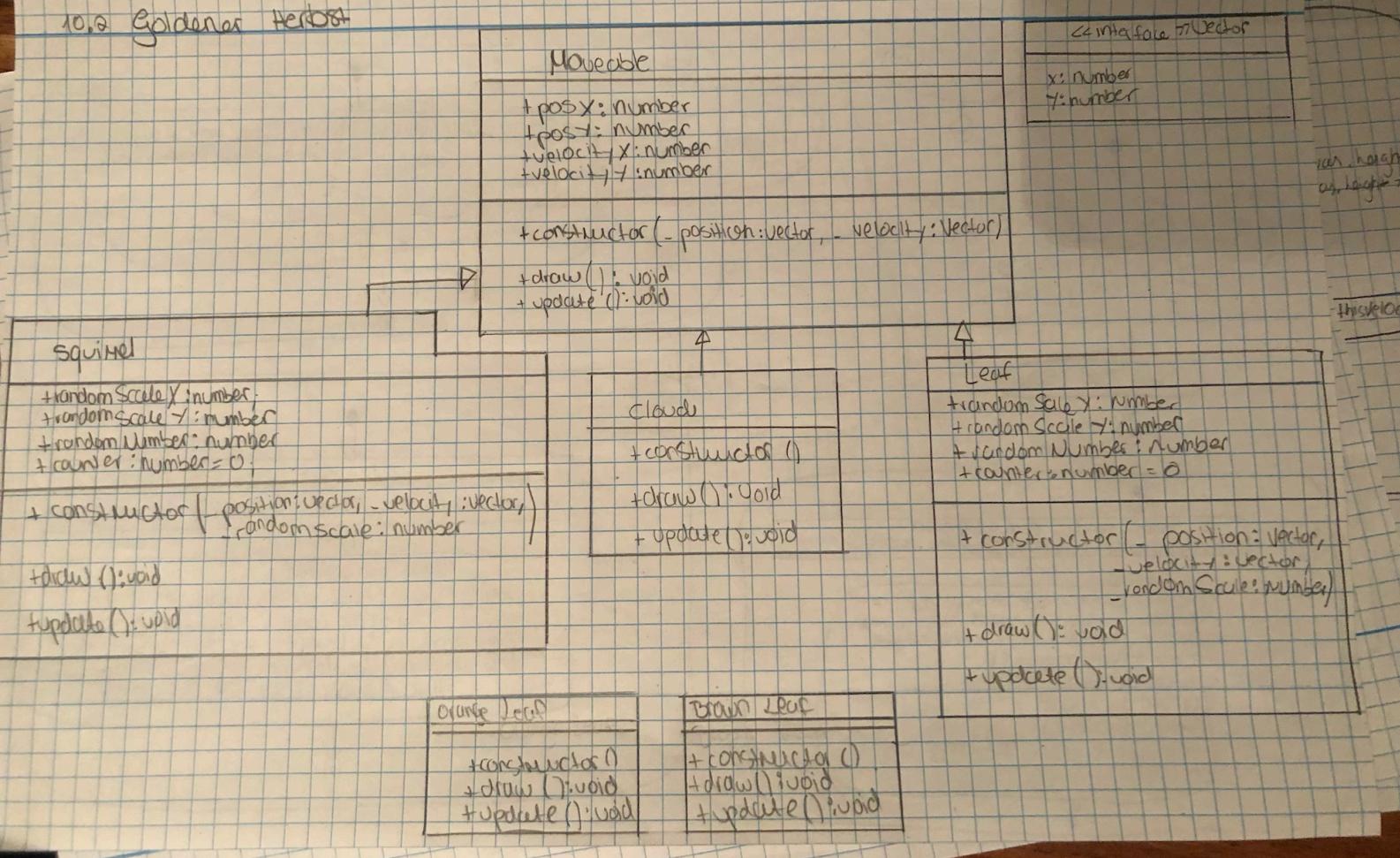
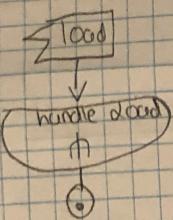


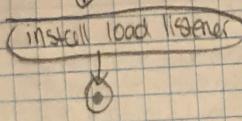
10.2 Goldenor Herbst



AD Main



```
let crc2: CanvasRenderingContext2D = document.querySelector("canvas").getContext("2d");
let goldenCut: number = 0.62;
let leafs: Leaf[] = [];
let squirrel: Squirrel = new Squirrel();
let image: ImageData = new ImageData();
```



handle load

```
let canvas: HTMLCanvasElement = document.querySelector("canvas");
let crc2: CanvasRenderingContext2D = canvas.getContext("2d");
let width: number = window.innerWidth;
let height: number = window.innerHeight;
createBackground();
createLeafs();
createSquirrel();
createCloud();
moveData = crc2.getImageData(0, 0, width, height);
```

create cloud

```
moveableCloud.push(newCloud({x: crc2.canvas.width * x / 1, y: crc2.canvas.height * y / 3, x2: y, y2: y}))
```

