

Funktionale  
Analyse

## Plattform PC

- bietet mehr Platz, so kann der Nutzer einfacher mit dem Spiel interagieren

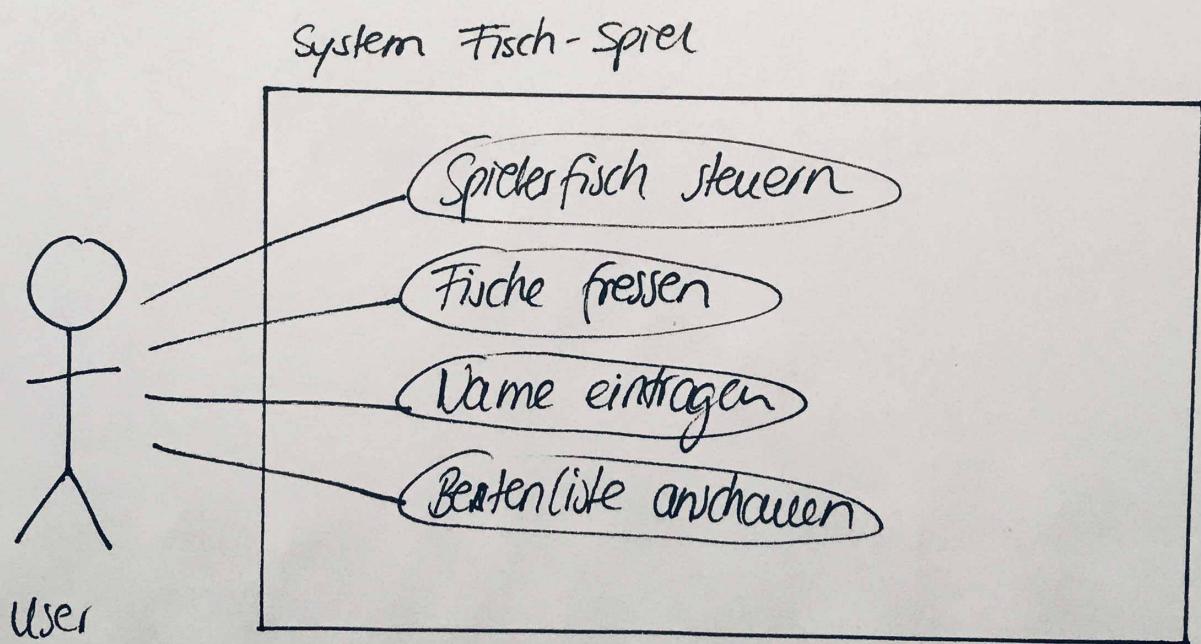
## Interaktion der Objekte untereinander

- Spielerfisch kann von Anfang an die kleinen blauen Fische fressen
- sobald der Spielerfisch genug kleine Fische gefressen hat / groß genug ist, kann er auch die großen, grünen Fische fressen
- wenn die großen, grünen Fische den Spielerfisch berühren, solange er noch nicht groß genug ist, fressen die grünen Fische den Spielerfisch und das Spiel endet

## Ablauf aus Sicht des Nutzers

- Nutzer kann die Spielanleitungen lesen
- steuert den gelben Spielerfisch mit den Pfeiltasten
- kann mit dem Spielerfisch anfangs die kleinen, blauen Fische fressen und muss versuchen den großen, grünen Fischen auszuweichen
- schafft er es nicht an den grünen Fischen vorbei zu schwimmen, solange er eine bestimmte Größe erreicht hat, endet das Spiel
- er kann hier trotzdem schon seinen erreichten Punktestand sehen und sich in die Bestenliste eintragen, welche er rechts anschauen kann
- schafft er es genug kleine Fische zu fressen, kann er nun auch die großen, grünen Fische fressen
- wenn der Nutzer alle Fische gefressen hat, gewinnt er das Spiel
- auch hier kann er sich nun in die Bestenliste eintragen

