

Skizze

↑ Q ← → → http://

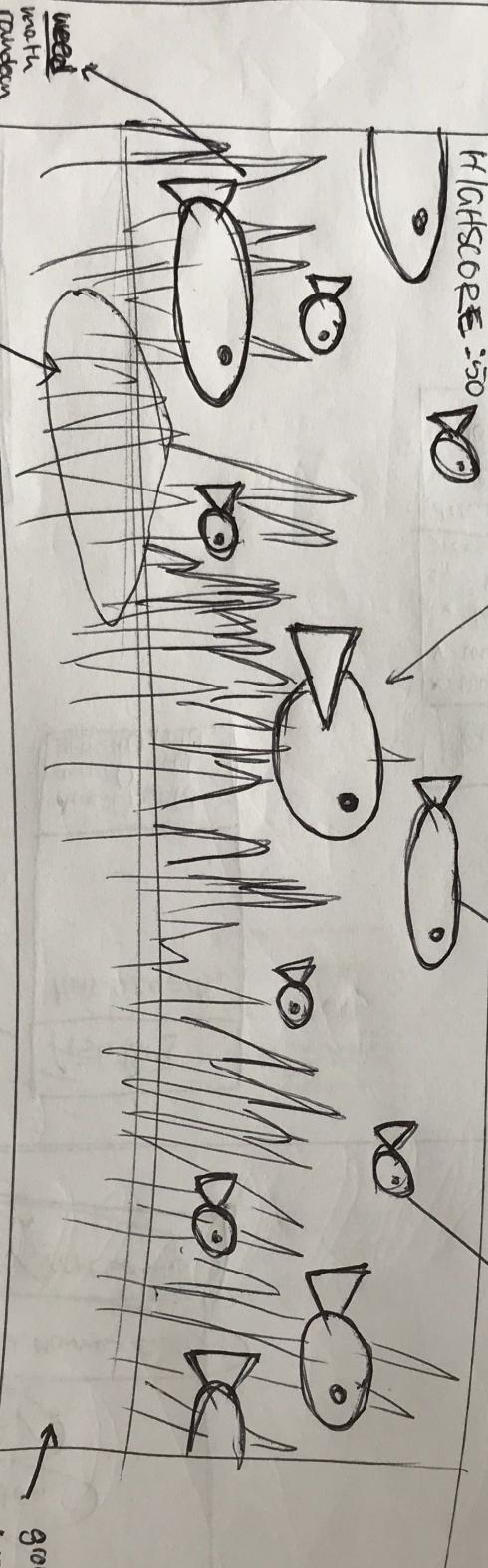
~~Paddlefish red~~
AQUA
~~10/55/1.5~~

AQUARIUM

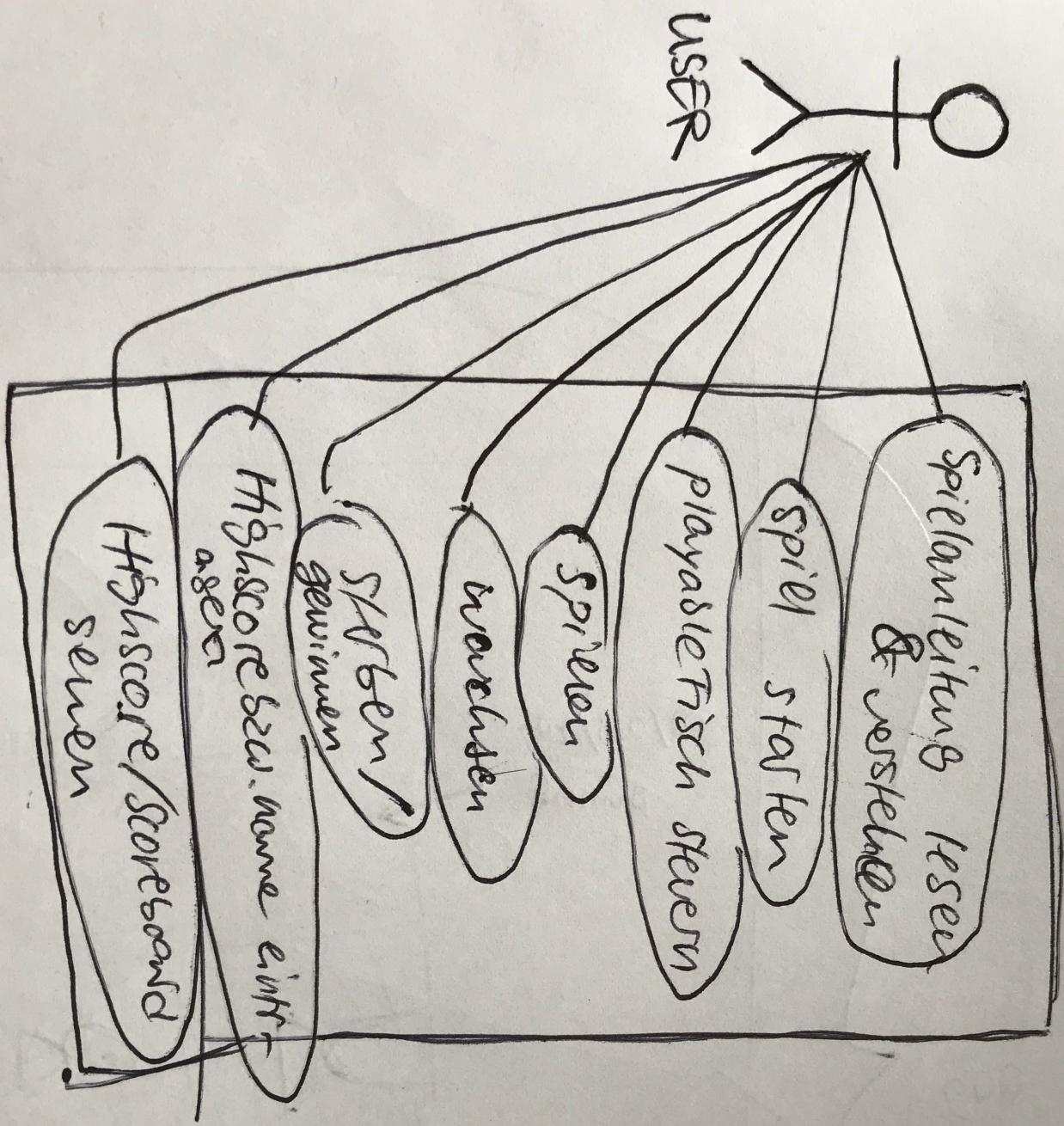
Fish Big / green
20/45, 1.5

Fish Small / purple
15, 25, 1.5

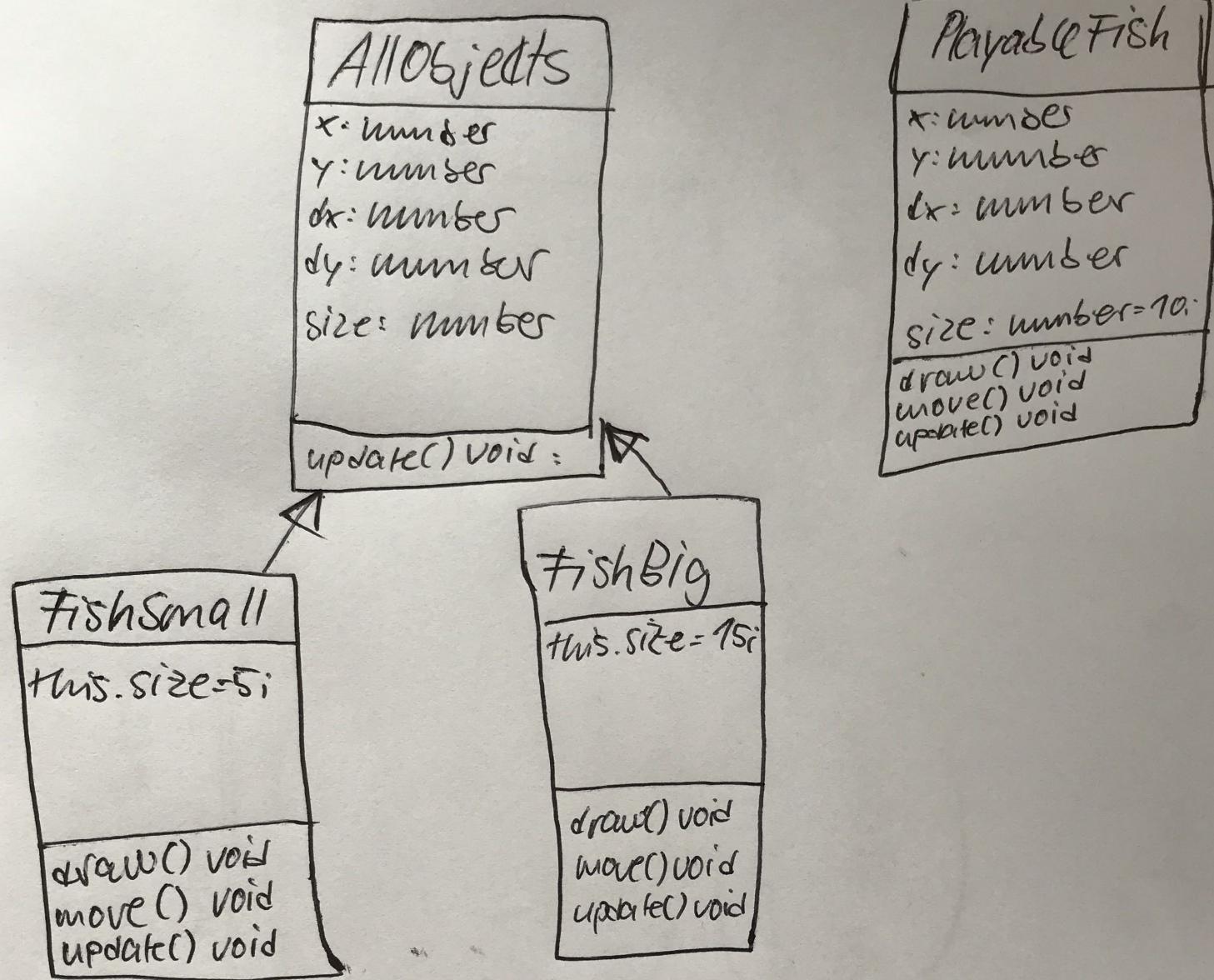
HIGHSCORE : 50

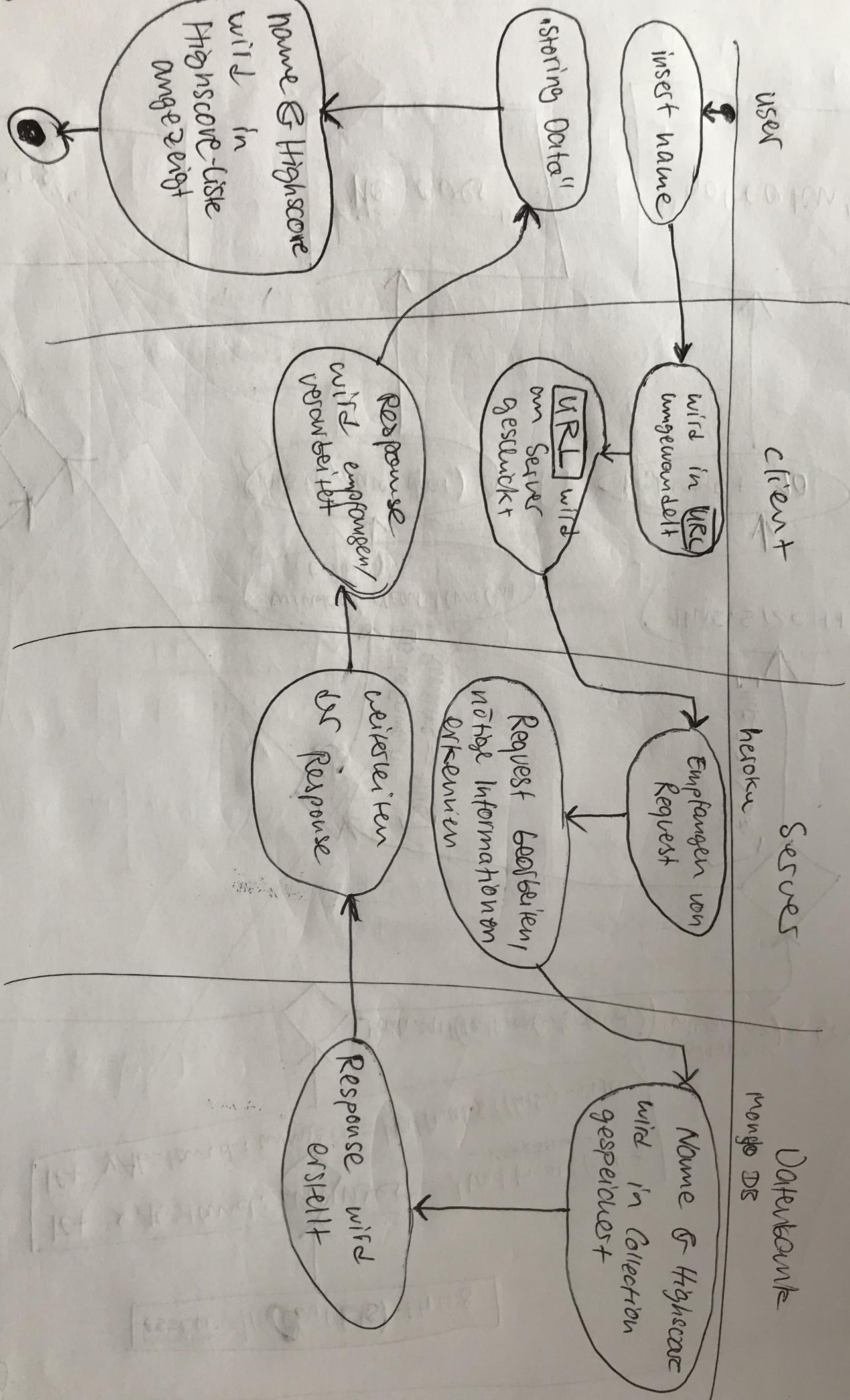


canvas width
800



classdiagramms





HauptProg./Main/Aqua.ts.

```
document.addEventListener("DOMContentLoaded", init);
document.addEventListener("keydown", MoveTheFish);

export let crc: CanvasRenderingContext2D;
export let canvas: HTMLCanvasElement;
export let highscore: number = 0;
export let younamehere: string;
export let time: number;
let fps: number = 30;

let imageData: ImageData
let ArrayAllObjects: AllObjects[] = [];
export let isTheGameStillRunning: boolean = true;
let playablefish: PlayableFish;
```

Interfaces

Scoreboard
[key:sking]:string

AquaHighScore
name: sking; Highscore: number;

init

```
canvas = document.getElementById('canvas')[0];
```

refresh()

drawBackground

```
