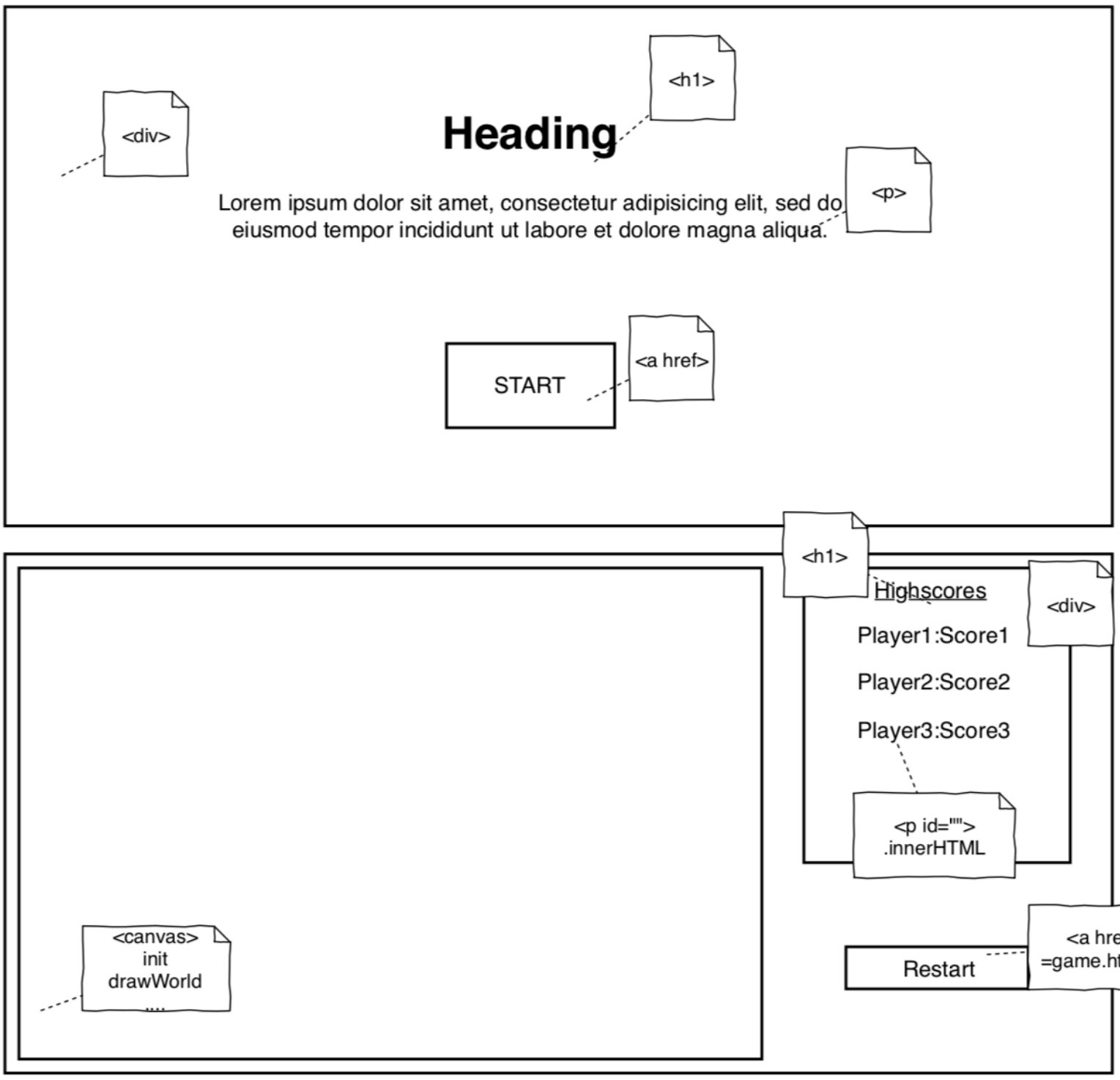
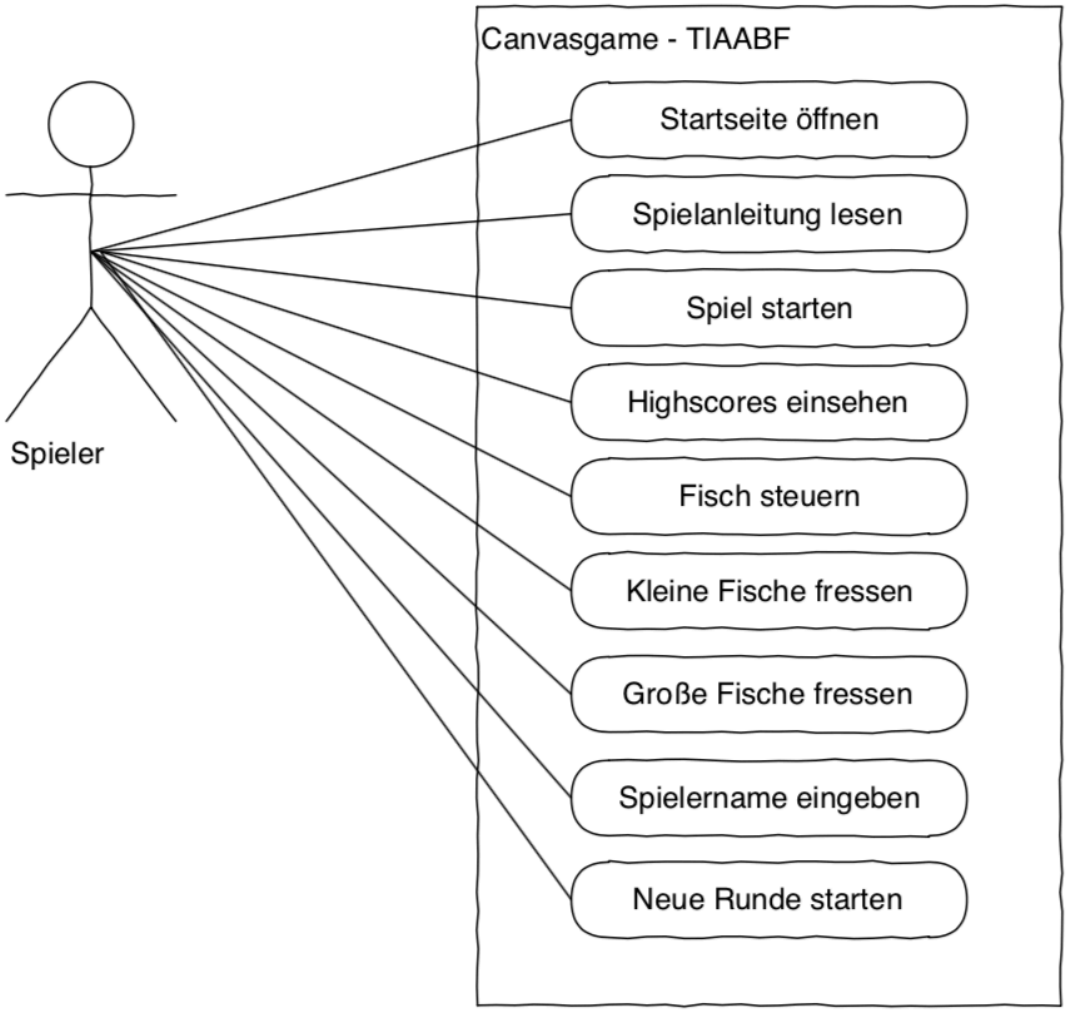
Technische Analyse:

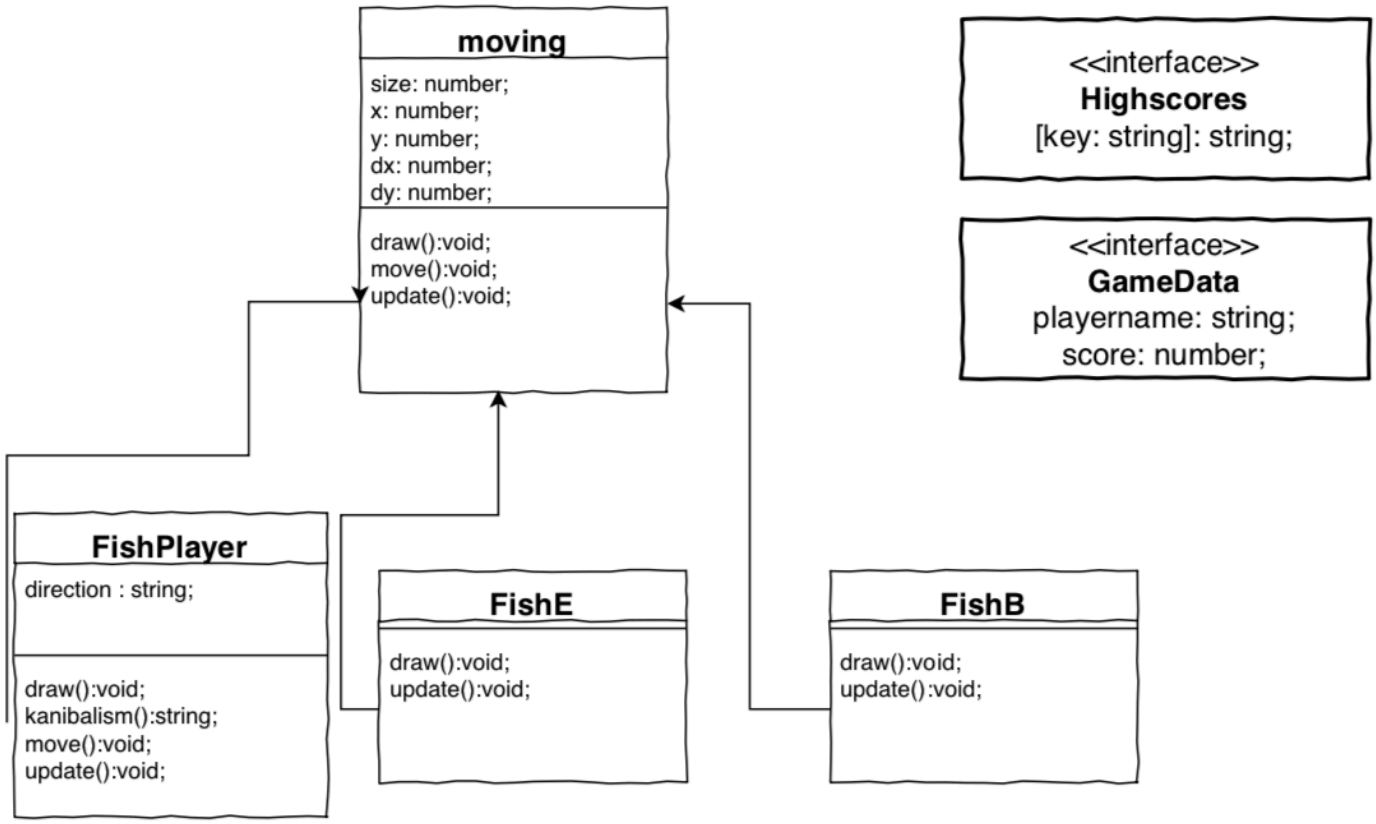
Skizze:



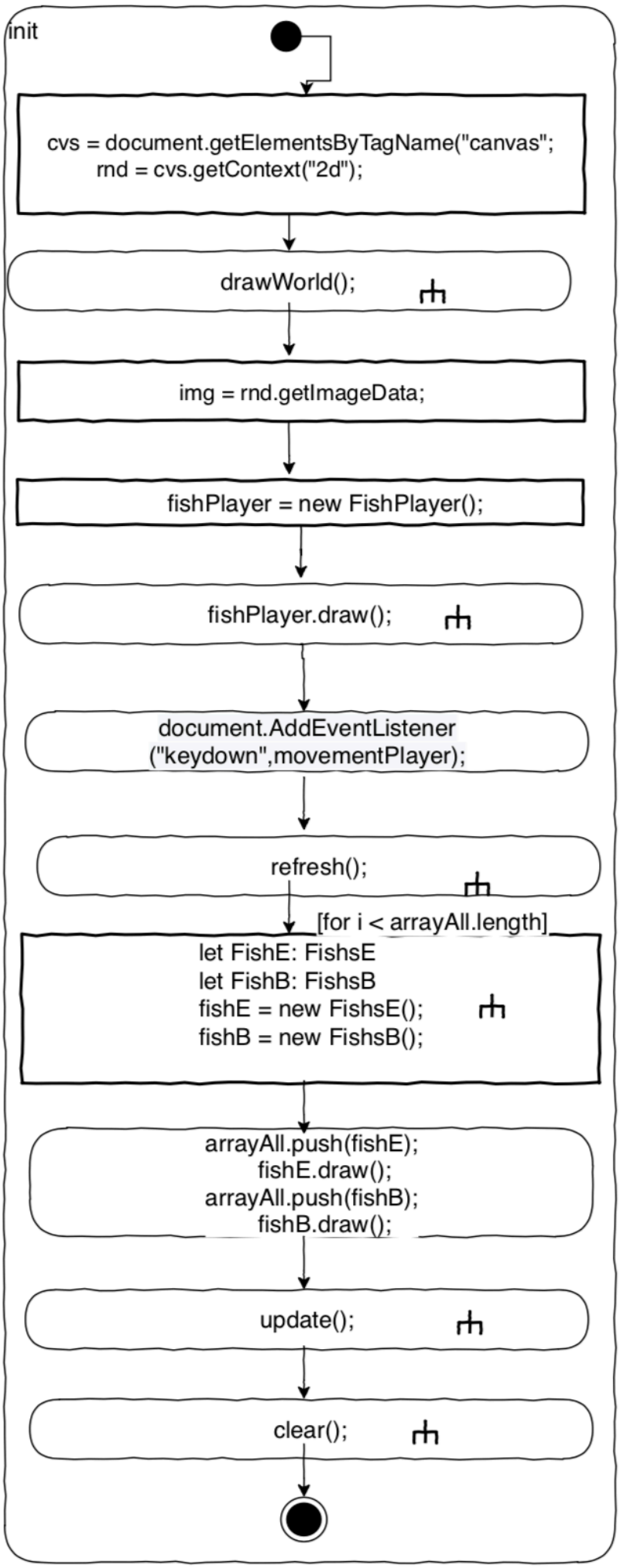
Anwendungsfall Diagramm:

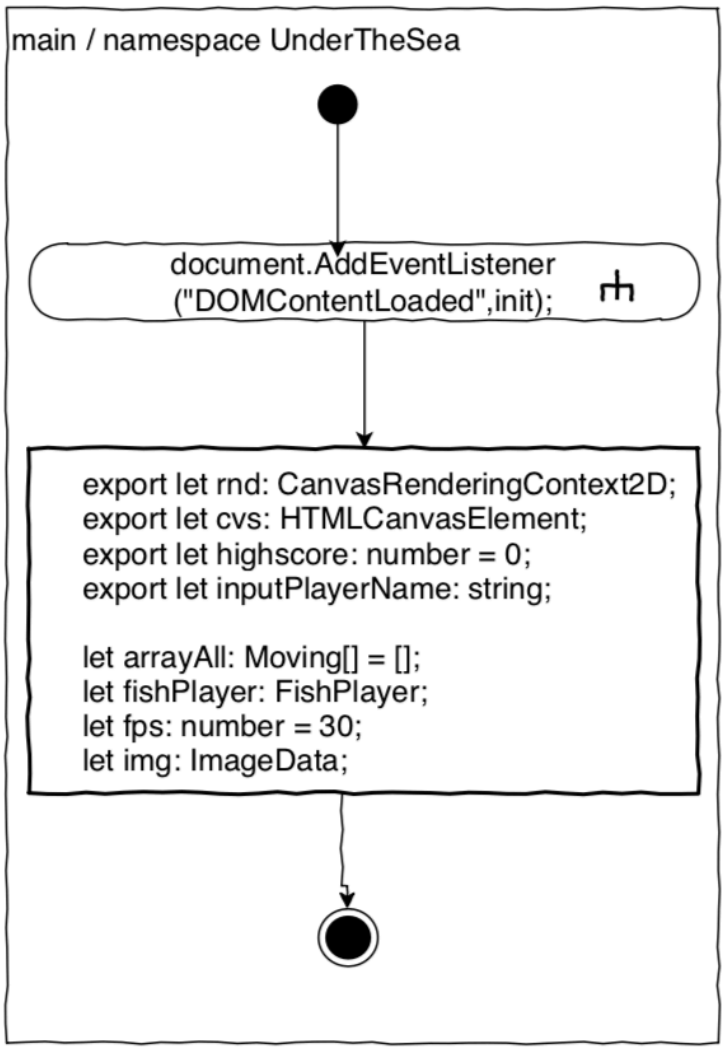


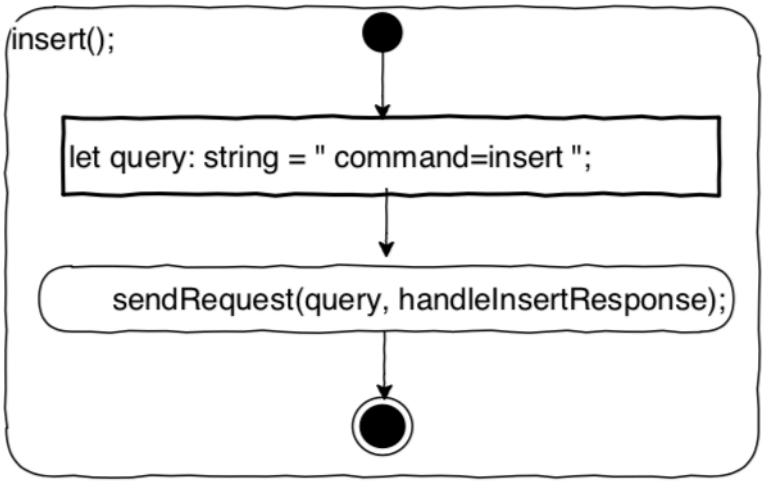
Klassendiagramm:

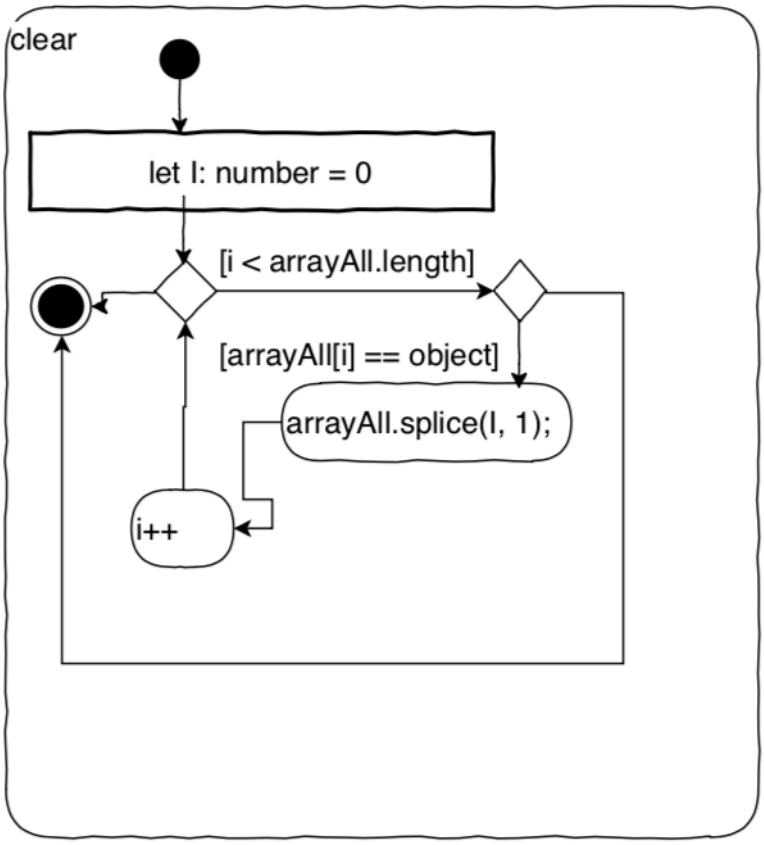


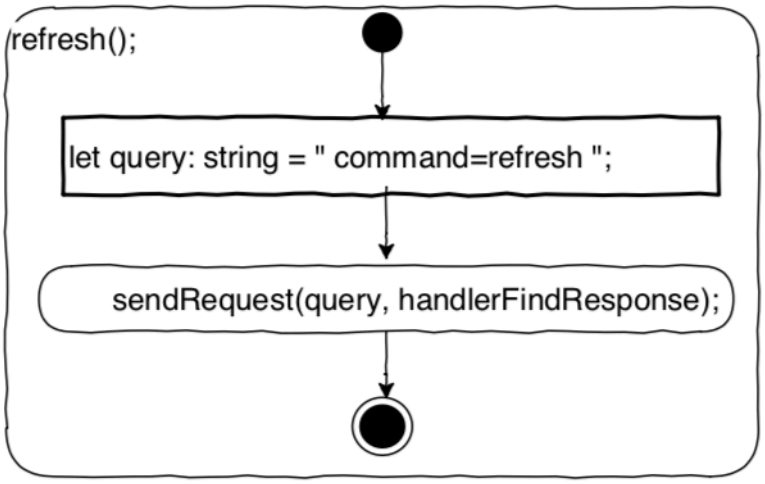
Aktivitäts-Diagramme

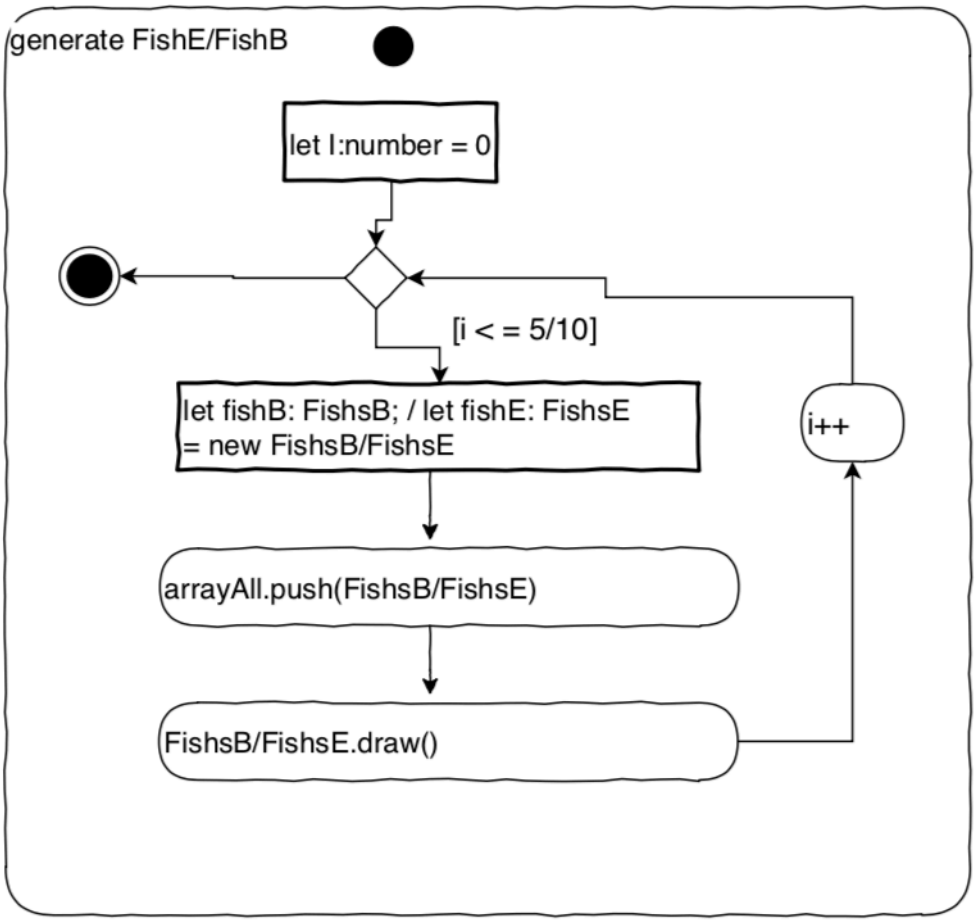