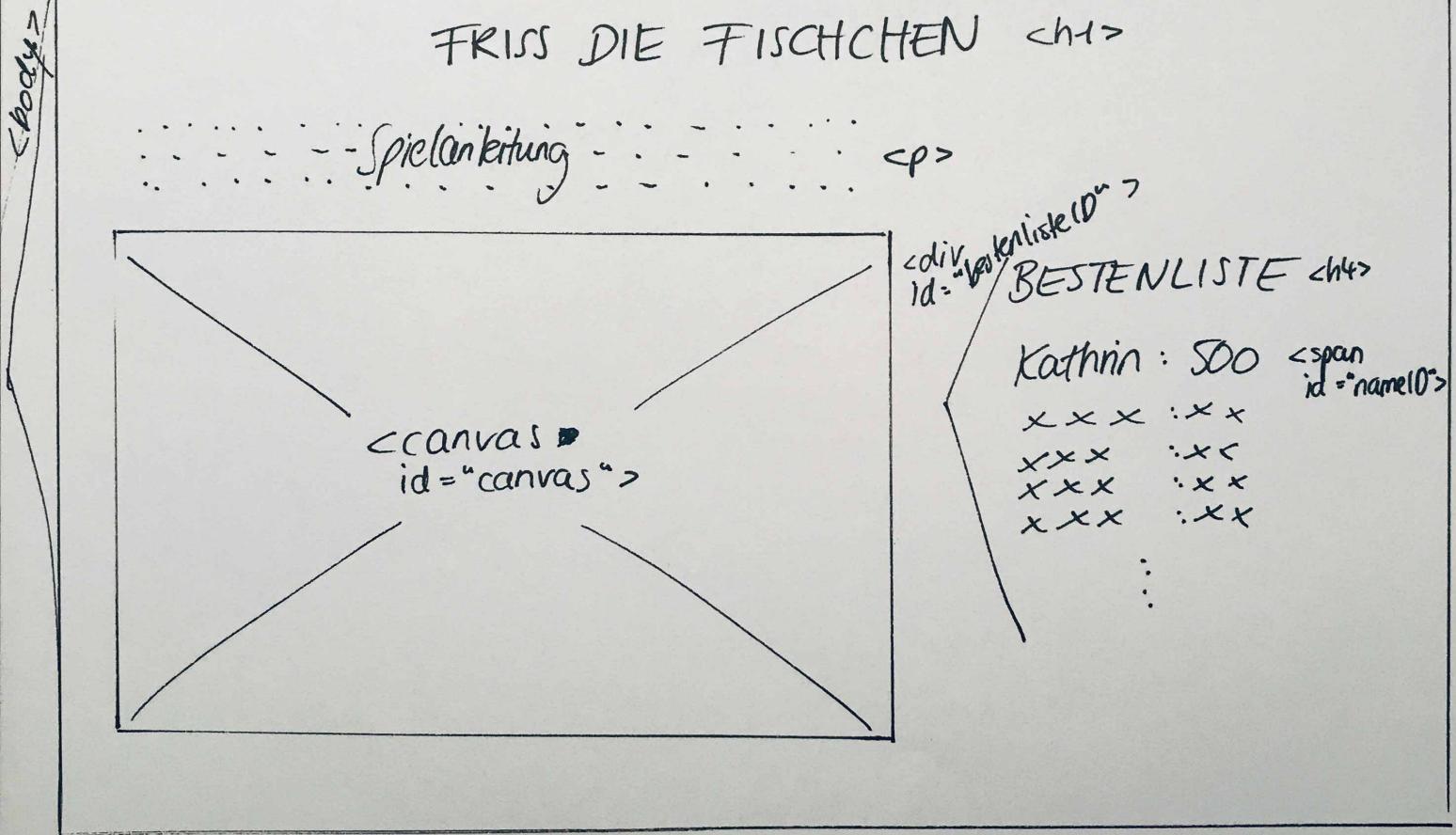
# Anwendungsfalldiagramm



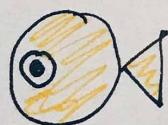
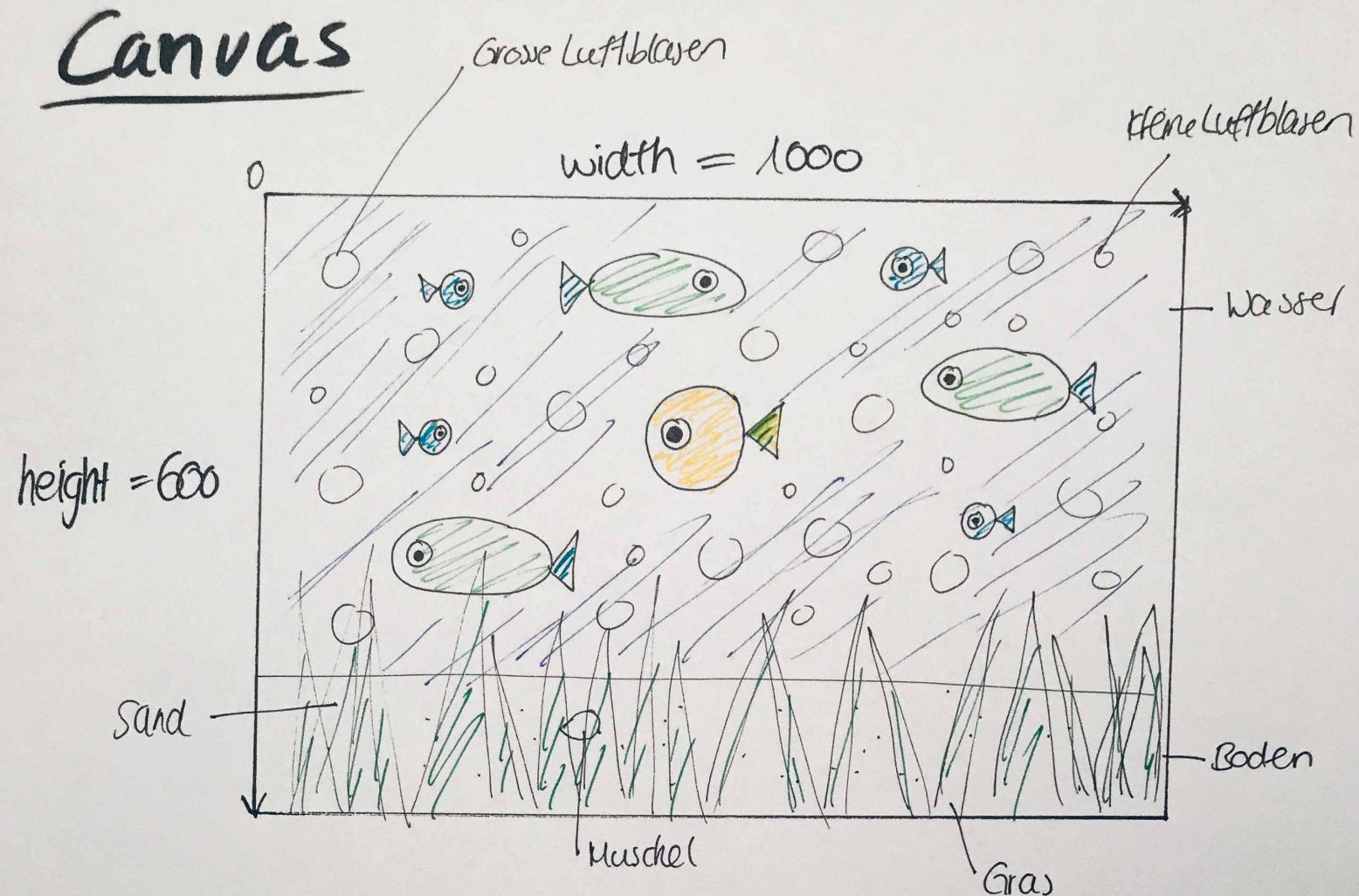
# Technische Analyse

