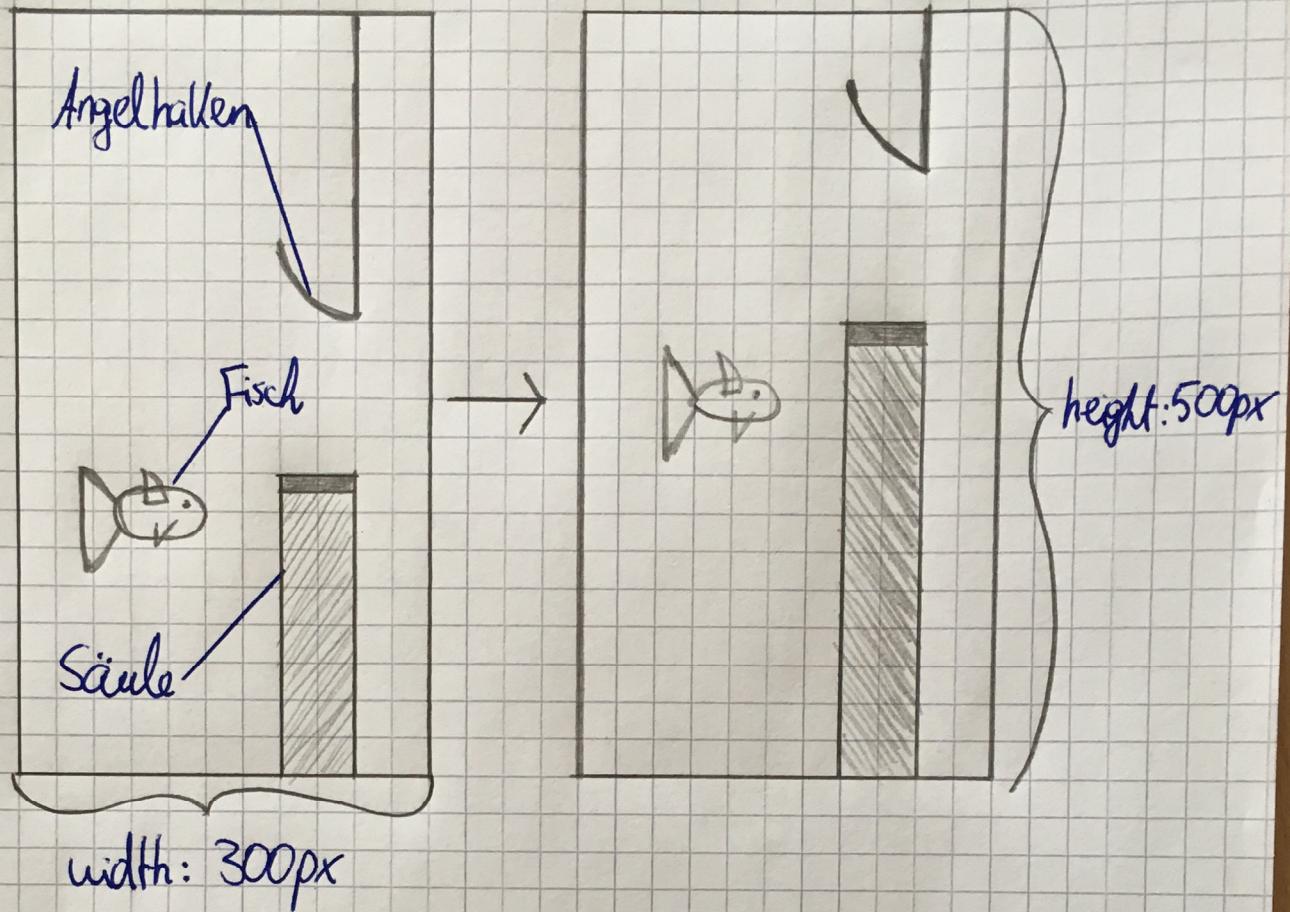


Abschlussaufgabe - Skizze und UCD

Skizze:



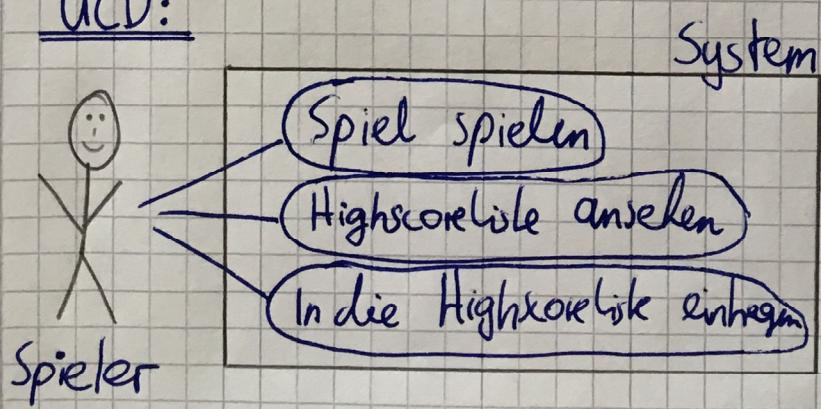
Lücke zwischen Hindernissen : 100px

Fish height : 50 px

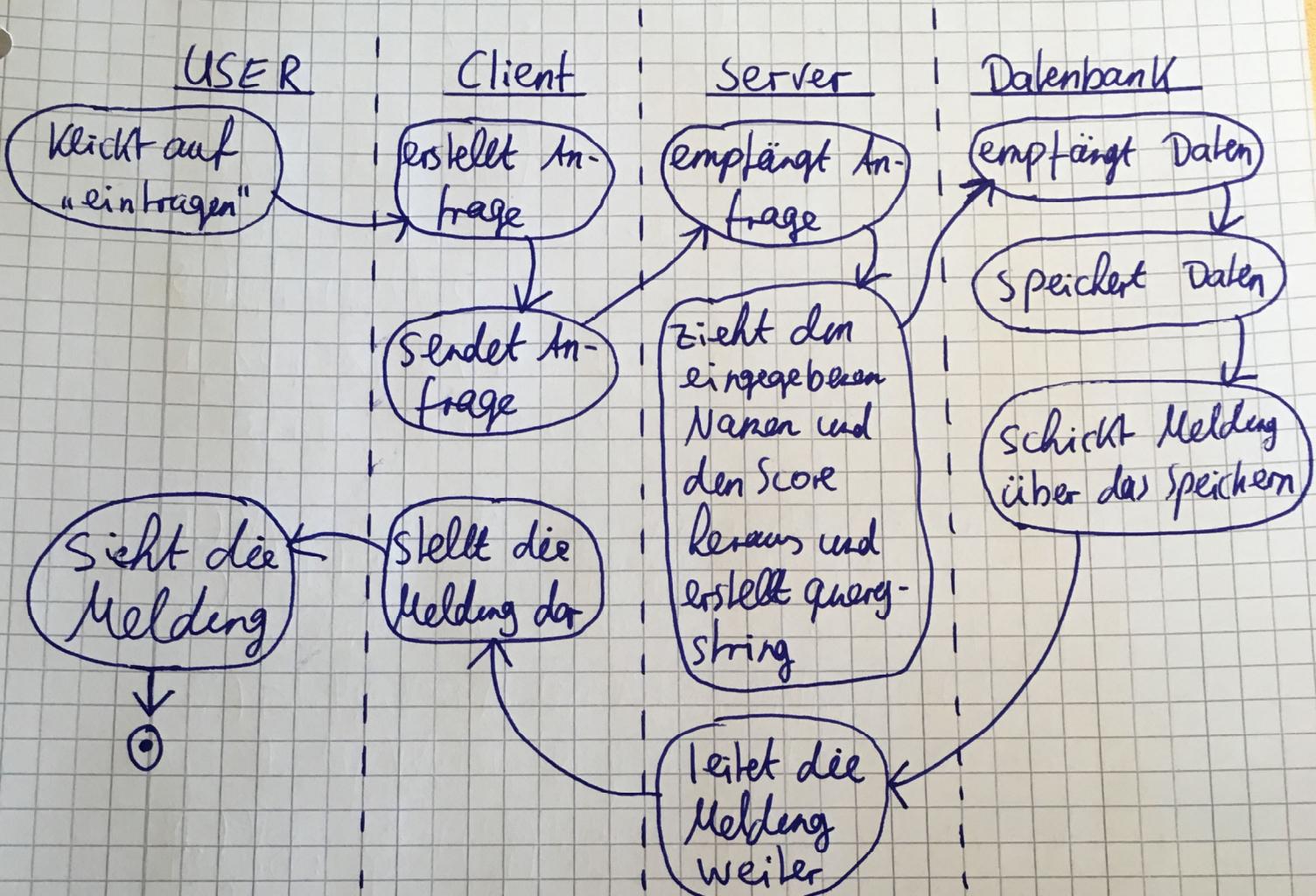
Hindernis width : 50 px

Hindernis height : variabel

UCD:



Domainübergreifendes AD:



Types:

```
interface AssocStringString {  
  [key: string]: string  
}
```

```
interface PlayerData {  
  name: string;  
  score: number;  
}
```

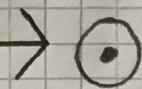
init ()



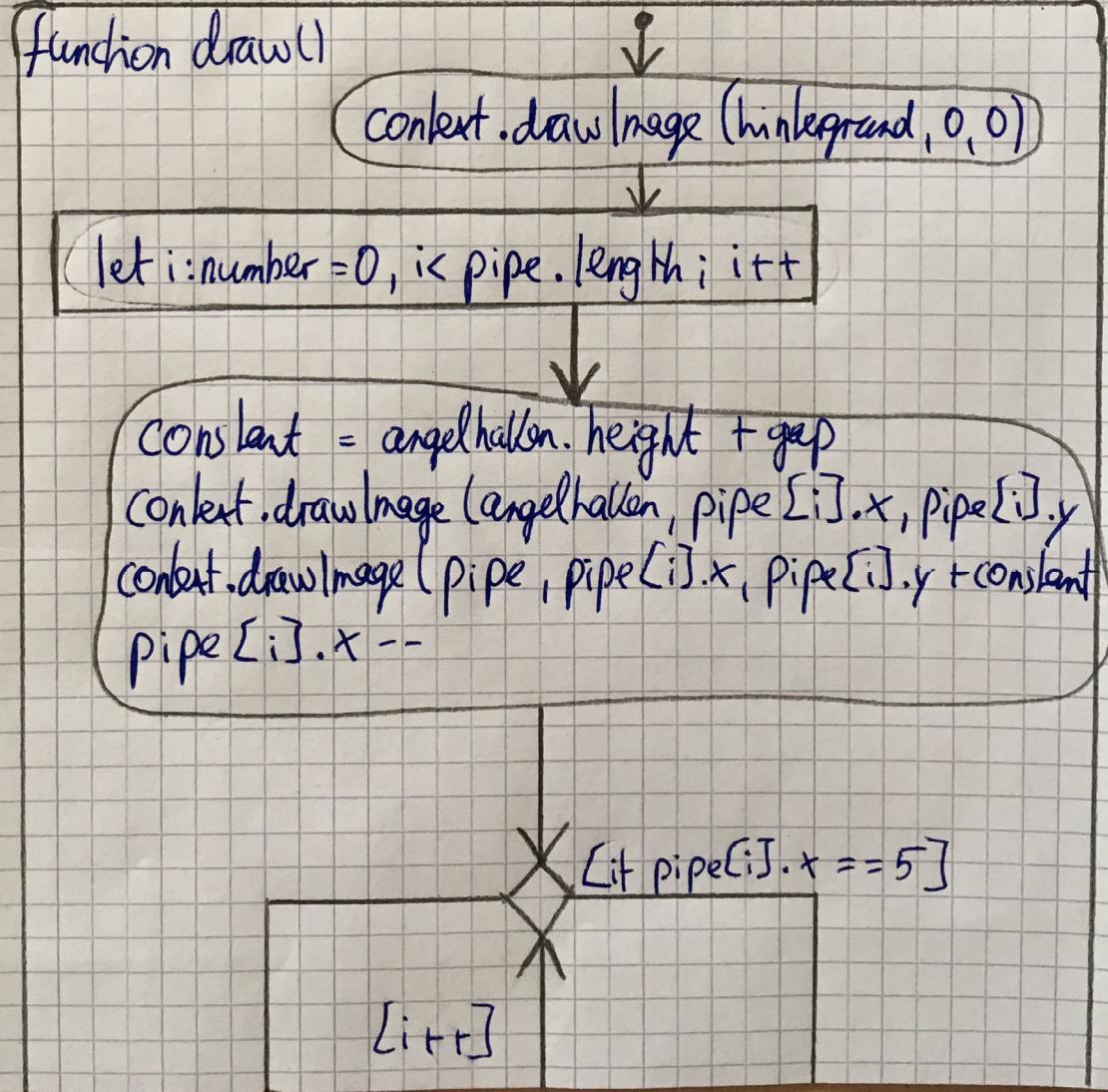
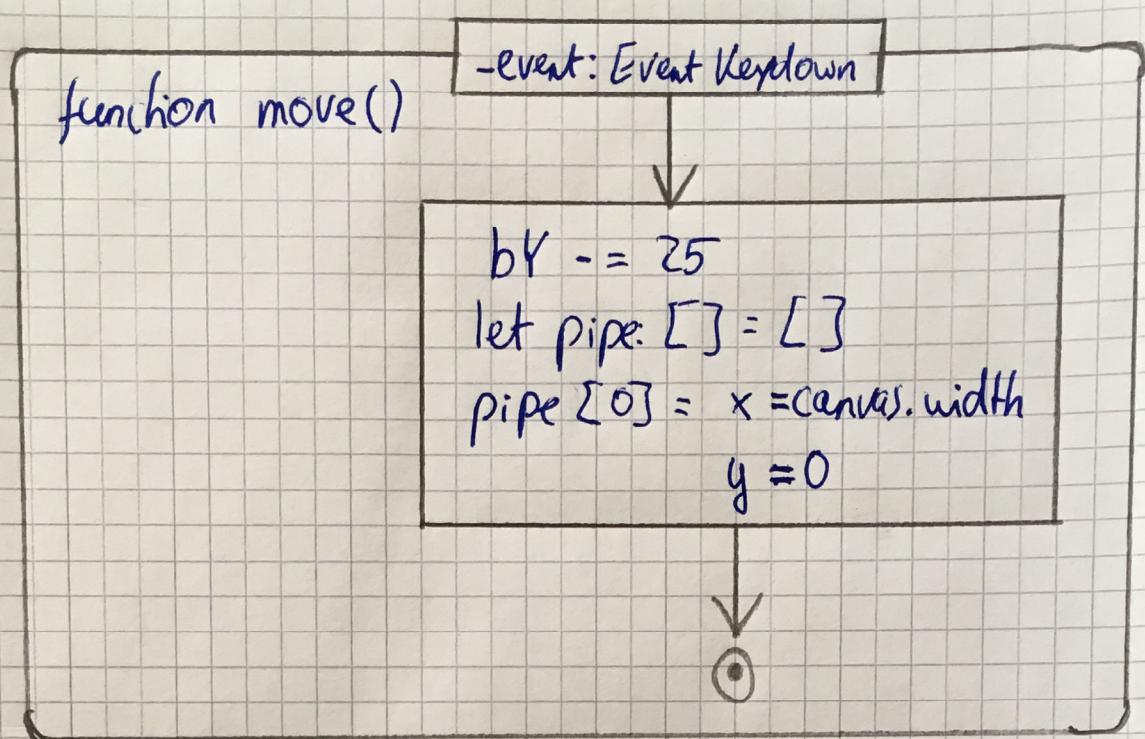
```
let canvas = getElementById ("canvas")
let context = canvas.getContext ("2D")
let fisch : HTMLImageElement = new Image()
let hintergrund : HTMLImageElement = new Image()
let Vordergrund : HTMLImageElement = new Image()
let angelhaken : HTMLImageElement = new Image()
let pipe : HTMLImageElement = new Image()
let sand : new Path2D
let gap : number = 85
let constant
let bX : number = 10
let bY : number = 150
let gravity : number = 1.5
let score : number = 0
let scor : HTMLAudio = new Audio()
let lose : HTMLAudio = new Audio()
```



```
fisch .src = "..."
hintergrund .src = "..."
Vordergrund .src = "..."
angelhaken .src = "..."
pipe .src = "..."
sand .scor .src = ...
lose .src = ..."
```



```
>.addEventListener ('keydown', move it) → ○
```



