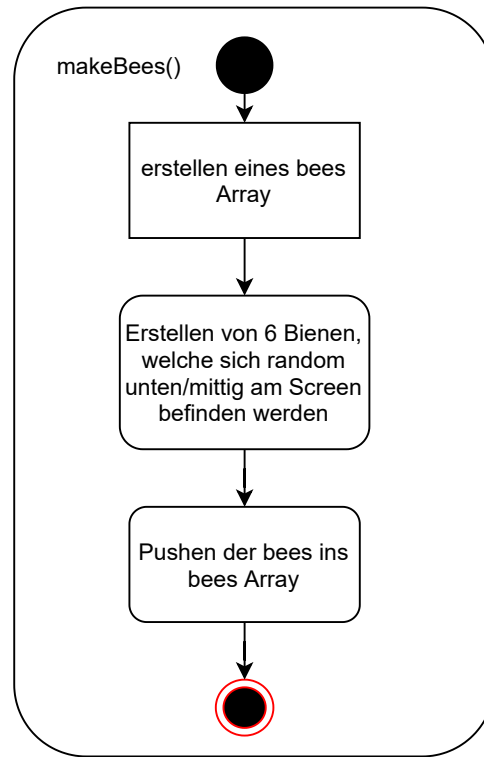
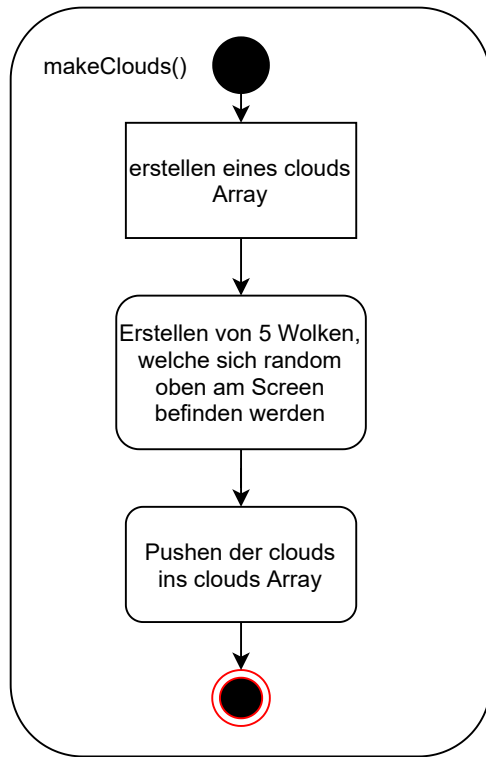


_canvas: HTMLCanvasElement
_timeScale: number

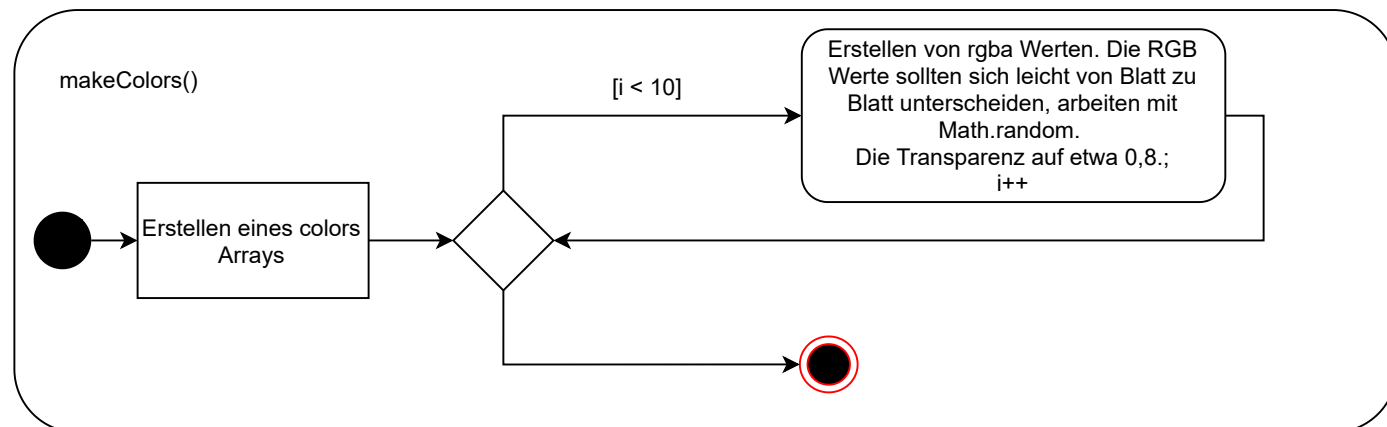
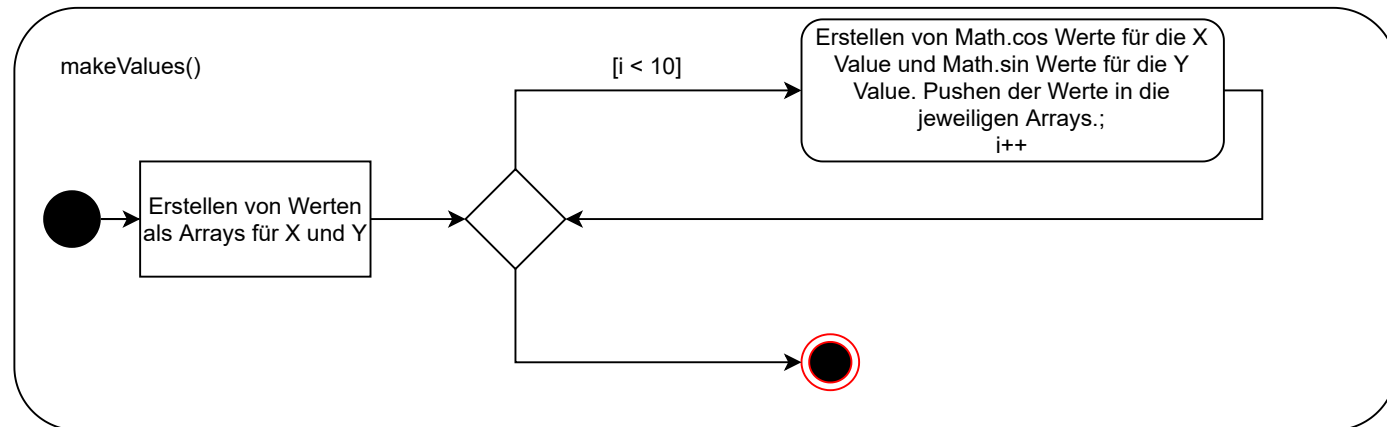
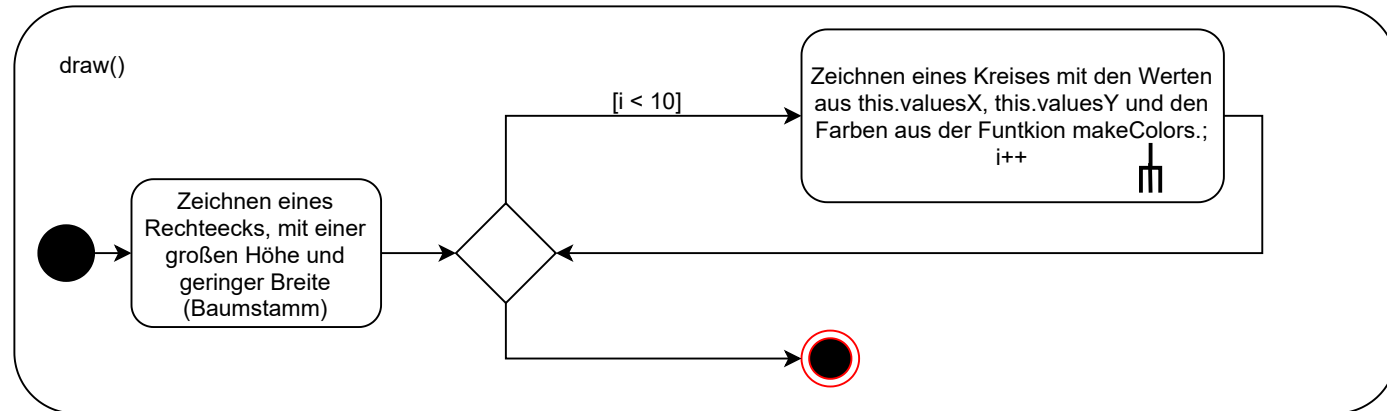
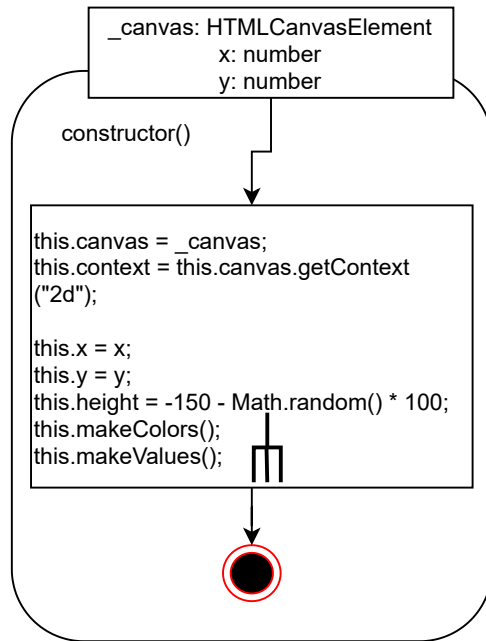
constructor()

```
this.canvas = _canvas;  
this.context = this.canvas.getContext  
("2d");  
  
this.dayNightCycle = new DayNightCycle  
(this.canvas, 70, Mathe.PI, _timeScale);  
  
this.mountain = new Mountain  
(this.canvas, 0, 150, this.canvas.width,  
300)  
  
this.mountain2 = new Mountain  
(this.canvas, 0, 300, this.canvas.width,  
150)  
  
this.makeTrees();  
this.makeFlower();  
this.makeBees();  
this.makeClouds();
```