# Konzept - Fress die Fischchen

## Skizze



# Canvas



Spielerfisch

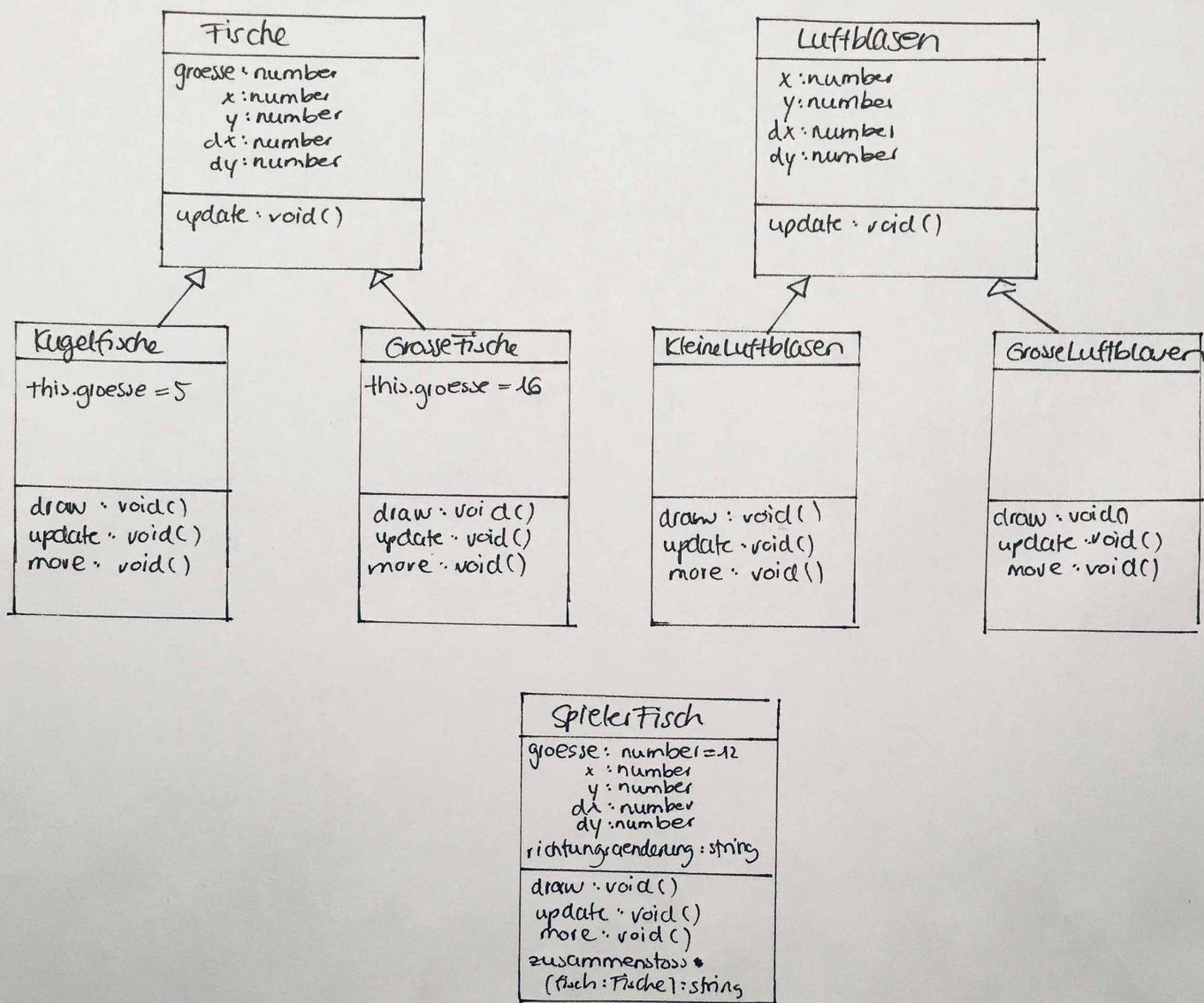


Kugelfische



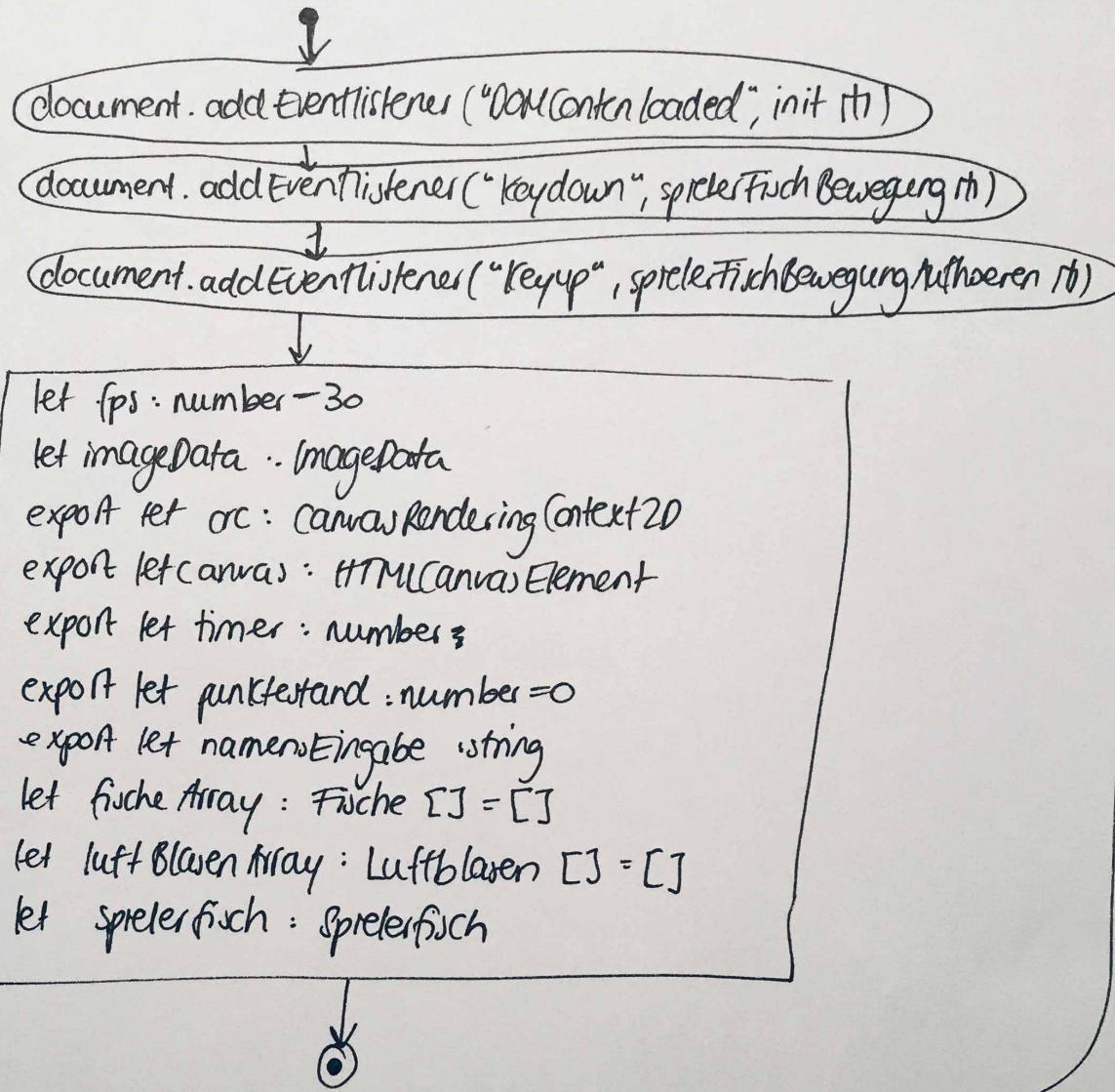
Große Fische

# Klassendiagramme



# Aktivitätsdiagramme

Hauptprogramm



init

```
canvas = document.getElementsByTagName("canvas")[0]  
crc = canvas.getContext("2d")
```

zeichne Hintergrund

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

```
sprecherfisch = new Sprecherfisch
```

sprecherfisch.draw

erstelle Kleine Luftblasen

erstelle Große Luftblasen

erstelle Kugelfische

erstelle Große Fische

refresh

update

update

```
timer = window.setTimeout(update, 1000/fps)  
orc.clearRect(0,0,canvas.width, canvas.height)  
orc.putImageData(imageData, 0, 0)
```

spielerfisch.update

```
let i : number = 0
```

i++

[fishArray[i < fishArray.length]]

fischeArray[i].update

fischeArray.splice(i, 1)

window.clearTimeout(timer)

alert

namenEingabe = prompt("Du hast verloren...")

internet

refresh

```
let i : number = 0
```

i++

[luftblasenArray.length]

luftblasenArray[i].update

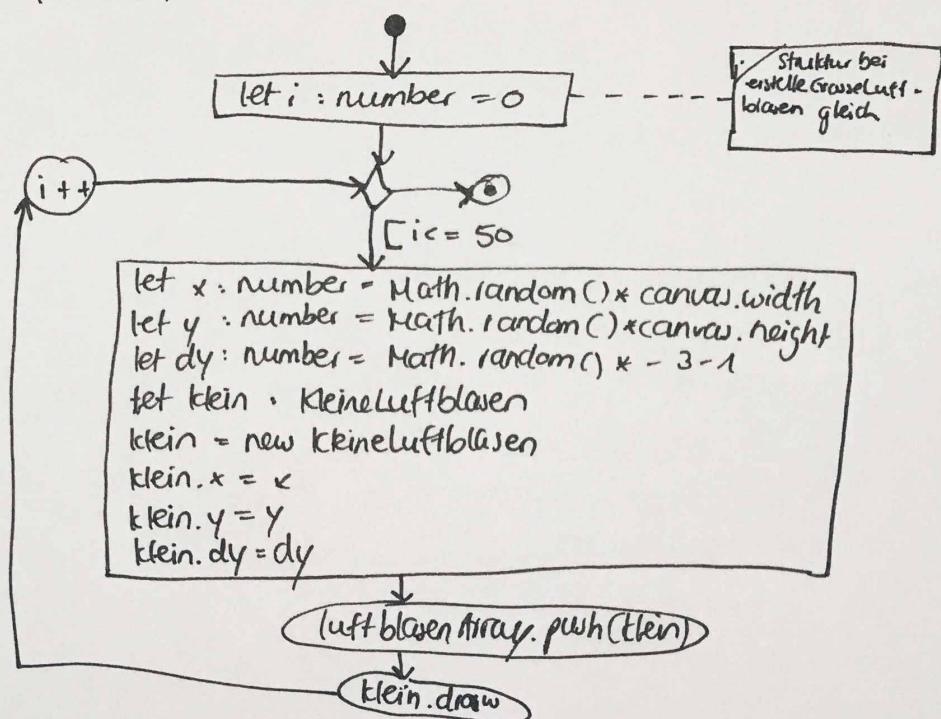
Punktestand darstellen

[fishArray.length == 0]