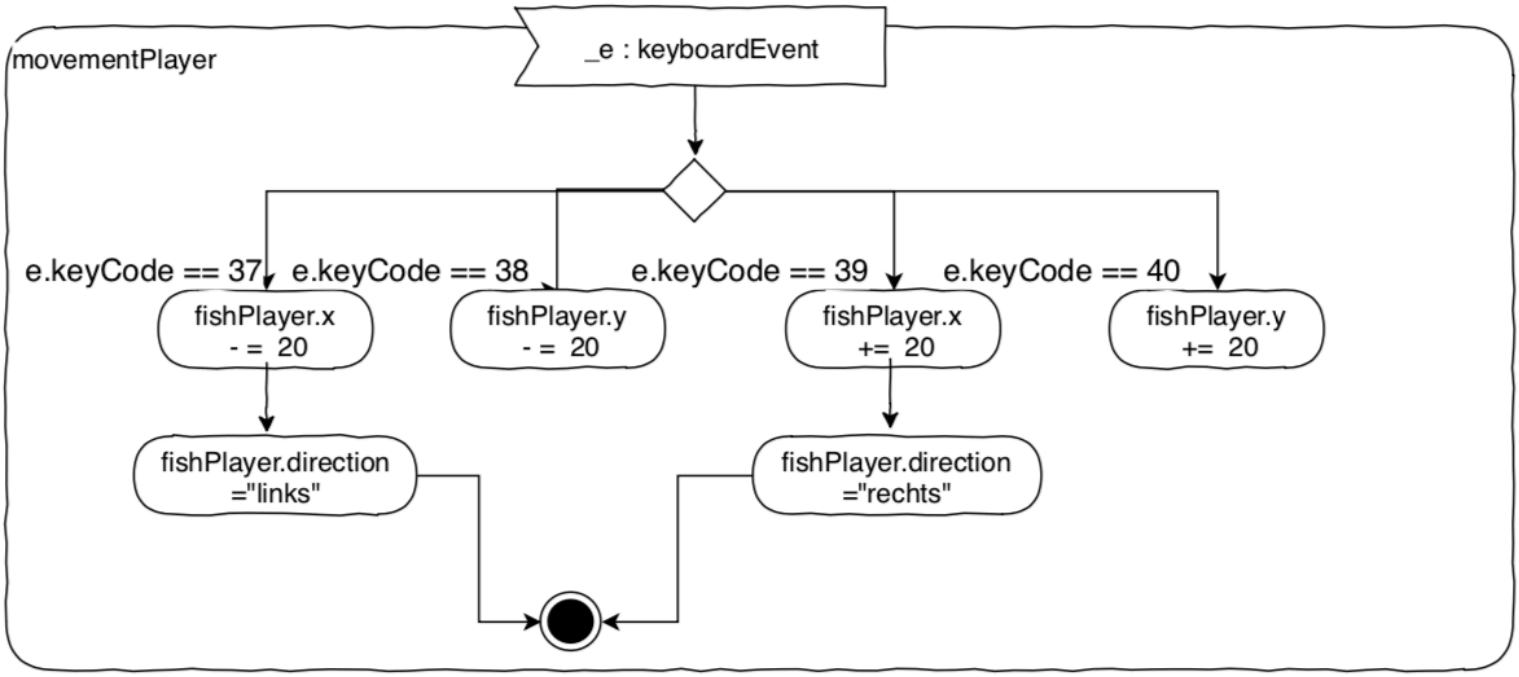


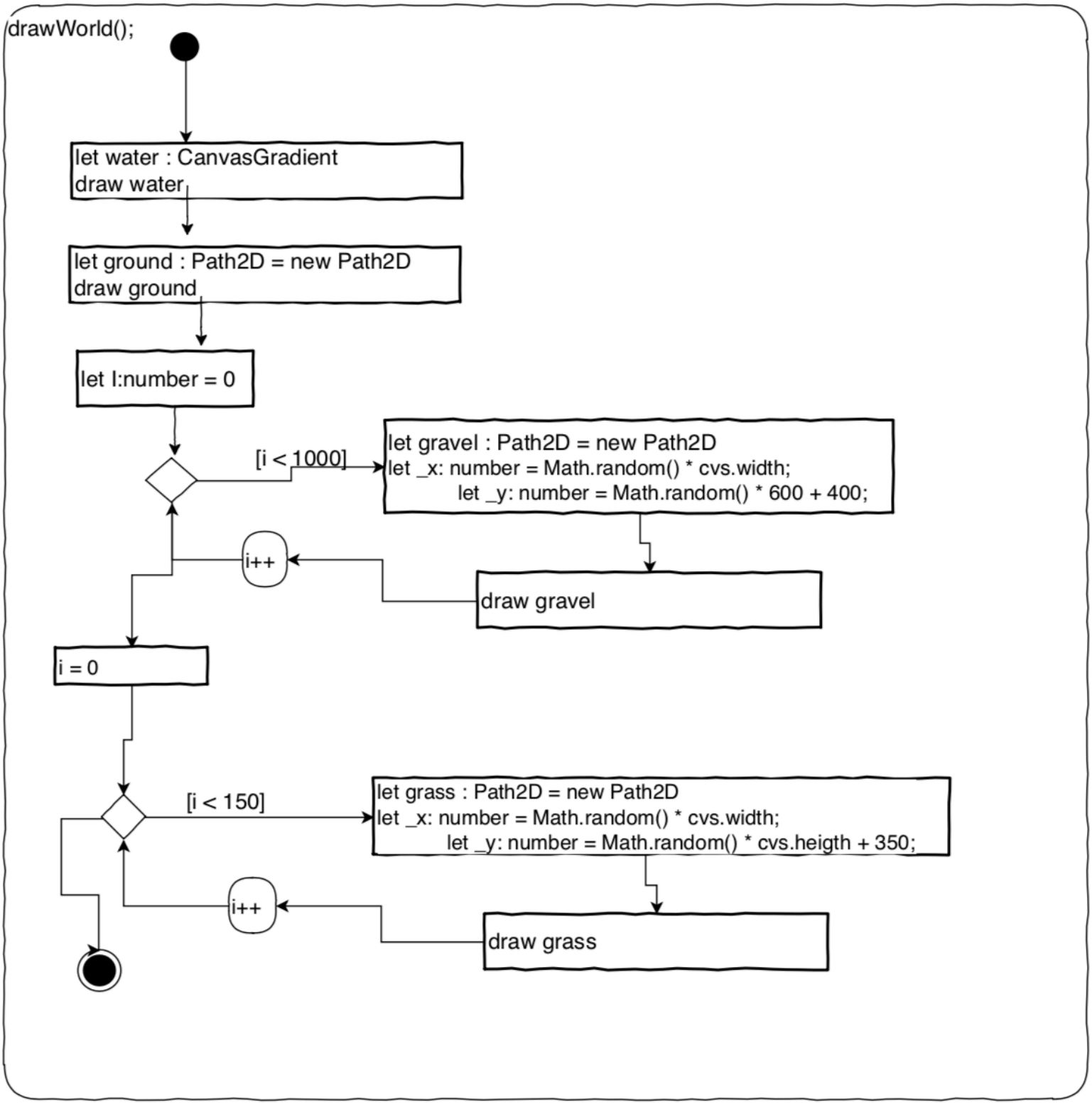

insert();
let query: string = " command=insert ";
sendRequest(query, handleInsertResponse);

