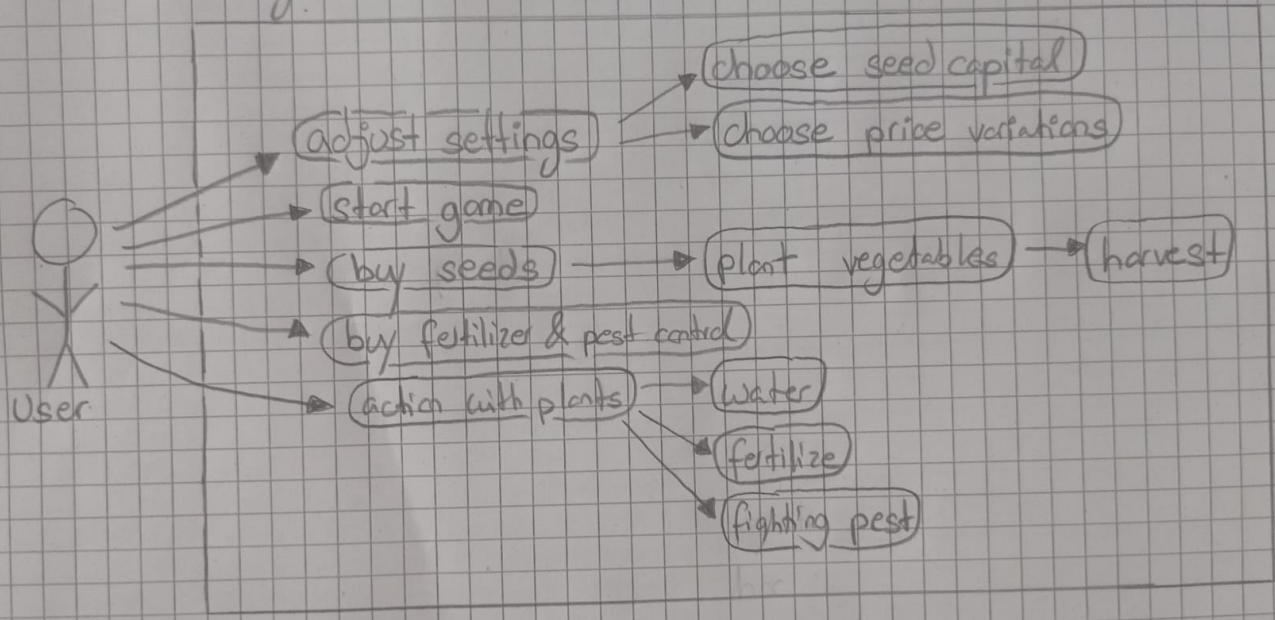
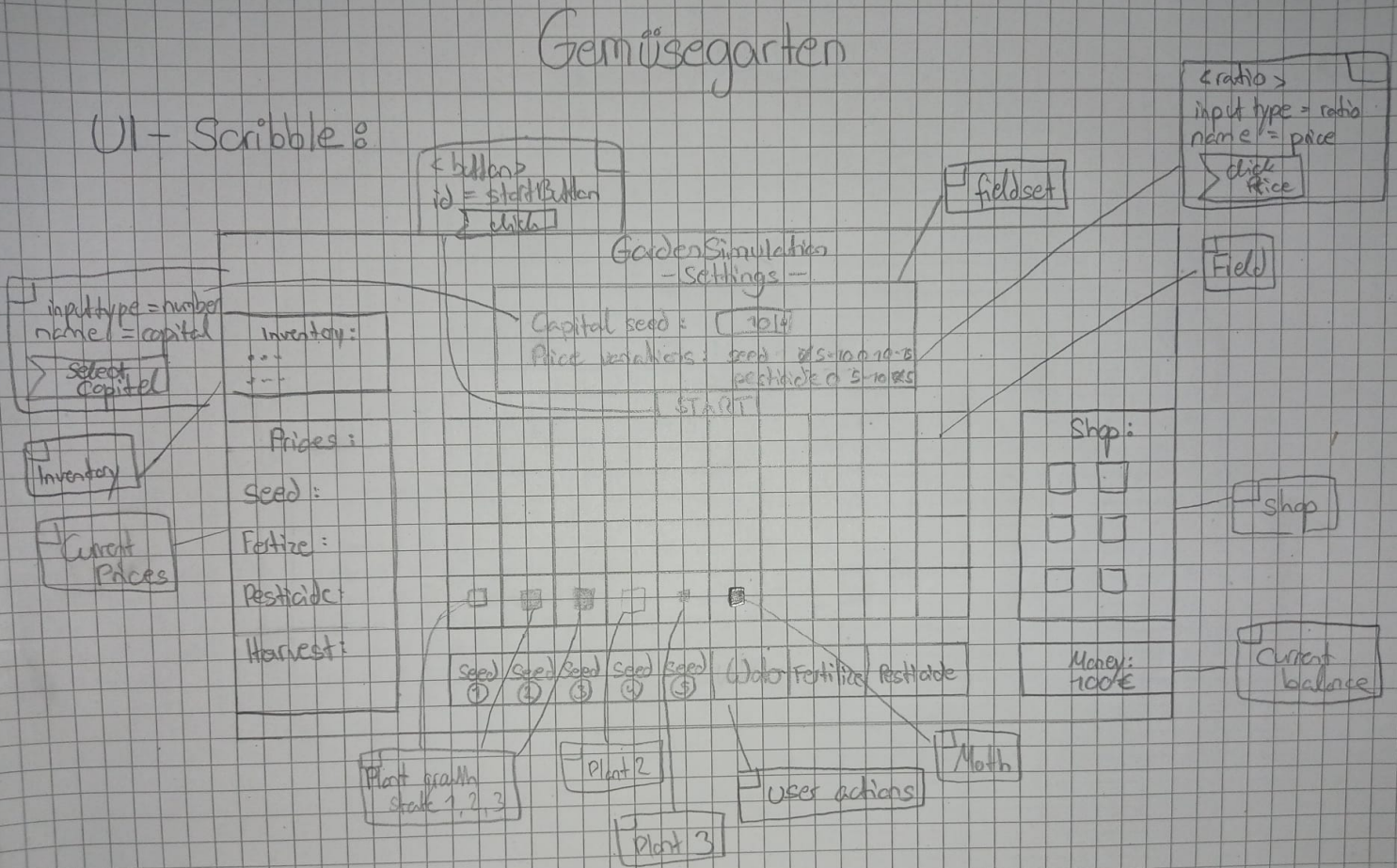
Gemüsegarten