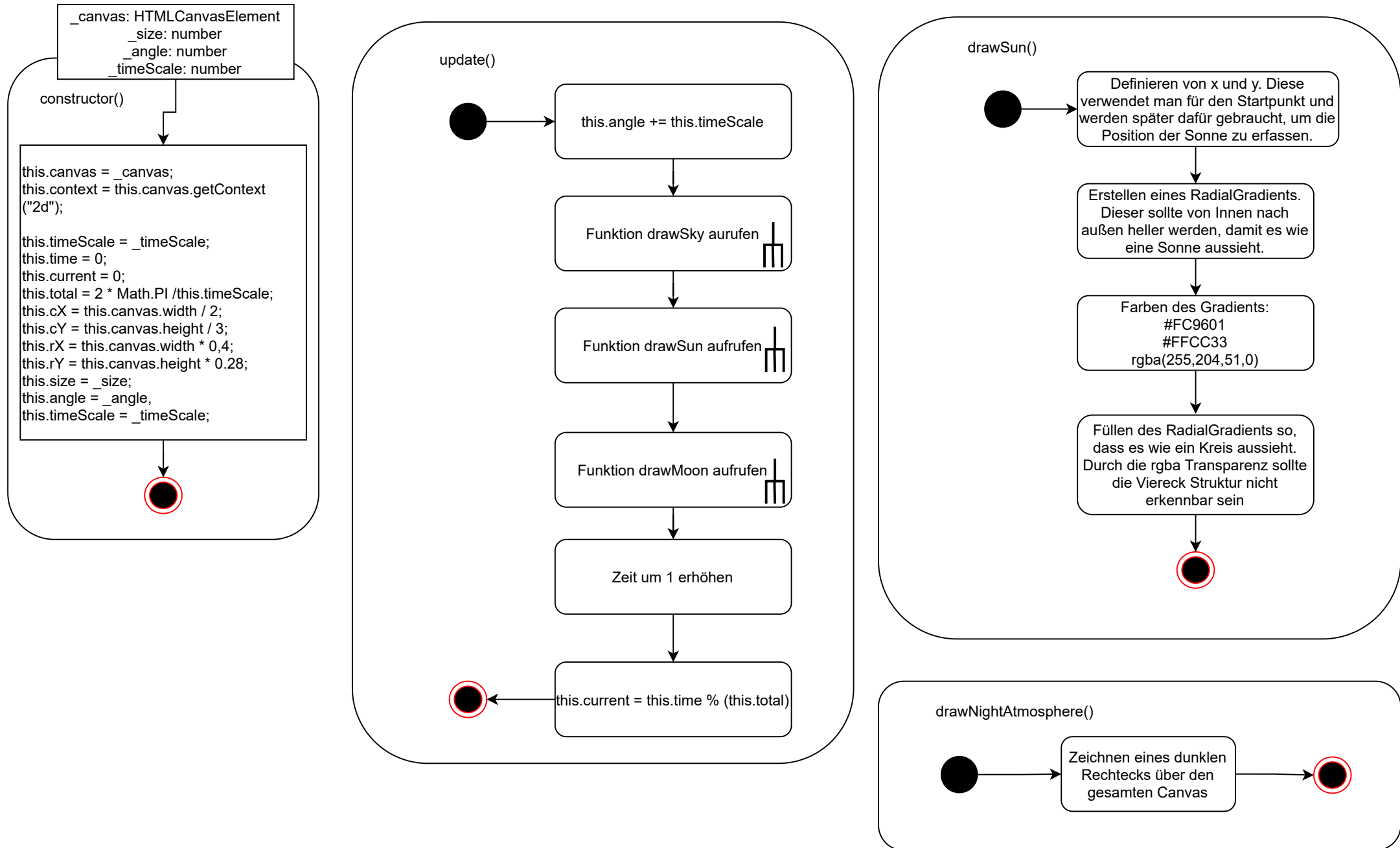


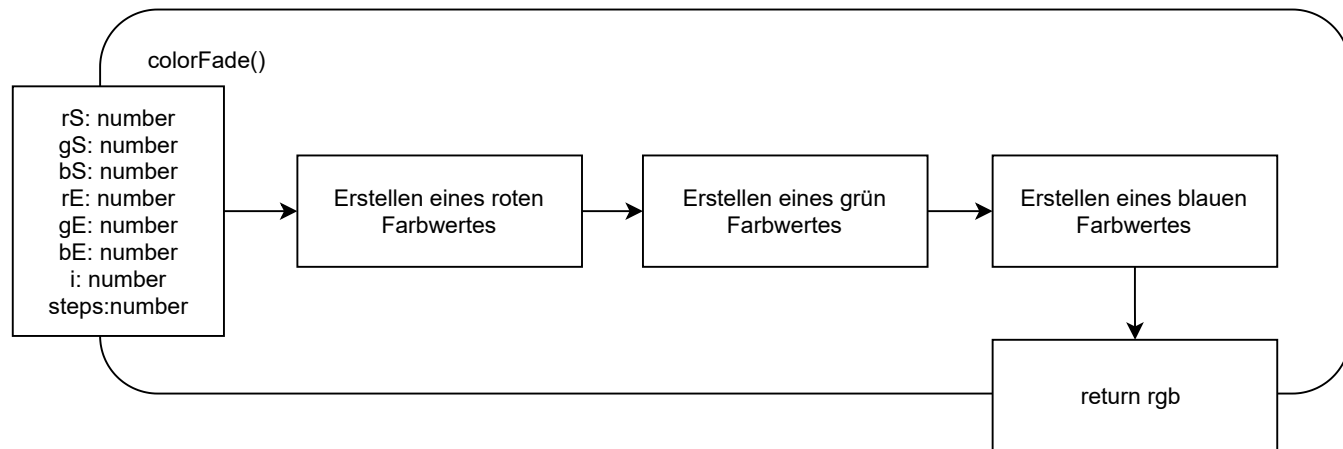
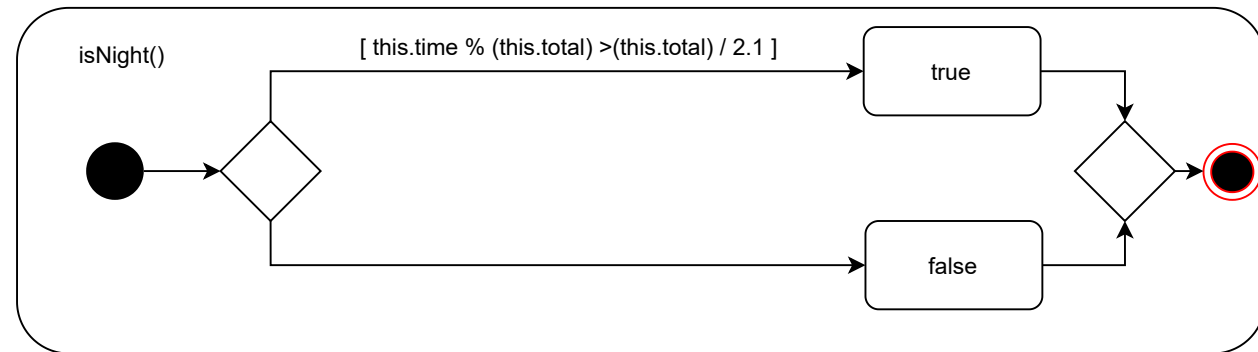
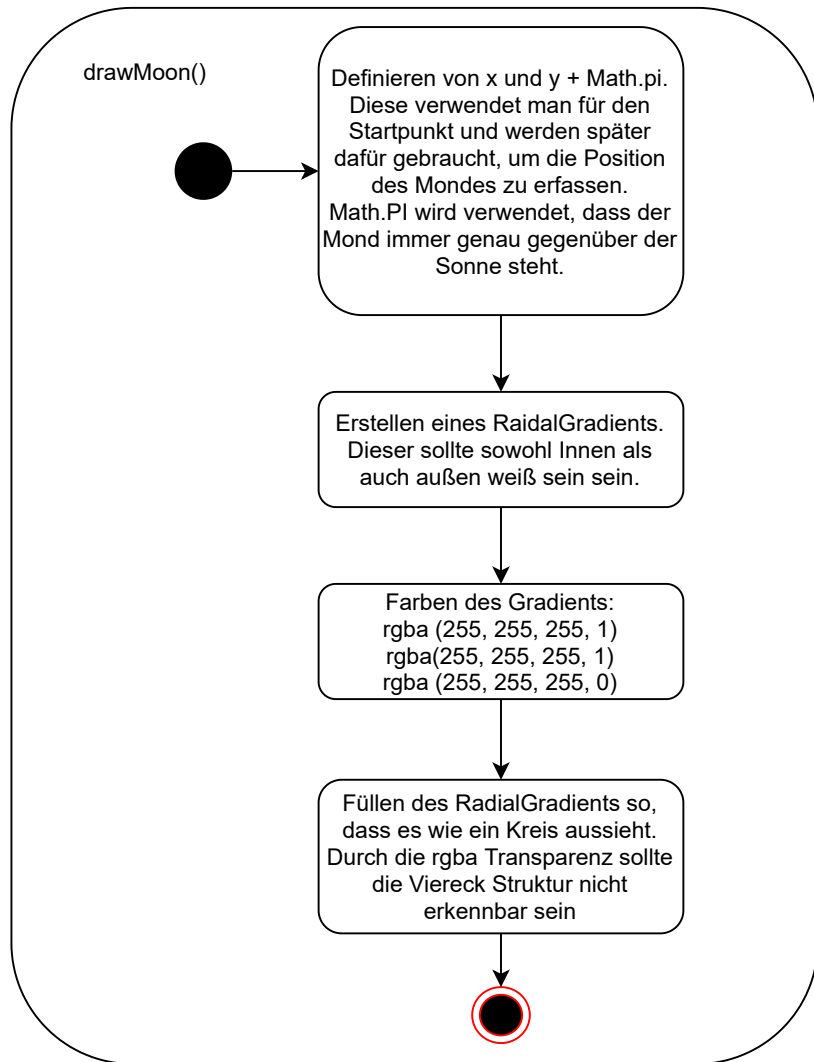


