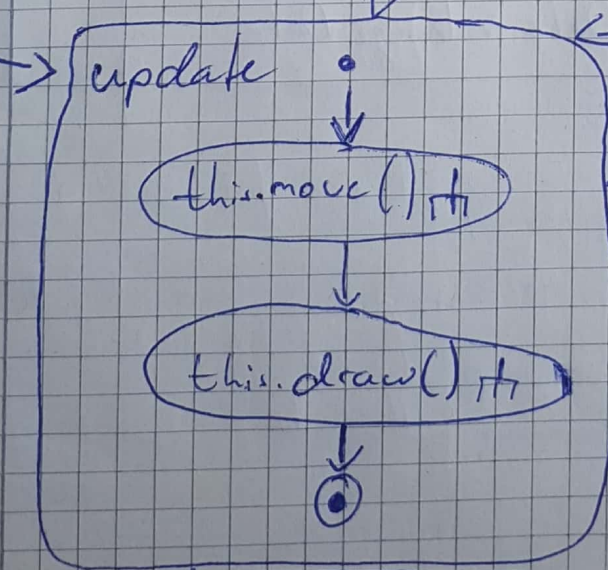
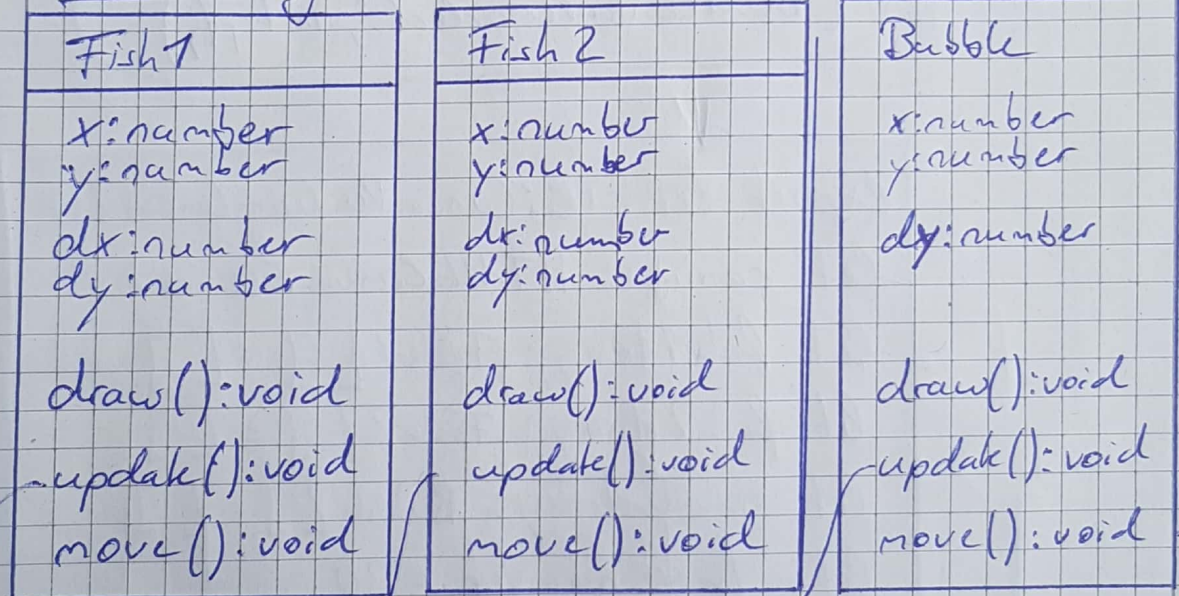
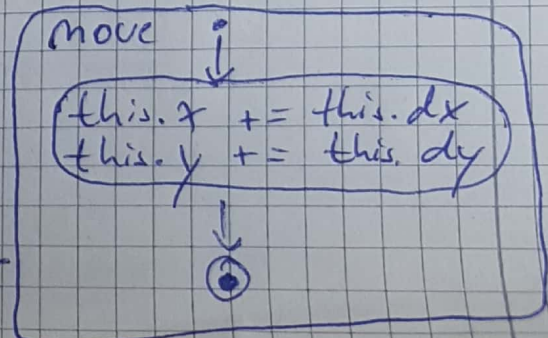


