1. Idea

This year's assignment is the design and creation of an interactive garden. Users should be able to plant, water and fertilize fields, while parasites try to destroy the crop.

2. User Experience:

First and foremost, the goal of the application is fun for the user. The user should be able to plant the garden and obtain a yield, while also challenging a race against time. With the given 40 fields, the first few plants are still relatively easy to keep track of, but from 20 upwards it becomes more and more difficult. In addition to fertilizing and watering, you also have to keep an eye on the parasites that fly to the planted fields. Accordingly, reaction and perception skills are essential for success in this game.

3. Platform:

To be able to develop all functionalities in time, the planned platform for the first draft is the PC. Later, after the game has been tested, a mobile version for smartphones can also be implemented, as recommended in the task. However, this cannot be guaranteed.

4. Audio:

For ease of use, the focus is mainly on the visual design. Audio files will therefore not be implemented.

5. General Functions

1.1. Plants

As required by the task, five different types of plants are integrated. Each plant has the basic functions:

- Timer
- Appearance (there are 3 different growth phases)
- Purchase price
- Final price
- Fertilizer
- Water

The values vary among the types. When a plant is planted via click, a seedling of the selected plant is placed on the field. It cannot be removed before the timer expires or the plant dies. After the timer exceeds half of the growth time, a check is made to see if half of the required fertilizer/water has been added to the plant. If not, the plant dies, if yes, the second growth phase starts. Here the appearance of the plant changes. At the end of the timer, it is checked again if all requirements are fulfilled. If yes, the image of the finished fruit is displayed, if not, the plant dies. Finally, the user can harvest the fruit with a mouse click. The sales money is credited to the capital. A dead plant must also be removed by clicking on it, but does not return any money.

1.2. Field

Fields have either the state planted or empty. If a field is empty, only placeholder values are displayed in the right information column. The Fertilize and Water functions have no effect. If the

field has the state planted, no other plant can be planted. This state will only change again when the plant dies or is harvested. After clicking on such a plant, all information in the field is reset.

1.3. Parasites

In a certain time interval, a parasite flies onto a field with the state planted. The field is chosen by math.random. As soon as the parasite reaches its destination a timer runs, which represents the time it needs to eat the plant. When this timer expires, the plant and the parasite are removed from the field. The normal representation of a dead plant is displayed. If the user reacts to the infestation in time and uses pesticides, the timer is reset and the parasite dies.

1.4. Fertilizing

Each plant has a required fertilizer measure. By clicking on fertilize, the counter is increased by the value 1. After each use, the price is subtracted from the capital and it is checked if the value required by the plant type has been exceeded. If yes, the plant dies.

1.5. Watering

Each plant has a required watering value. With a click on water, the counter is increased by the value 1. After each use it is checked if the value required by the plant type has been exceeded. If yes, the plant dies.

6. Basic

The user opens the html page where the game is implemented. As soon as the page is opened, the game field is generated. On the left side of the page a table is created, which consists of 4 columns and 10 rows. It forms the field slots in the further course. On the right side there are two blocks. In one of them all the values of the selected plant are displayed, in the other one the store with the 5 types of plants, fertilizers and pesticides. The game starts with planting the first seedling.

- The user clicks on the desired plant
- The "seedling" image of the plant is shown & the timer starts. The values are on the right.
- The user plants more plants
- As soon as min. one field has the state planted, beetles fly on the plants in 4 second intervals.
- The user must treat infested plants with pesticides within 5 seconds or they will die.
- The first plant reaches half the growth time and checks if half of the required fertilizer/water has been given. If not, it dies. If yes, its appearance changes to "young plant".
- The user continues planting plants and fighting parasites.
- After the timer expires and the fertilizer/water is checked, the appearance of the plant changes to "ripe".
- The user clicks on the plant.
- The plant is harvested, which means the yield is added to the capital and the field resets.
- The user continues to play as long as he wants. There is no fixed goal.