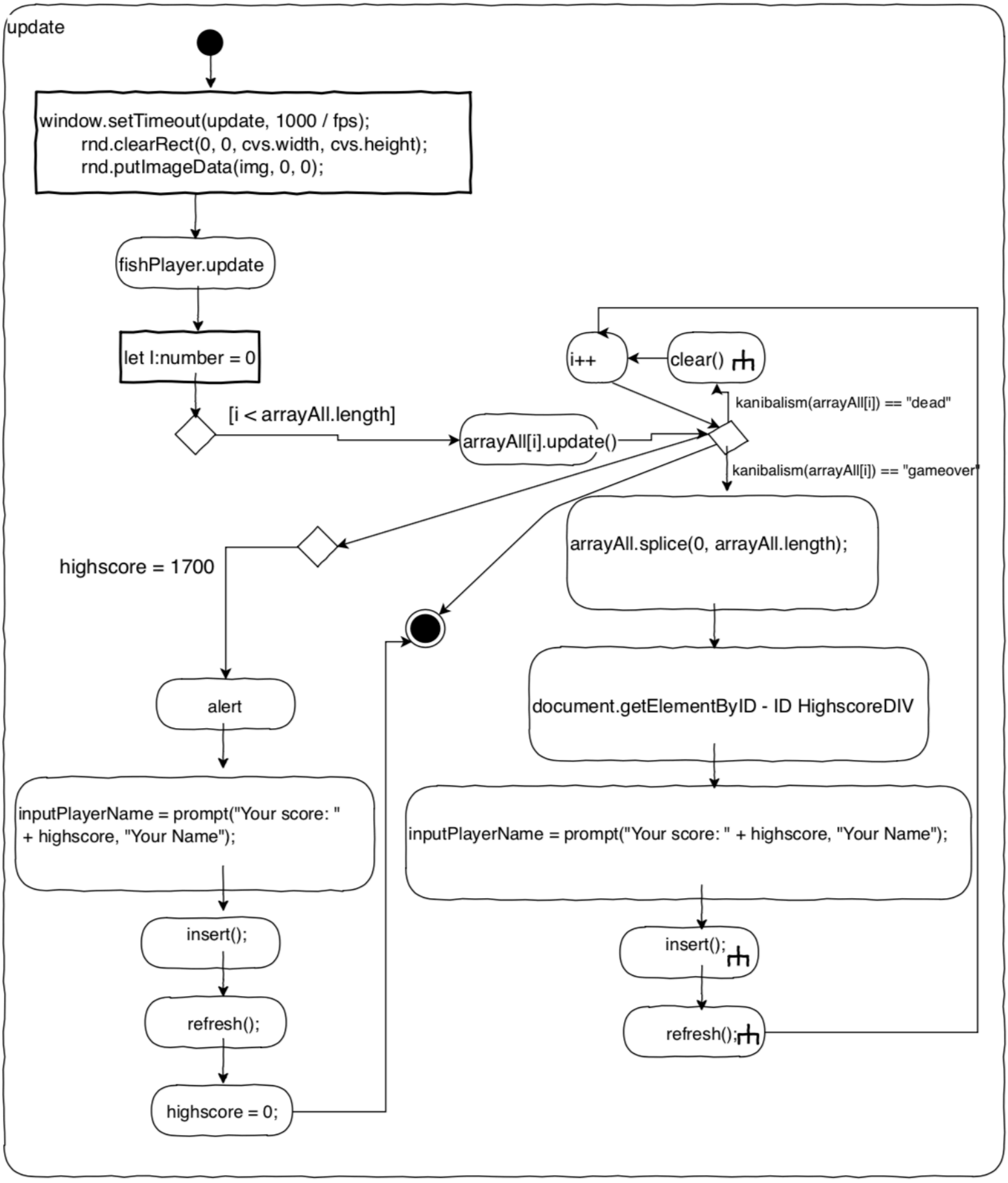

clear
let I: number = 0
[i < arrayAll.length]
[arrayAll[i] == object]
arrayAll.splice(I, 1);
i++


refresh();
let query: string = " command=refresh ";
sendRequest(query, handlerFindResponse);

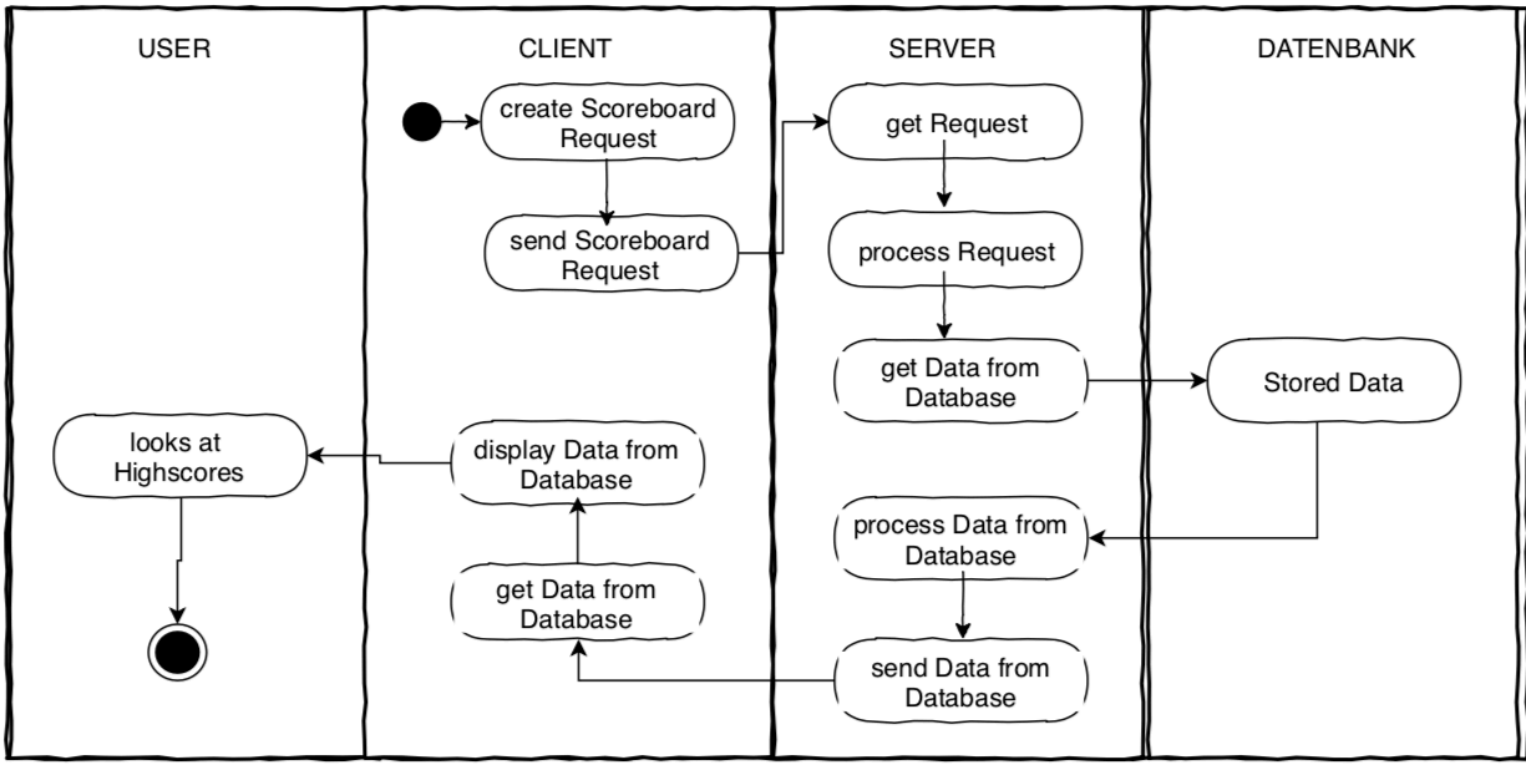
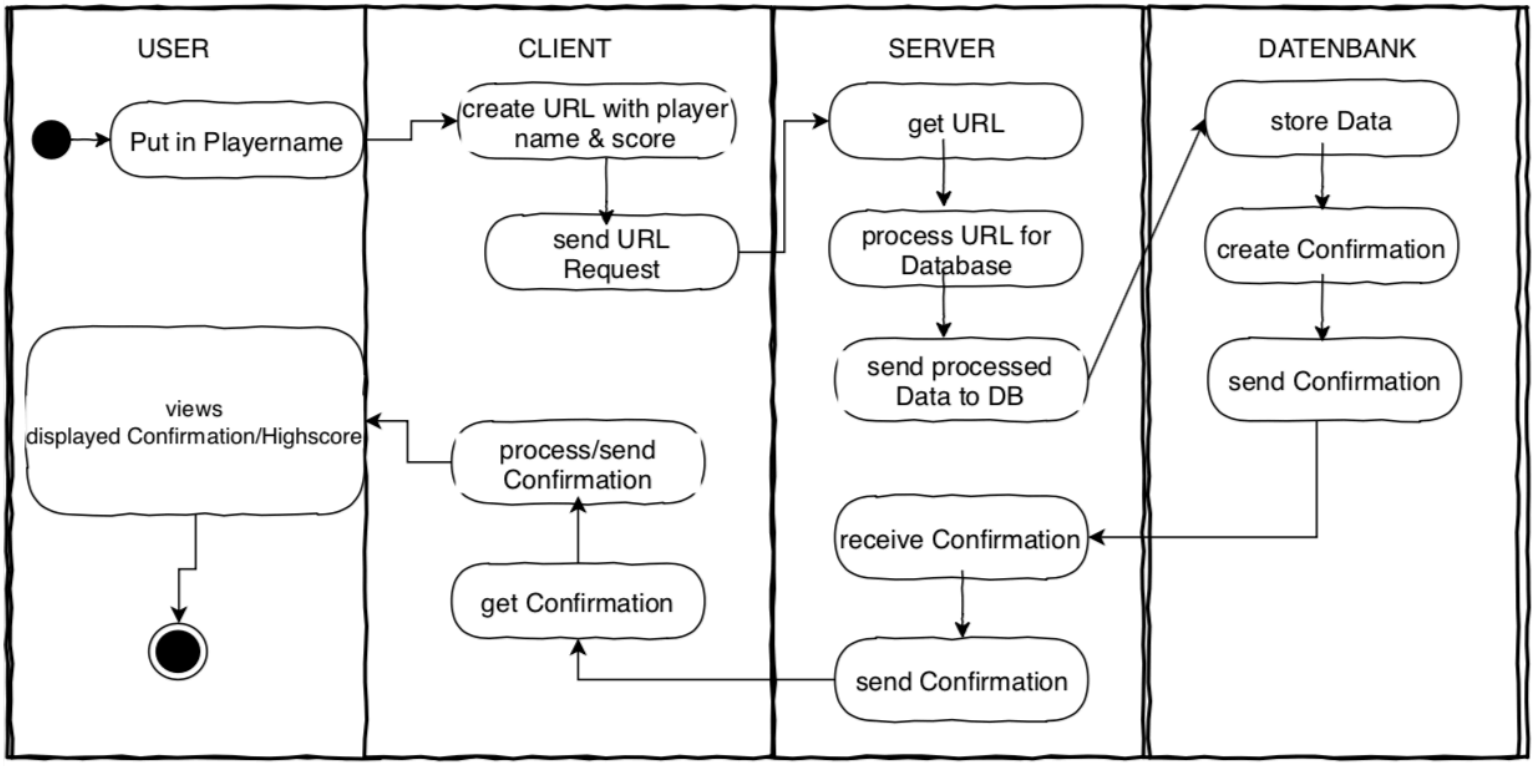