* 4

create background

```
let horizon: number = crc2.canvas.height * goldenCut;
let posMountains: Vector = {x: 0, y: horizon};
```

drawBackground()

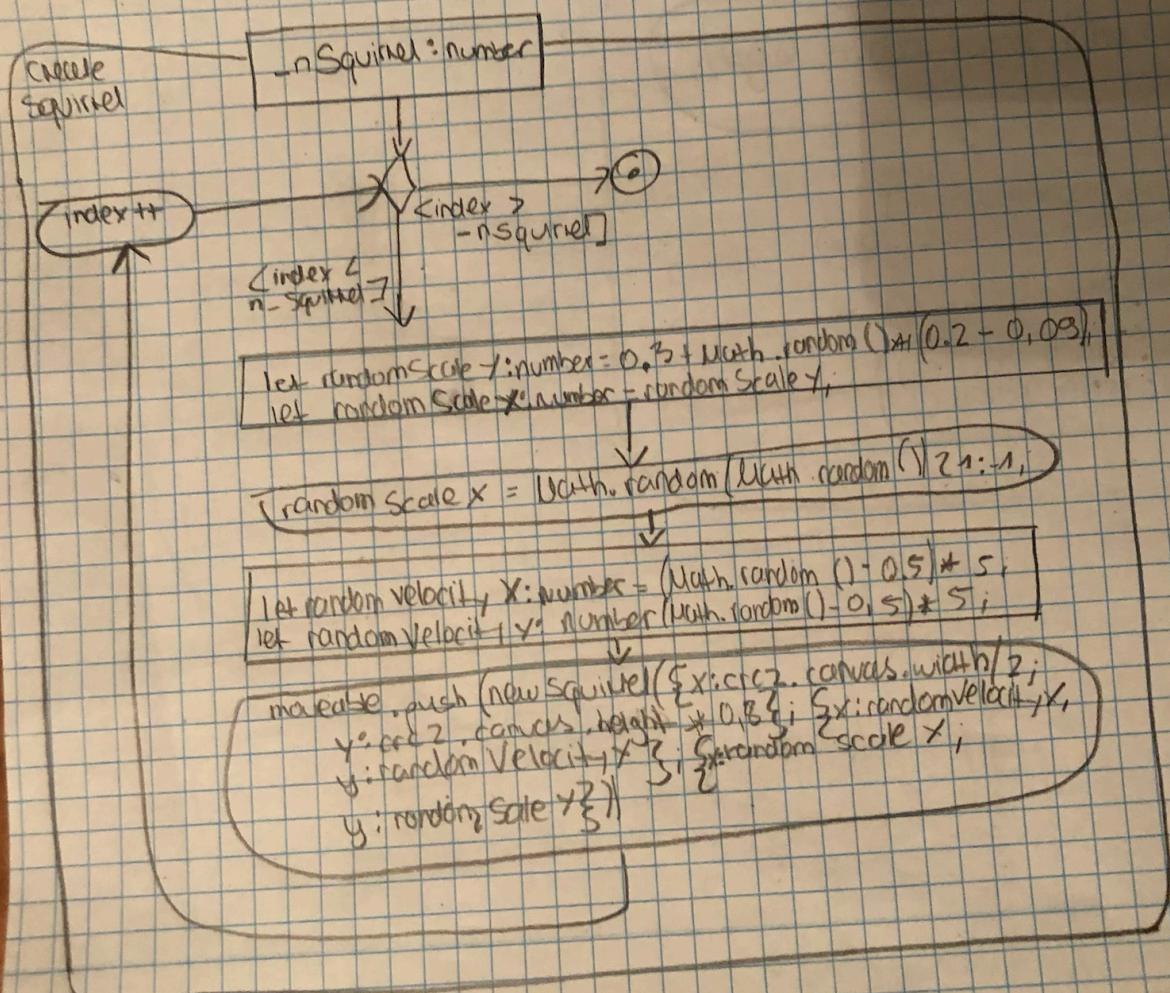
```
drawSun({x: crc2.canvas.width * 0.25, y: crc2.canvas.height * 0.95})
```

```
drawMountains(posMountains, 75, 200, "brown", "grey")
```

```
drawMountains(posMountains, 50, 150, "grey", "white")
```

```
drawEvergreen(-30, -50, 0.6, 0.5)
```