7. Interactions

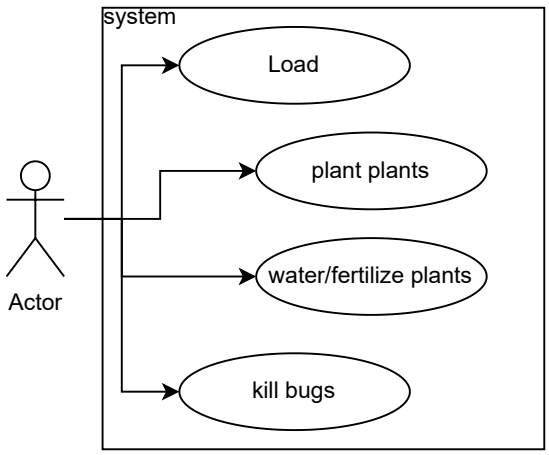
Activities

Arbeitspaket	Aktivität
AP 1 Creation of the Pitch	The playing field (fields, store, values, capital, timer) is created.
AP 2 Selecting a field	The user must select the desired field by mouse click.
AP 3 Planting a plant	The user selects a plant he wants to plant by mouse click.
AP 4 Subtract capital	All entries in the store cost money.
AP 5 Parasites fly	Small parasites fly over the field at intermittent intervals.
AP 6 Parasite infestation	If a parasite lands on a plant, a timer starts how long it takes for the plant to be eaten.
AP 7 Pesticides	Pesticides eliminate parasites on plants.
AP 8 Watered halftime	Each plant needs a certain amount of water to grow. If it has not been watered after half of the growth period, the plant dies.
AP 9 fertilized halftime	Each plant needs a certain amount of fertilizer to grow. If it has not been fertilized after half of the growth period, the plant dies.
AP 10 Plant dies	The plant is erased from the field and gives no money.
AP 11 new appearance plant	The appearance of a seedling changes to the young plant.
AP 12 water	Each plant needs a certain amount of water to grow. With a click of the mouse, the user can select this.
AP 13 fertilize	Each plant needs a certain amount of fertilizer to grow. With a click of the mouse, the user can select it.
AP 14 watered	Each plant needs a certain amount of water to grow. If it has not been watered after the growth period, the plant dies.
AP 15 fertilized	Each plant needs a certain amount of fertilizer to grow. If it has not been fertilized after the growth period, the plant dies.
AP 16 harvested	A ripe fruit can be harvested at the click of a mouse.
AP 17 add capital	Ripe fruit brings new capital.
AP 18 planted	A field with plant gains the status planted
AP 19 Unplanted	A field without a plant gains the status unplanted
AP 20 Halftime	The Timer reaches halftime
AP 21 finished	The Timer is finished