Use-Case-Diagram:

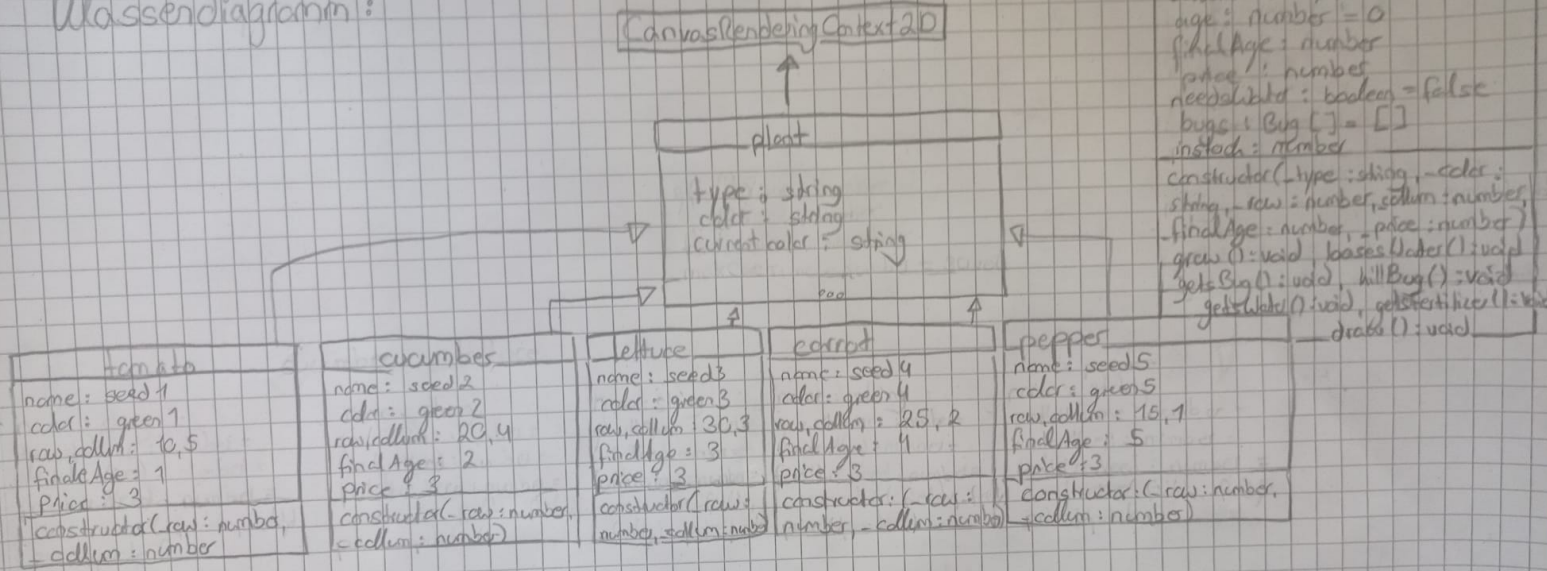


Gemüsegarten

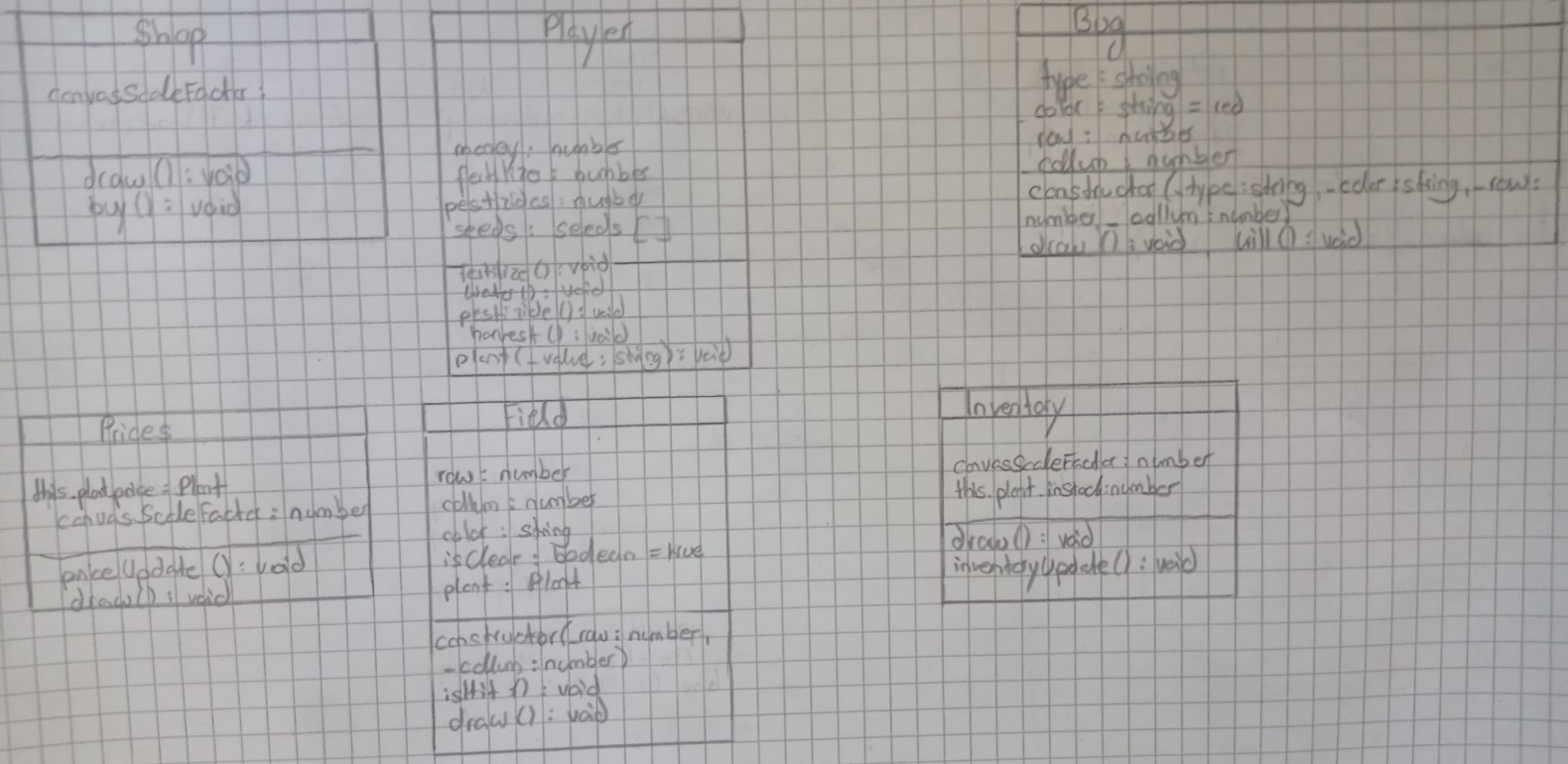


Gemüsegarten

Klassendiagramm:



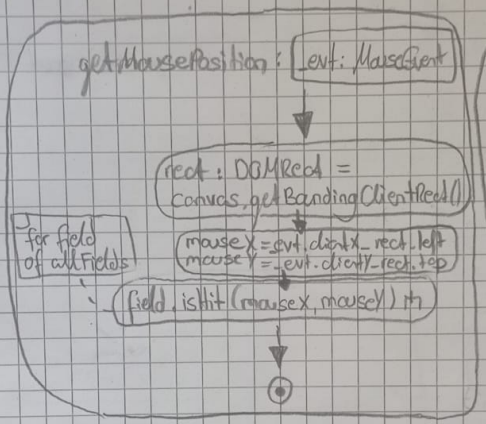
Klassendiagramm
– Gemüsegartensimulation –