AD Main

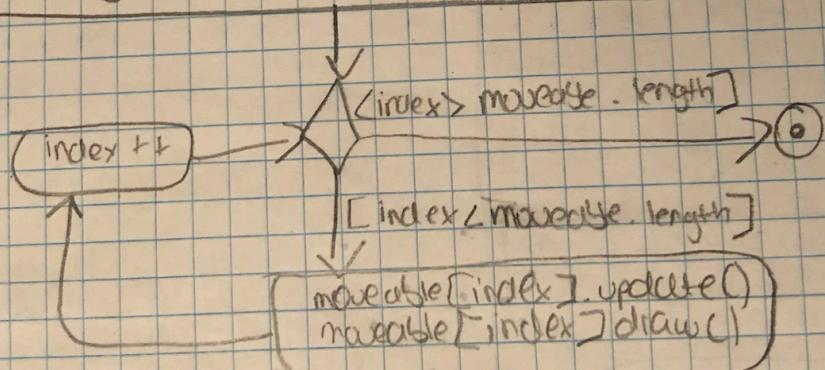


Leaf

AD MAIN

animate

```
crc2.clearRect(0,0,crc2.canvas.width, crc2.canvas.height);  
crc2.putImageData()
```



AD Leaf

AD Main

AD

or

createLeaves

nLeaf: number

index++

[index > nLeaf] → ①
[index ≤ nLeaf]

let randomScaleY: number = 0.12 + Math.random() * (0.2 - 0.01);
let randomScaleX: number = randomScaleY;

randomScaleX ← Math.round(Math.random() ? 1 : -1);

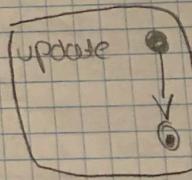
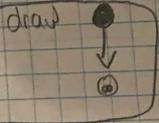
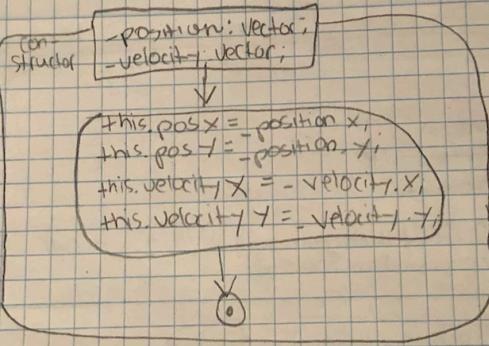
let randomVelocityX: number = ((Math.random() - 3) * 10,
let randomVelocityY: number = ((Math.random() - 3) * 10,

moveable.push(newLeaf({x: crc2.canvas.width, y:
crc2.canvas.height * 0.2}; {x: randomVelocityX,
y: randomVelocityY}; {x: randomScaleY, y: randomScaleY / 3}))

AD Leaf

AD Moveable

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



AD Squid

```
public randomScaleX:number;
public randomScaleY:number;
public randomNumber:number;
public counter:number
```

draw

```
crc2.save();
crc2.translate(this.posX + this.posY);
crc2.scale(this.randomScaleX, this.randomScaleY);
drawing();
crc2.restore();
```

Constructor

```
-position:Vector
-velocity:Vector
-randomScale:Vector
```

```
super(position, velocity)
this.randomScaleX = -randomScaleX
this.randomScaleY = -randomScaleY
```

update

```
[this.posX > crc2.canvas.width ||  
this.posX < 0]
```

```
this.velocityX =  
-this.velocityX
```

```
[this.counter =  
this.randomNumber]
```

```
this.velocityX =  
-this.velocityX  
this.velocityY =  
-this.velocityY
```

```
this.counter = 0  
(randomNumber =  
number = (Math.floor  
(Math.random()) * 300)  
100);
```

```
[this.posX > crc2.canvas.height ||  
this.posY < crc2.canvas.height - 0.8]
```

```
this.velocityY =  
-this.velocityY
```

```
this.posX += this.velocityX  
this.posY += this.velocityY  
counter++  
this.draw();
```

A) Leaf

```
public randomScaleX:number;  
public randomScaleY:number;  
public randomNumber:number;  
public counter:number
```

draw

```
crc2.save();  
crc2.translate(this.posX, this.posY);  
crc2.setScale(this.randomScaleX, this.randomScaleY);  
drawingOrangeLeaf(), drawingBrownLeaf();  
crc2.restore();
```

constructor

```
-position:vector  
-velocity:vector  
randomScale:vector
```

```
super(position, velocity);  
this.randomScaleX = -randomScaleX  
this.randomScaleY = -randomScaleY
```

update

```
[this.posX >  
crc2.canvas.width]||  
[this.posX < 0]
```

```
[this.velocityX =  
-this.velocityX]
```

```
[this.counter =  
this.randomNumber]
```

```
[this.velocityX =  
-this.velocityY;  
this.velocityY =  
-this.velocityX;  
this.counter = 0]
```

```
[randomNumber =  
number = (widthFor  
(width + random) * 800)  
500];
```

```
[this.posX += this.velocityX  
this.posY += this.velocityY  
counter++  
this.draw();]
```