```
Snowflake          Trees             children1          children2
x : number         x : number        x : number         x : number
y : number         y : number        y : number         y : number
dx : number                          dx : number        dx : number
                                     dy : number        y : number
draw() : void      draw() :          draw() : void      draw() : void
move() : void      void              move() : void      move() : void
```

```
let crc2 = CanvasRenderingContext
let fps : number = 25
let snowflakes : Snowflake[]
let childrenDown : ChildDown[]
let childrenUp : ChildUp[]
```

init() th

●

init

let canvas: HTMLCanvasElement = document.getElementsByTagName[0]

createBackground.ti

~~update Snowflake children down and up~~ (scribbled out)

let i: number = 0

jewels und Children und Trees neu für style

[i > Anzahl des jeweils Objekts] → A

[i < Anzahl des jeweils Objekts]

i++

let NewSnowflake: Snowflake =
new Snowflake();
NewSnowflake.x = Math.random
* crc2.canvas.~~height~~ Width;
NewSnowflake.y = Math.random
* crc2.canvas.~~height~~ height;
NewSnowflake.dx = Math.
random * 4-2;
NewSnowflake.dy = Math.
random() * 4-2;

Snowflakes.push(NewSnow-
flake)

A → update Snowflake.ti
update childrendown.ti
update children up.ti

◎

Create Background

↓

(Himmel erstellen)

↓

(Hügel erstellt)

↓

(Sonne erstellen)

↓

(Wolken erstellen)

↓

(Sonne erstellen)

↓

(Bild Speichern)

↓

(•)

Wird
für
kinder
einmal
ausgeführt

Update Snowflakes / Childrenup / children down

↓

(window.setTimeout(update, 1000/fps))

↓

gespeichertes
Bild
einfügen

(ctx.clearRect(0,0, ctx2.canvas.
width, ctx2.canvas.height)

↓

[let number = 0]

↓

◇ — [i < Anzahl] — (•)

let flake:
Snowflake=
Snowflakes[i] → (flake.move();
flake.draw();) → (i++)