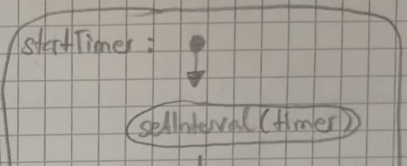


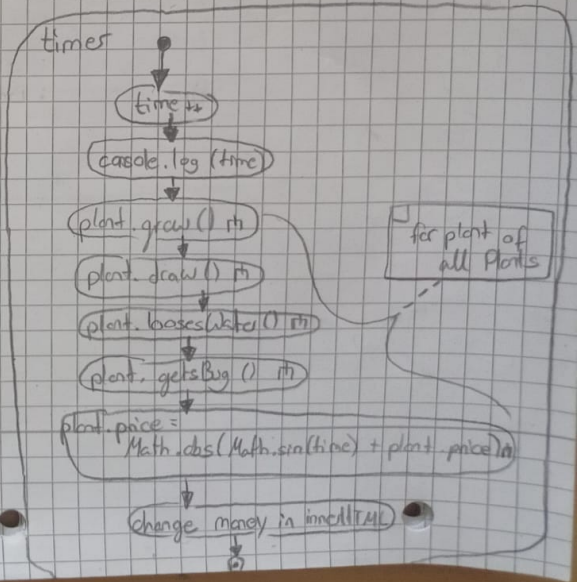
Aktivitätsdiagramm
-Gemüsegartensimulation-

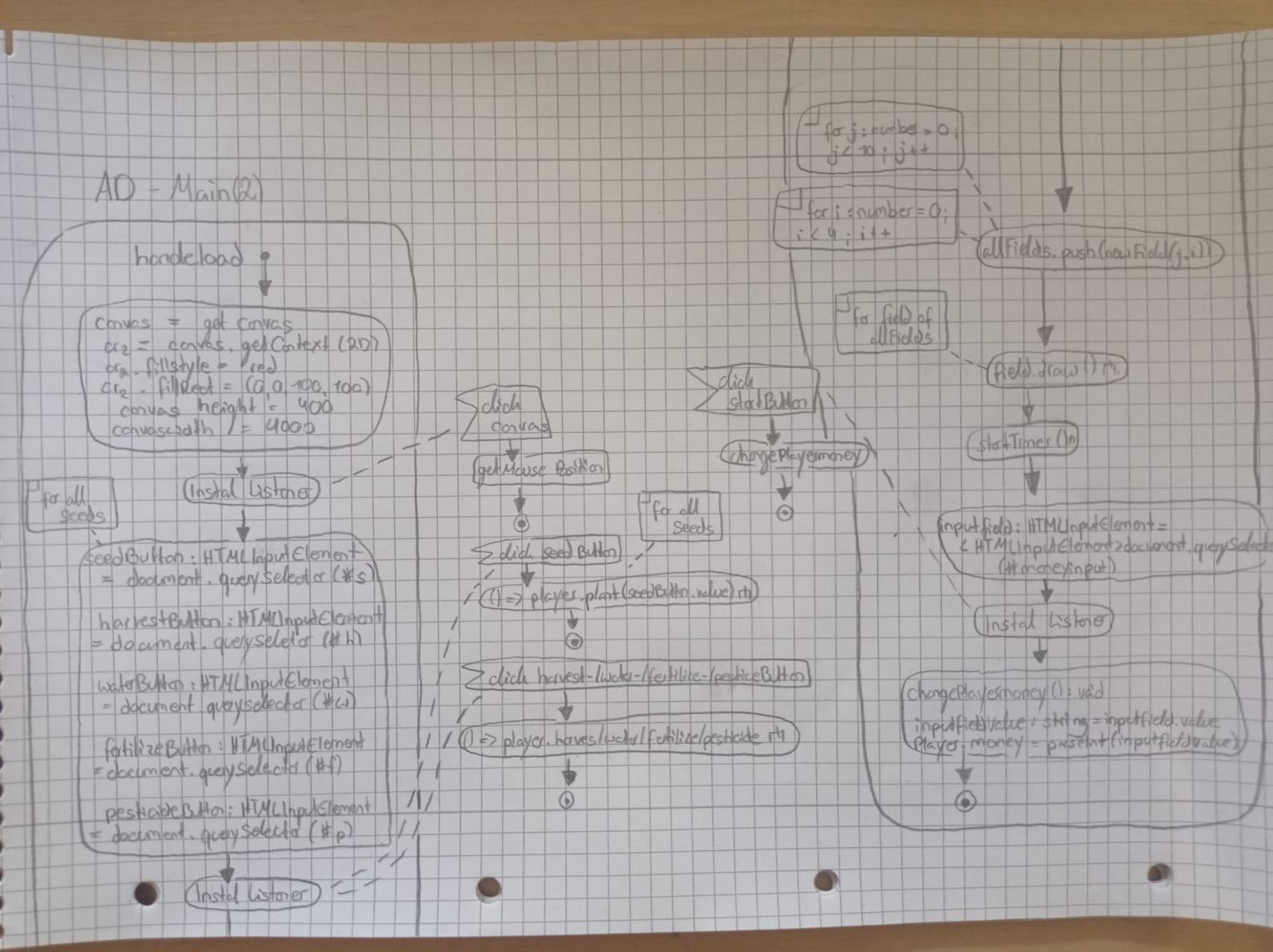
Main
(gardensimulation.ts)

```
export :
  cr2 : CanvasRenderingContext2D
  canvas : HTMLCanvasElement
  mouseX : number
  mouseY : number
  player : Player = new Player()
  prices : Prices = new Prices
  shop : Shop = new Shop
  allPlants : Plant[] = []
  allFields : Field[] = []
  time : number = 0
```

