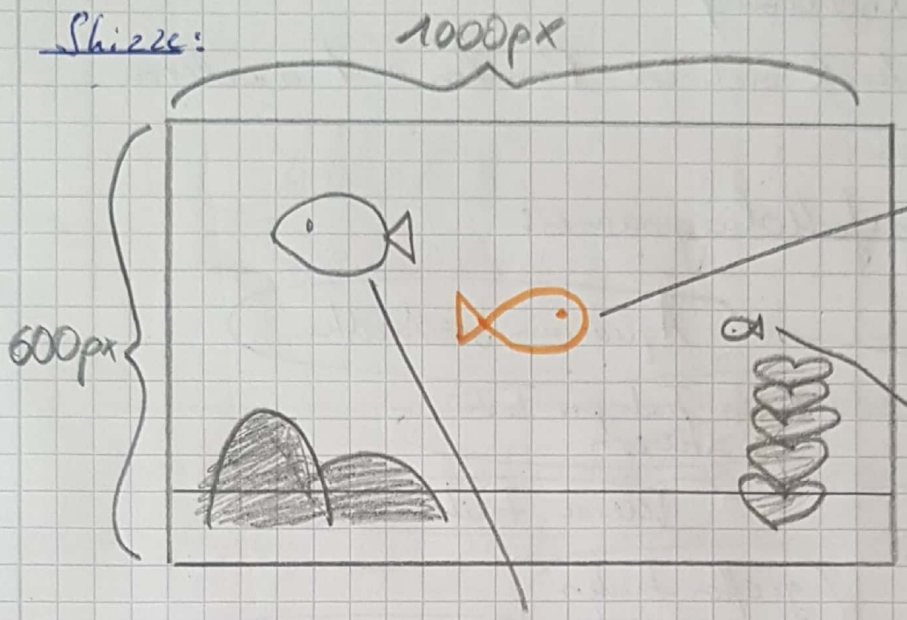
Endabgabe - EIA2

Manuel Matern

Skizze:



Wie mache ich die Kollision!?

Basis: Code von Aufgabe 12.

Nutzer:

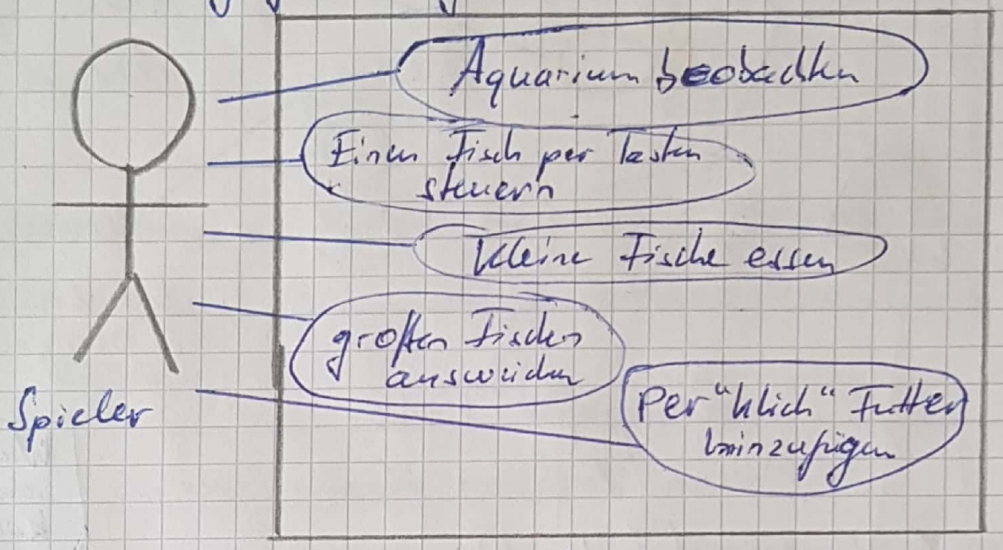
Startet das Programm und kann den Fisch mit WASD oder den Pfeiltasten steuern. Ziel ist einen höheren Score als die der in der Scoreliste stehenden Spieler zu haben. Durch Klick Futter hinzufügen. (Vielleicht später einbauen, dass essbar).