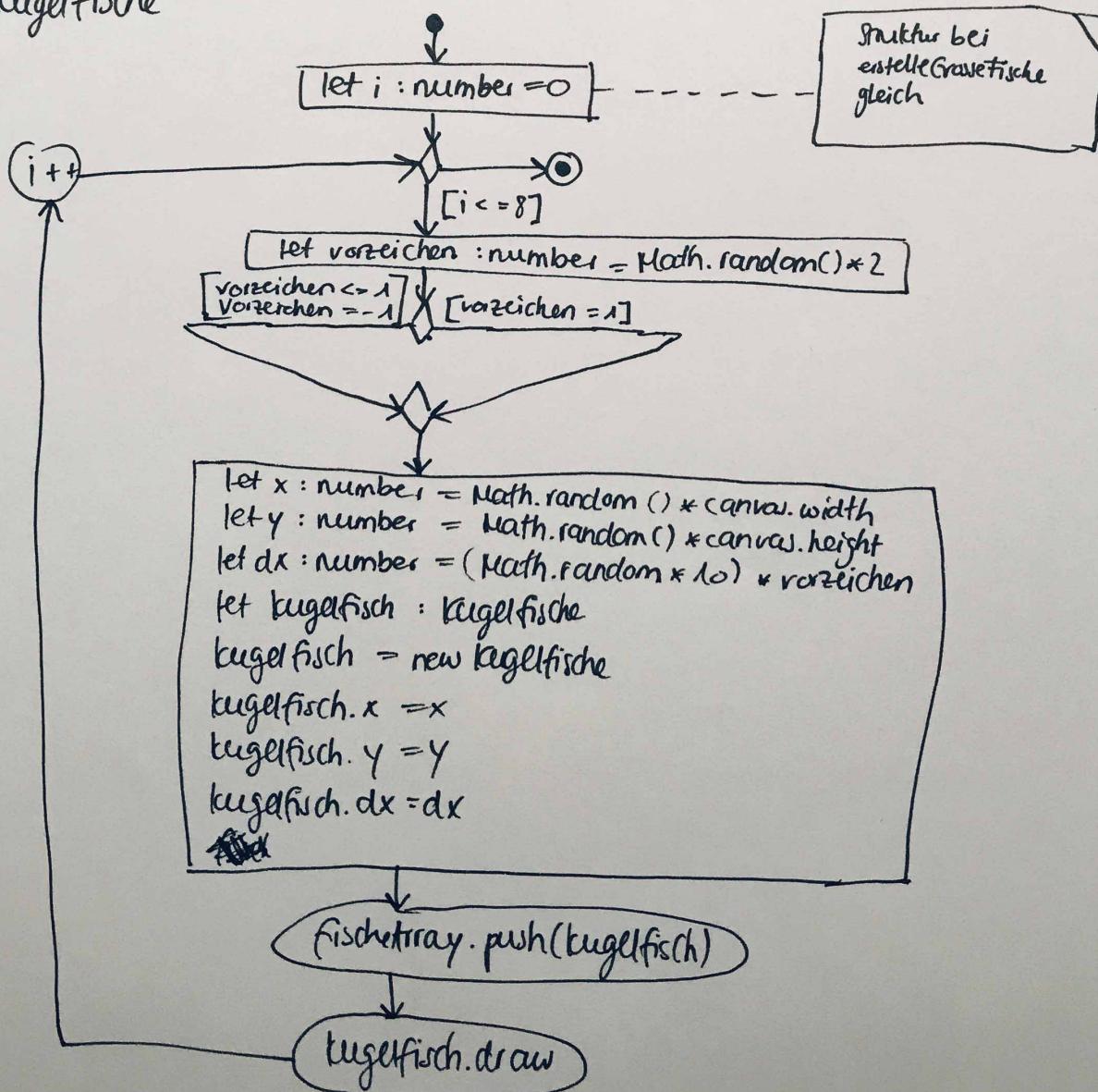
spielende

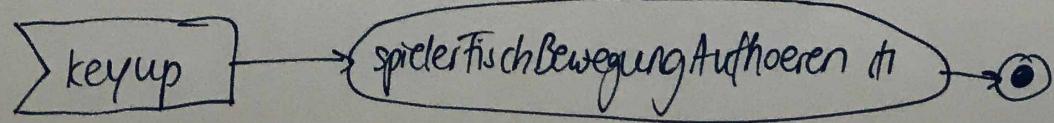