SCORE ++
SCOR . play()

[if bx + fisch.width >= pipe[i].x &&
bx <= pipe[i].x + angelhaken.width &&
(bx <= pipe[i].y + angelhaken.height ||
bx + fisch.height >= pipe[i].y +
constant) ||
bx + fisch.height >= canvas.height -
Vordergrund.height]

lose . play

toggle Highscore

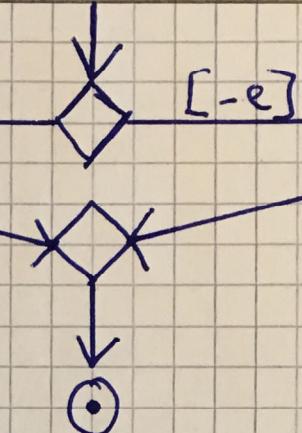
> .addEventListener("DOMContentLoaded", init) → ○

antwort

- e: Mongo.MongoError
AssocStringString: PlayerData []

Callback (JSON
stringify)

Callback „Error“

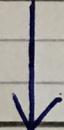


handle Request

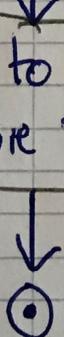
- request: Http.IncomingMessage,
- response: Http.ServerResponse



```
let query: AssocStringString = URL.parse(request.url,  
true).query;  
let command: string = query[„command“]
```



let highscore: PlayerData



insert to Database
and store Data