generate FishE/FishB
let I:number = 0
[i < = 5/10]
let fishB: FishsB; / let fishE: FishsE
= new FishsB/FishsE
arrayAll.push(FishsB/FishsE)
FishsB/FishsE.draw()
i++


movementPlayer
_e : keyboardEvent
e.keyCode == 37
e.keyCode == 38
e.keyCode == 39
e.keyCode == 40
fishPlayer.x
- = 20
fishPlayer.y
- = 20
fishPlayer.x
+= 20
fishPlayer.y
+= 20
fishPlayer.direction
="links"
fishPlayer.direction
="rechts"


drawWorld();
let water : CanvasGradient
draw water
let ground : Path2D = new Path2D
draw ground
let I:number = 0
[i < 1000]
let gravel : Path2D = new Path2D
let _x: number = Math.random() * cvs.width;
let _y: number = Math.random() * 600 + 400;
i++
draw gravel
i = 0
[i < 150]
let grass : Path2D = new Path2D
let _x: number = Math.random() * cvs.width;
let _y: number = Math.random() * cvs.heigth + 350;
i++
draw grass


update
window.setTimeout(update, 1000 / fps);
rnd.clearRect(0, 0, cvs.width, cvs.height);
rnd.putImageData(img, 0, 0);
fishPlayer.update
let I:number = 0
[i < arrayAll.length]
arrayAll[i].update()
i++
clear()
kanibalism(arrayAll[i]) == "dead"
kanibalism(arrayAll[i]) == "gameover"
arrayAll.splice(0, arrayAll.length);
highscore = 1700
alert
inputPlayerName = prompt("Your score: " + highscore, "Your Name");
insert();
refresh();
highscore = 0;
document.getElementByID - ID HighscoreDIV
inputPlayerName = prompt("Your score: " + highscore, "Your Name");
insert();
refresh();

Highscore - Display on Start



Highscore - Input after Win/Death