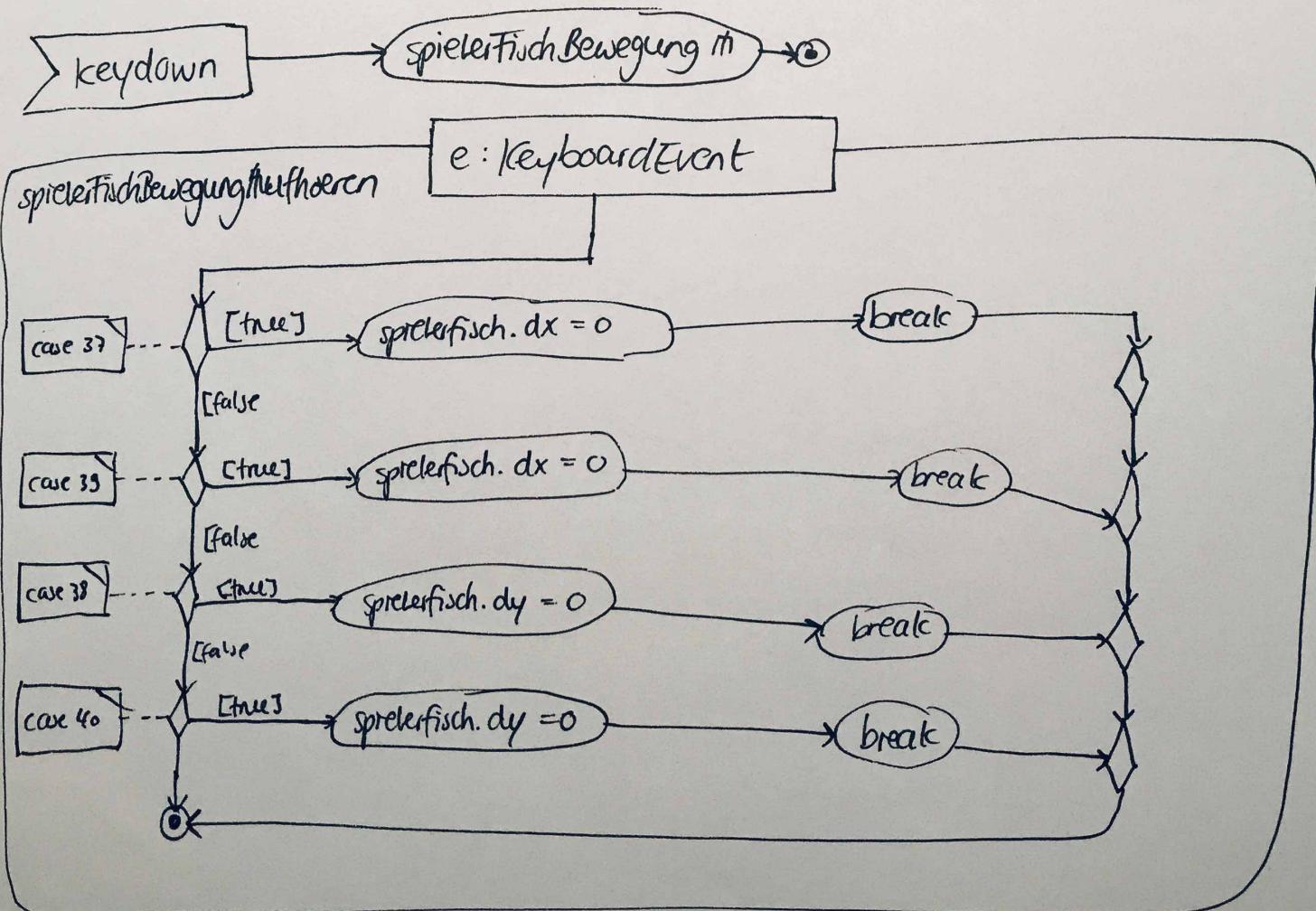
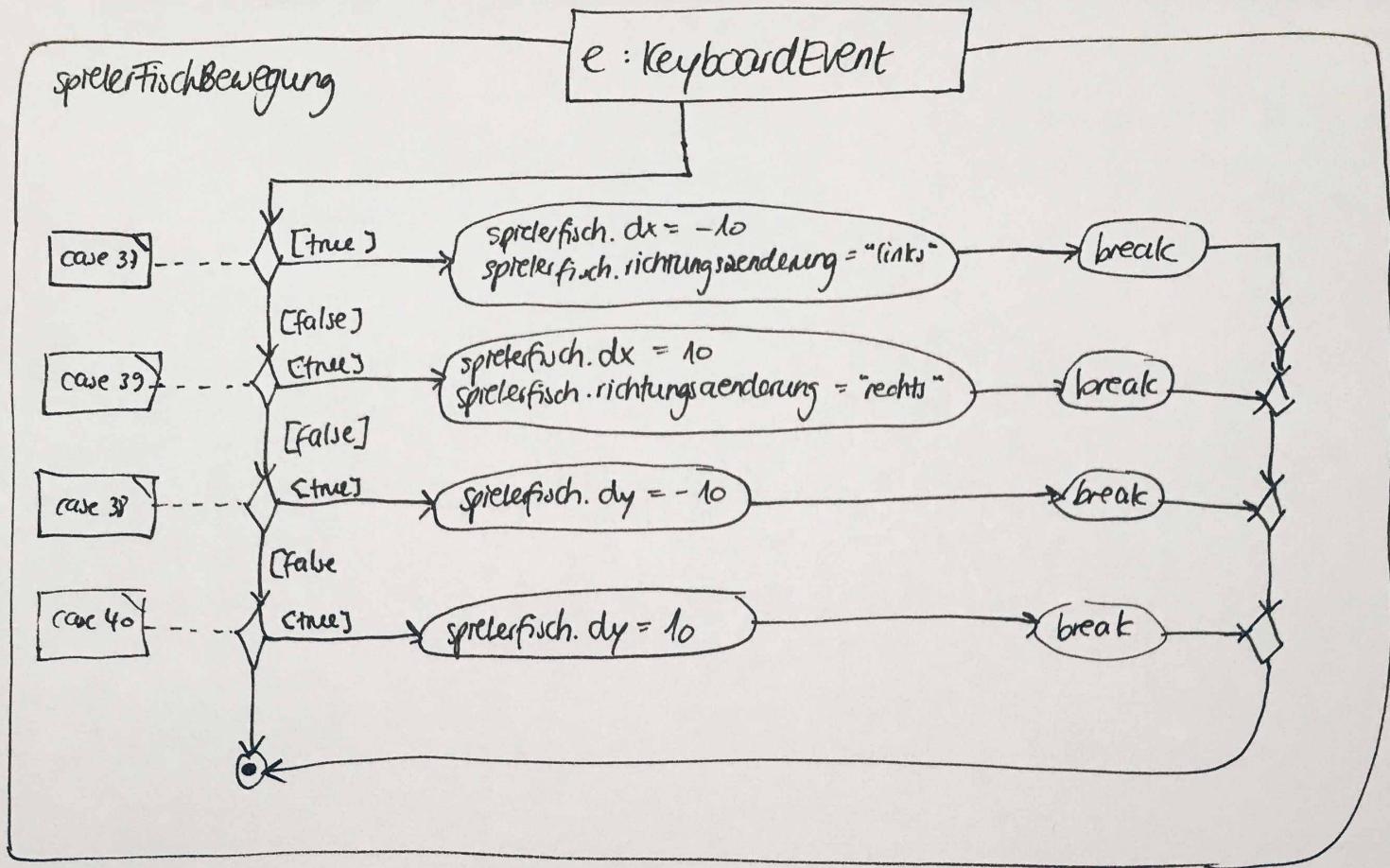


