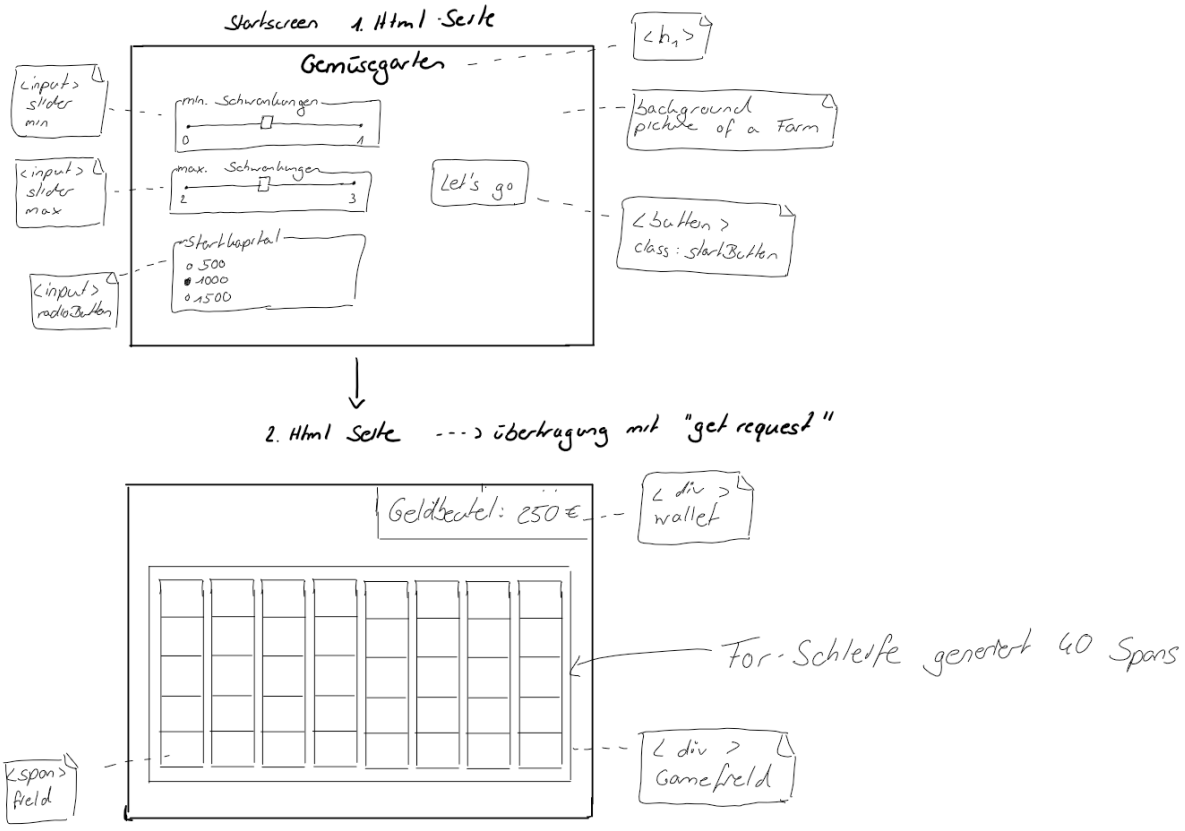
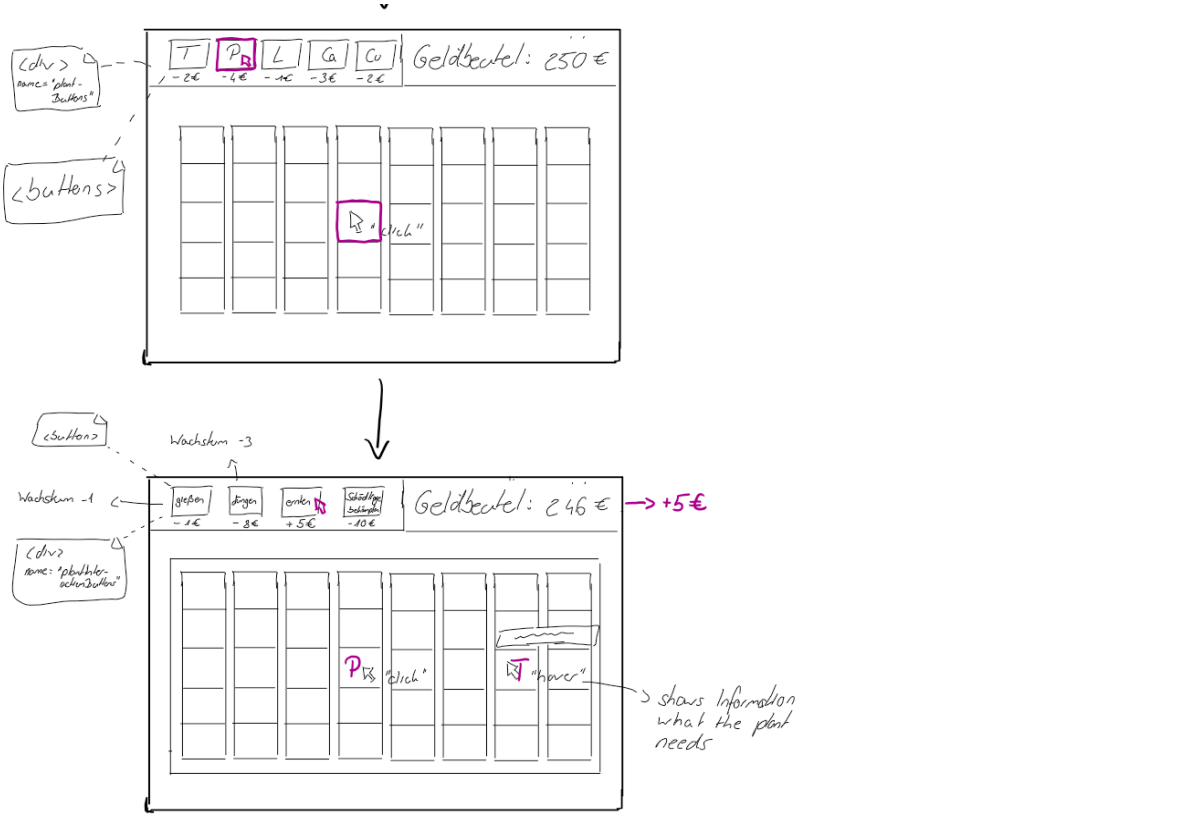
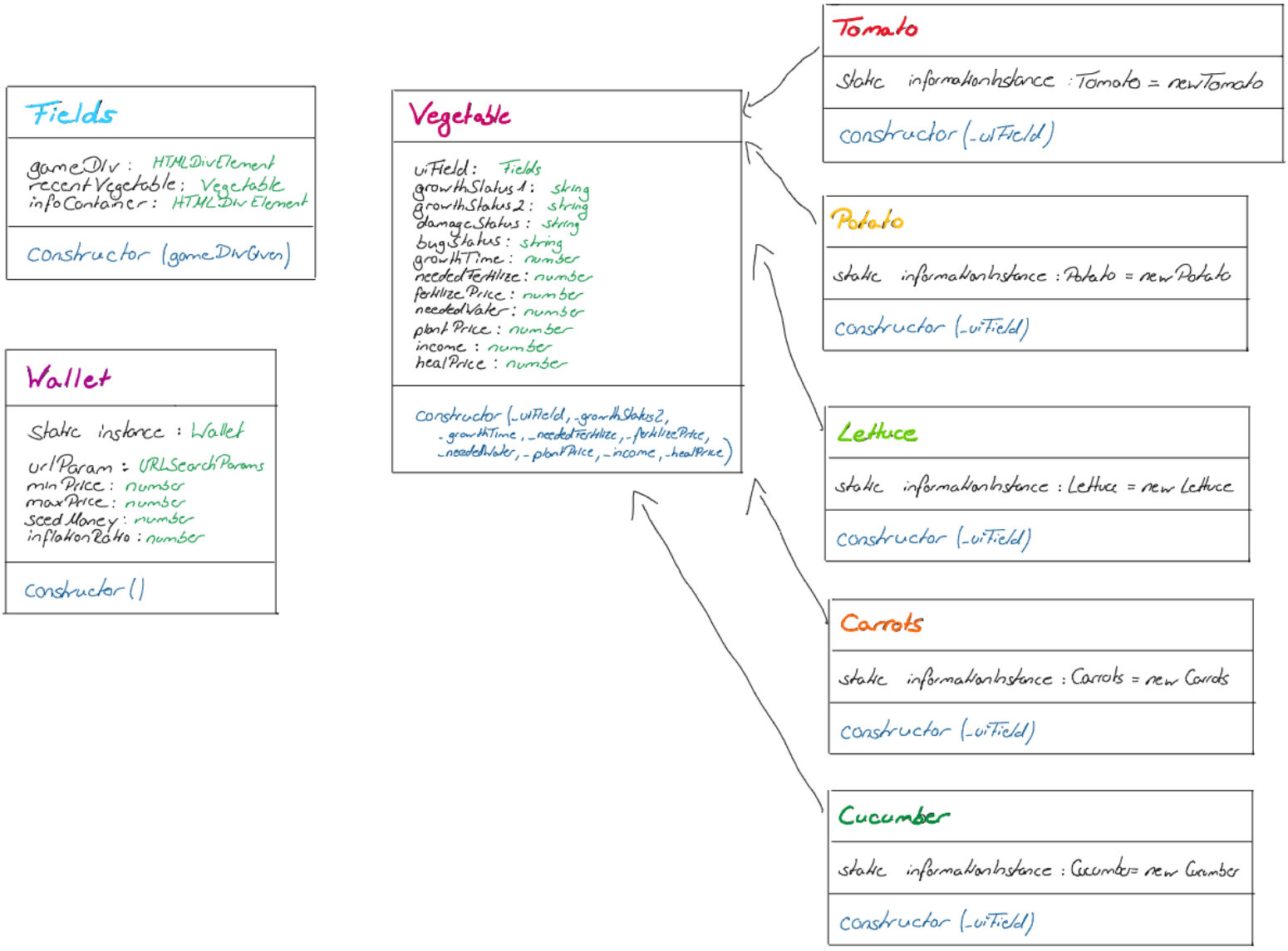
UI-Scribble

Mittwoch, 29. Juni 2022 16:05

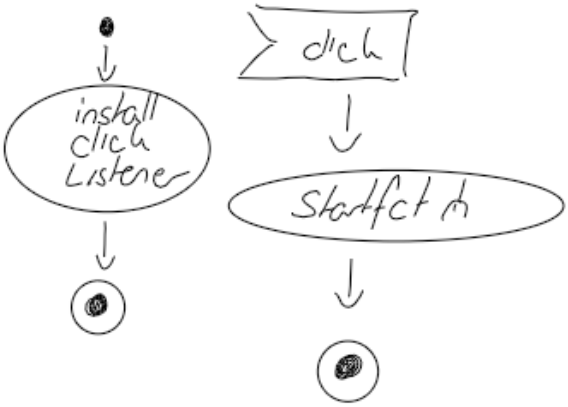


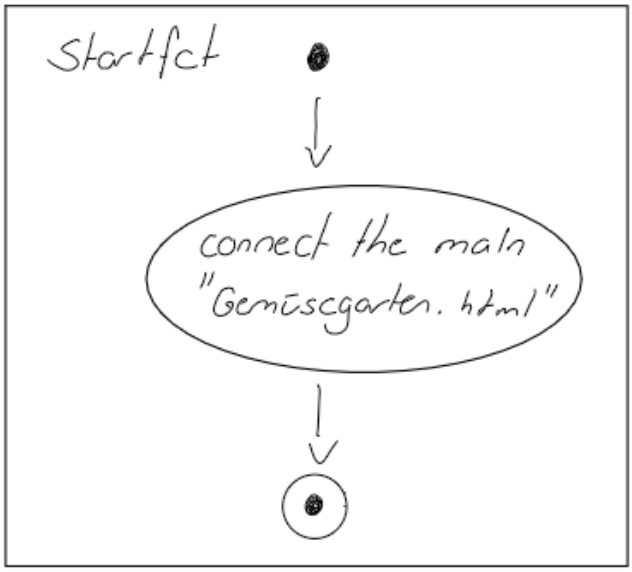



Fields
gameDiv : HTMLDivElement
recentVegetable : Vegetable
infoContainer : HTMLDivElement
Constructor (gameDivGiven)
Wallet
Static instance : Wallet
urlParam : URLSearchParams
minPrice : number
maxPrice : number
seedMoney : number
inflationRatio : number
constructor()
Vegetable
uiField : Fields
growthStatus1 : string
growthStatus2 : string
damageStatus : string
bugStatus : string
growthTime : number
neededFertilizer : number
fertilizePrice : number
neededWater : number
plantPrice : number
income : number
healPrice : number
constructor (_uiField, _growthStatus2, _growthTime, _neededFertilizer, _fertilizePrice, _neededWater, _plantPrice, _income, _healPrice)
Tomato
Static informationInstance : Tomato = new Tomato
constructor (_uiField)
Potato
static informationInstance : Potato = new Potato
constructor (_uiField)
Lettuce
static informationInstance : Lettuce = new Lettuce
constructor (_uiField)
Carrots
static informationInstance : Carrots = new Carrots
constructor (_uiField)
Cucumber
static informationInstance : Cucumber = new Cucumber
constructor (_uiField)

Start.ts from Startscreen.html

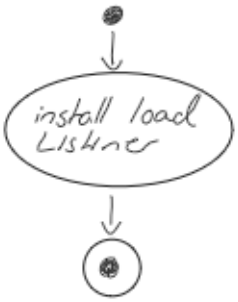
```
let startButton: Button = getElementById
```

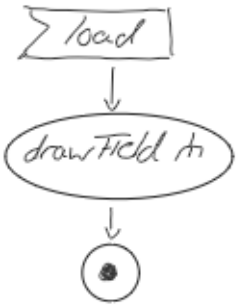


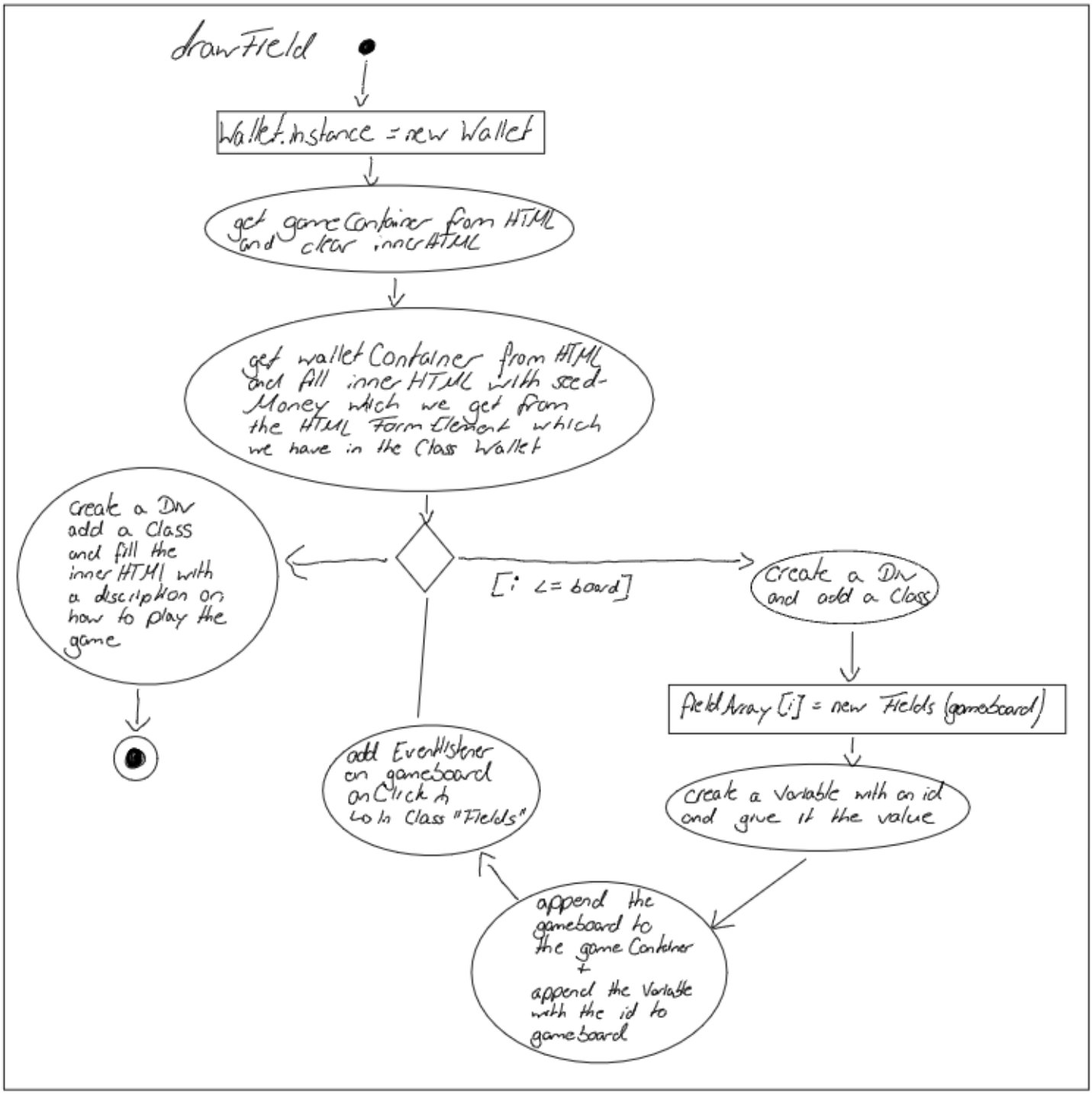


Activity Diagram - Main

```
let board: number
let gameboard: HTMLDivElement
let gameContainer: HTMLDivElement
let fieldArray: Fields[] = []
let walletContainer: HTMLDivElement
let textContainer: HTMLDivElement
```

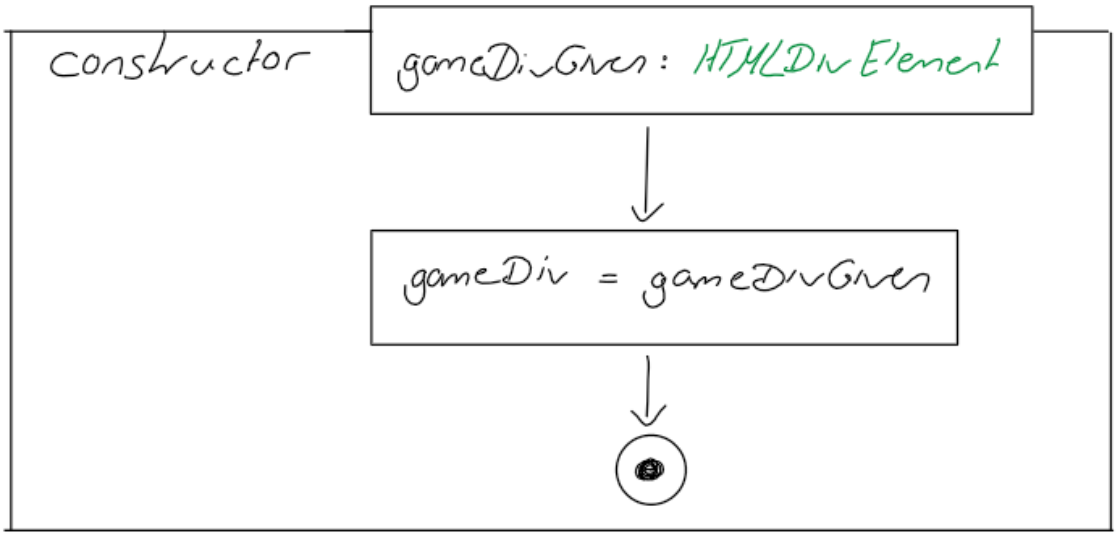


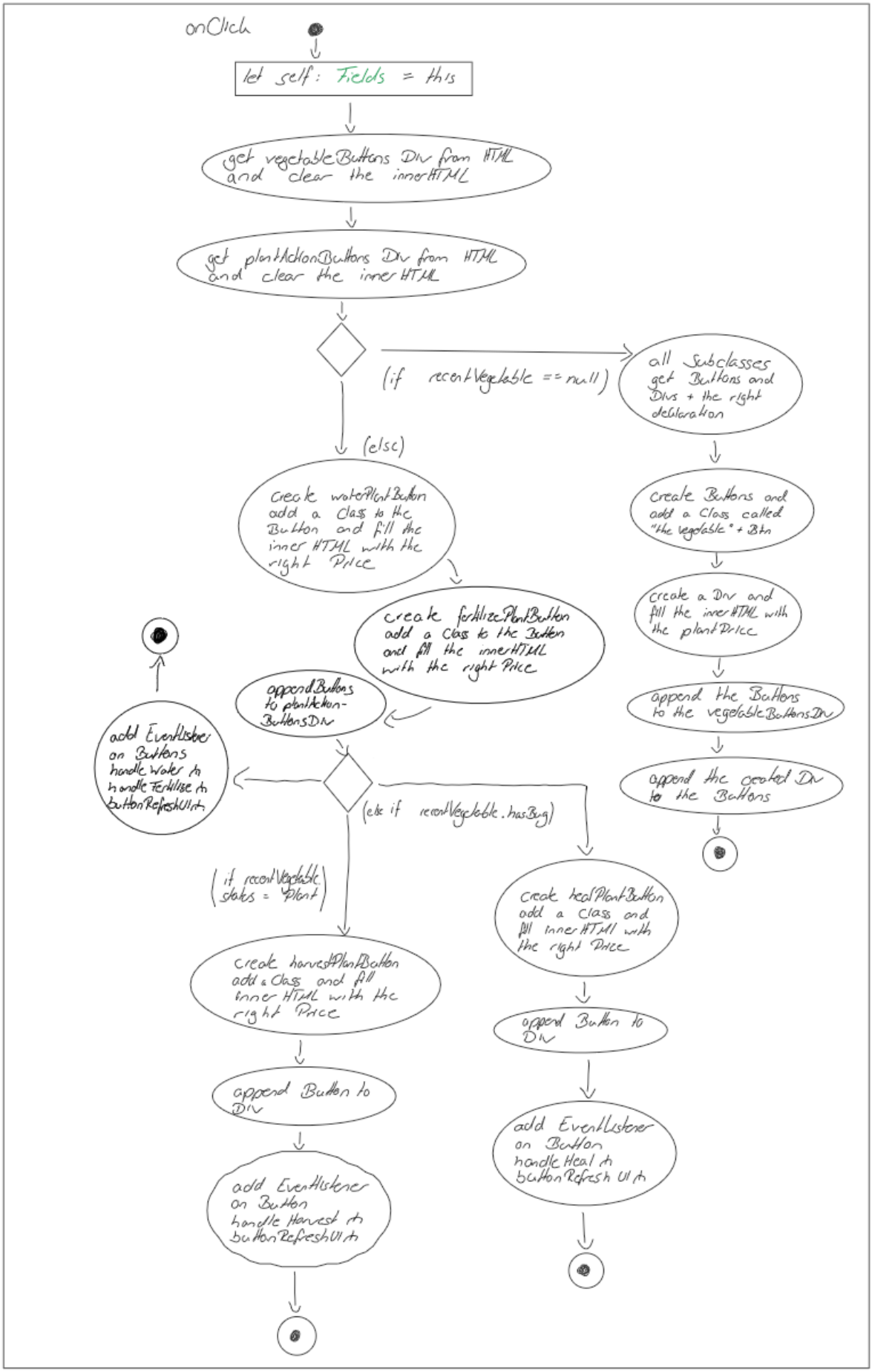


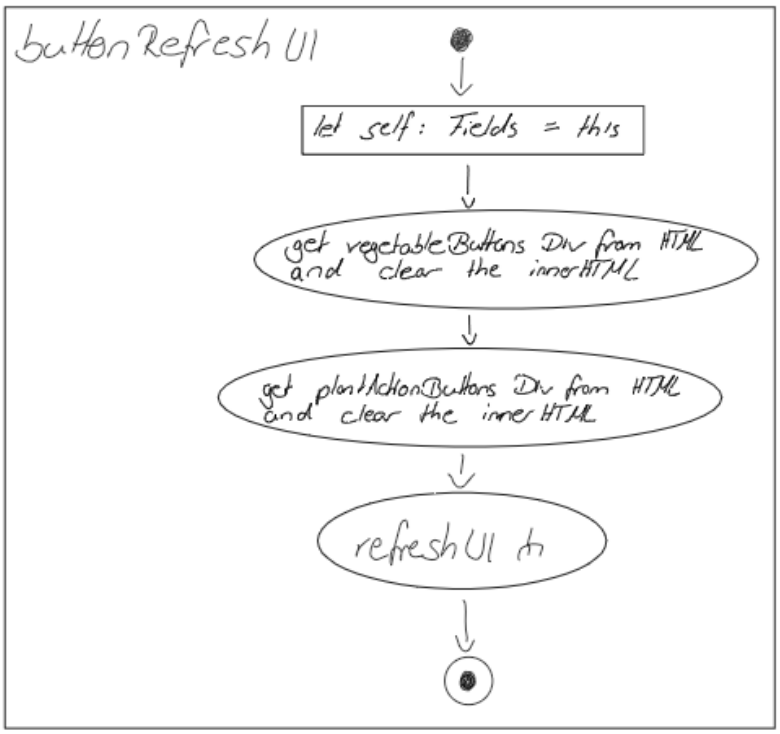


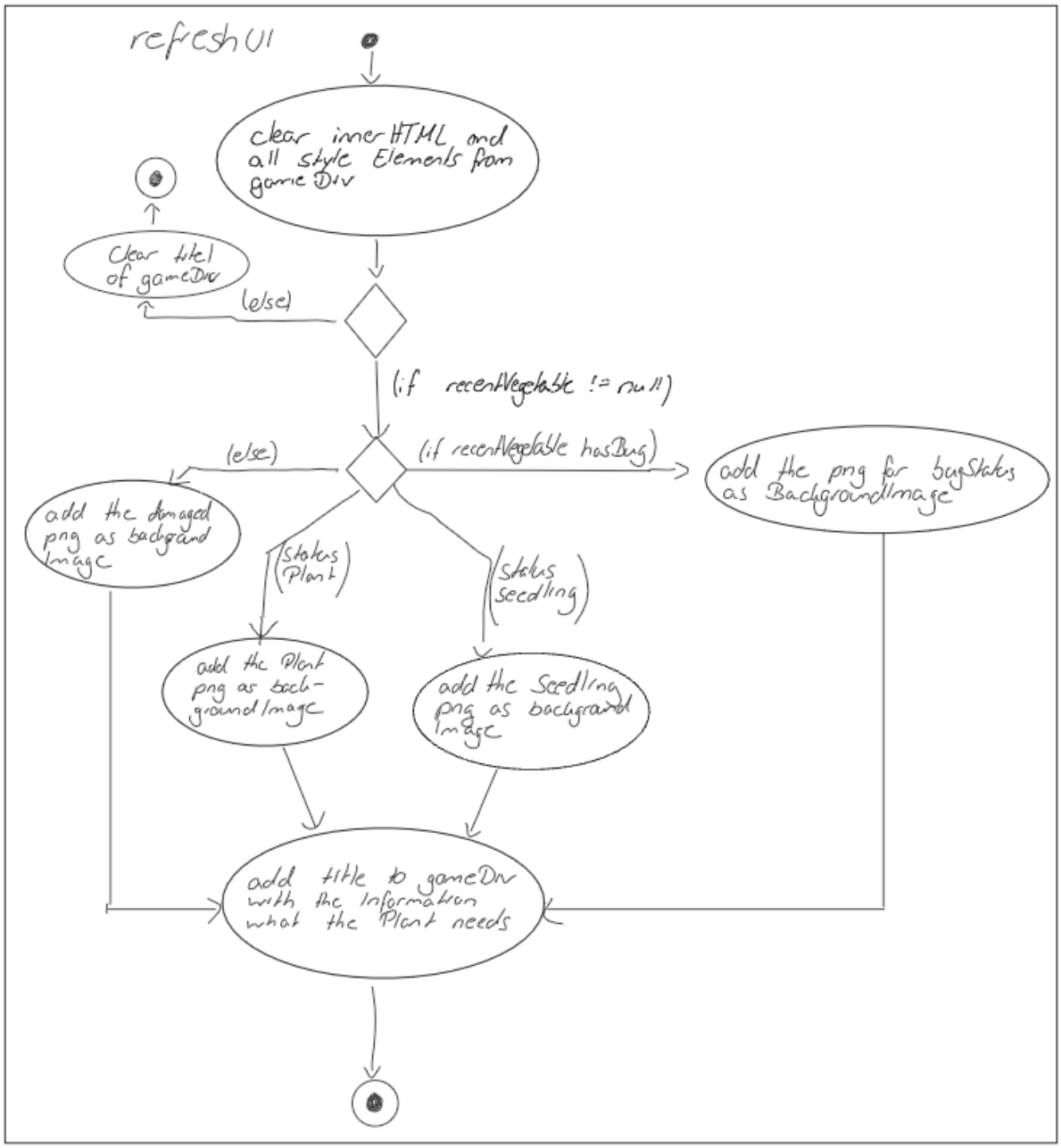
Class - Fields

```
gameDiv : HTMLDivElement
recentVegetable : Vegetable | null
infoContainer : HTMLDivElement
```

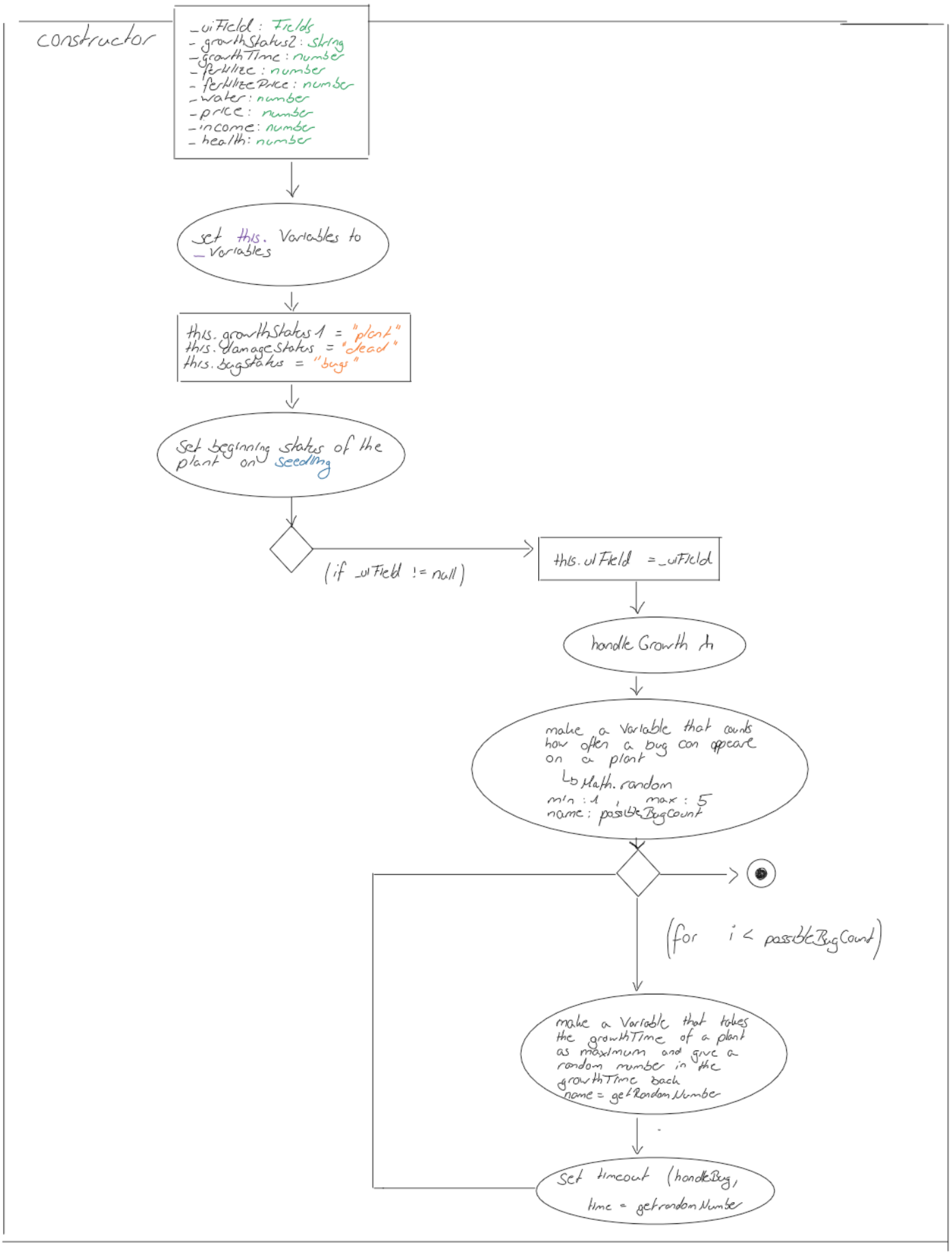


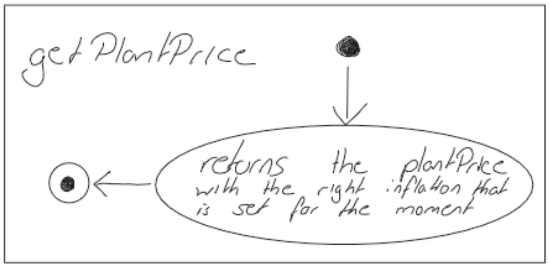

onClick
let self: Fields = this
get vegetableButtons Div from HTML and clear the innerHTML
get plantActionButtons Div from HTML and clear the innerHTML
(if recentVegetable == null)
all Subclasses get Buttons and Divs + the right deklaration
create Buttons and add a Class called "the vegetable" + Btn
create a Div and fill the innerHTML with the plantPrice
append the Buttons to the vegetableButtonsDiv
append the created Div to the Buttons
(else)
create waterPlantButton add a Class to the Button and fill the innerHTML with the right Price
create fertilizePlantButton add a Class to the Button and fill the innerHTML with the right Price
appendButtons to plantAction-ButtonsDiv
add EventListener on Buttons handle Water th handle Fertilise th buttonRefreshUI th
(else if recentVegetable.hasBug)
create healPlantButton add a Class and fill innerHTML with the right Price
append Button to Div
add EventListener on Button handle Heal th buttonRefresh UI th
(if recentVegetable.state = "plant")
create harvestPlantButton add a Class and fill innerHTML with the right Price
append Button to Div
add EventListener on Button handle Harvest th buttonRefreshUI th

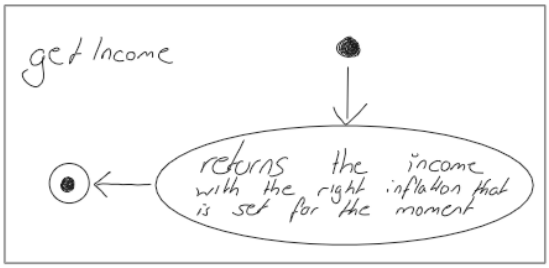

buttonRefreshUI
let self: Fields = this
get vegetableButtons Div from HTML and clear the innerHTML
get plantActionButtons Div from HTML and clear the innerHTML
refreshUI fn

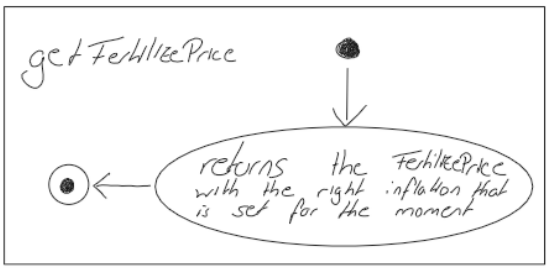

refreshUI
clear innerHTML and all style Elements from gameDiv
Clear titel of gameDiv
(else)
(if recentVegetable != null)
(else)
(if recentVegetable hasBug)
add the png for bugStates as BackgroundImage
add the damaged png as backgrand Image
(Status Plant)
(Status Seedling)
add the Plant png as back-ground Image
add the Seedling png as backgrand Image
add title to gameDiv with the information what the Plant needs

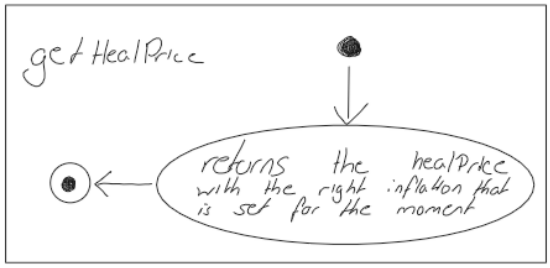
Activity Diagram – Super Class Vegetable

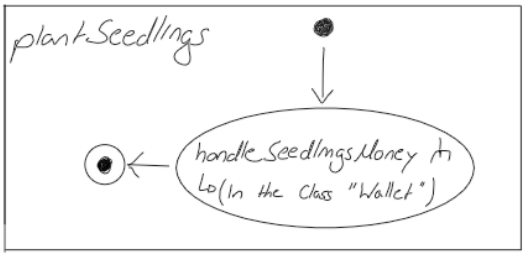


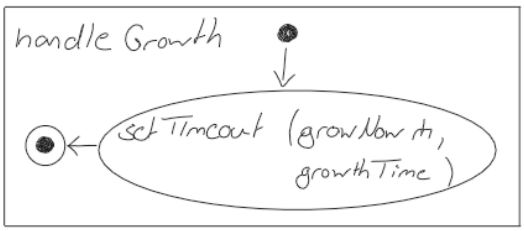

getPlantPrice
returns the plantPrice with the right inflation that is set for the moment

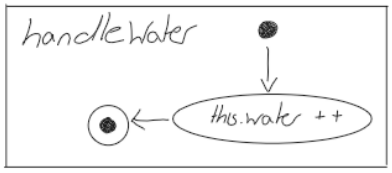

getIncome
returns the income with the right inflation that is set for the moment

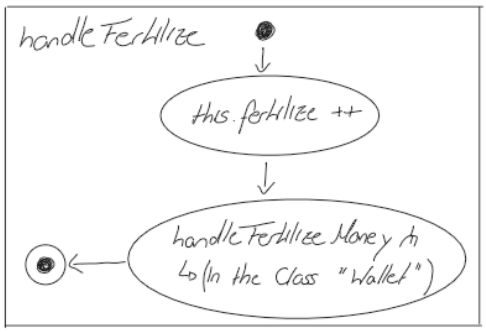

getFertilizePrice
returns the FertilizePrice with the right inflation that is set for the moment

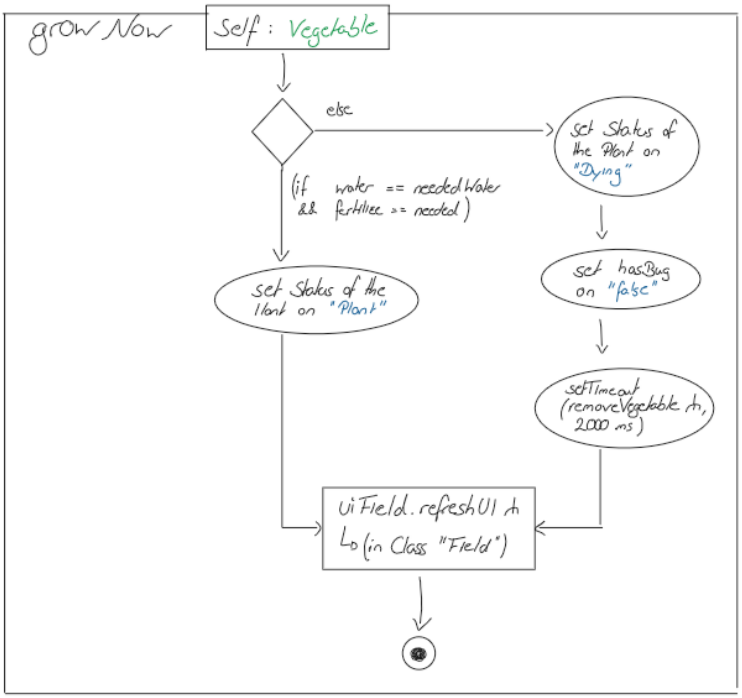

getHealPrice
returns the healPrice with the right inflation that is set for the moment

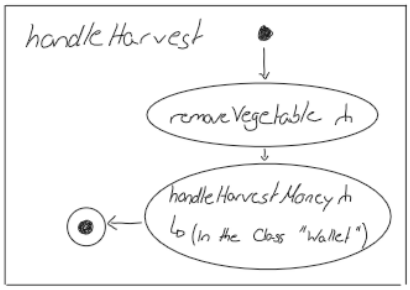

plantSeedlings
handleSeedlingsMoney th
Lo(In the Class "Wallet")

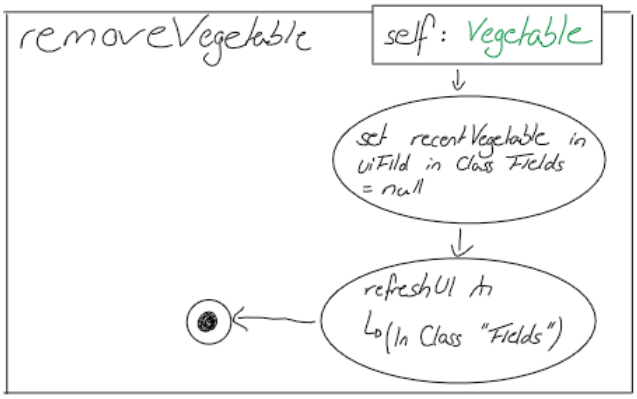

handleGrowth
setTimeout (growNow th, growthTime)

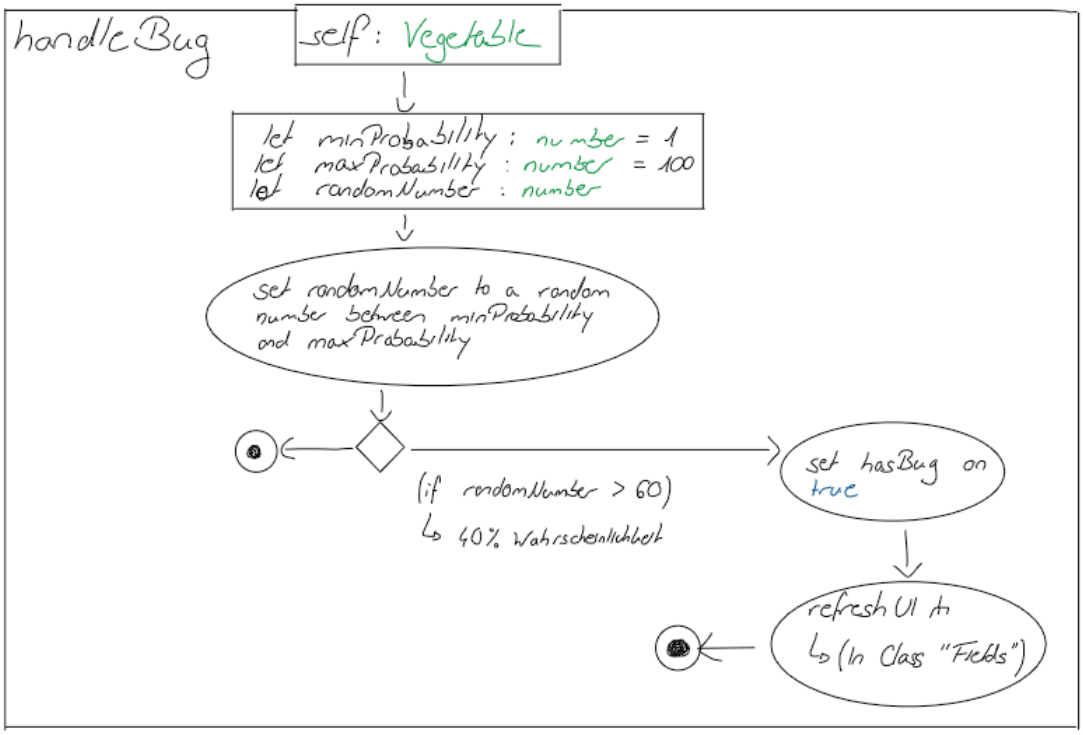

handleWater
this.water ++

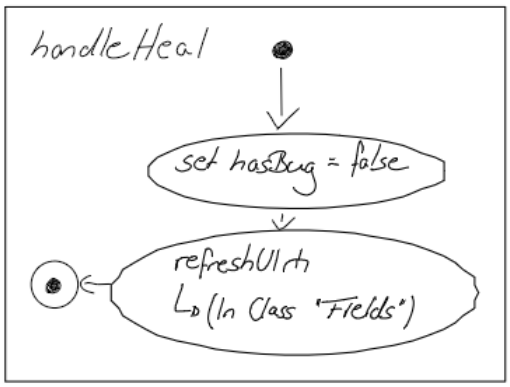

handleFertilize
this.fertilize ++
handleFertilizeMoney th
Lo(In the Class "Wallet")


growNow
Self : Vegetable
else
(if water == neededWater && fertilize == needed)
set Status of the Plant on "Dying"
set hasBug on "false"
setTimeout (removeVegetable th, 2000 ms)
set Status of the Plant on "Plant"
uiField.refreshUI th
Lo(in Class "Field")


handleHarvest
removeVegetable th
handleHarvestMoney th
Lo(In the Class "Wallet")

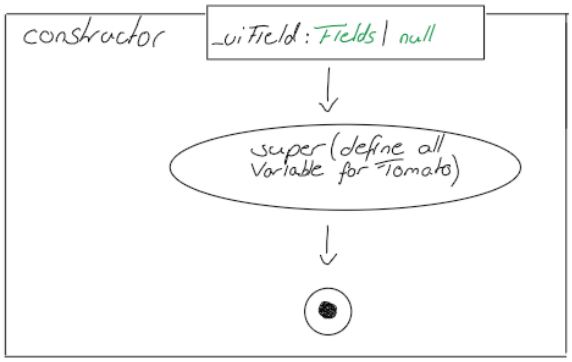

removeVegetable
self: Vegetable
set recentVegetable in uiFild in Class Fields = null
refreshUI in
↳(In Class "Fields")


handleBug
self: Vegetable
let minProbability : number = 1
let maxProbability : number = 100
let randomNumber : number
set randomNumber to a random number between minProbability and maxProbability
(if randomNumber > 60)
↳ 40% Wahrscheinlichkeit
set hasBug on true
refreshUI in
↳(In Class "Fields")


handleHeal
set hasBug = false
refreshUI in
↳(In Class "Fields")

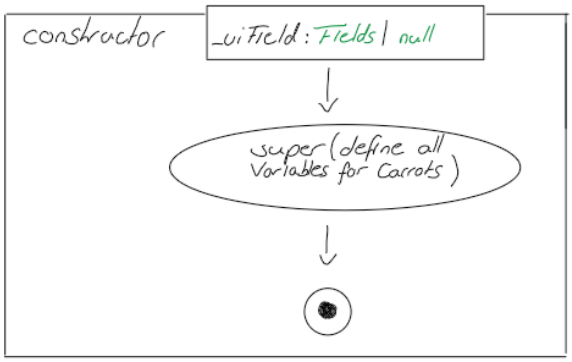
Subclass – Tomate

```
static informationInstance : Tomato = new Tomato(null)
```



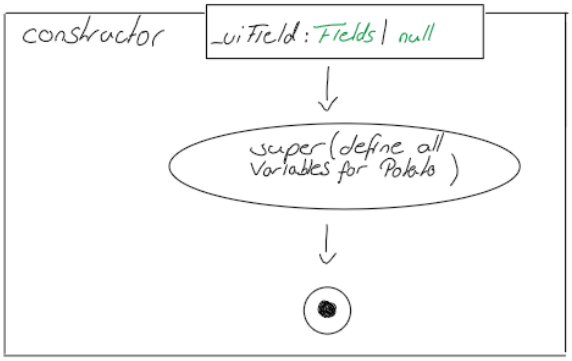
Subclass – Carrots

```
static informationInstance : Carrots = new Carrots(null)
```



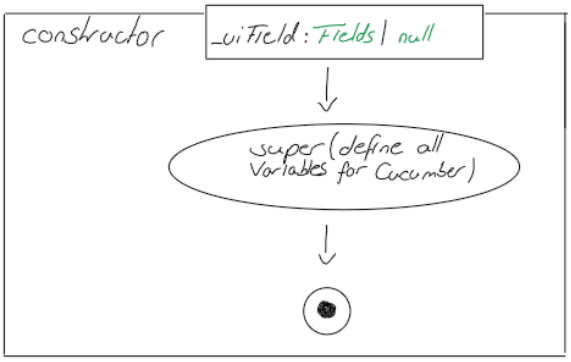
Subclass – Potato

```
static informationInstance : Potato = new Potato(null)
```



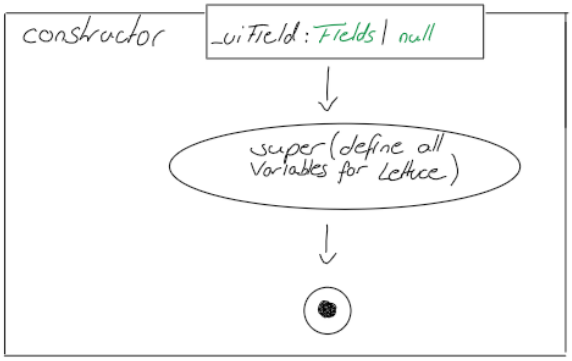
Subclass – Cucumber

```
static informationInstance : Cucumber = new Cucumber(null)
```



Subclass – Lettuce

```
static informationInstance : Lettuce = new Lettuce(null)
```

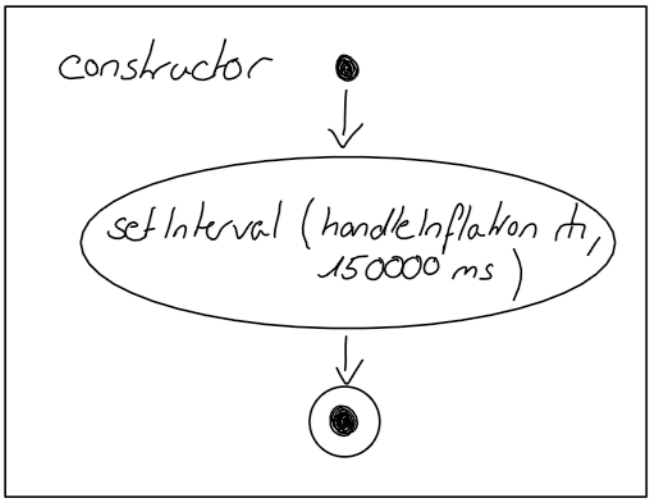


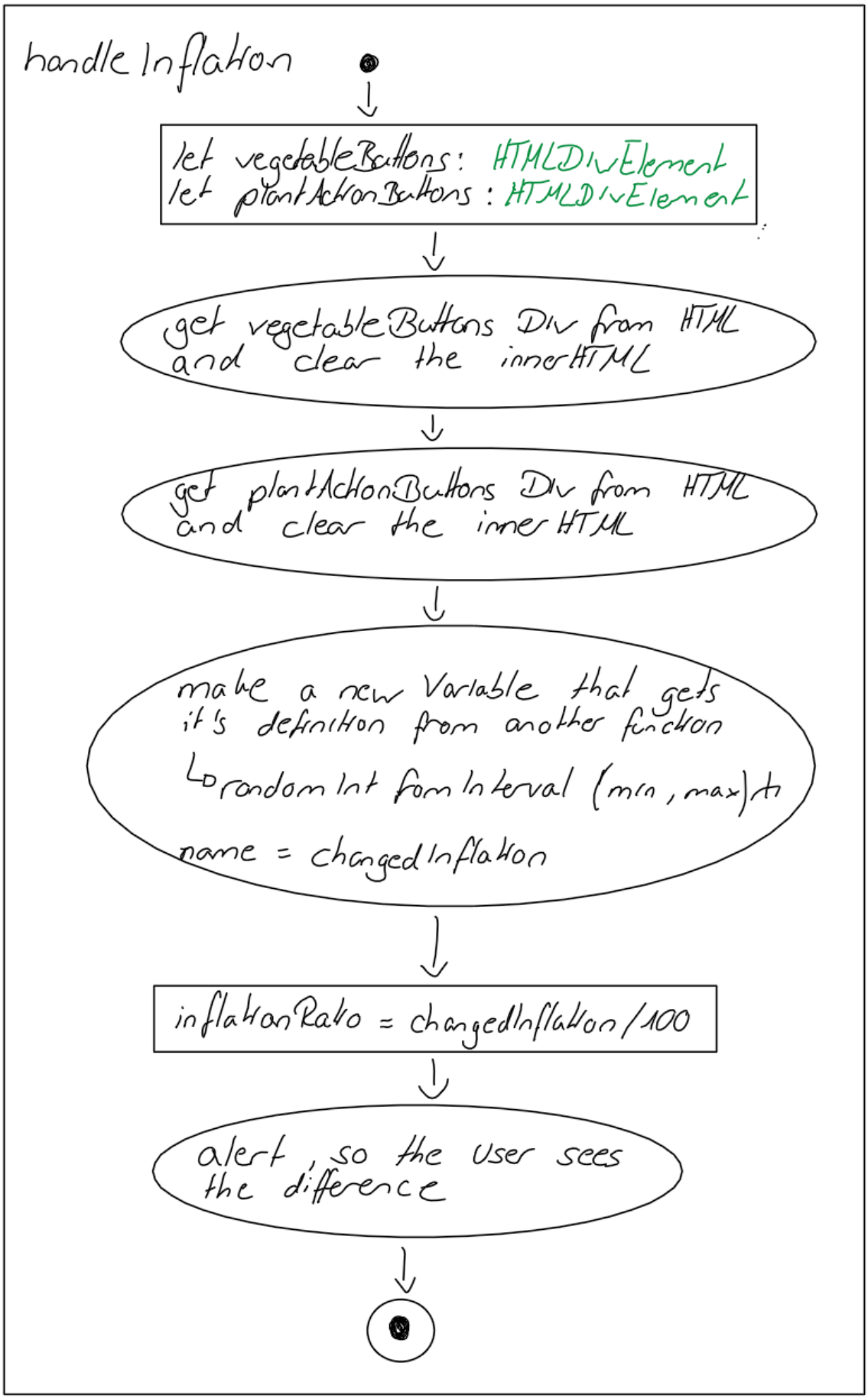
Class – Wallet

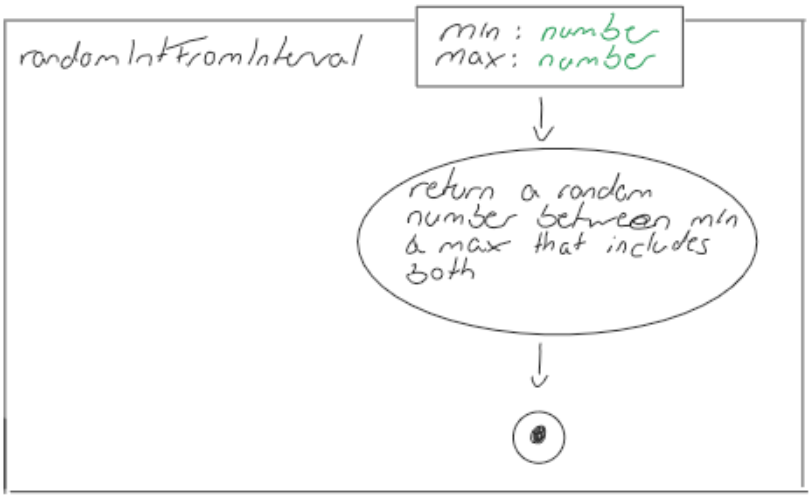
static instance : Wallet

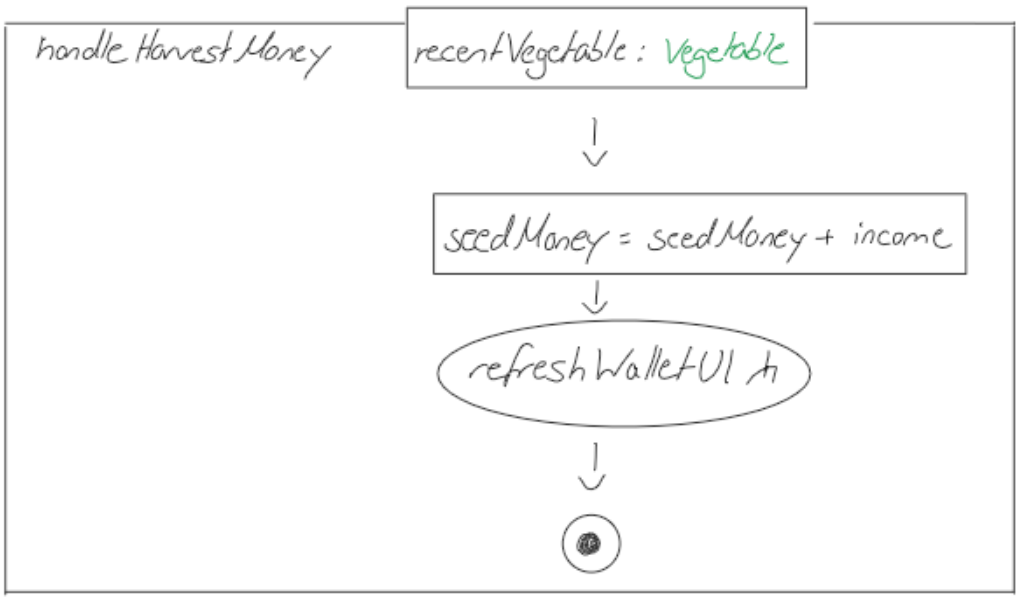
walletContainer : HTMLDIVElement = document.querySelector

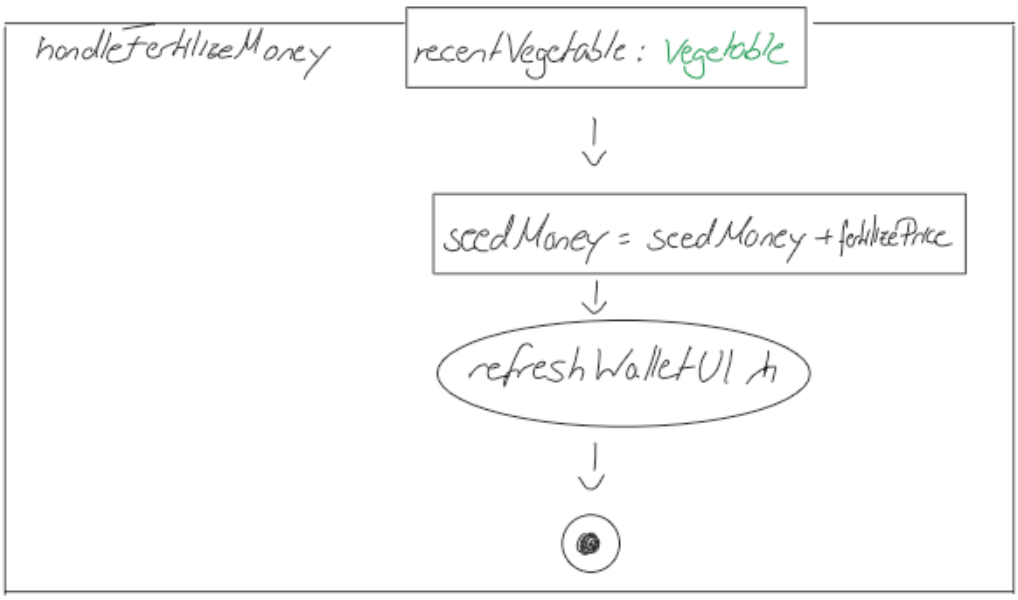
urlParam : URLSearchParams = new URLSearchParams(window.locale.search)
minPrice : number = get value from Form Element via urlParam
maxPrice : number = "
seedMoney : number = "
inflationRate : number = 1

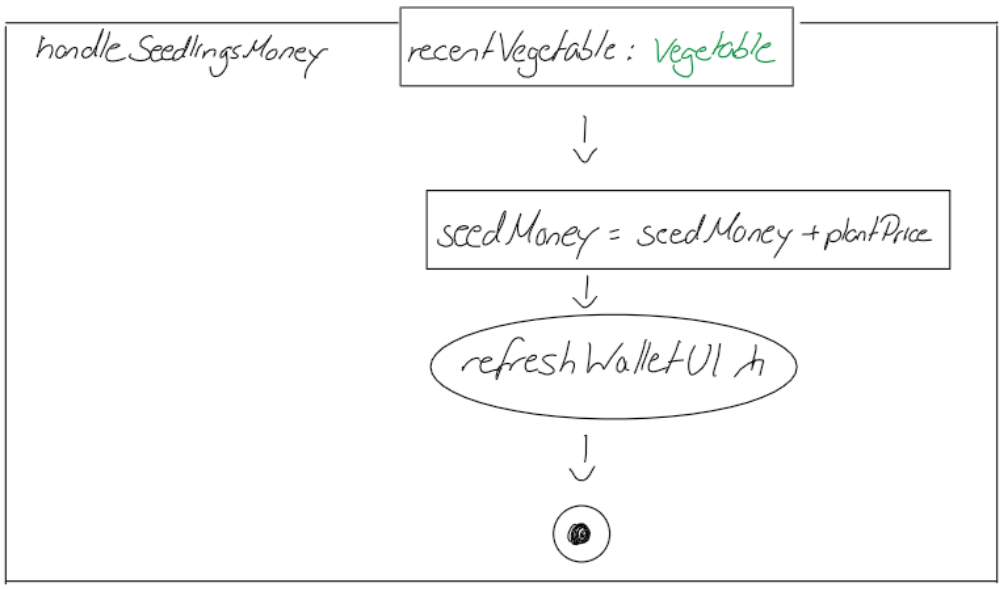


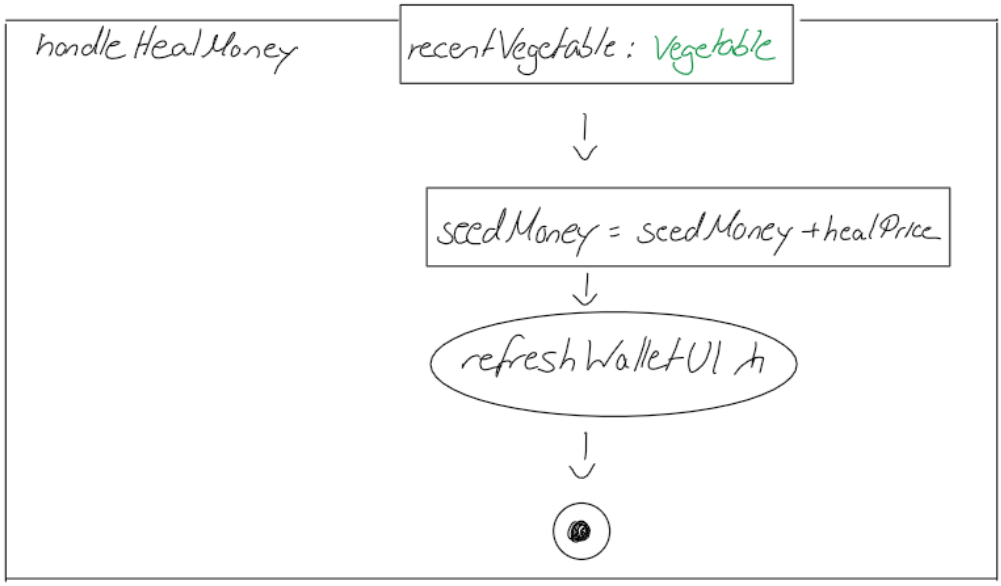

handle Inflation
let vegetableButtons: HTMLDivElement
let plantActionButtons: HTMLDivElement
get vegetableButtons Div from HTML and clear the innerHTML
get plantActionButtons Div from HTML and clear the innerHTML
make a new Variable that gets it's definition from another function
Lo random Int from Interval (min, max) th
name = changedInflation
inflationRatio = changedInflation / 100
alert, so the user sees the difference

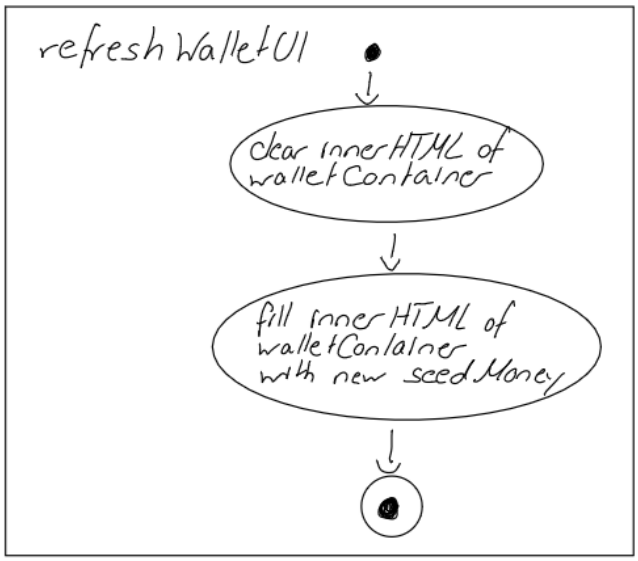

randomIntFromInterval
min : number
max : number
return a random number between min & max that includes both


handleHarvestMoney
recentVegetable : Vegetable
seedMoney = seedMoney + income
refreshWalletUI ()


handleFertilizeMoney
recentVegetable : Vegetable
seedMoney = seedMoney + fertilizePrice
refreshWalletUI ()


handleSeedlingsMoney
recentVegetable : Vegetable
seedMoney = seedMoney + plantPrice
refreshWalletUI ()


handleHealMoney
recentVegetable : Vegetable
seedMoney = seedMoney + healPrice
refreshWalletUI ()


refreshWalletUI
clear innerHTML of walletContainer
fill innerHTML of walletContainer with new seedMoney