Aufgabe 11 - Animation - Konzept

Klassendiagramme

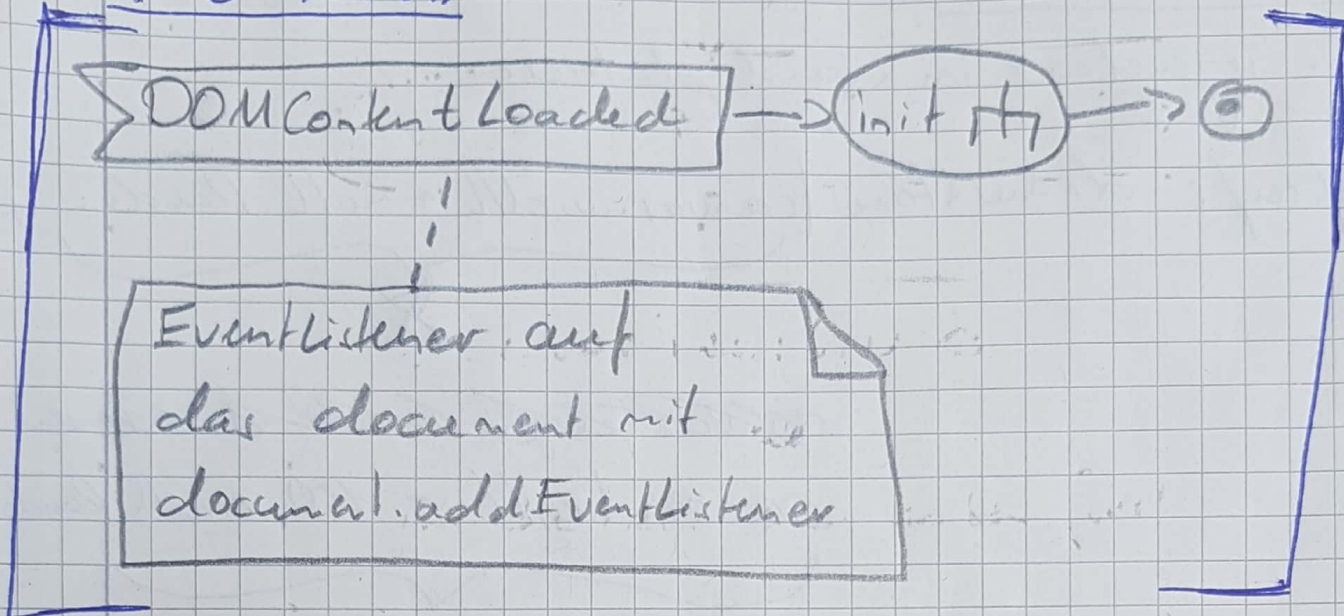


update Funktionen
in den jeweiligen
ts-Dateien von
Fish1, Fish2 und Bubble.

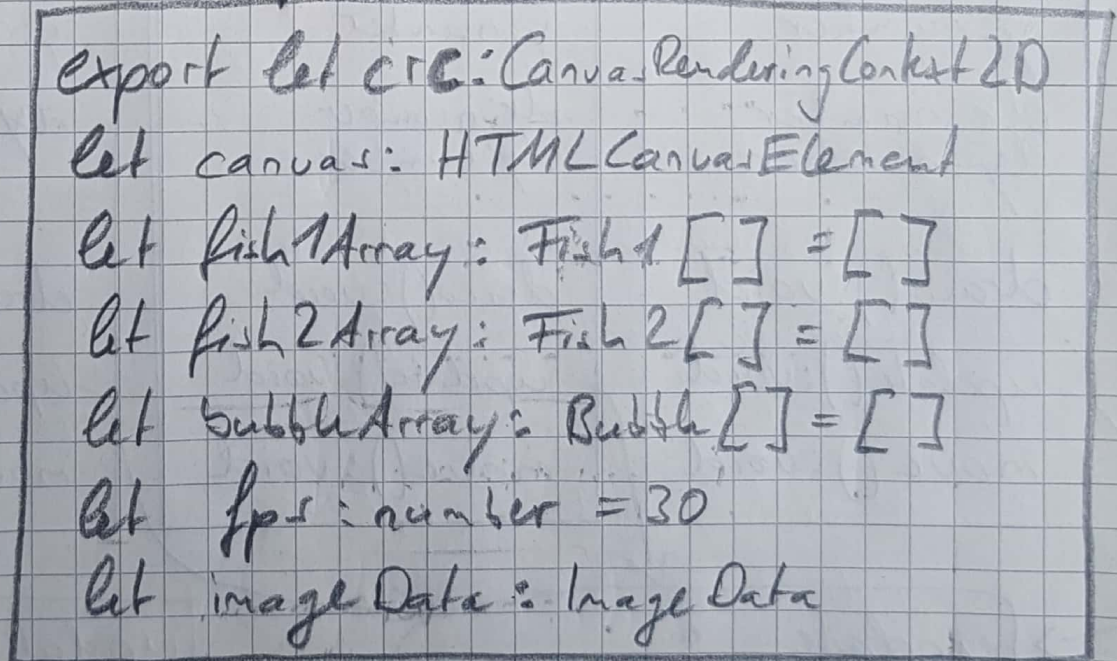
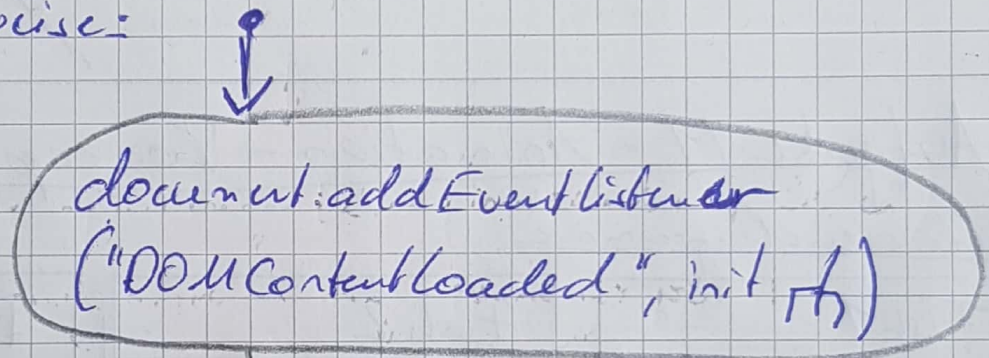


move Funktion zu
den jeweiligen
ts-Dateien

AD-Main



Beziehungswiese:



init

```
canvas = document.getElementById("canvas")  
cnc = canvas.getContext("2d")
```

AD nicht
vorhanden, da
es lediglich den
Code der
vorherigen Aufgabe
enthält

drawBackground()

```
imageData = cnc.getImageData(0, 0, canvas.width,  
canvas.height)
```

```
let i = number = 0; i < 10; i++
```

update()

[i > 10]

[i < 10]

for-Schleifen
für die
Bubbles und
ggf. andere
Fische noch
hinzufügen.

```
let x: number = Math.random() * canvas.width  
let y: number = Math.random() * canvas.height  
let dx: number = Math.random() * 10 - 5  
let dy: number = Math.random() * 10 - 5  
let fish1: Fish1;  
fish1 = new Fish1();  
fish1.x = x  
fish1.y = y  
fish1.dx = dx  
fish1.dy = dy
```

i++

fishArray.push(fish1)

fish1.draw()

Neues fish Objekt
in fish Array
pushen

draw funktion von fish
aufrufen

update

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

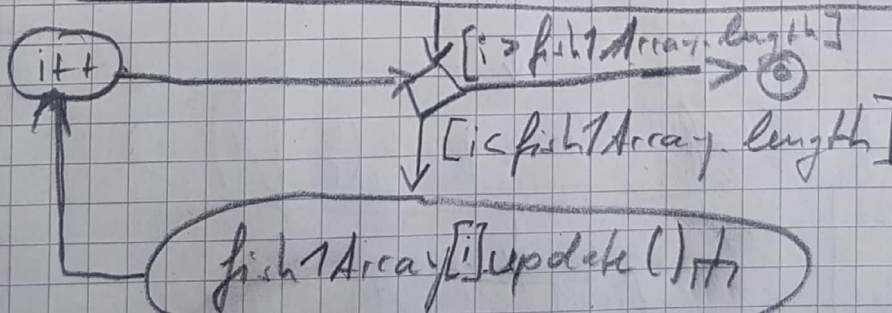
`crc.clearRect(0, 0, canvas.width, canvas.height)`

Den Canvas leeren, damit die vorherigen Objekte nicht stehen bleiben (Wollen kein Windows XP-Effekt).

`crc.putImageData(imageData, 0, 0)`

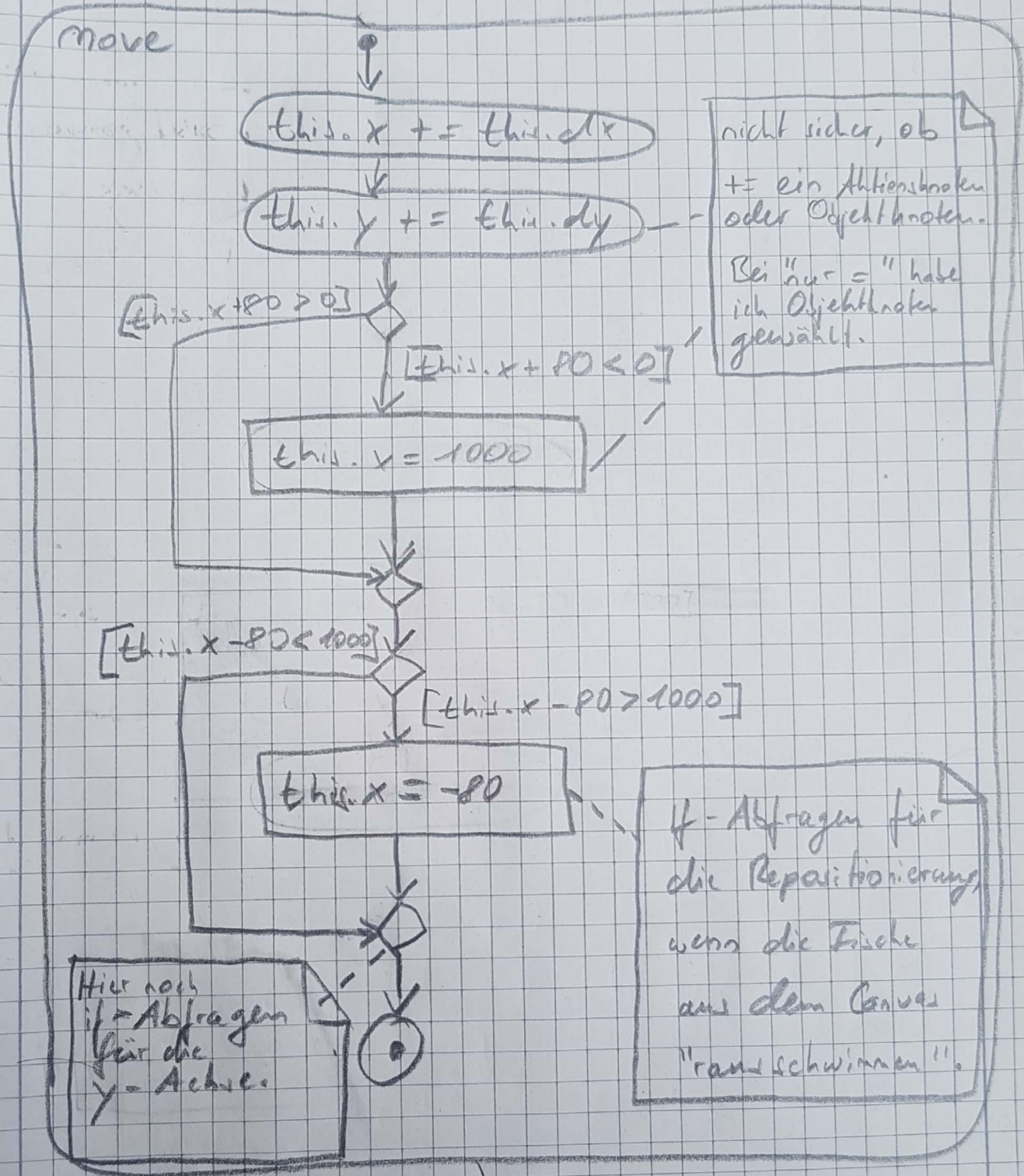
Hintergrund in imageData gespeichert, wird hier wieder eingezeichnet, um nervigen draw-Codi zu sparen

`let i: number = 0; i < fishArray.length; i++`



Diese for-Schleife kommt einmal für fish1, für fish2 und für bubble. Hier beispielhaft für fish1

Angepasste move Funktion für die Richtungsänderung,
falls Objekt am Rand des Canvas:



Beispielhaft für
die Fische gemacht.
Selbes, nur für y-Achse,
für die Bubbles.