## erstelle Kleine Luftblähen

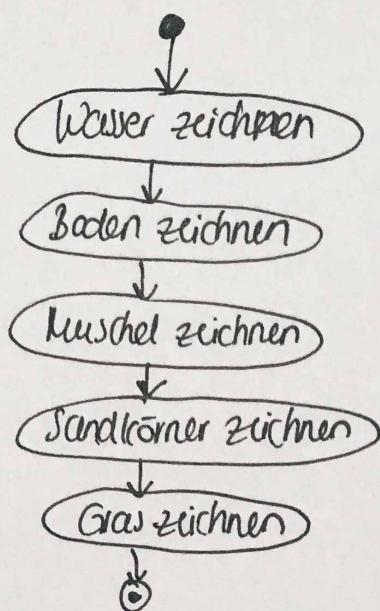


## erstelle Kugelfische

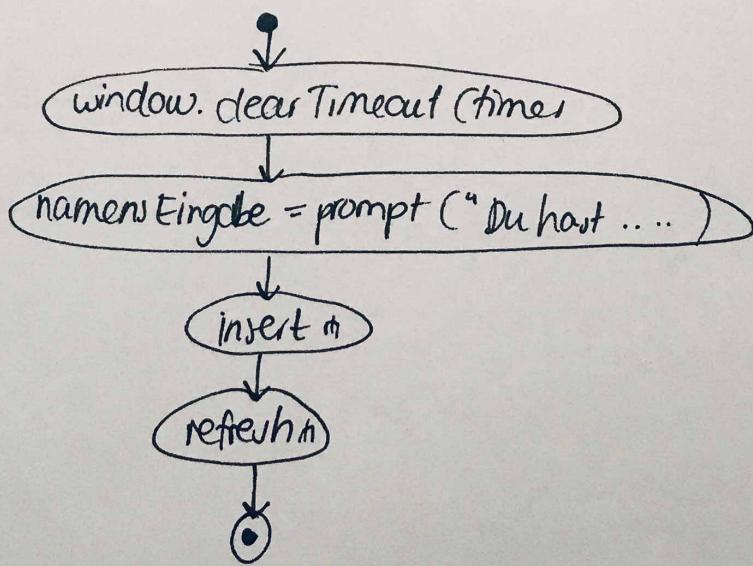




## zeichnetHintergrund

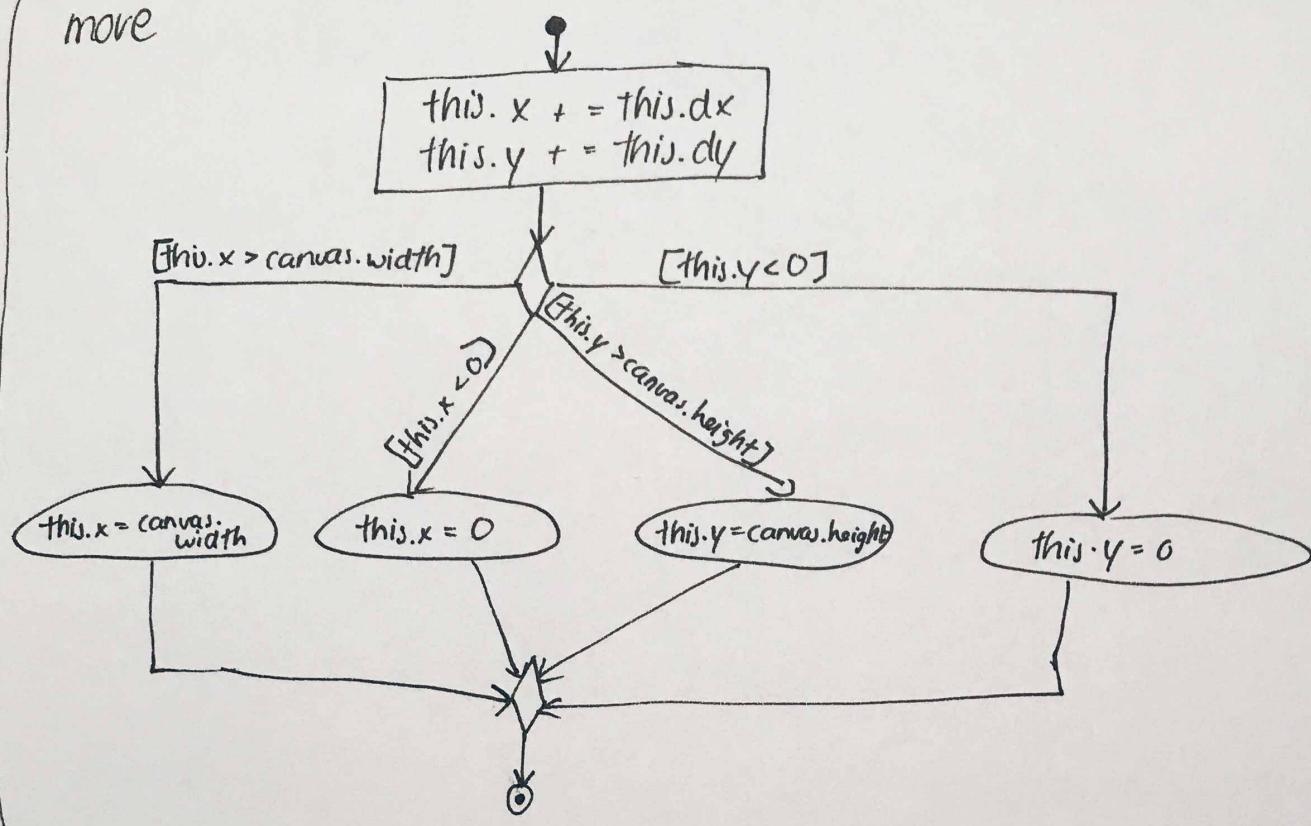


## spielzuEnde

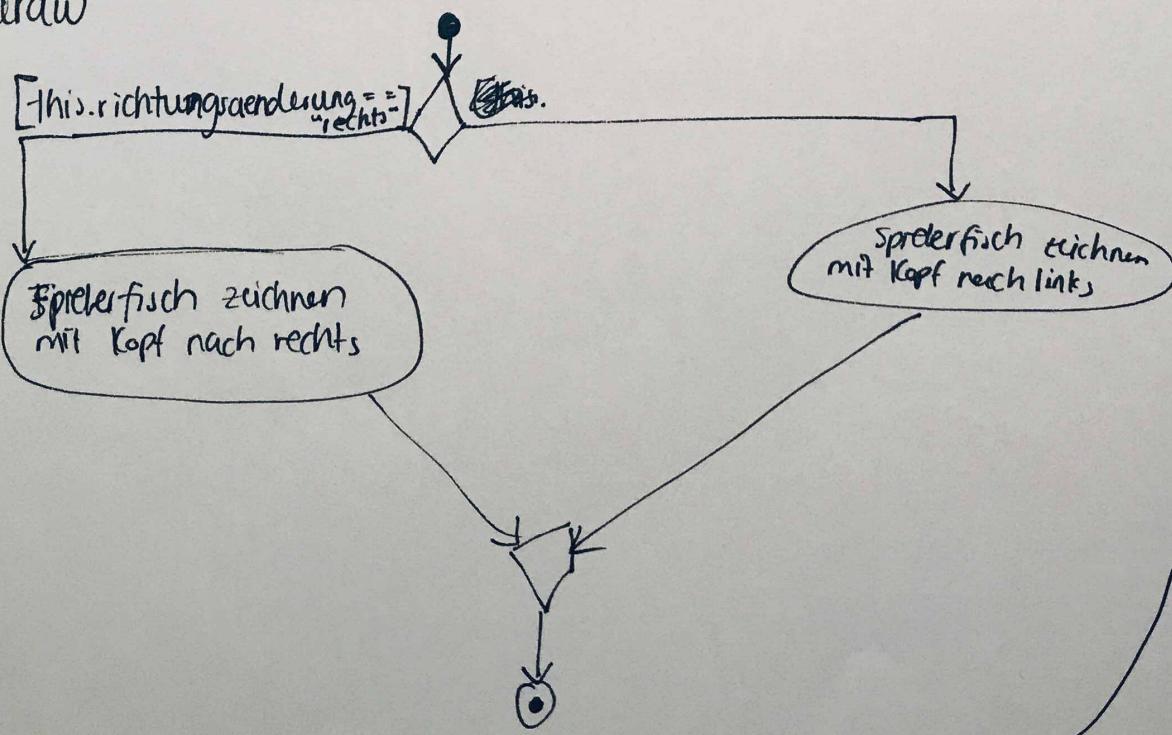


# Spielderfisch

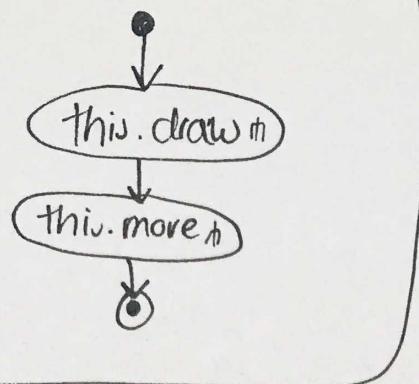
move



draw



update



zusammenstoß

fisch : Fische

```
let xEntfernung : number = Math.abs(this.x - fisch.x)  
let yEntfernung : number = Math.abs(this.y - fisch.y)
```

[ $\sqrt{x^2 + y^2} < 60$ ]

[ $this.groesse > fisch.groesse$ ]

