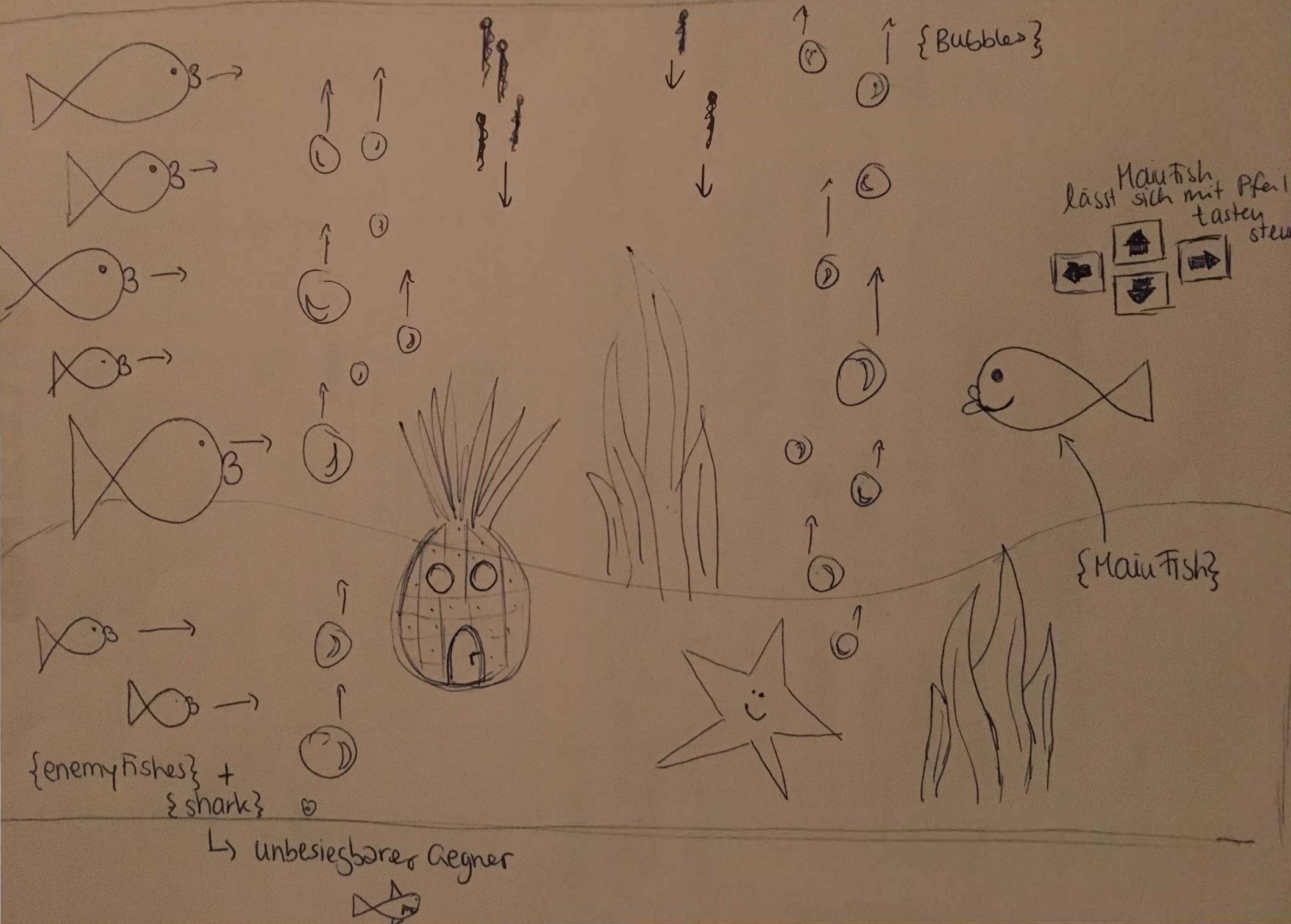


{Food} bei Leertasten click



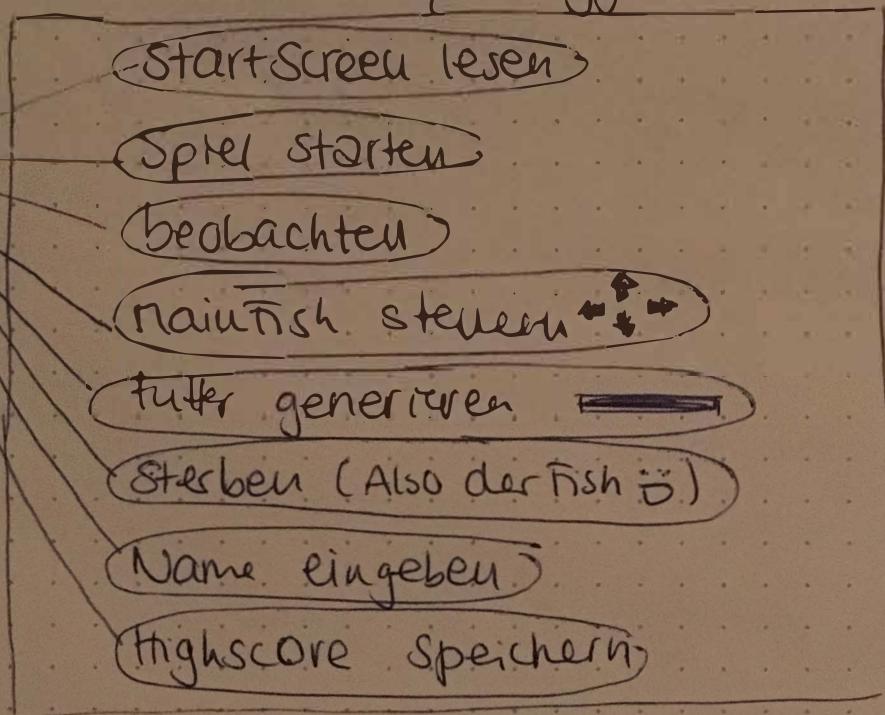
- Spielelemente:
- Hai Fish = kann mit Pfeiltasten nach oben, unten, rechts, links bewegt werden
 - = kann food & kleinere Fische als ei selbst essen
→ er wächst
 - = wird von Hai & größeren Fischen gegessen
→ stirbt ... GAME OVER
- Enemy Fish = bewegen sich von allein von links nach rechts
- = kann Maiufish fressen wenn größer
 - = wird von Maiufish gegessen wenn kleiner
- Shark = wie Enemy Fish, nur unbestieghbar
- Food = kann durch Leertaste random auf der x-pos. erzeugt werden
- Bubbles & andere statische Elemente = können nicht zur Interaktion genutzt werden

Technische Gegebenheiten : Desktop, PC

There is always a bigger fish



USER



Spieldlauf:

- > bei Aufruf erscheint Startbildschirm mit kurzer Erklärung & Start Button
- > eigentliches Spiel:
 - Spieler steuert mit Pfeiltasten den Mainfish (größere Fische & shark vermeiden, Futter & kleinere Fische fressen)
 - wächst bei Futter und kleineren Fischen
 - stirbt bei shark & größeren Fischen
 - kann durch Leertaste Futter generieren
- > nach Spiel (bei Tod) od. high score:
 - kann Name eingeben
 - kann Highscore speichern