-> Canvas muss groß genug sein. Steuerung angenehm, damit man besser navigieren kann (Schnelligkeit/Ruckelfrei).

- Pfeiltaste Links ≙ Fisch bewegt sich nach Links
- " Rechts ≙ " rechts
- " Oben ≙ " oben
- " ~~Unten~~ unten ≙ " unten.
- Nutzer klickt ≙ Futter spawnt an Stelle des Cursors
- Nutzer / Spieler navigiert zwischen den zu großen Fischen hindurch zu kleineren Fischen, um sie zu essen.

- Nutzer versucht so weit zu kommen wie er geht. (überleben)
- ⇒ L> eventuell mehrere Runden mit anderen Fischen.

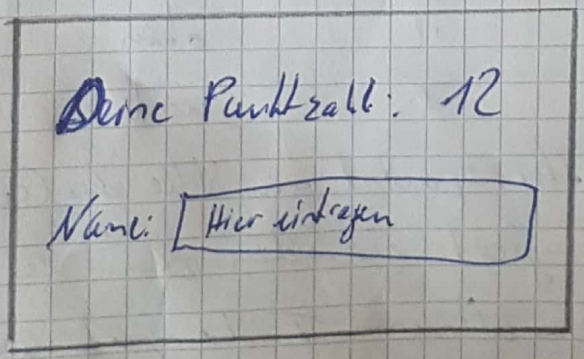
Anwendungsfalldiagramm:



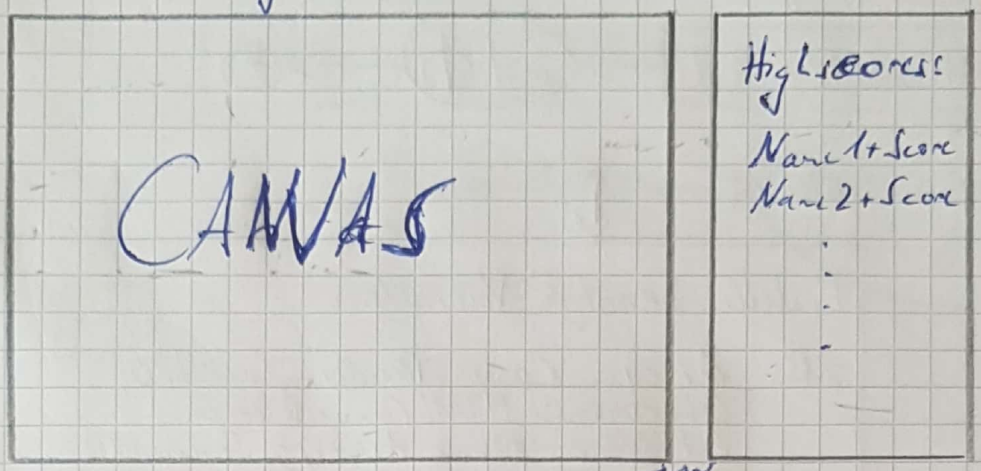
System / Aquarium Spiel
Anwendung zum Zeitvertreib / Spaß.

Weitere Nutzer Interaktionen und Reaktionen des Spiels:

- Schwimmt man aus dem Canvas heraus (egal ob oben, unten, rechts oder links), kommt man auf der gegenüberliegenden Seite wieder heraus.
- Bei Fall GameOver, erscheint ein Textfeld mit der Möglichkeit seinen Namen einzutragen und dem erreichten Punkten.
  → Neustart ≙ Reload der Seite