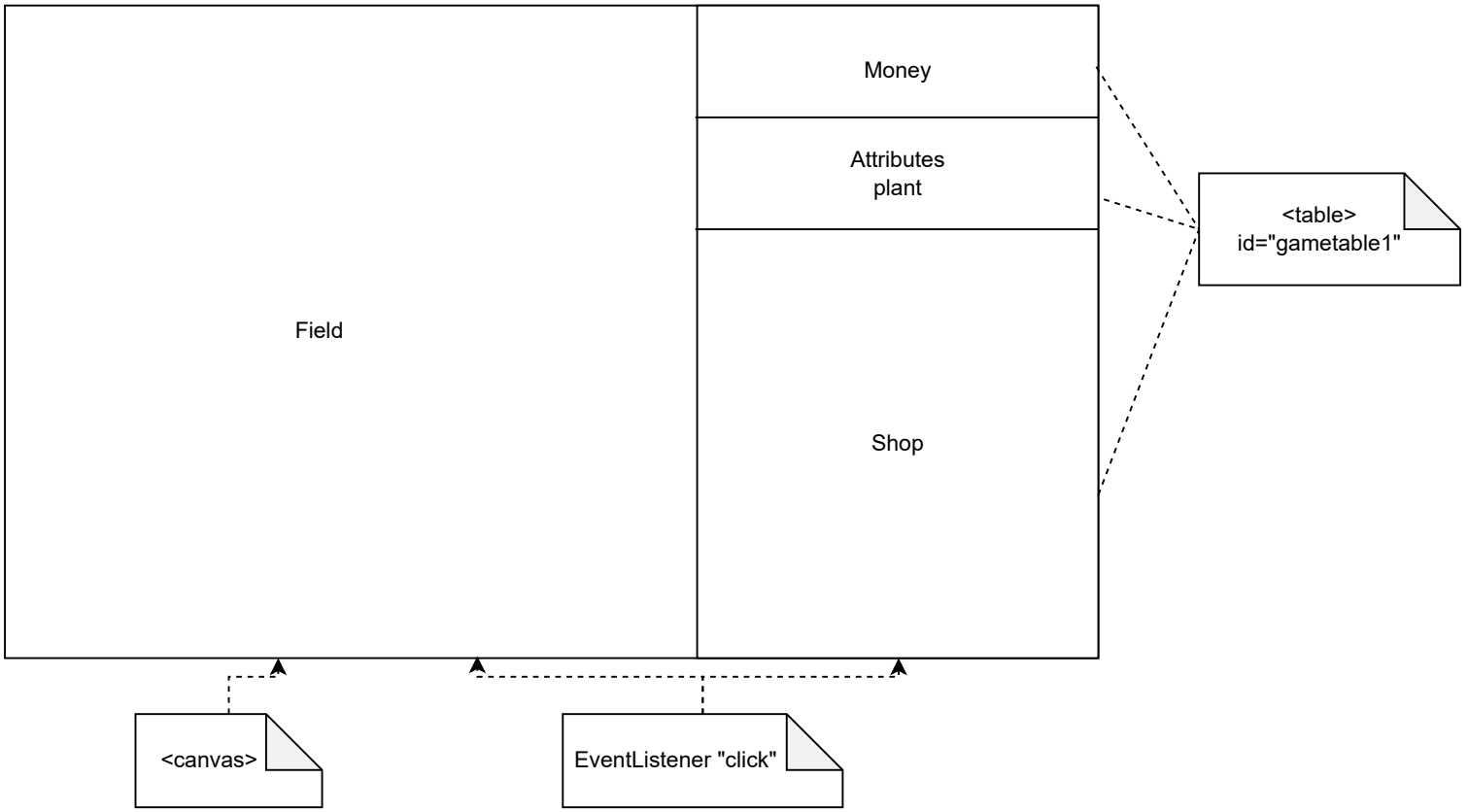
Dependencies

AP 1 -> AP2; AP 3; AP 4; Ap 5; AP 6; AP 7; AP 8; AP 9; AP 10; AP 11; AP 12; AP 13; AP 14; Ap 15; AP 16; AP 17; AP 18; AP 19; AP 20; AP 21	To use any function in the game, you must first create the playing field.
AP 2 -> AP 3; AP 7; AP 12; AP 13; AP 16	To perform an action, the user must select a field to apply it to.
AP 3 -> AP 4; Ap 5; AP 6; AP 7; AP 8; AP 9; AP 10; AP 11; AP 12; AP 13; AP 14; Ap 15; AP 16; AP 17; AP 18	In order to perform functions related to a plant, there must be a plant on the selected field.
AP 3; AP 7; Ap 12; AP 13 -> AP 4	Each interaction with store items costs a certain amount of money.
AP 3; AP 18 -> AP 5	The parasites fly onto the field only when at least one slot has the value planted.
AP 5 -> AP 6	The parasites must be able to fly across the field to settle on a plant.
AP 6; AP 4 -> AP 7	If a parasite settles on the plant, it can be eliminated by pesticides. These can be purchased for money in the store.
AP 18; AP 20 -> AP 8	A plant that has been in the field for half of the total duration and has not yet died is checked. It must have received at least half of the required water.
AP 18; AP 20 -> AP 9	A plant that has been in the field for half of the total duration and has not yet died is checked. It must have received at least half of the required fertilizer.
AP 6; AP 8; AP 9; AP 14; AP 15 -> AP 10	If the period in which the user can act expires without the required function being selected, the plant dies.
AP 8; AP 9; AP 20 -> AP 11	To reach the next stage, the plant must be healthy.
AP 14; AP 15; AP 21 -> AP 16	If the plant is healthy and the time is up it can be harvested.
AP 16 -> AP 17	After the Plant is harvested, the earned money is added to the capital

