

Klassendiagramm Krabbe:

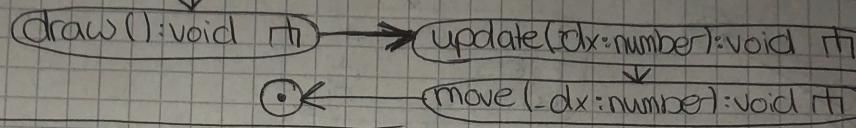
Class Krabbe

x: number
y: number
dx: number
dy: number

→ Krabbe soll sich von rechts nach links bewegen & Richtung wechseln, wenn sie den Rand erreicht.



AD Move Funktion Klasse Krabbe:



move [-dx: number]

this.x += this.dx;
this.y += this.dy;

[this.x <= 0 || this.x >= 1000]

this.dx = this.dx * -dx

Klassendiagramm Fisch 1:

Class Fisch

x: number
y: number
dx: number
dy: number

→ Fisch bewegt sich in zufällige Richtung & soll wenn er den canvas verlässt, in der Mitte wieder angezeigt werden

AD Move Funktion für Fisch 1:

move

-x: number
-y: number

[this.x <= 0 || x >= 1000 || y <= 0 || y >= 700]

this.x += this.dx
this.y += this.dy

this.x = -x
this.y = -y

Ablauf Canvas.ts

DOMContentLoaded

init() →

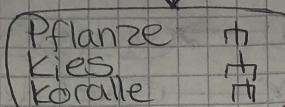
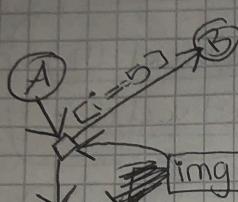
then

bitte laufe nach

canvas = HTMLCanvasElement
fishArray = Fisch[] = []
crabArray = Krabbe[] = []
fps = number = 30
img = ImageData

init

let canvas = getElementByTagName("canvas")



```

let i: number = 0
let y: number = random() * canvas.height
let x: number = random() * canvas.width
let dx: number = random() * 10 - 5
let dy: number = random() * 10 - 5
let fish: Fisch
    
```

→ fish = new Fisch()

\downarrow
fish.x = x
fish.y = y
fish.dx = dx
fish.dy = dy
fishArray.push(fish)

A

i++

fish.draw() ⚡

praktisch zufällig die Größe und Farbe ist

B ~~Eine Fische~~ ~~ein Krabbe~~ ~~ein Fisch~~ ~~ein Krabbe~~ ~~ein Fisch~~ ~~ein Krabbe~~

```

let i: number = 0
let y: number = random() * canvas.height
let x: number = random() * canvas.width
let dx: number = 0
let dy: number = random() * -5;
    
```

~~let~~ crab: Krabbe

```

crab = new Krabbelt()
crab.x = x
crab.y = 650
crab.dx = dx
crab.dy = dy
    
```

crabArray.push(crab) → crab.draw():

i++

update

window.setTimeout(update, 1000 /fps)

crc.clearRect

crc.putImageData(img 0,0)

i > fishArray.length

fishArray[i].update(fishArray[i].x, fishArray[i].y) i

i++

[fishArray[i].x <= 0 || x >= 1000 ||
.y <= 0 || y >= 700]

fishArray[i].update(500, 350)

i++

i: number = 0

CrabArray[i].update(CrabArray[i].dx);

[CrabArray[i].x <= 0 || x >= 1000]

(CrabArray[i].update(-1))