`this.groesse++`

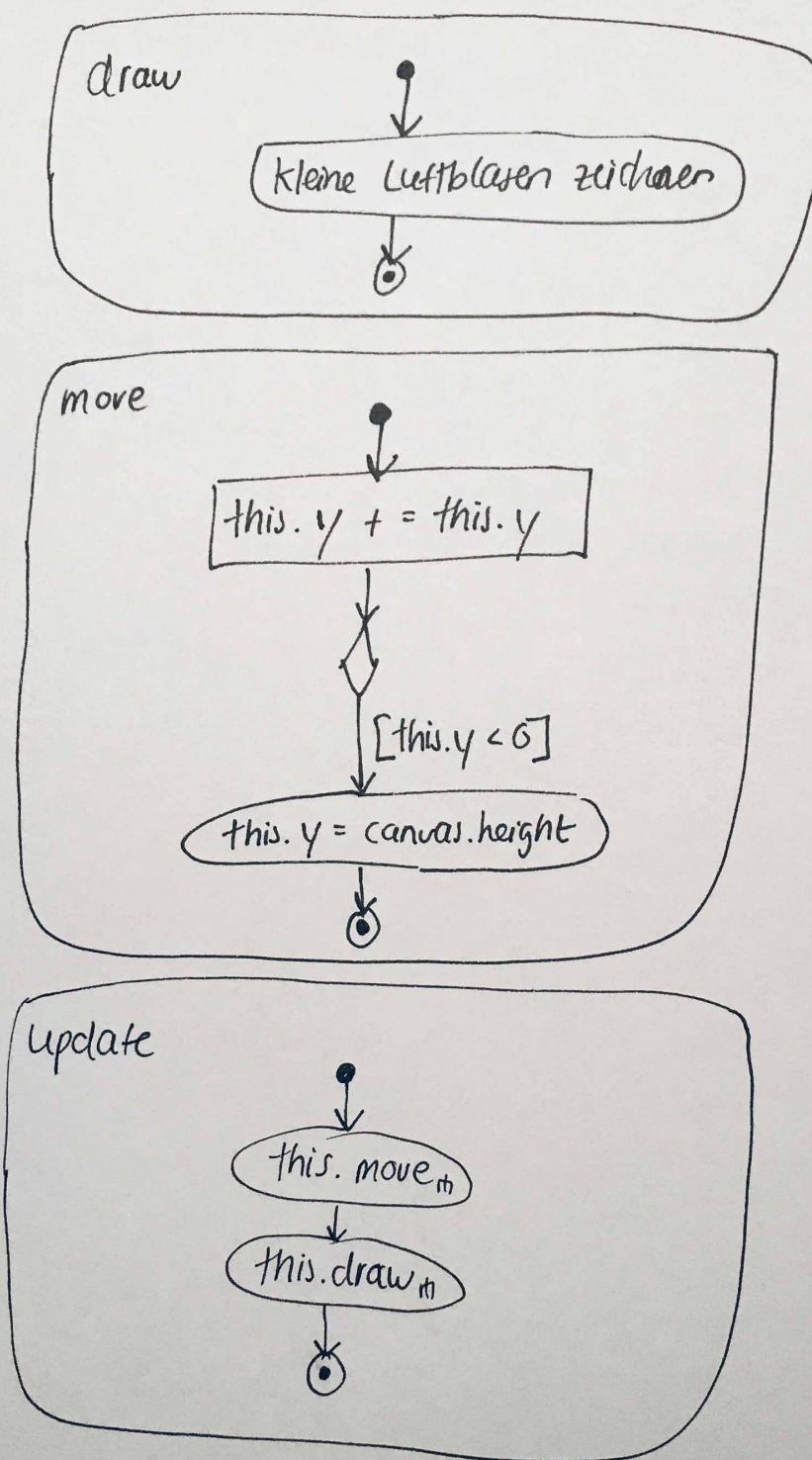
`punktestand += 25`

return 1

return 2

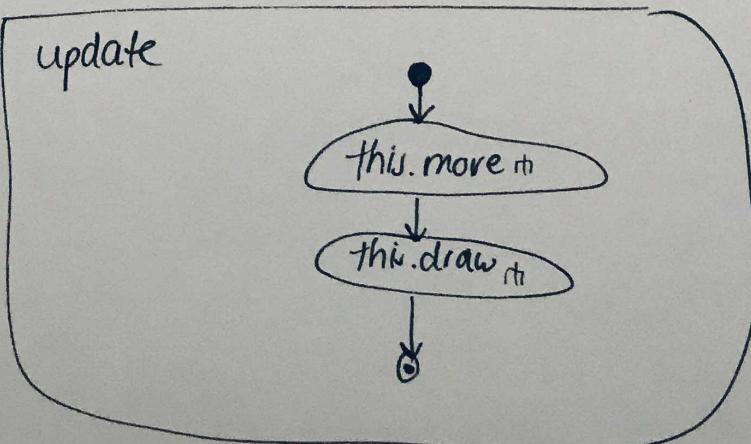
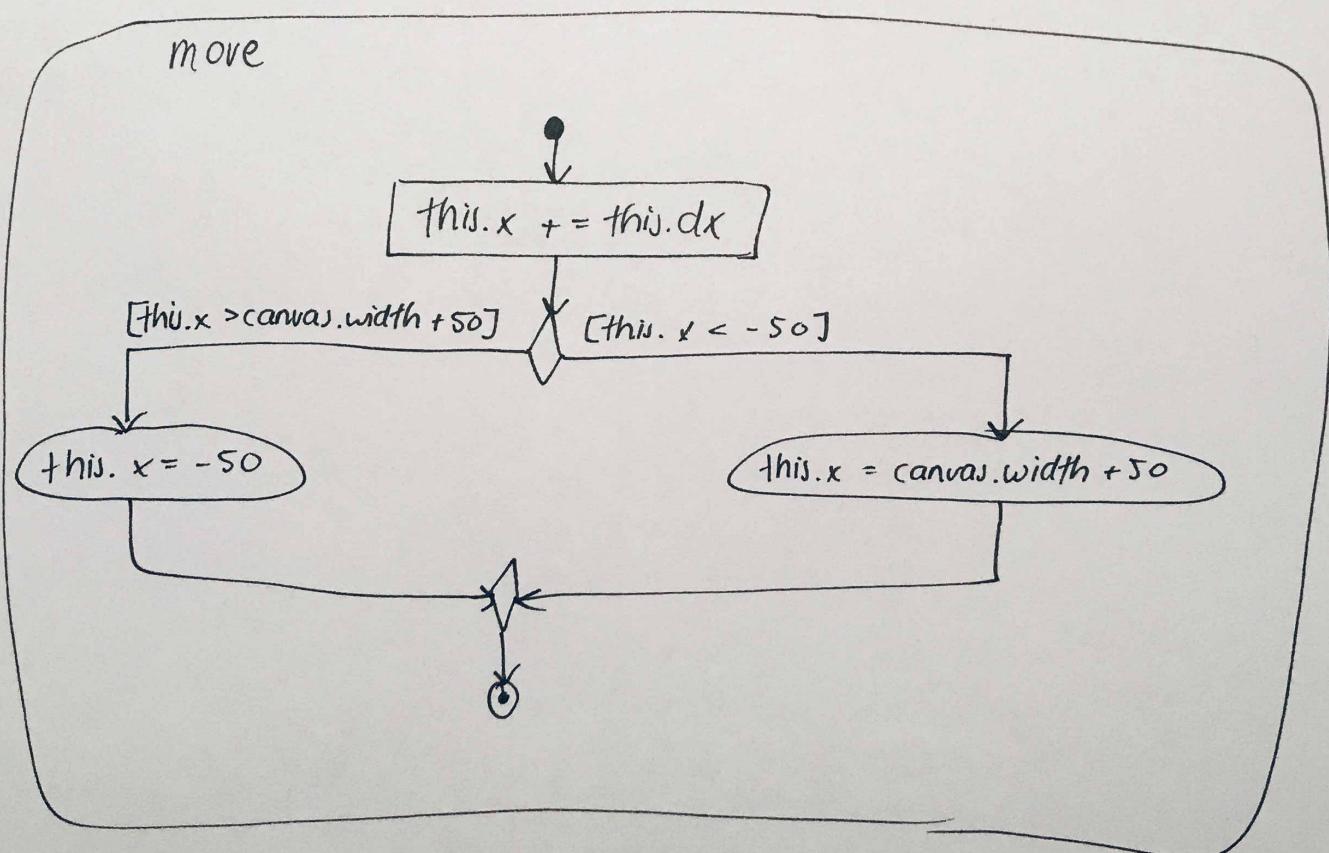
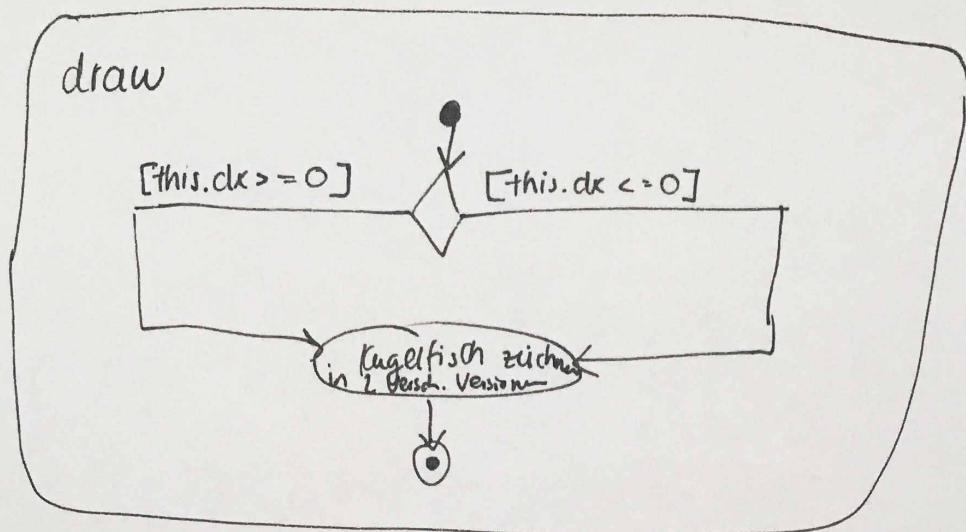
return 0