Aktivitätsdiagramm: Trees



Aktivitätsdiagramm: DayNightCycle





drawSky()



Erstellen eines Vierecks
das die obere
Hälfte des Screens
einnimmt

$\text{this.current} > 0 \ \&\& \ \text{this.current} < \text{this.total} * 0.02$

$\text{this.context.fillStyle} =$

`DayNightCycle.colorFade(39, 33, 78, 255, 107, 62, this.time % this.total, this.total * 0.02);`

$\text{this.current} > \text{this.total} * 0.02 \ \&\& \ \text{this.current} < \text{this.total} * 0.6$

$\text{this.context.fillStyle} =$

`DayNightCycle.colorFade(255, 107, 62, 181, 214, 224, this.time % this.total - (this.total * 0.02), this.total * 0.04);`

$\text{this.current} > \text{this.total} * 0.06 \ \&\& \ \text{this.current} < \text{this.total} * 0.44$

$\text{this.context.fillStyle} = \text{"rgb(181, 214, 224)"};$

$\text{this.current} > \text{this.total} * 0.44 \ \&\& \ \text{this.current} < \text{this.total} * 0.47$

$\text{this.context.fillStyle} =$

`DayNightCycle.colorFade(181, 214, 224, 255, 107, 62, this.time % this.total - (this.total * 0.44), this.total * 0.03);`

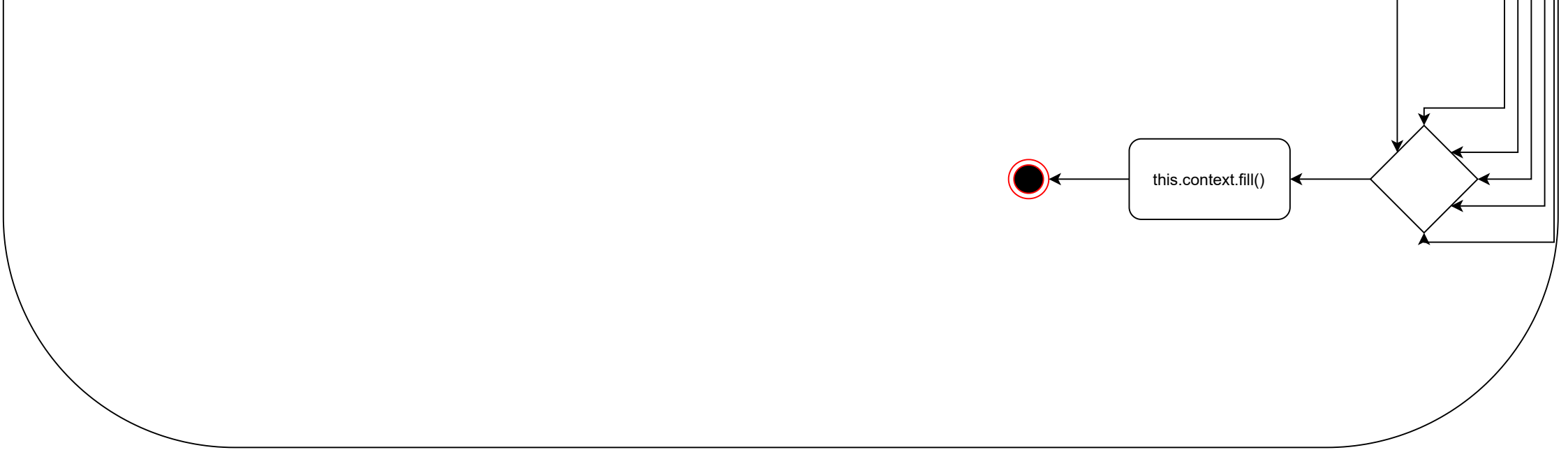
$\text{this.current} > \text{this.total} * 0.02 \ \&\& \ \text{this.current} < \text{this.total} * 0.6$