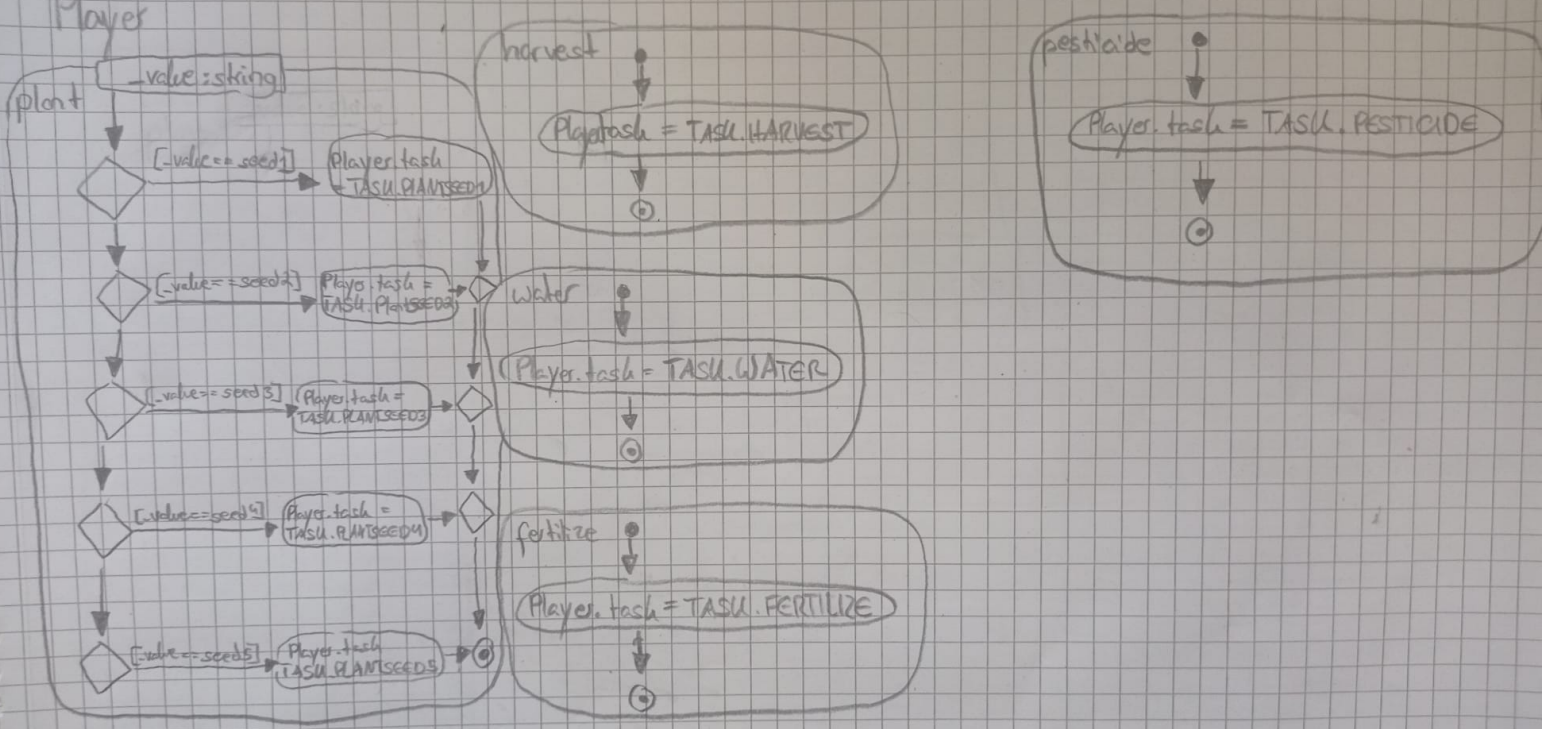




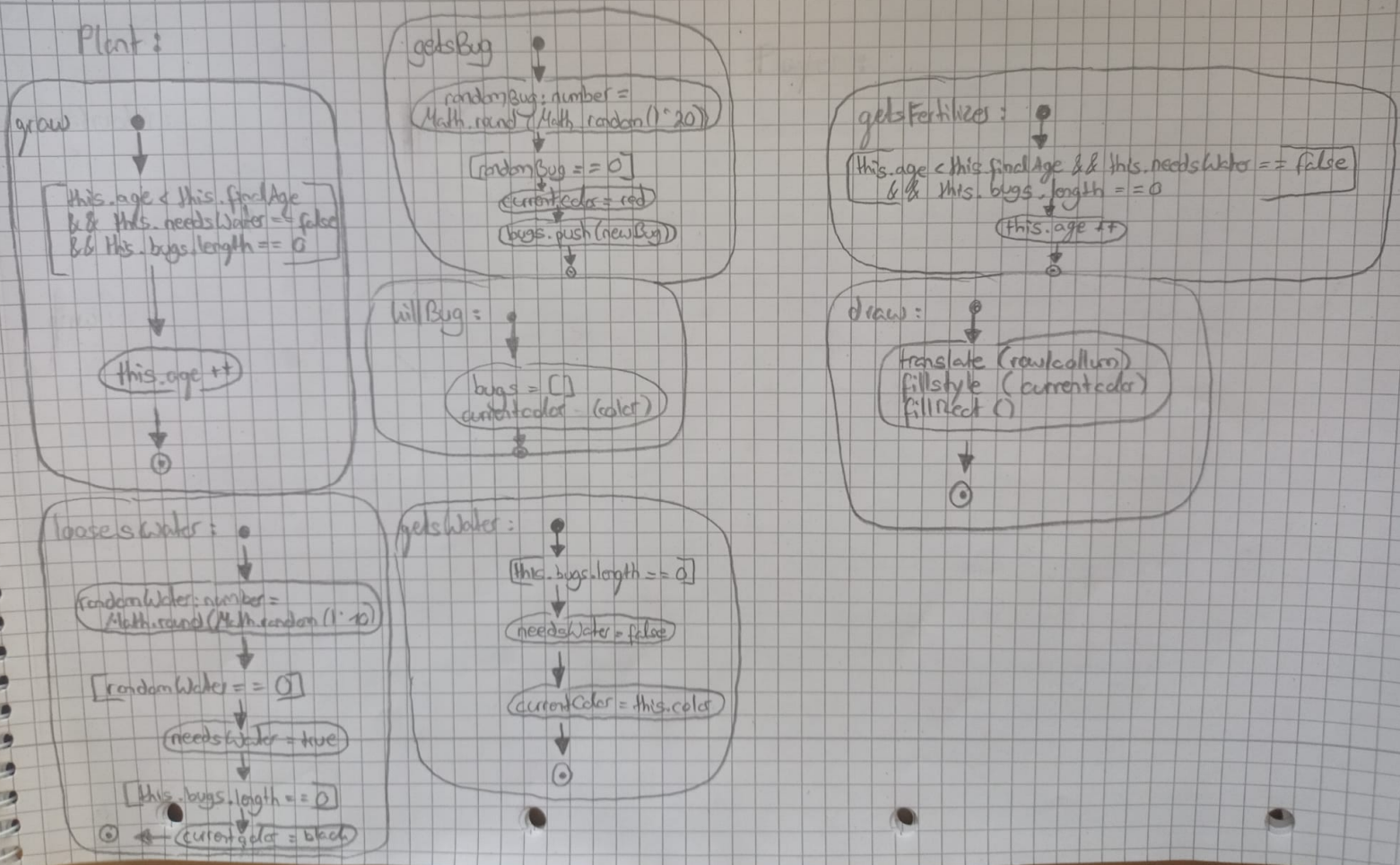



AD - Main(2)
handleload
canvas = get canvas
cr2 = canvas.getContext('2d')
cr2.fillstyle = 'red'
cr2.fillRect = (0,0,100,100)
canvas.height = 400
canvas.width = 400
Instal Listener
for all seeds
seedButton: HTMLInputElement = document.querySelector('#s')
harvestButton: HTMLInputElement = document.querySelector('#h')
waterButton: HTMLInputElement = document.querySelector('#w')
fertilizeButton: HTMLInputElement = document.querySelector('#f')
pesticideButton: HTMLInputElement = document.querySelector('#p')
Instal Listener
click canvas
getMousePosition
for all seeds
click seedButton
() => player.plant(seedButton.value) rh
click harvest-/water-/fertilize-/pesticideButton
() => player.harvest/water/fertilize/pesticide rh
click startButton
changePlayermoney
for j: number = 0; j < 10; j++
for i: number = 0; i < 4; i++
allFields.push(new Field(j,i))
for field of allFields
field.draw() rh
startTimer() rh
inputfield: HTMLInputElement = <HTMLInputElement>document.querySelector('#money-input')
Instal listener
changePlayermoney(): void
inputfield.value: string = inputfield.value
player.money = parseInt(inputfield.value)