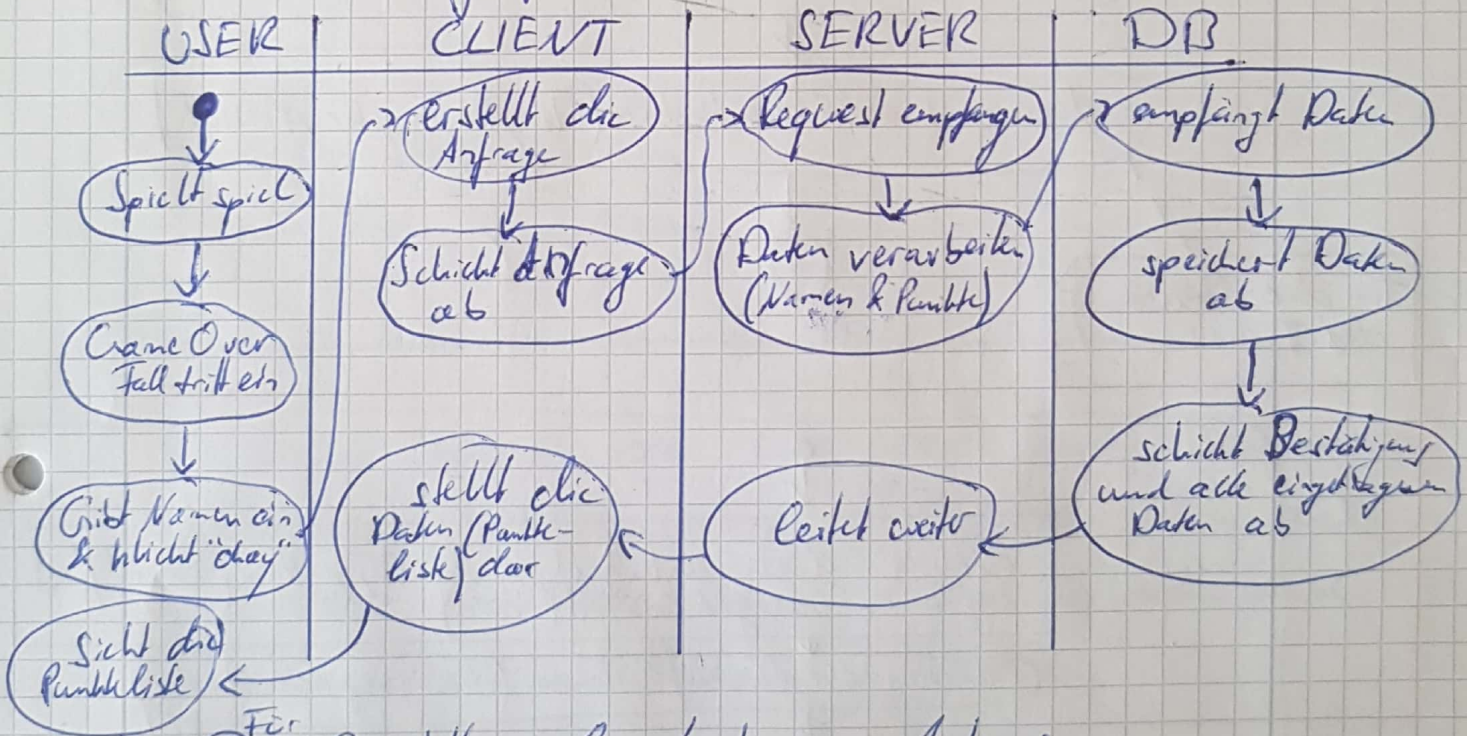


Skizze ~~Ko~~ Highscoreliste:



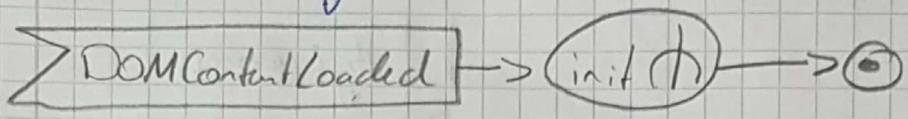
Anordnung mit CSS margin & Flexbox
→ display: flex und flex-direction: row im <Body>

Domänen-übergreifendes AD:

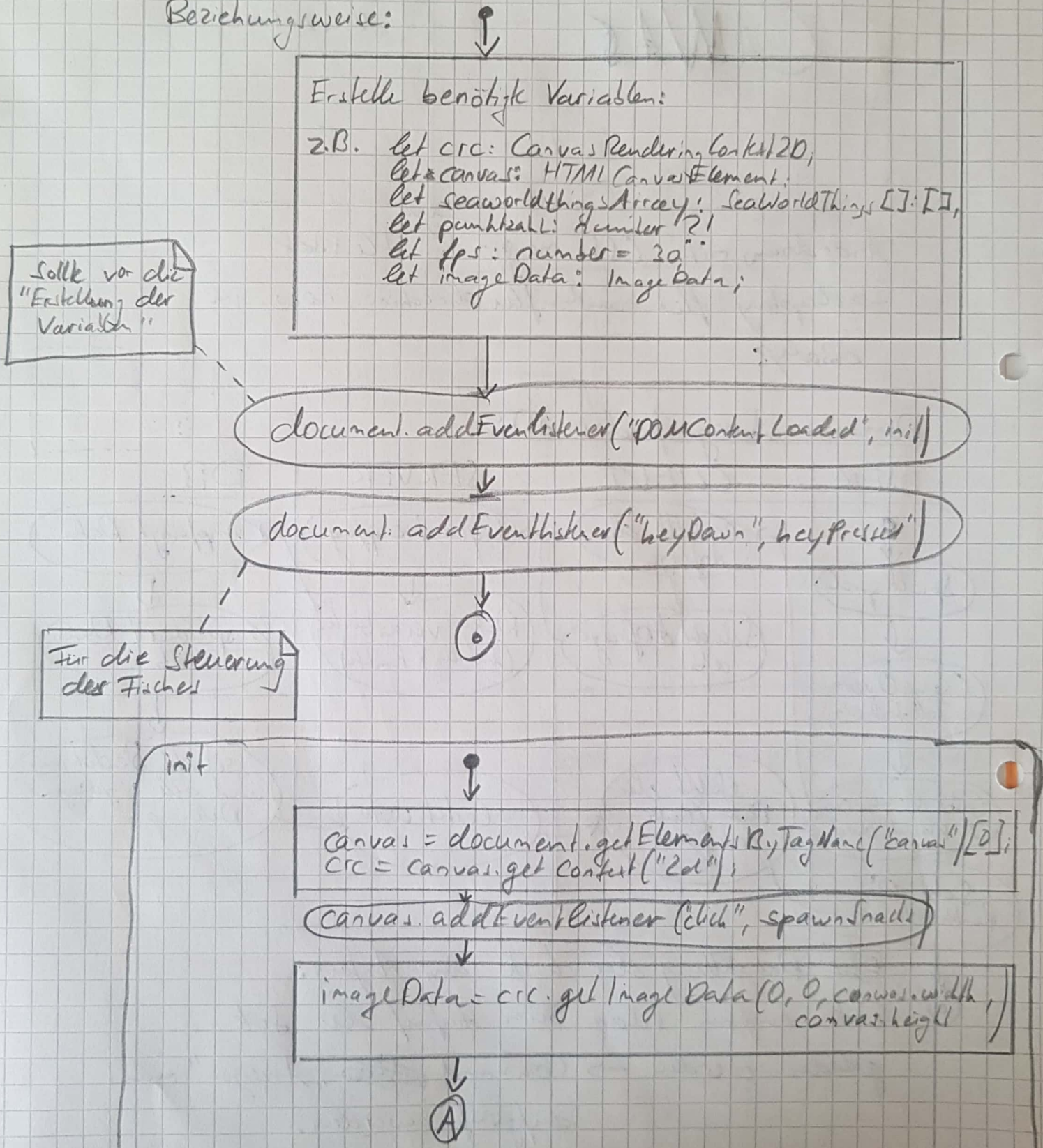


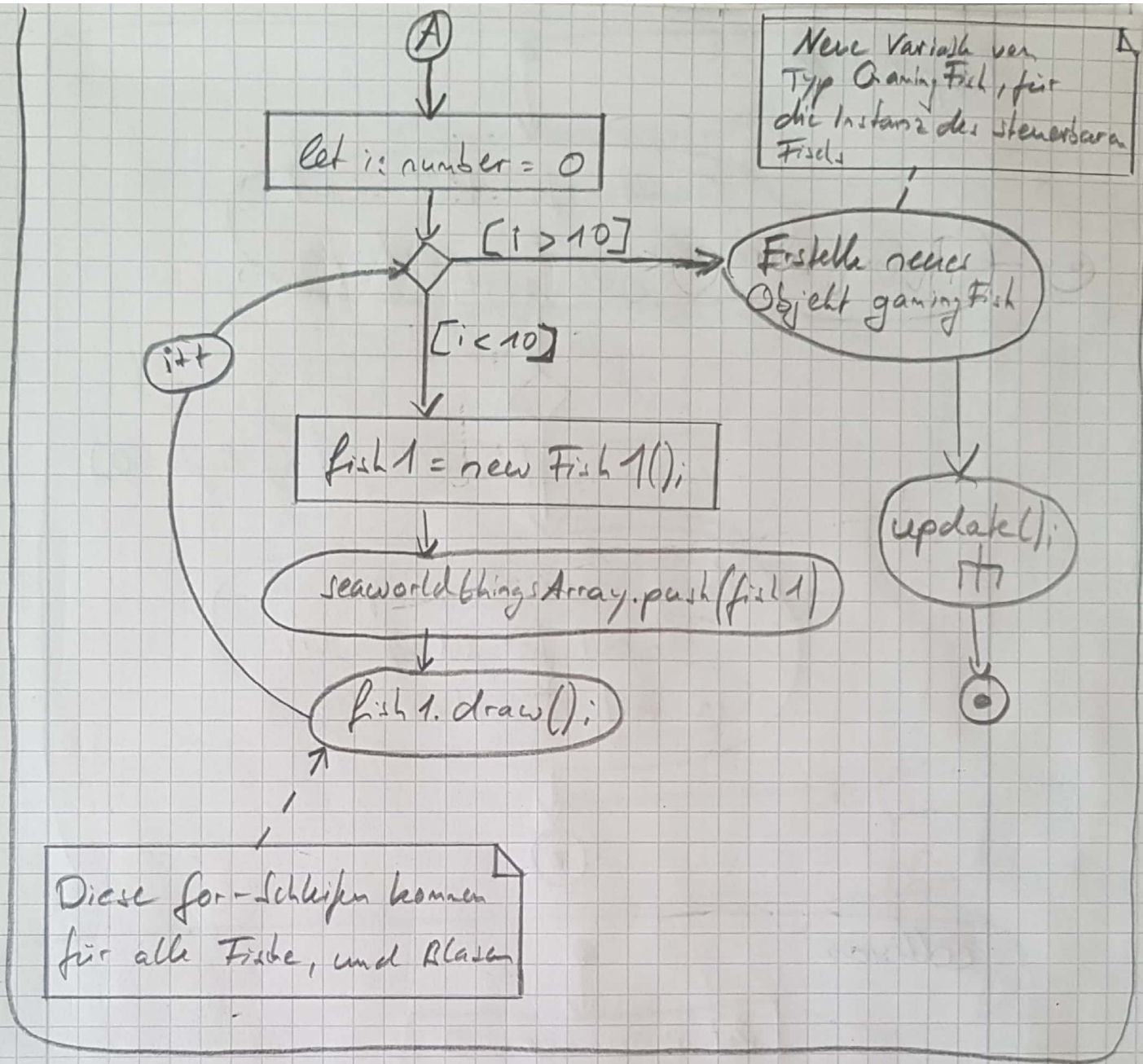
Für Darstellung der Liste am Anfang:
Daten müssen schon beim Aufruf der Seite geladen werden. → Command ~~Show~~ „Anzeigen" muss aufgerufen werden.