$\text{this.context.fillStyle} =$

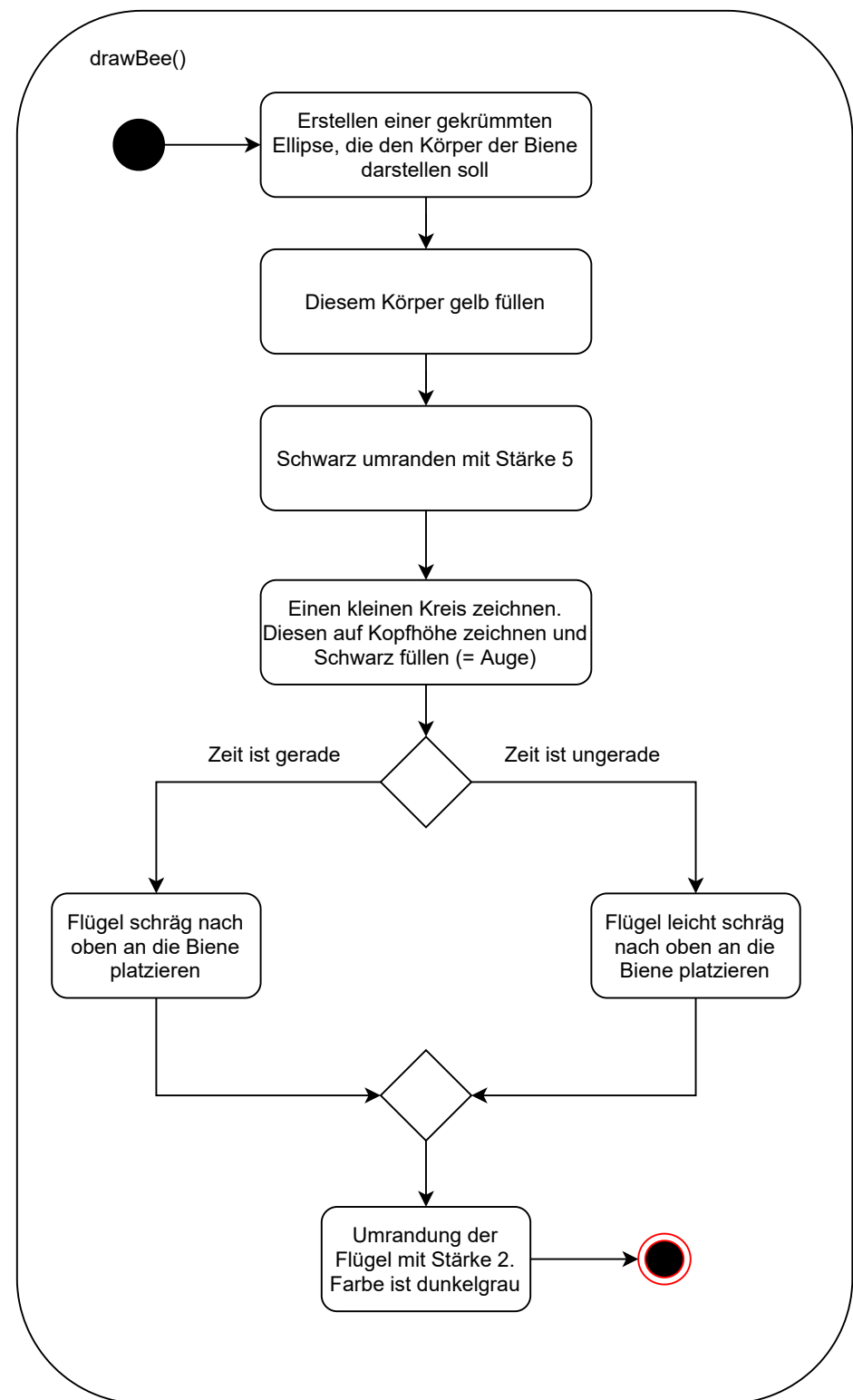
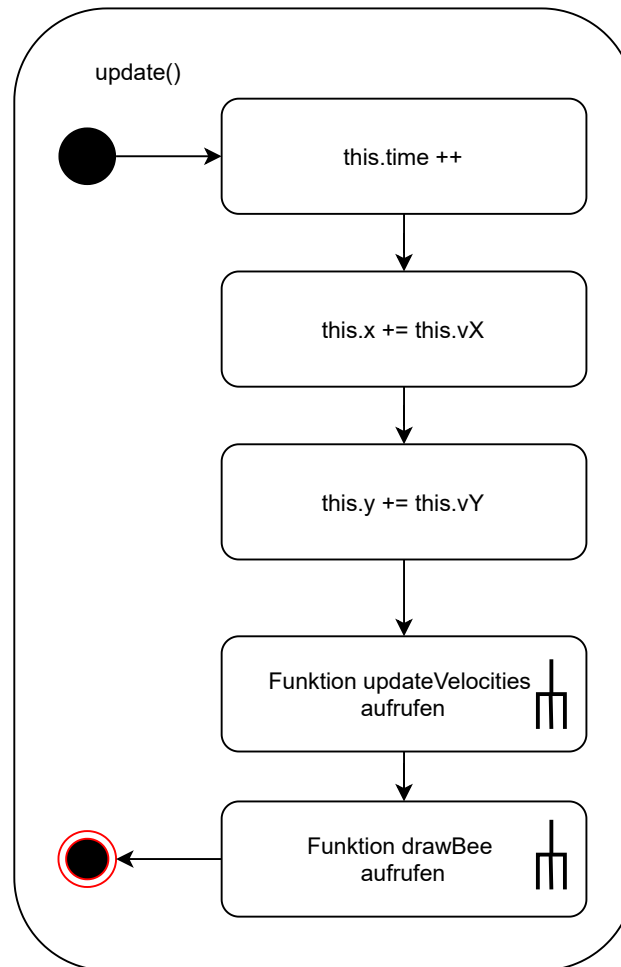
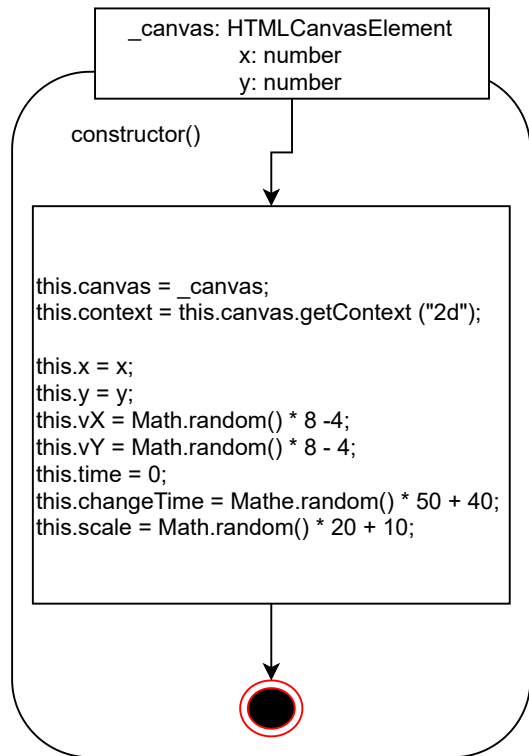
`DayNightCycle.colorFade(255, 107, 62, 39, 33, 78, this.time % this.total - (this.total * 0.47), this.total * 0.03);`

$\text{this.current} > \text{this.total} * 0.5 \ \&\& \ \text{this.current} < \text{this.total}$

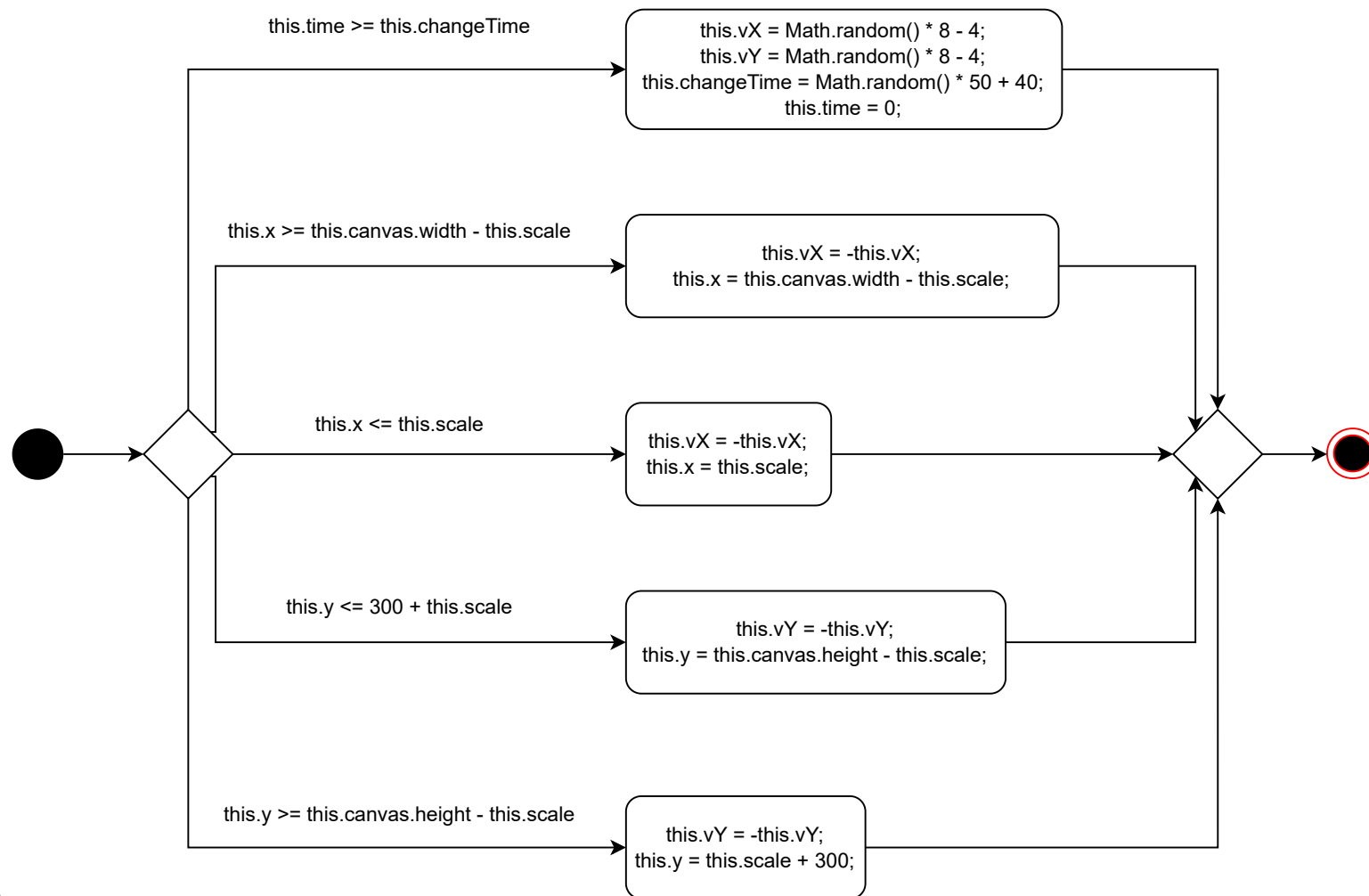
$\text{this.context.fillStyle} = \text{"rgb(39,33,78)"};$