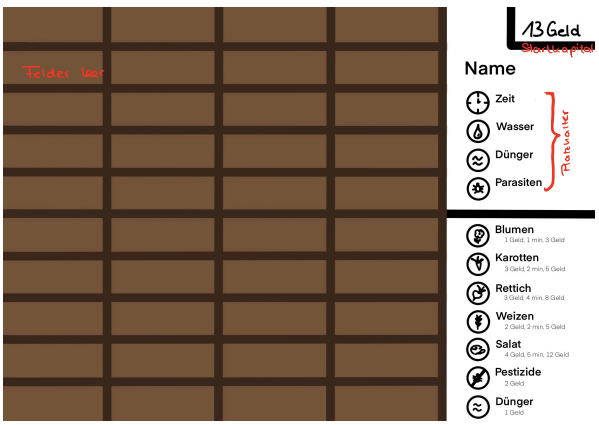
Use Case



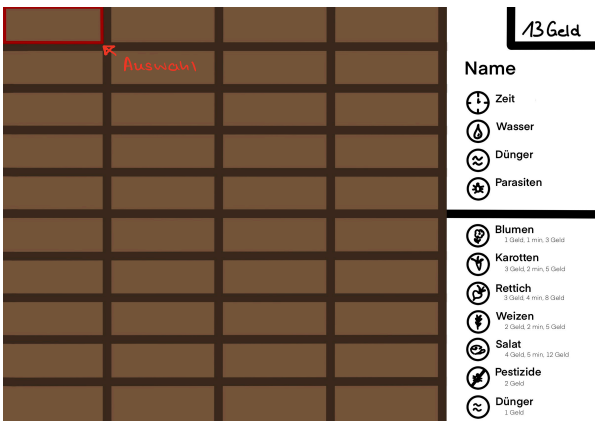
UI-Scribble



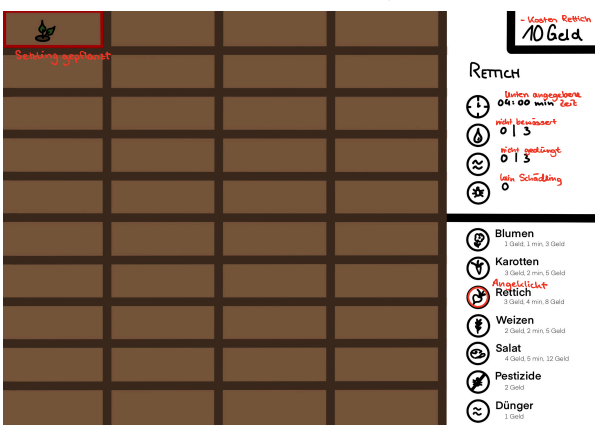
Start-Position: Spielfeld generiert, Startkapital hinzugefügt, Werte Standart



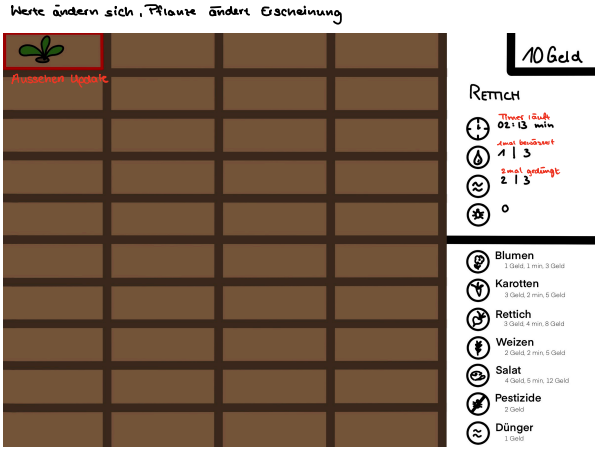
Nutzer wählt Feld aus (klick); Feld markiert



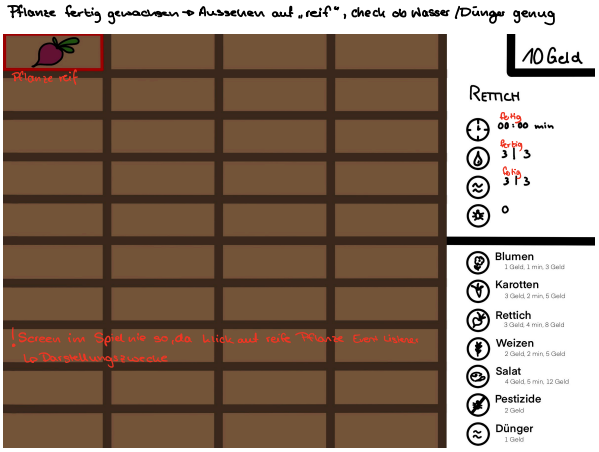
Nutzer klickt auf Pflanze, Setzling auf Feld gesetzt, Werte eingefügt, Kapital abgezogen