Aktivitätsdiagramme:

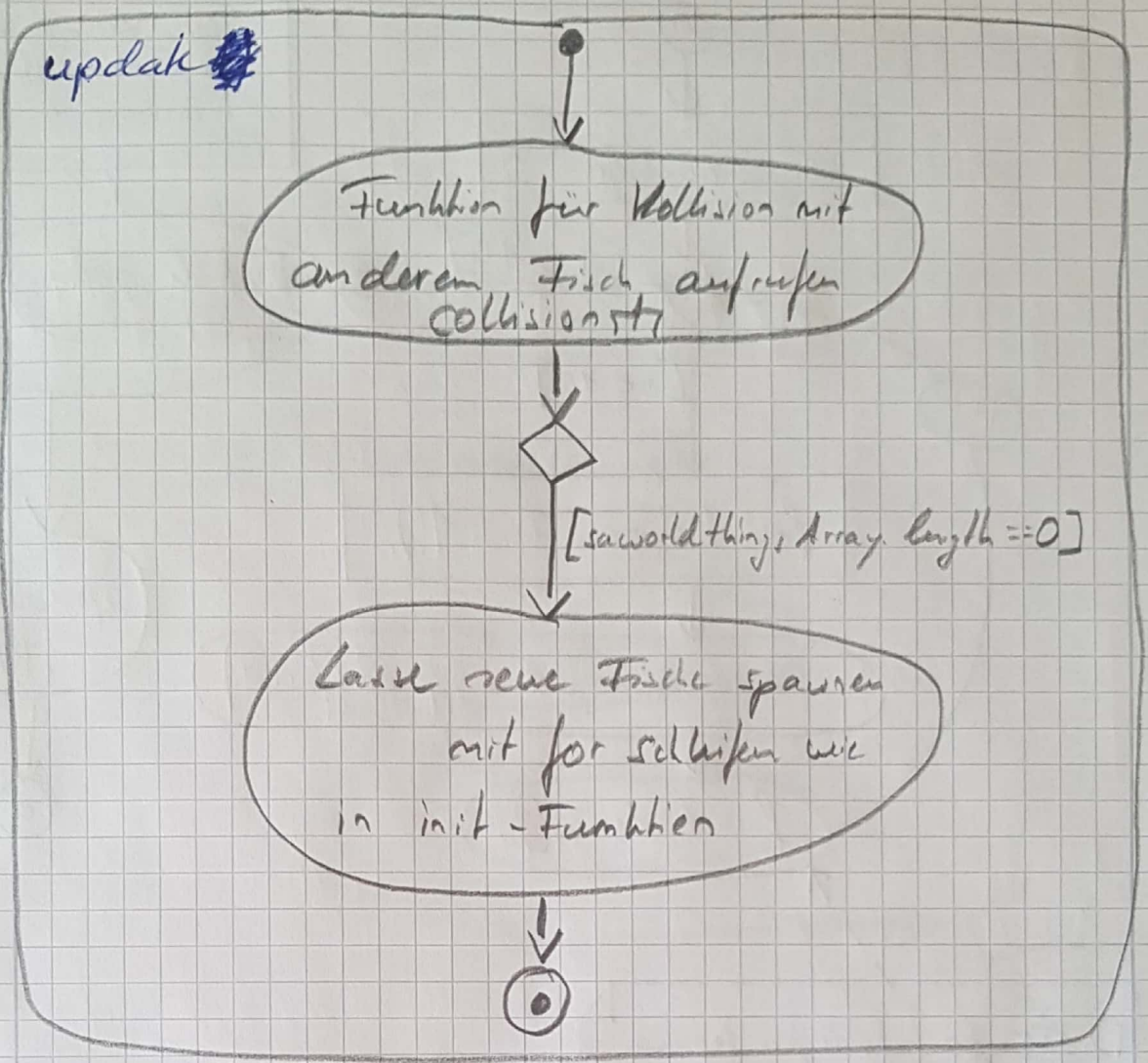