Superklasse

Objects

typ: string
xPos: number
yPos: number
width: number
height: number
size: number

update(): void
draw(): void

|-src:string|

Moving Objects

xSpeed: number
ySpeed: number
xMax: number
xMin: number
yMax: number
yMin: number
img: HTMLImageElement

update(): void
move(): void
draw(): void

Main Fish

-src, typ, width, height, size

update(): void

draw(): void

handle KeyEvent(): void

check Distance To Others(): void

Enemy Fishes

-src, typ, width, height, size

update(): void

move(): void

draw(): void

Bubble

-src, -x, -y

radius: number
color1: string
color2: string

update(): void
move(): void
draw(): void

-src, -typ, -width, -height, size

update(): void
move(): void
draw(): void

Shark

-src, -typ, -width, -height, size

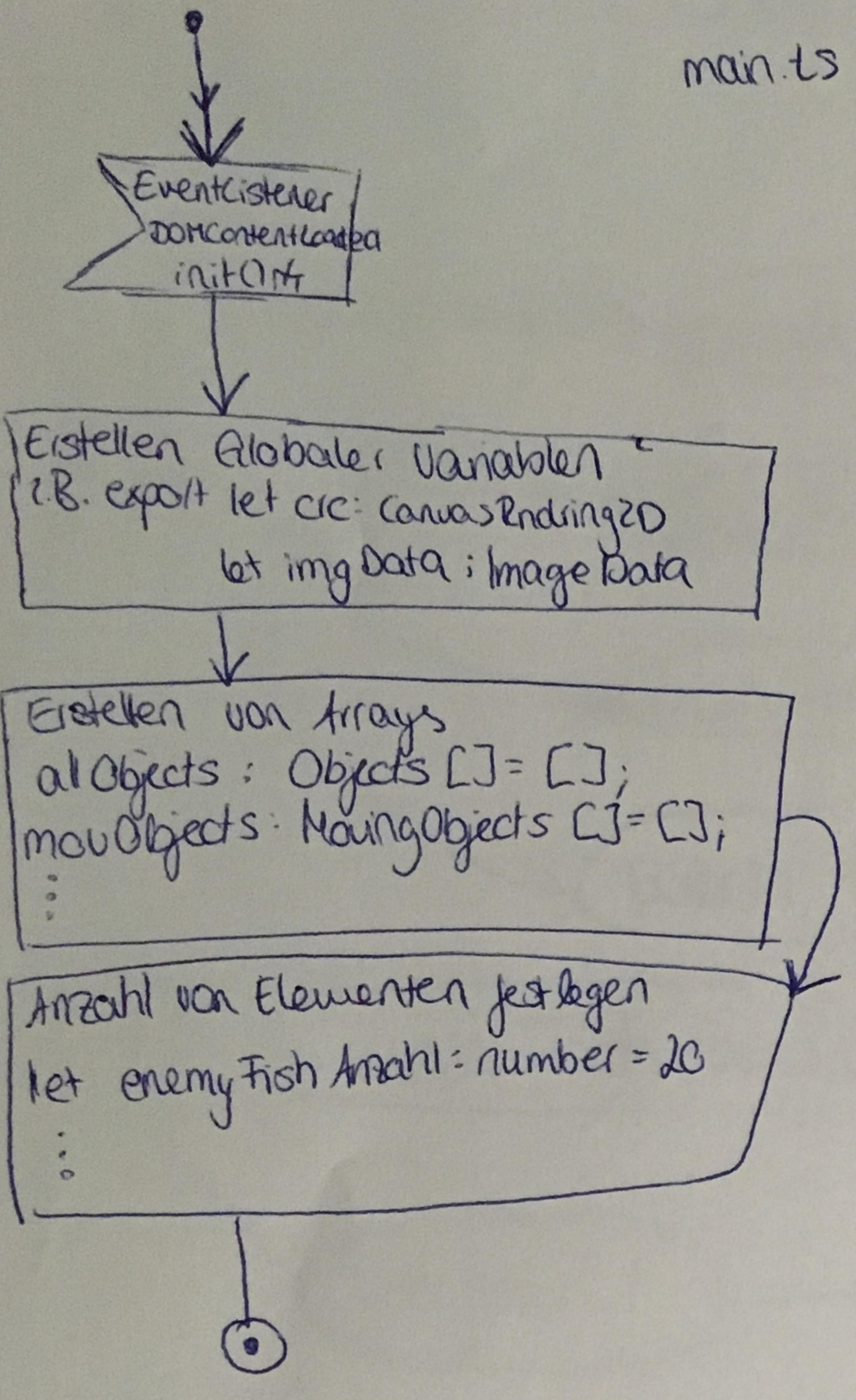
update(): void
move(): void
draw(): void