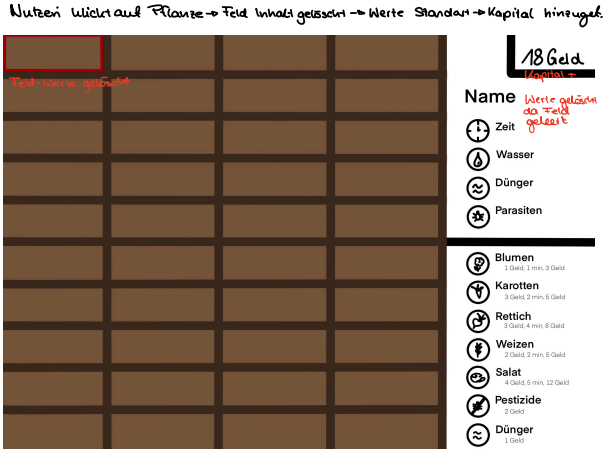
Werte ändern sich, Pflanze ändert Erscheinung



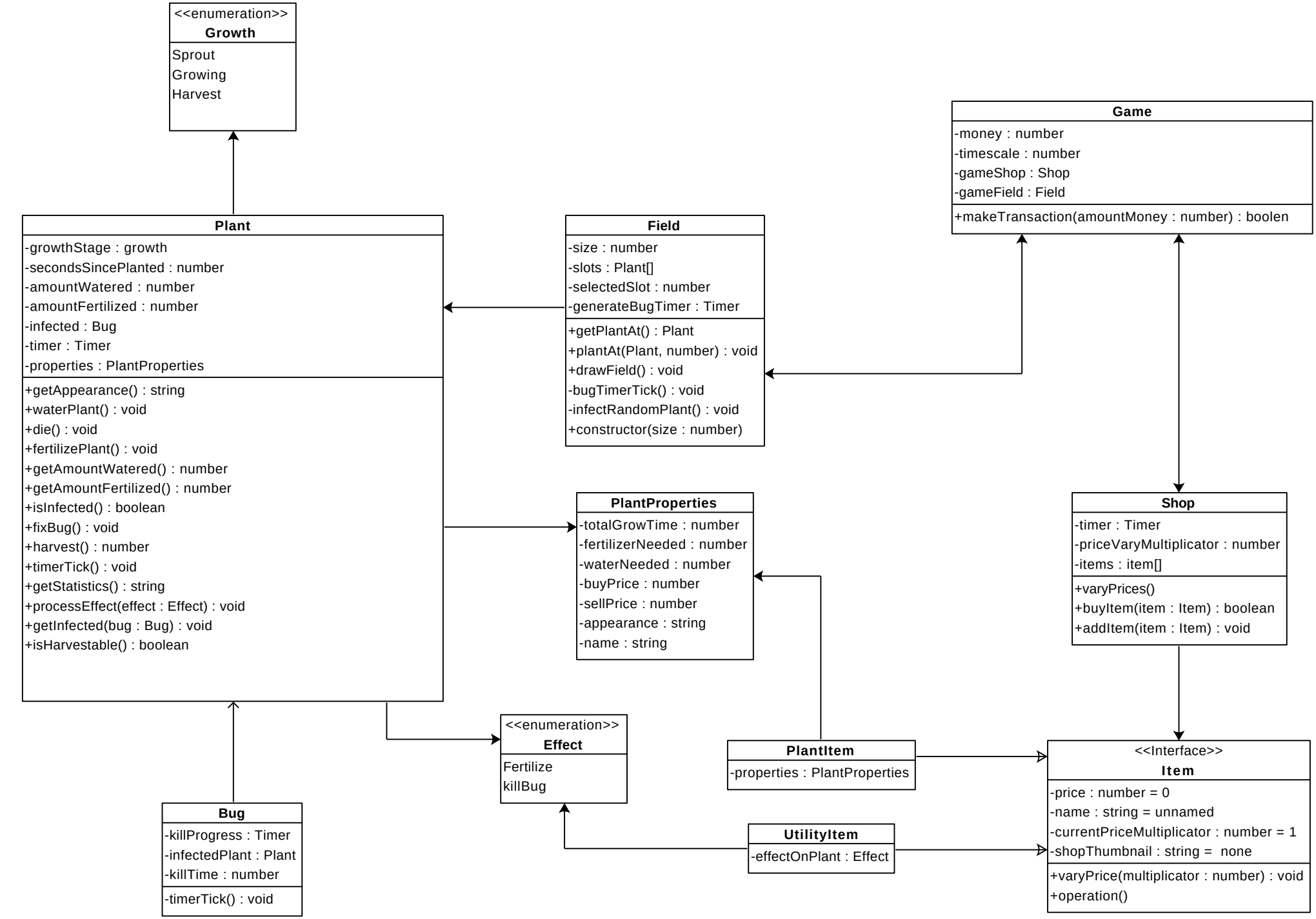
Pflanze fertig gewachsen → Aussehen auf „reif", check ob Wasser/Dünger genug