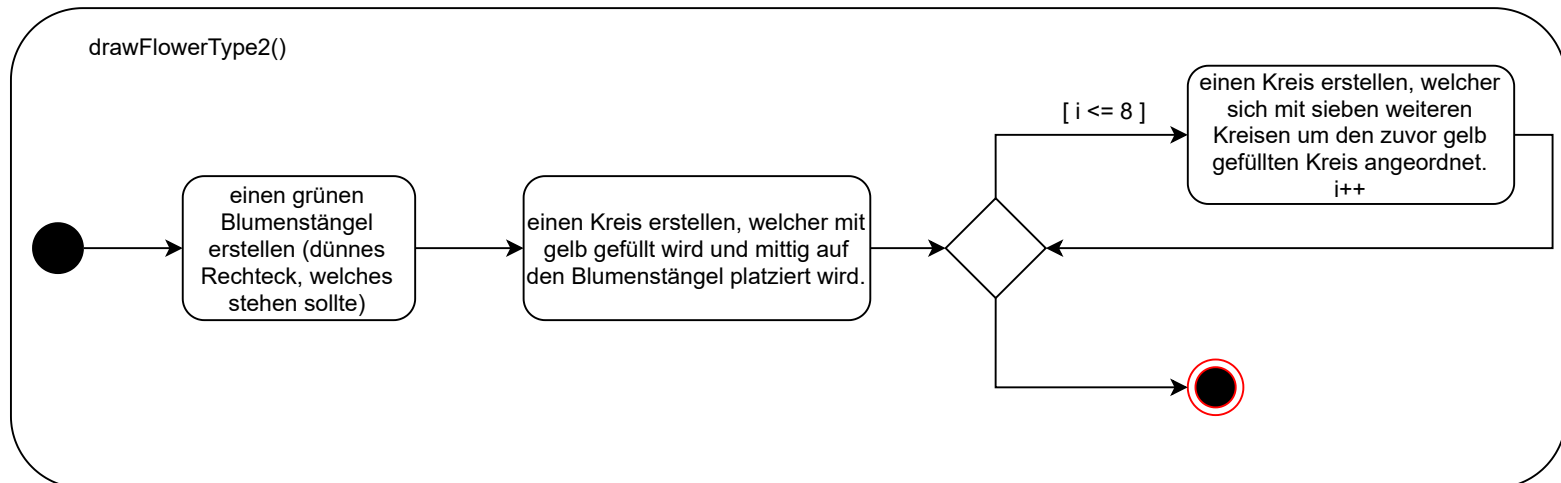
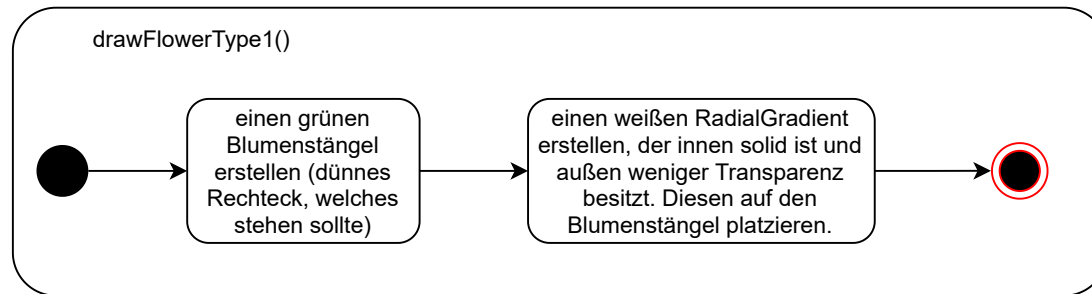
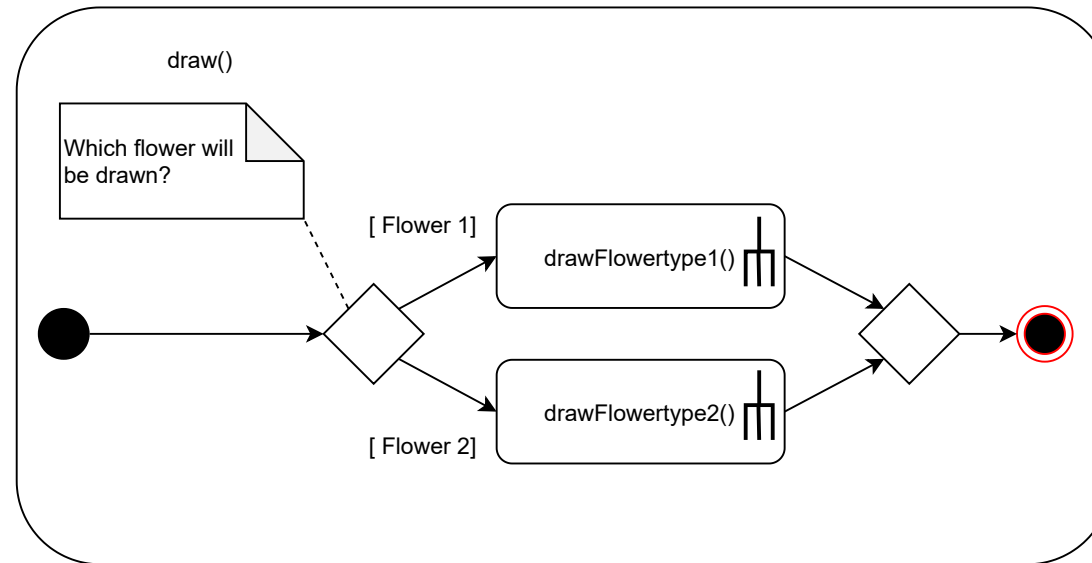
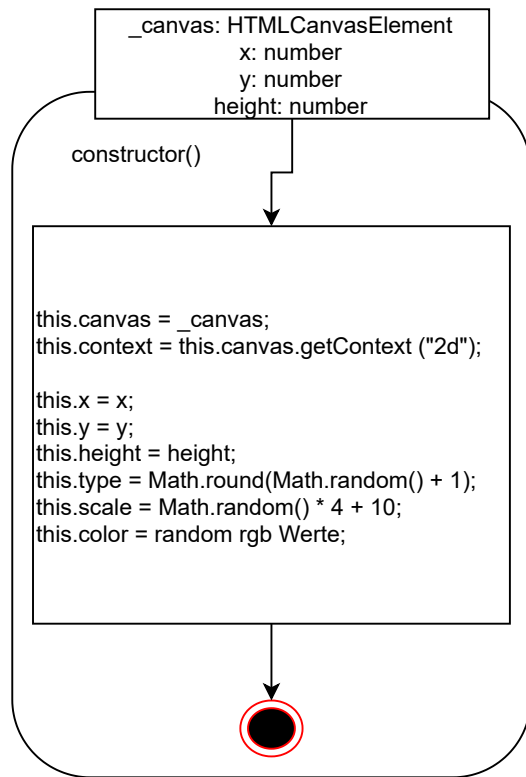
Aktivitätsdiagramm: Bee