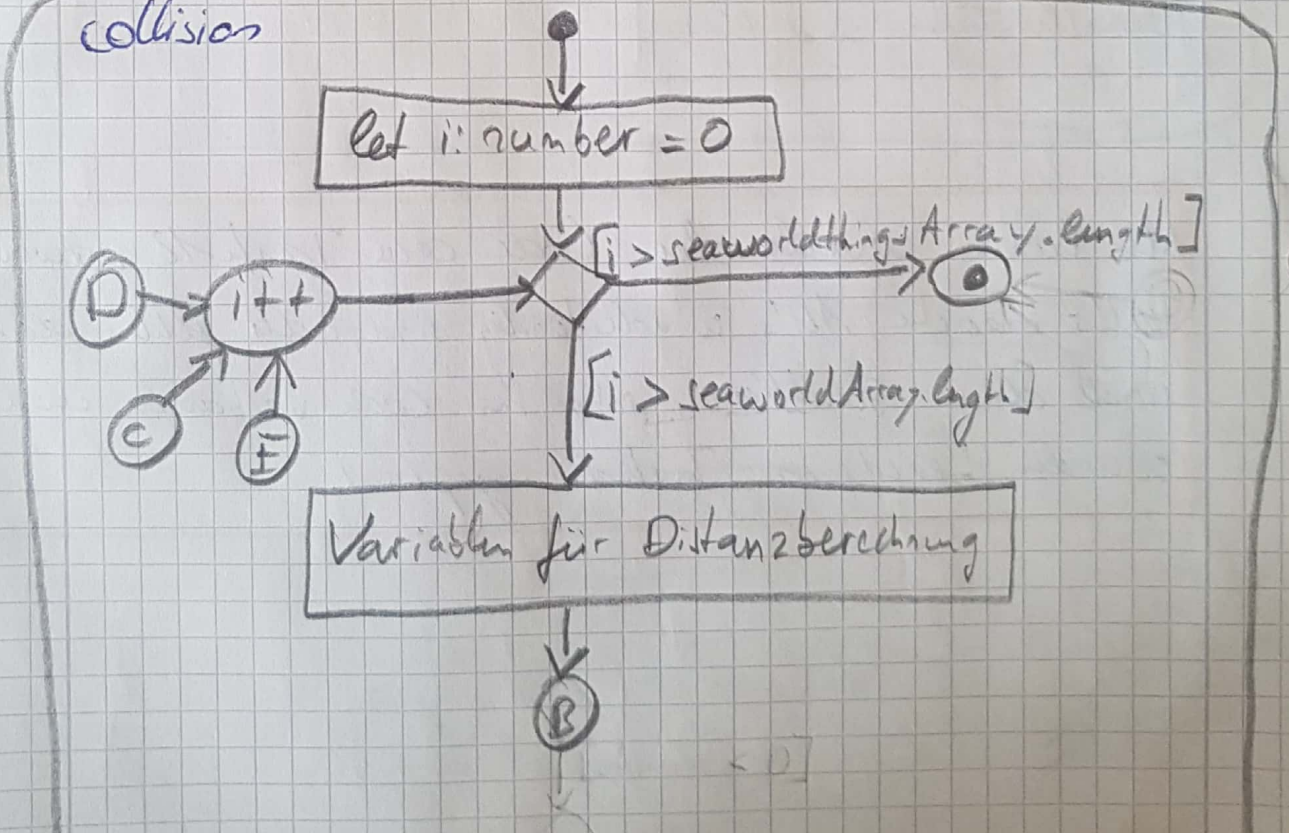
Beziehungsweise:

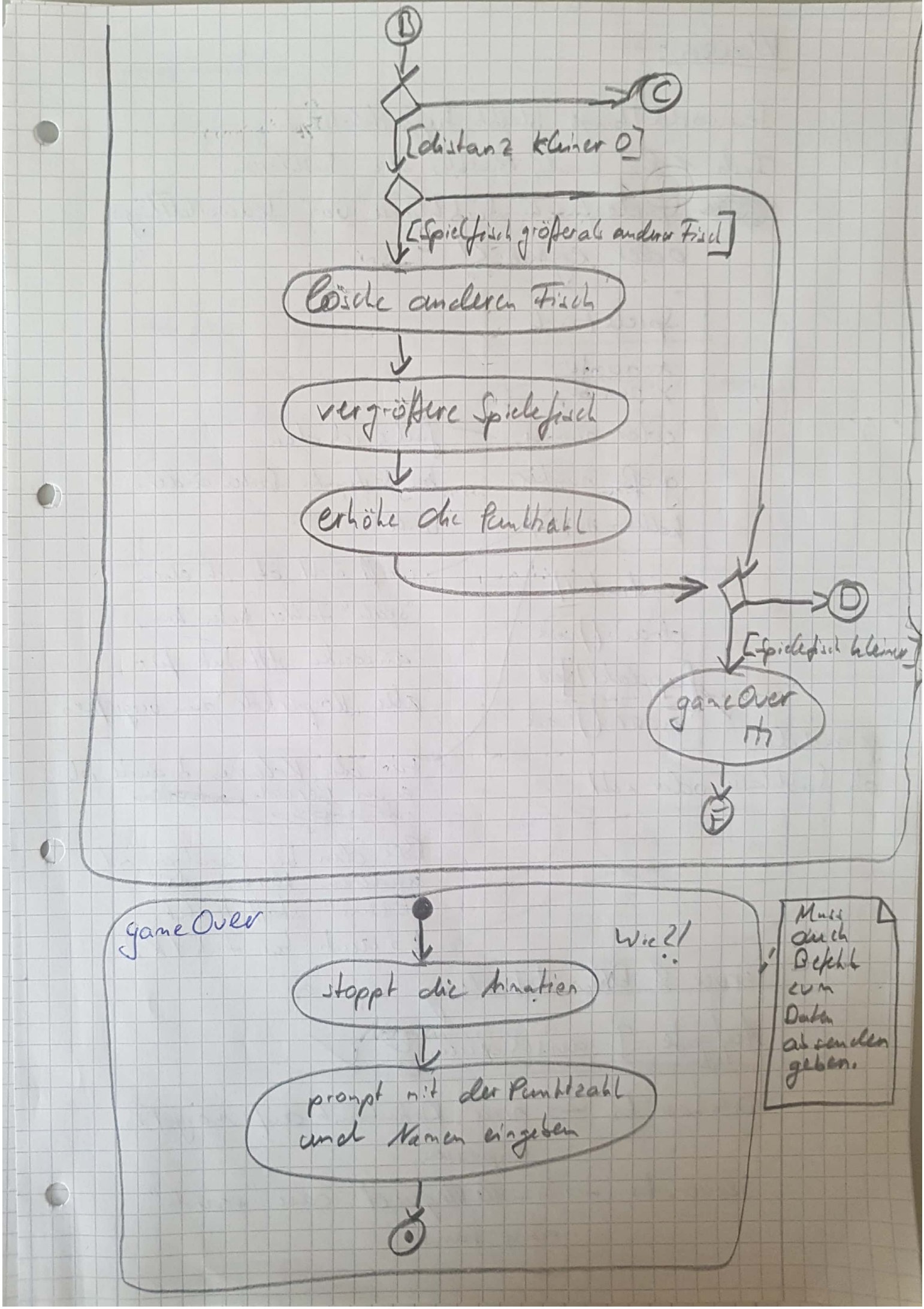




Als Basis wurde der Code aus Task 12 verwendet. Falls manche AD's unvollständig sind oder fehlen, dann sind diese Funktionen schon im Code vorhanden und wurden bereits in Task 12 geplant.







Klassen:

SeaworldThings bleibt Superklasse.
Fish 1 & 2 und Bubbles die Subklassen.
Spielefisch Subklasse von SeaworldThings oder eigene?!

SpieleFisch
x: number; y: number; color: number; größe: number; kollision: number;
constructor(); draw(): void [update(): void] move(): void

falls ich diese bei bestimmten Sachen ändern will

weiß nicht ob ich einen "scale" Faktor haben kann, ansonsten Attribute für alle „Körperteile" zum vergrößern

je nachdem ob Subklasse, oder nicht

Für die Kollision brauche ich einen Bereich

Bei dem bei Berührung mit anderem Bereich das "Essen" statt findet.
↳ auch bei Fish 1 & 2

Server & DB - Anbindung:

Aufgabe 9 ausdenken ✓

Database.ts → neue Datenbank auf mongoDB. anpassen

Server.ts → handleRequest case "search" muss raus.

Types.ts

```
interface Punktzahl {
    [key: string]: string;
}

interface Spieler {
    name: string;
    punktzahl: number;
}
```

← brauche ich für DB und zum Anzeigen.

Steuerung: