export class ~~trees~~ snowflakes

Attribute:

x: number / y: number
dx: number / dy: number
color: string

Animation:
move() : void {
  this.x += this.dx;
  this.y += this.dy; }

Abbildung:

draw(): void {
crc2.fillStyle = this.color
crc2.strokeStyle = ...
path Beginnen
in Canvas abhängig von this.x und this.y zeichnen

---

export class trees

Attribute:
x: number / y: number / color: string

Abbildung:
draw(): void {
crc2.fillStyle = this.color
path Beginnen und Objekt zeichnen

---

export class children

Attribute:
x: number / y: number / dx: number / dy: number / color: string

Animation:
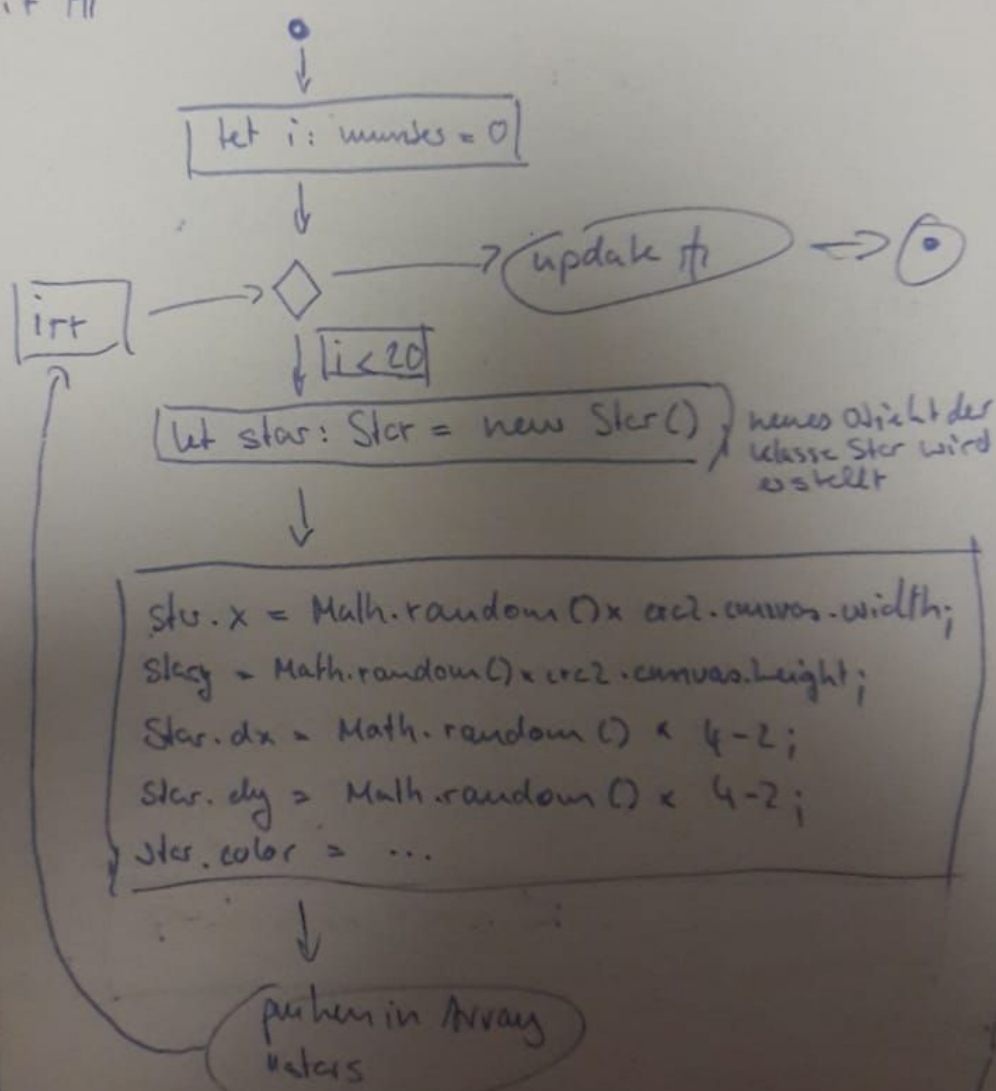move() : void { this.x += this.dx; this.y += this.dy; }

Main.ts

```
let fps: number = 25;
let stars: Star[] = [];
let snowflakes: Snowflake[] = [];
let children: Child[] = [];
```
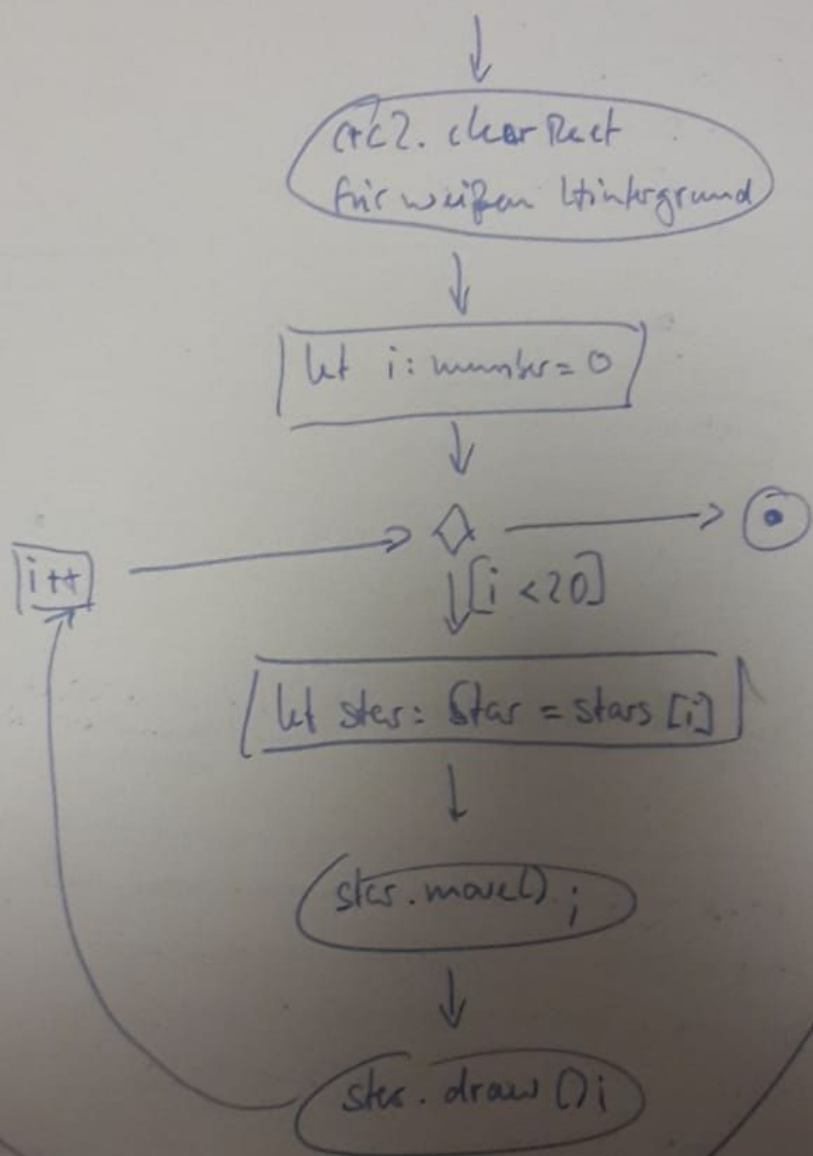
init rh

```
let i: number = 0
```

◇ ──────→ update rh ⟹ ◉

irr

`i < 20`

```
let star: Star = new Star()   neues Objekt der
                              Klasse Star wird
                              erstellt
```

```
stv.x = Math.random() x erct.canvas.width;
star.y = Math.random() x crc2.canvas.height;
star.dx = Math.random() x 4 - 2;
star.dy = Math.random() x 4 - 2;
star.color = ...
```

pushen in Array
stars

→ in der init-Funktion werden die anderen
Schleifen für die anderen Klassen Objekte auch
durchlaufen

update — & window.setTimeout(update, 1000/fps);

↓

ctx2. clear Rect
für weißen Hintergrund

↓

let i: number = 0

↓

i++ ——→ ◇ ——————→ ⊙
        ↓ [i < 20]

let star: Star = stars [i]

↓

star.move();

↓

star.draw()i

auch hier weitere Schleifen für die weiteren klassen Objekte