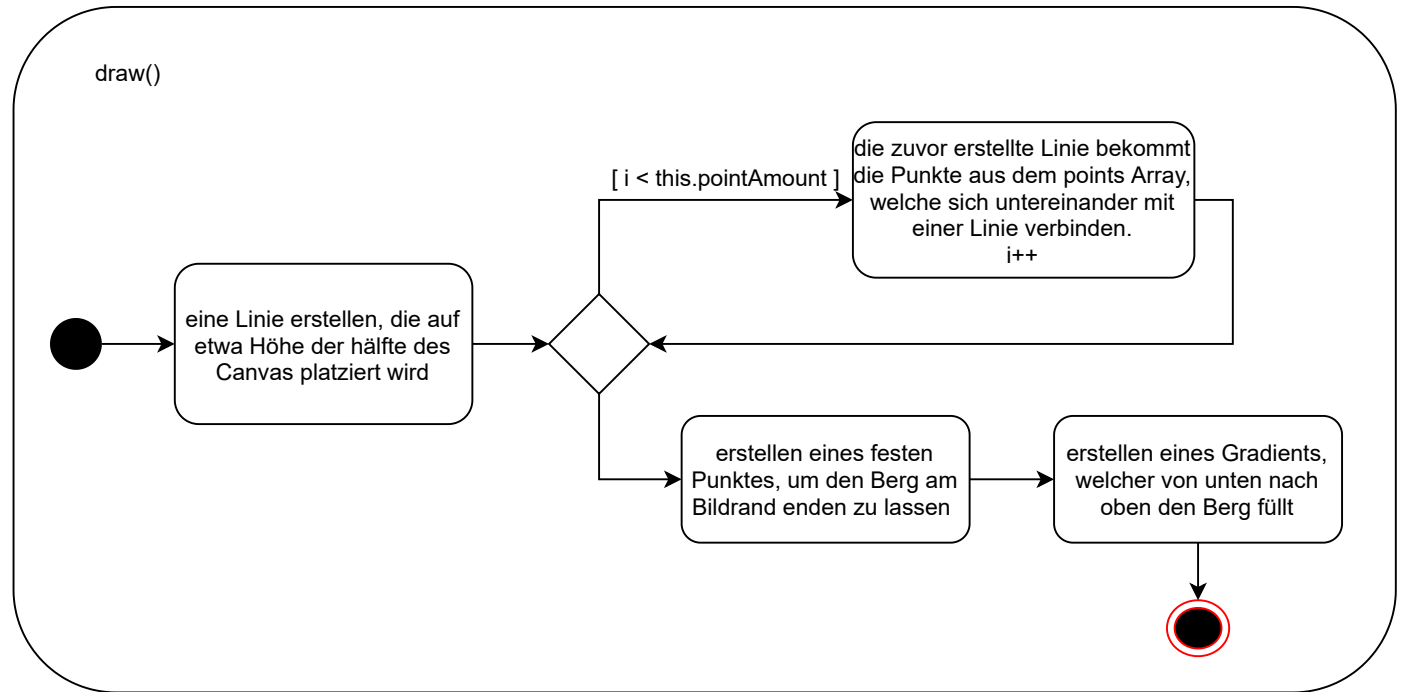
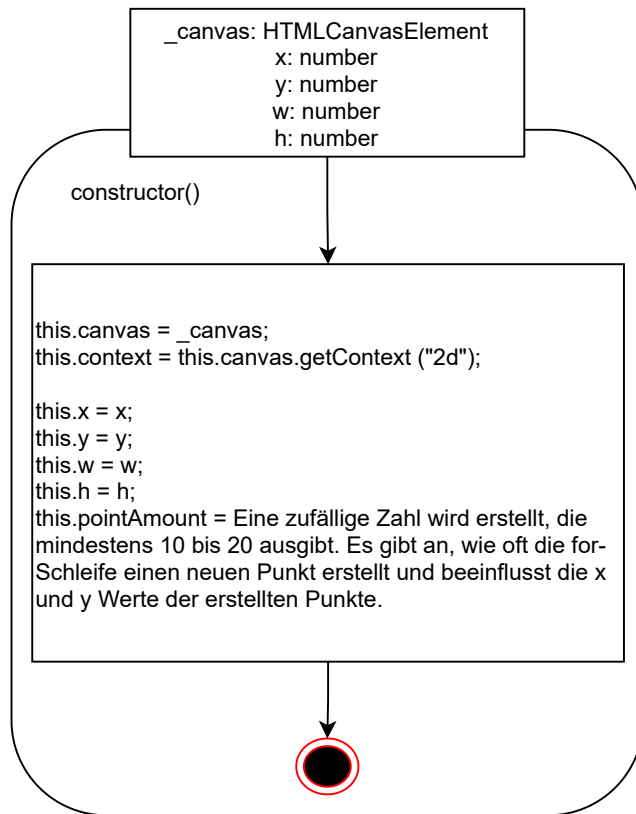
updateVelocities()



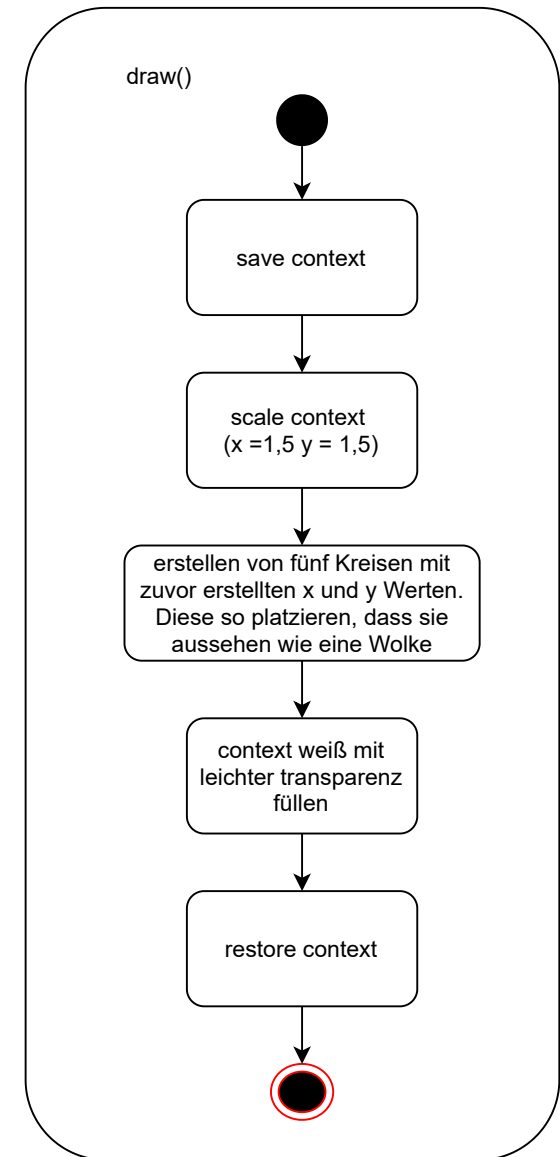
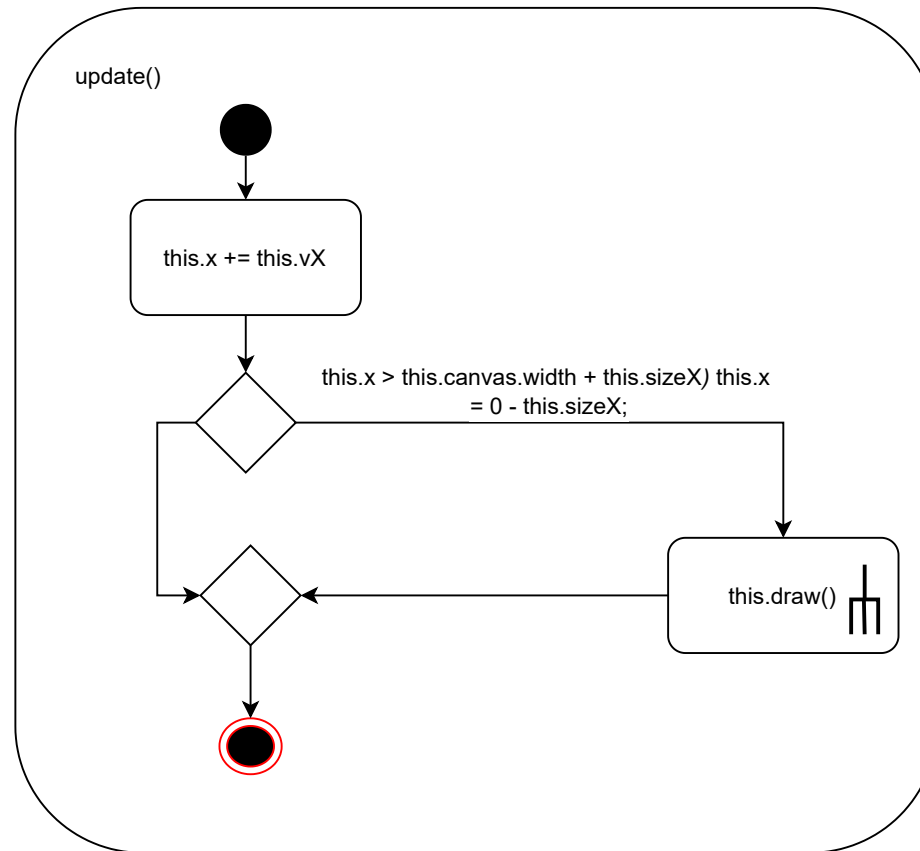
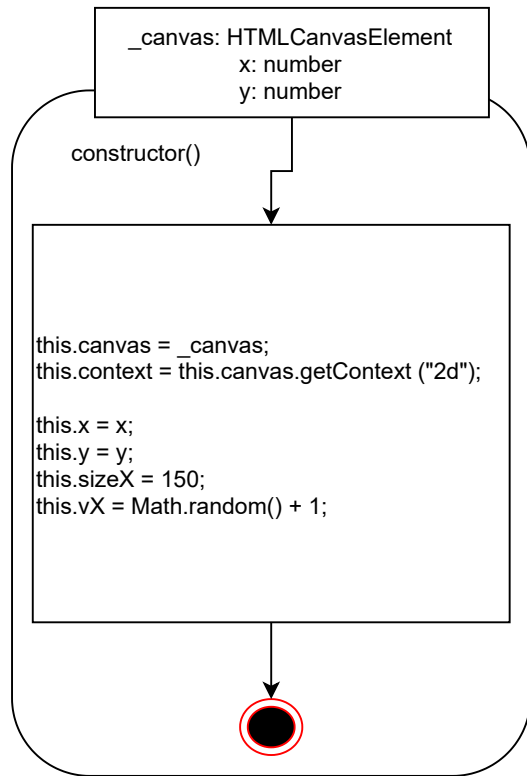
Aktivitätsdiagramm: Flower



Aktivitätsdiagramm: Mountain



Aktivitätsdiagramm: Cloud



Aktivitätsdiagramm: Scene

```
let canvas: HTMLCanvasElement =  
<HTMLCanvasElement>document.getElementById("canvas");  
const timeScale: number = 0.005; (kann angepasst werden,  
beeinflusst die Geschwindigkeit, in der der Tag zur Nacht wird  
und umgekehrt)  
const scene: Scene = new Scene(canvas, timeScale);
```