Food

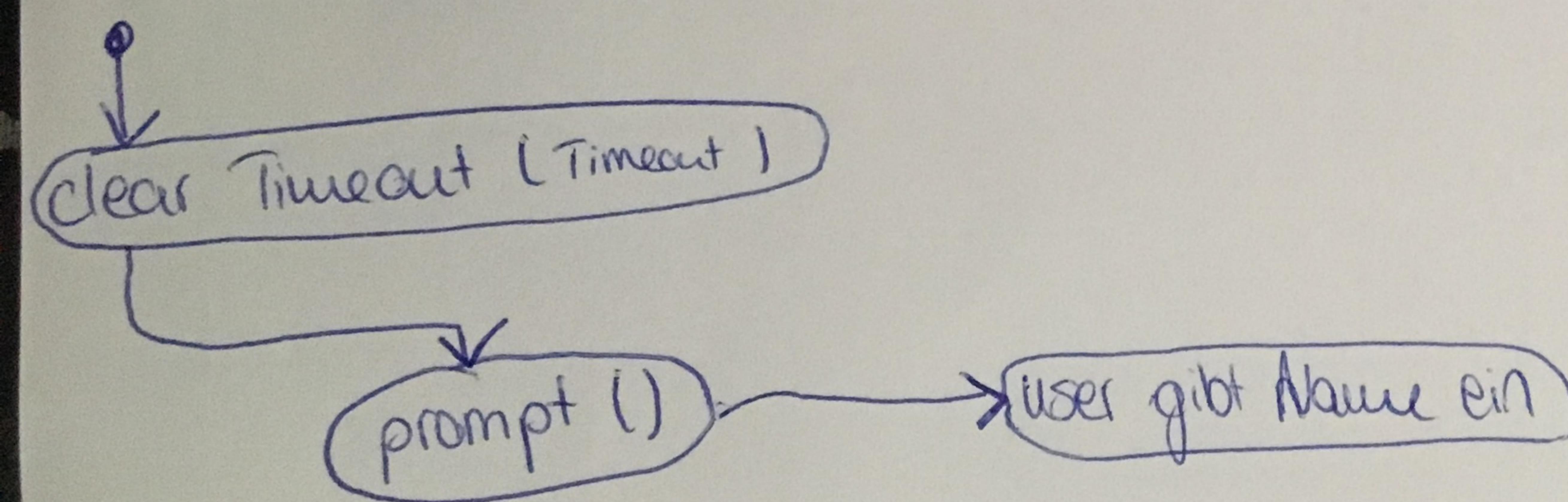
-src, -typ, -width, -height, size

update(): void
move(): void
draw(): void

main.ts



handle Game Over



init()

gamePage = display "none"
introPage = display "block"

button.addEventListener

> "click" --> startGame()