imageData = crc.getImageData(0, 0, canvas.width, canvas.height);  
playableFish = new PlayableFish();
```

playableFish.draw

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

[i <= 10]

```
let fish : FishBig;
```

```
ArrayAllObjects.push(fish);
```

fish.draw(i)

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

[i <= 10]

```
let fishes: FishSmall
```

```
ArrayAllObjects.push(fish)
```

i++

i++

fishie.draw

update(i);

drawBackground



```
let water: Path 2D
```

werte auslegen, Zeichnen & Füllen

```
let drawGround: Path 2D
```

werte auslesen, zeichnen & füllen

```
let: Stone: Path 2D
```

werte auslesen, zeichnen & füllen.

```
let: drawWeed,
```

let x: number = Math.random
let y: number = Math.random



eating

tastyfish:AllObjects : String

```
[let x:AbsCoord = number = Math.abs(this.x-tastyfish.x);  
let y:AbsCoord = number = Math.abs(this.y-tastyfish.y);]
```

```
[math.sqrt(Math.pow(AbsCoord,2)+AbsCoord,2)<50]
```

"nothing"



"it's over"

"got eaten"



window.clearTimeOut
(time)

if (theGame.isRunning == true)



clear (you've died!)

this.size++

[this.size > tastyfish]



Playable fish move

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

[this.x < 0]

this.x = 0

this.y = canvas.height
this.y = canvas.height

[this.x > canvaswidth]

this.x = canvas.width
this.x = canvas.width

[this.y < 0]

this.y = 0

[default]

moveTheFish

keyBoardEvent

{if. keycode == 65 }

[else. if. keycode
== 95]

playable-fish
·x == 25

playable-fish.
direction = links

playable-fish
·x += 25

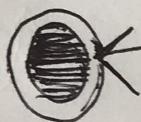
[else. if. keycode
== 87]

playablefish.y -= 25

[else. if. keycode
== 95]

playablefish
·y += 25

playable.fish direction = "rechts"



Server
Daten

refresh()

let query: string = "command
=refresh"

Send Request(query, handlefind
Response)

handlefindResponse - event: ProgressEvent

let xhr: XMLHttpRequest....

[xhr.readyState == XMLHttpRequest.DONE]

let ArrayGamers: Array<Highscore> = JSON

document.getelementsbyID("playername")
document.getelementsbyID("score")....

let i: number = ArrayGamers.length - 1

[i < ArrayGamers.length - 1]

document.getelementsbyID("playername").innerHTML ...

i++

Fish Big move

move func.
gleich bei
fishsmall !

~~this.x += this.dx;~~

[this.x + 30 > 900]

this.x = 0

insert()

Server
Database

let query : string = "command =
insert"

query += "name = " + yournamehere +
"highscore = " + highscore

sendRequest(query, handleInsertResponse)

update()

```
window.setTimeOut(update 1000/fps);
crc.clearRect(0,0,canvas.width,canvas.height);
crc.putImageData(imageData,0,0);
```

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

i++

[i < ArrayAllObjects.length]

ArrayAllObjects[i].update()

[playablefish.eating.ArrayAllObjects[i] == goerten]

ArrayAllObjects.splice

[playablefish.eating(ArrayAllObjects[i]) == itsOver & isTheGameStillRunning == true]

yournamehere = prompt("YourScore" + highscore + "insert your name")

insert(i,highscore)

refresh(i,highscore)

[isTheGameStillRunning == false]

HighScore in
Canvas anzeigen

[ArrayAllObjects.length == 0 & isTheGameStillRunning == true]

youwon(i,highscore)

