

Klassendiagramme

interface vector

x: number
y: number

canvas rendering context

koaf

positionY: Number
positionY-1: Number
velocityX: Number
velocityY: Number
type: Number

+constructor (position: Vector,
velocity: Vector,
type: number)

+draw()

+update()

cloud

+posX: number
+posY: number
+velocityX: number
+velocityY: number
+constructor (position: Vector)
+draw()
+update()

squirrel

+posX: number
+posY: number
+velocityX: number
+velocityY: number

+constructor()
+draw()
+update()

AD Main

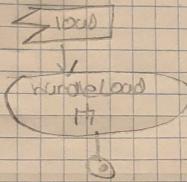
handleLoad



```
let canvas: HTMLCanvasElement  
HTMLCanvasElement>document.querySelector
```



```
cr2 = canvas.getContext("2d")  
let horizZon: number  
let posMountain: Vector  
  
drawBackground(), m  
drawLeaves(), m  
drawSquirrels(), m  
drawClouds(), m  
imageData = cr2.getImageData  
animate() m
```



drawBackground

```
let horizon: number  
let posMountains: Vector
```

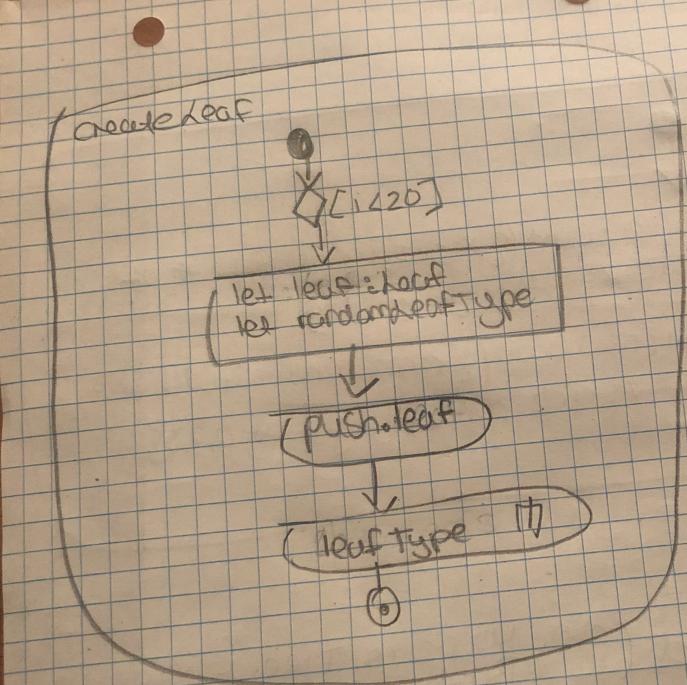


```
drawBackground  
drawSun({x: 570, y: 553})  
drawMountains(posMountains, 75, 150, color)  
drawMountains(posMountains, 50, 110, color)  
drawTree({x: 0, y: 392, z: 100, y: 100}, 250, 300)  
drawEvergreen({x: 100 + random(-1536, 1536), y: 100 + random(120, 220)})
```



```
drawBush({x: 300, y: 300}, {x: 70, y: 70}, 250, 300)
```

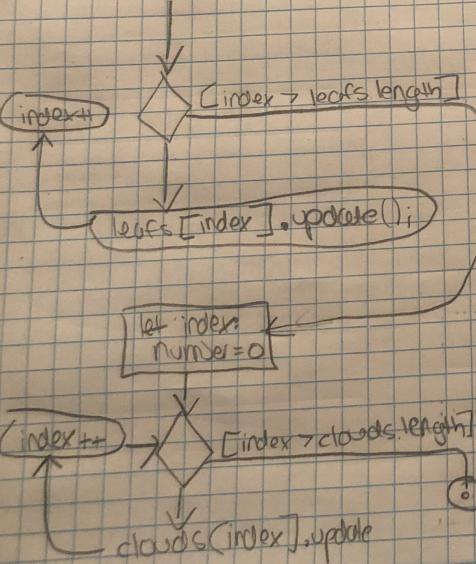




A1 Main

animate

```
requestAnimationFrame(animate)  
crc2.putImageData  
crc2.clearRect
```



create cloud

```
clouds.push(new Cloud({x: crc2.canvas.width * 0.07, y:  
crc2.canvas.height * 0.10}))
```

```
clouds.push(new Cloud({x: crc2.canvas.width * 0.10,  
y: crc2.canvas.height * 0.23}))
```

A1 Background (sun, Mountains, Tree, Evergreen, Bush)

→ siehe L_09_02 Goldenes Herbst

AD Cloud

```

public posX: number
public posY: number
public velocityX: number
public velocityY: number
  
```

constructor
position: vector

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

draw

```
let gradient: ConvexGradient
```

circle.setScale(1);
circle.translate(this.xPos, this.yPos)

cloud.draws
circle.setOutline()

update

[this.posX >
circle.convex.
width ||
this.posY < 0]

[this.posY > circle.convex.height ||
this.posY < 10]

[this.velocityX >
this.velocityY]

[this.velocityY > -this.velocityX]

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

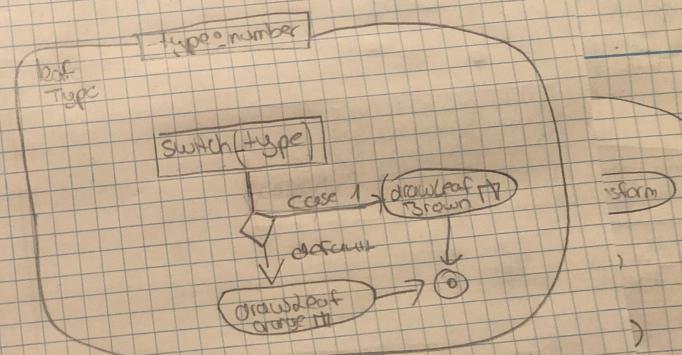
AD Leaf

```
posX: number  
posY: number  
velocityX: number  
velocityY: number  
type: number
```

constructor

```
position: vector, - velocity: vector, - type: number
```

```
this.posX = position.x  
this.posY = position.y  
this.velocityX = velocity.x  
this.velocityY = velocity.y  
this.type = type
```



AD Leafs

→ Seite L-06.02 Goldener Herbst

A1 Squirrel

```
public pos x : number  
public pos y : number  
public velocity x : number  
public velocity y : number  
public random scale x : number  
public random scale y : number  
public counter : number  
public random number : number
```

constructor

```
this.pos x = position.x  
this.pos y = position.y  
this.random scale x = random scale x  
this.random scale y = random scale y  
this.velocity x = velocity.x  
this.velocity y = velocity.y
```

draw

```
(crc2. reset Transform)  
↓  
(crc2. save())  
↓  
crc2. translate  
↓  
(crc2. begin path)  
↓  
(crc2. ellipse())  
↓  
(crc2. close Path())  
↓  
(crc2. fill())  
↓  
①
```

Update