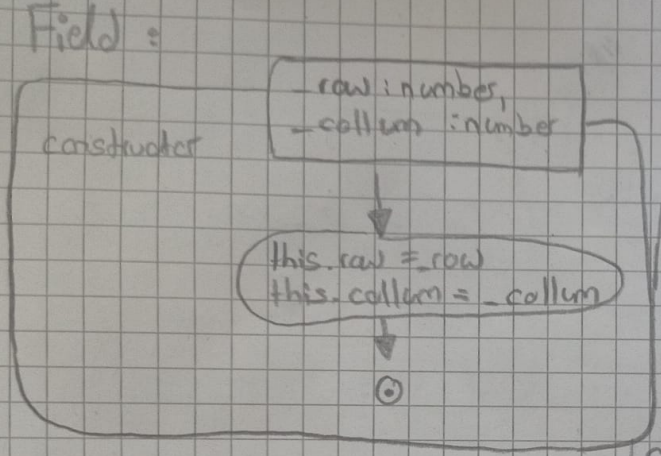
Aktivitätsdiagramm
- Gemüsegartensimulation -

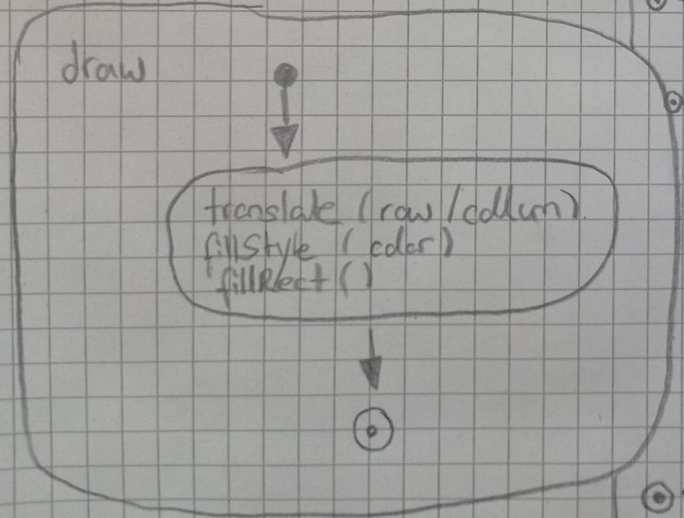


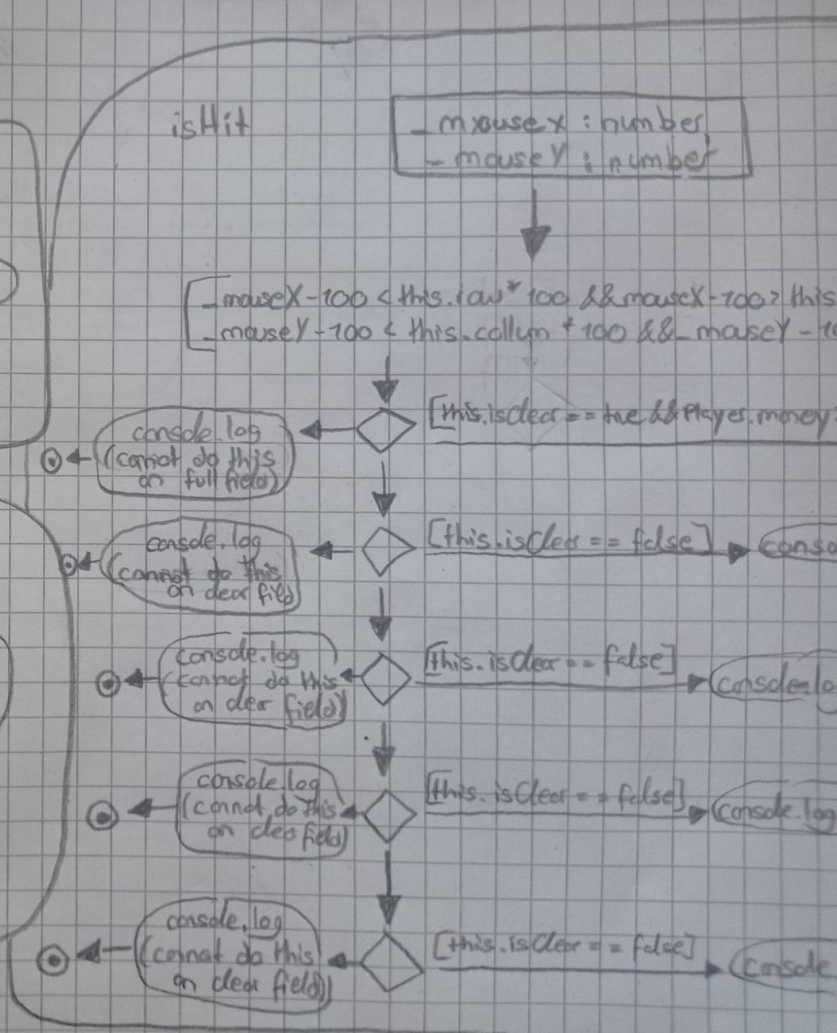
Aktivitätsdiagramm
– Gemüsegartensimulation –