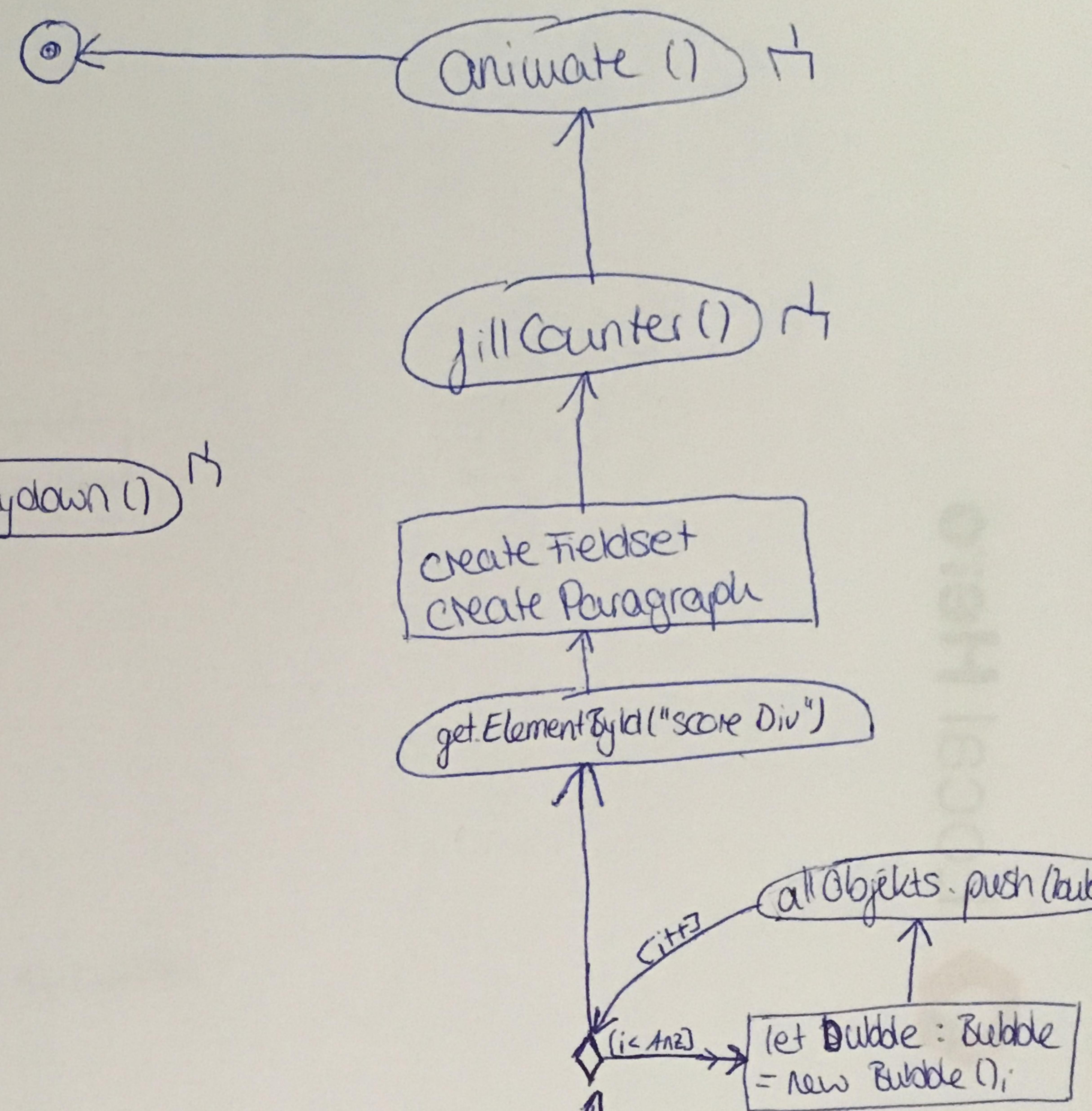
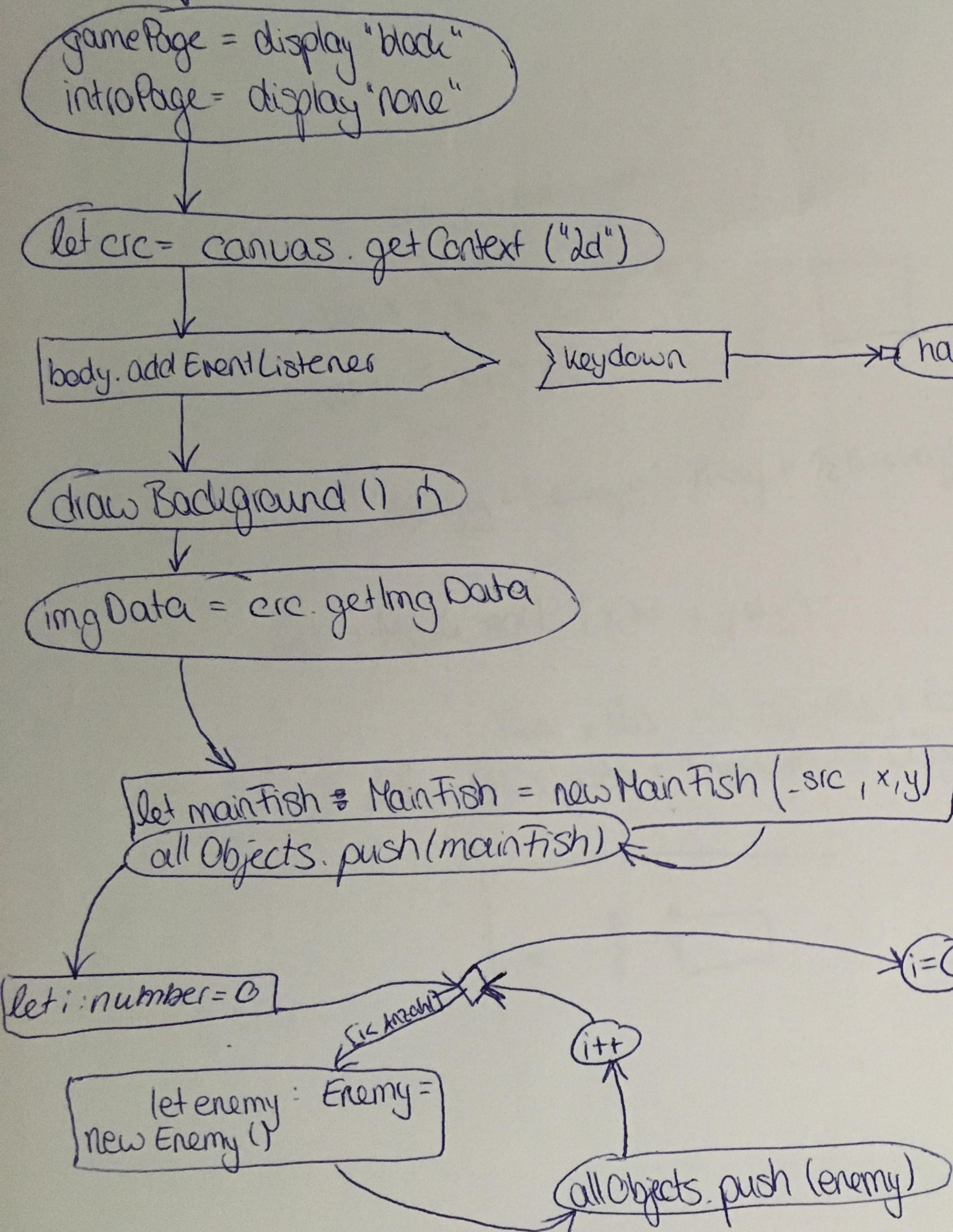
createFood()


let food: Food = newFood (-src, -typ, -x, -y)

allObjects.push (food)


food.draw()

Build Game





Handle Keyboard Event



Main.Fish. Check Distance

let i: number = 0

[allObjects.length]

[allObj[i] == enemy || shark || food]

let distance: number
between this
and allObj[i]

$|x_1 - y_1|$

$d = \text{distance}$

$|x_1 - x_2|$

$x_{\text{pos}_1} - x_{\text{pos}_2}$

$y_{\text{pos}_1} - y_{\text{pos}_2}$

$\text{var } d = \text{Math.sqrt}(a * a + b * b)$

[distance < 10]

[allObj[i]

[size > this.size || allObj[i] == "shark"]

death = true

[allObj[i].size <= this.size]
[allObj[i].type == "food"]

delete food
from array

this.size += 0.2

score += 10



