

Goldener Herbst Class Diagram

CanvasRenderingContext

cloud

posX: number,
posY: number,
velocityX: number,
velocityY: number

constructor (- position vector)
draw(): void,
update(): void

Leaf

posX: number,
posY: number,
velocityX: number,
velocityY: number,
~~counter: number~~,
Type: number,
constructor (- Type: number,
 - position vector)

draw(): void,
update(): void

Activity MAIN

```
let circ2: CanvasRenderingContext2D;  
let backgroundPosition: number = 0.5;  
let imageData: ImageData;  
let clouds: Cloud[] = [];  
let leaves: Leaf[] = [];  
interface Vector for x & y
```

install load-listener

load

handleLoad

handleLoad

```
let canvas: HTMLCanvasElement;
```

```
drawBackground(),  
drawSun(),  
drawMountains((x, y, x2, y2) => ...),  
drawTree1(position),  
drawTree2(position),  
drawBush(position),  
drawSquirrel(squirrel),  
imageData = circ2.getImageData(...),  
createCloud(),  
createLeaves(),  
animate(),
```


Activity - MAIN

create Cloud

clouds.push(newCloud({x: crc2.canvas * width * 0, y: crc2.canvas * height * 0.5})),
\$ex new position



create leafs



let number = 0, i < 20, i++

let randomleaf: number = Math.floor(Math.random() * 2)
let leaf: leaf = newLeaf

leaf.push(leaf)

draw random leaf



Activity: Clouds

posX: number,
posY: number,
velocityX: number = 0.13,
velocityY: number = 0.13,

constructor

position Vector

this.posX = position.x
this.posY = position.y

draw

save transform

translate to -position

restore transform

update

[this.posX > 400 || this.posY < 0]
|| this.posX < 0

this.velocityY =
-this.velocityY

[this.posX > 400 || this.posY < 0 ||
this.velocityY < 0]

this.velocityX =
-this.velocityX

this.posX += this.velocityX
this.posY += this.velocityY
this.draw()

Activity: LEAF

pos.x: number,
pos.y: number,
velocity.x: number,
velocity.y: number,
type: number

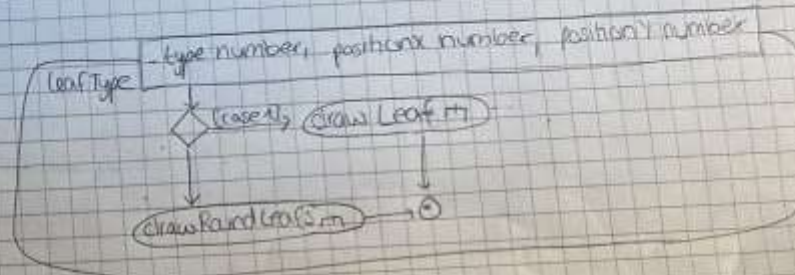


- velocity: vector

constructor

position: vector, type: number

this.pos.x = position.x,
this.pos.y = position.y,
this.velocity.x = velocity.x,
this.velocity.y = velocity.y,
this.type = type;



* drawLeaf & drawRoundLeaf in Aufgabe 8.2