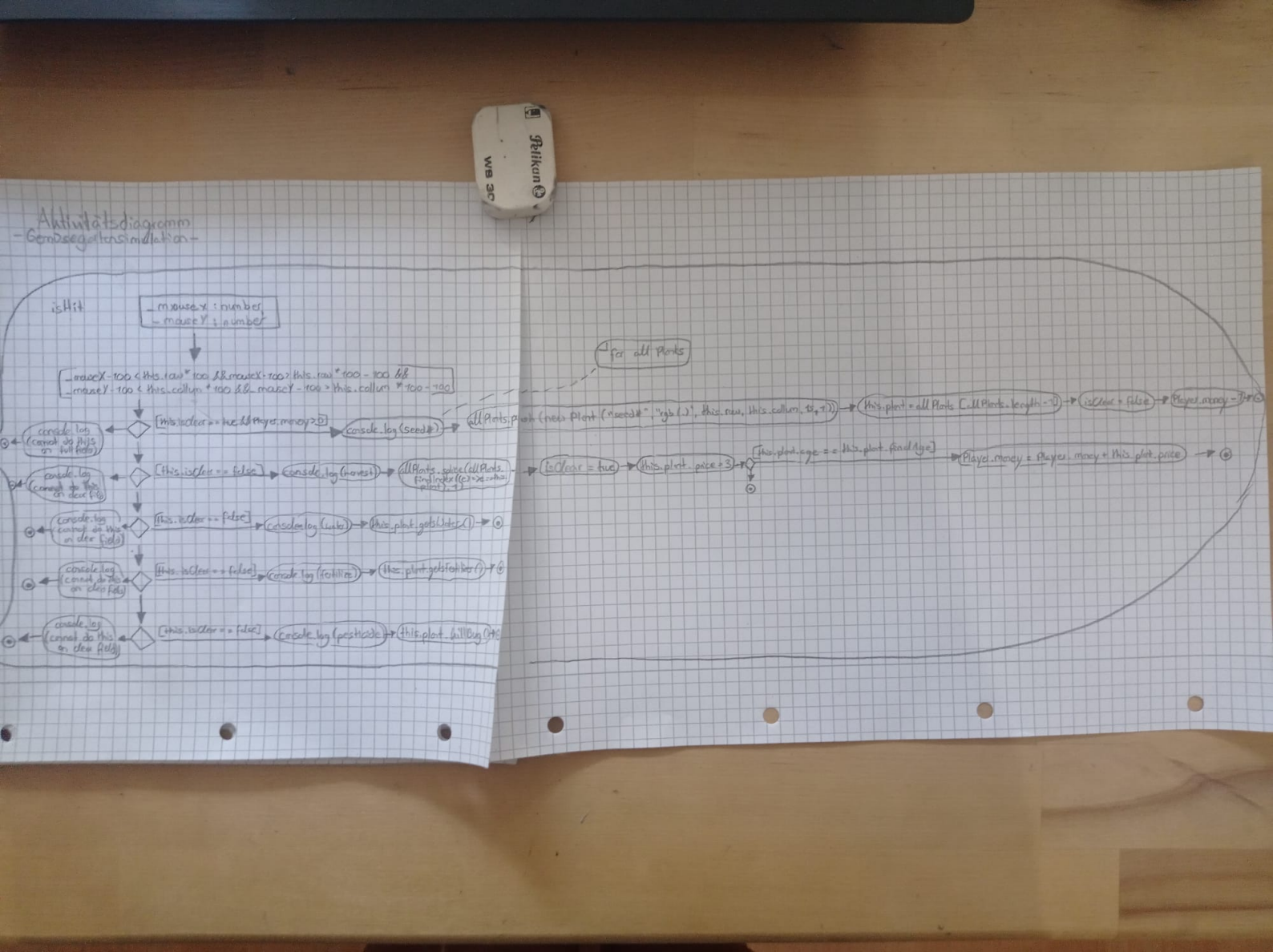


Aktivitätsdiagramm
-Gemosegaltensimulation-








Aktivitätsdiagramm
-Gemüsegartensimulation-
isHit
_mouseX : number
_mouseY : number
_mouseX-100 < this.raw * 100 && mouseX-100 > this.raw * 100 - 100 &&
_mouseY-100 < this.collum * 100 && _mouseY-100 > this.collum * 100 - 100
this.isClear == true && Player.money >= 0
console.log(seeds#)
allPlants.push(new Plant("seed#", "rgb(,)", this.raw, this.collum, 0, 1))
this.plant = allPlants[allPlants.length-1]
isClear = false
Player.money -= 1
console.log(cannot do this on full field)
this.isClear == false
console.log(harvest)
allPlants.splice(allPlants.findIndex((e)=>e==this.plant),1)
console.log(cannot do this on clear field)
this.isClear == false
console.log(water)
this.plant.getsWater()
console.log(cannot do this on clear field)
this.isClear == false
console.log(fertilize)
this.plant.getsFertilizer()
console.log(cannot do this on clear field)
this.isClear == false
console.log(pesticide)
this.plant.killBug()
for all Plants
isClear = true
this.plant.price += 3
this.plant.age == this.plant.finalAge
Player.money = Player.money + this.plant.price