# Kleine Luftblasen



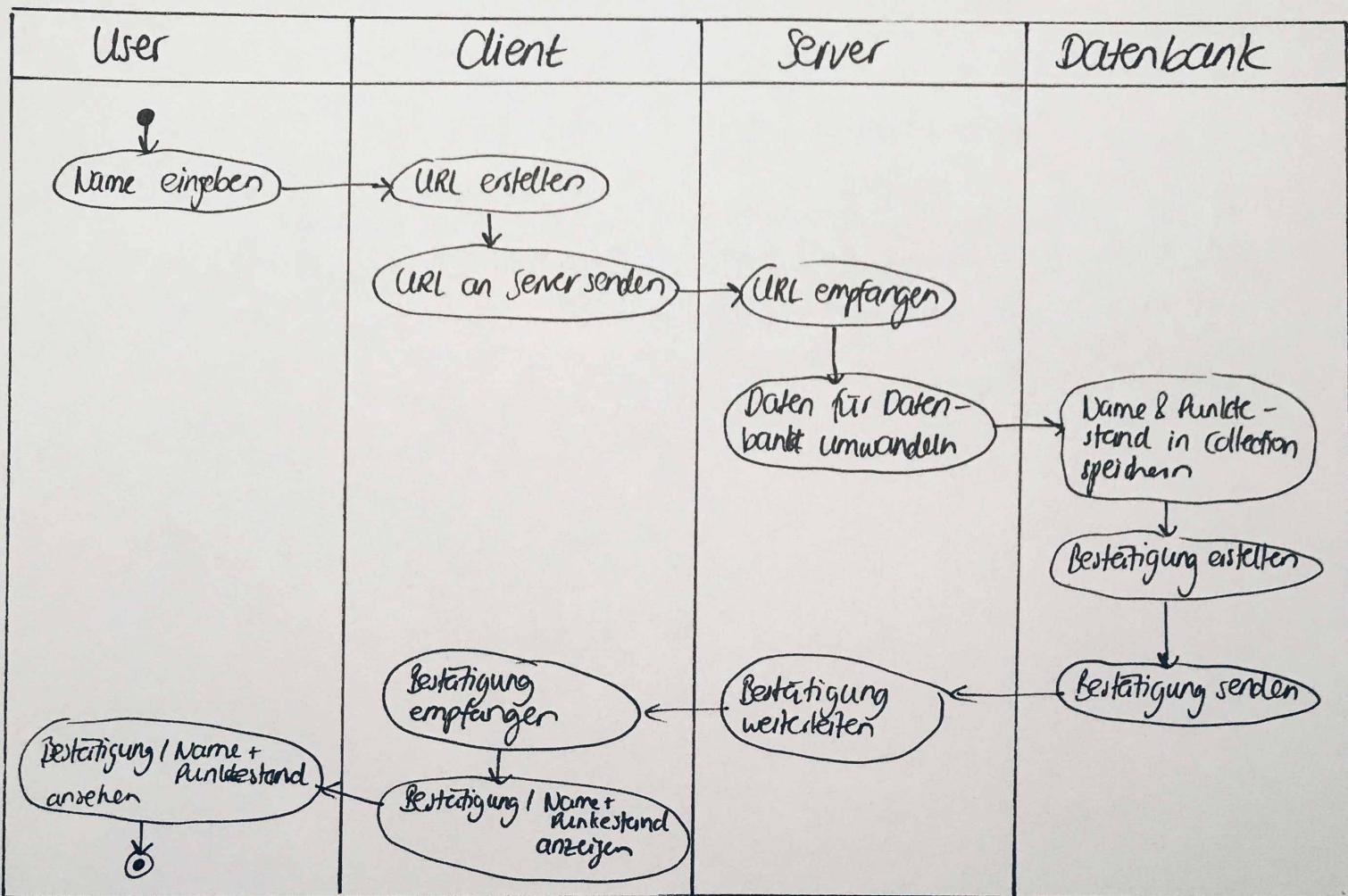
Bei GrossLuftblasen gleiche Struktur.

# Kugelfische



Bei GoseFische gleiche Struktur.

# Domänenübergreifenes AD



## Interface

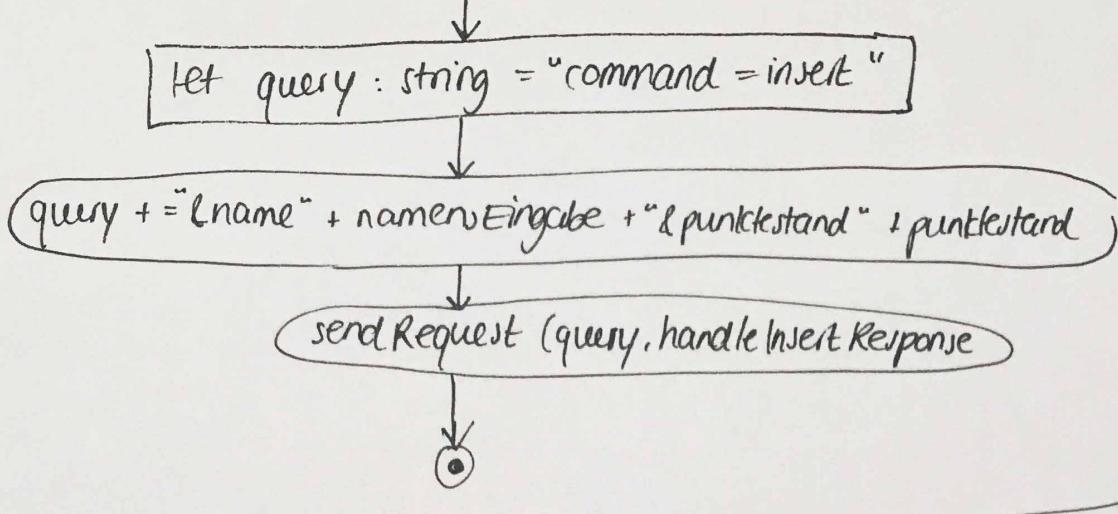
Bestenliste

```
[key: string]: string
```

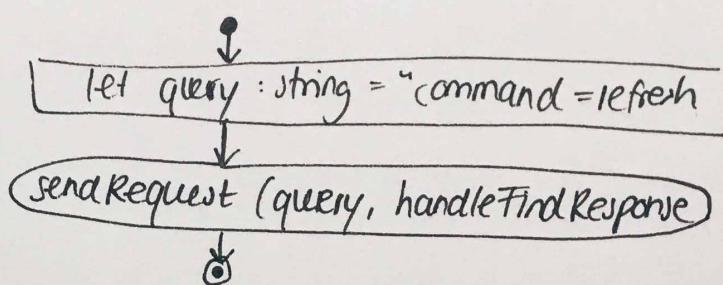
Spieler

```
name: string  
punktestand: number
```

insert



refresh



handleFindResponse

- event: ProgressEvent

```
let xhr : XMLHttpRequest = (< XMLHttpRequest > event.target)
```

[xhr.readyState == XMLHttpRequest.DONE]

```
let fischSpielerArray : Spieler[] = JSON.parse(xhr.response)
```

```
document.getElementById("namelD").innerHTML = ""
```

```
let i : number = fischSpielerArray.length - 10
```

i++

X

[i < fischSpielerArray.length]

```
document.getElementById("namelD").innerHTML += `<div>${fischSpielerArray[i].name}: ${fischSpielerArray[i].punktestand}</div>`
```