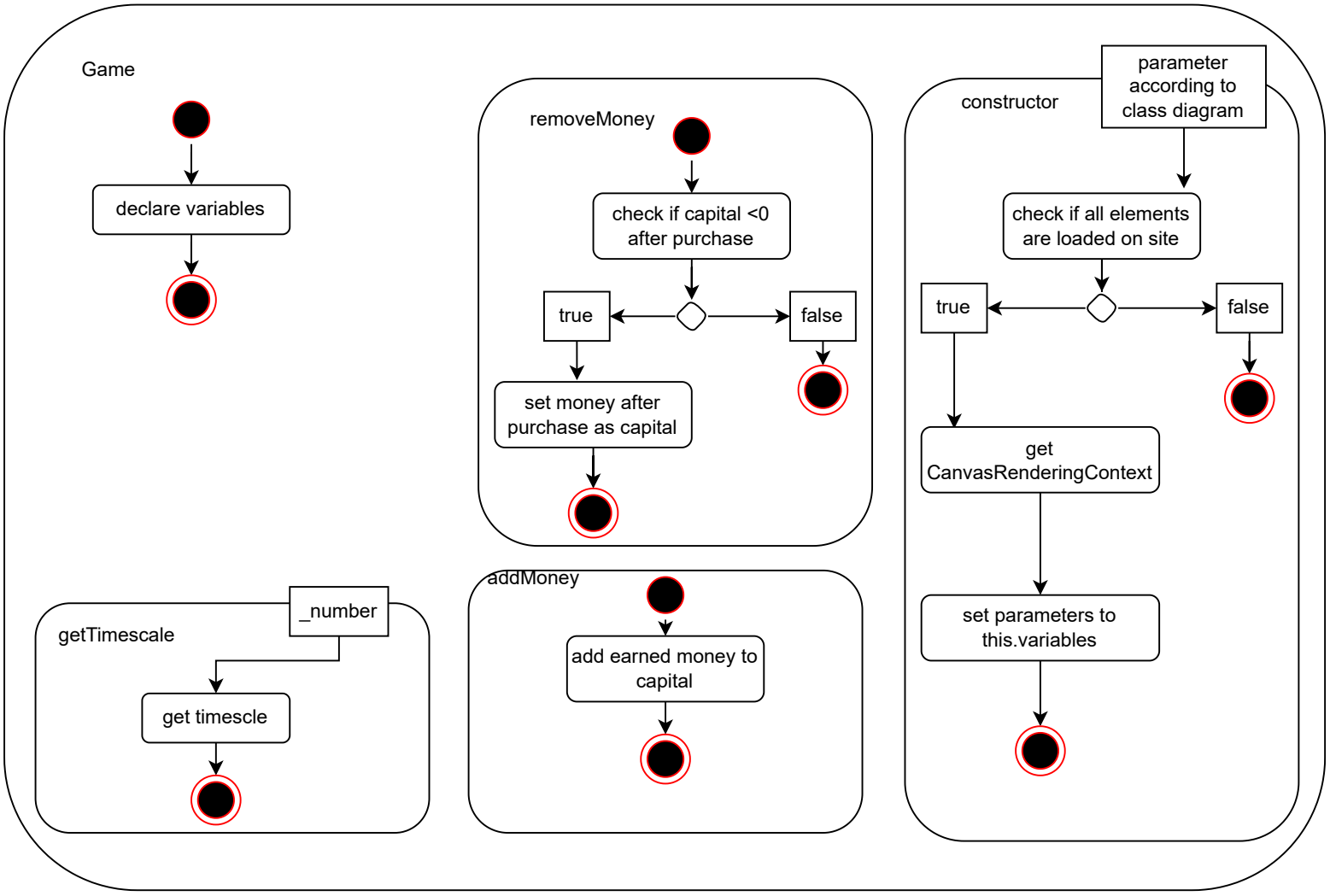
Nutzer klickt auf Pflanze → Feld Inhalt gelöscht → Werte Standart → Kapital hinzugef.

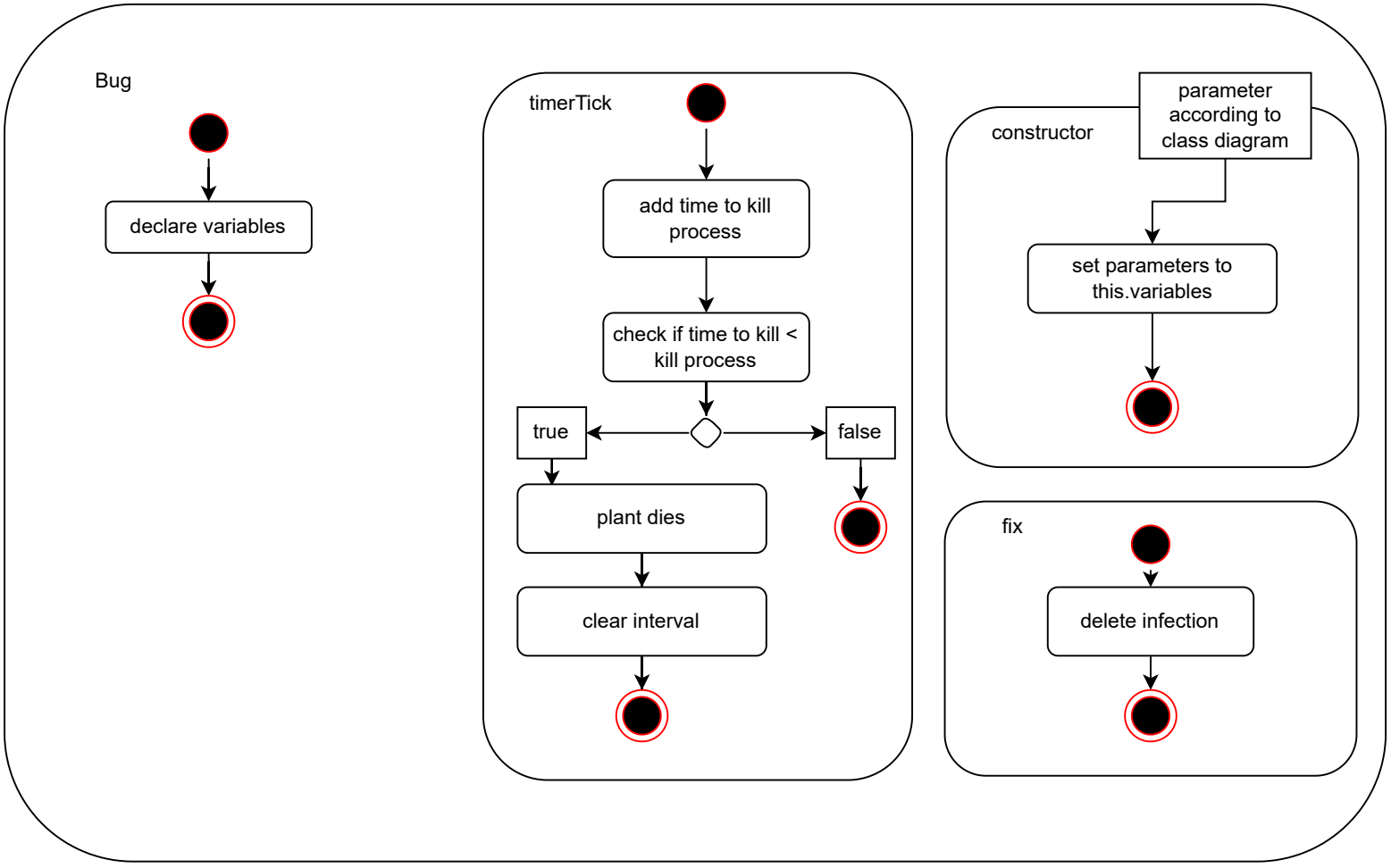


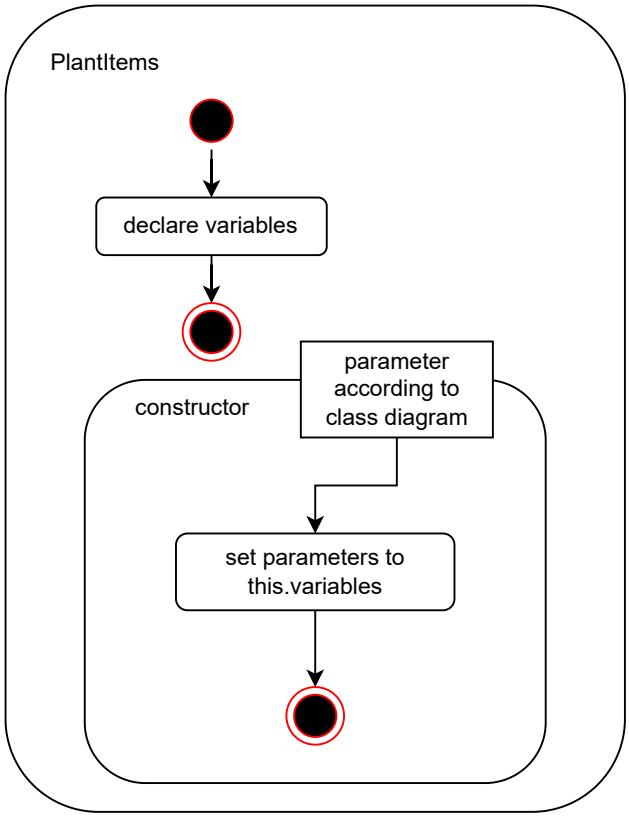
Classdiagram

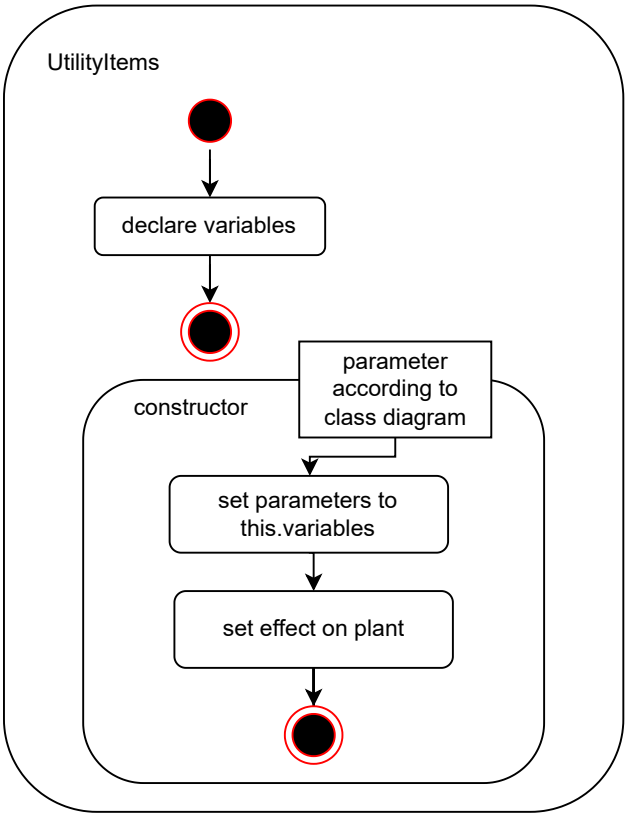


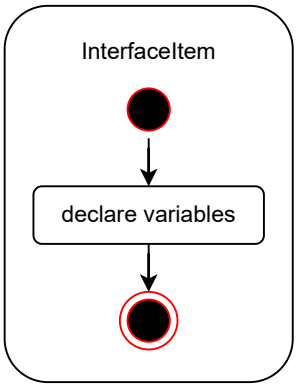
Activitydiagram

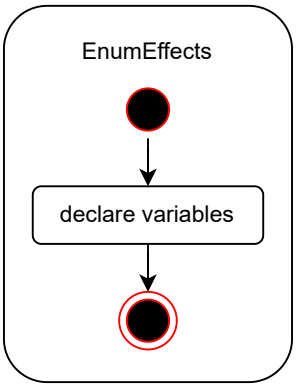


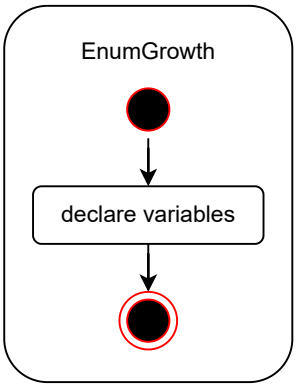


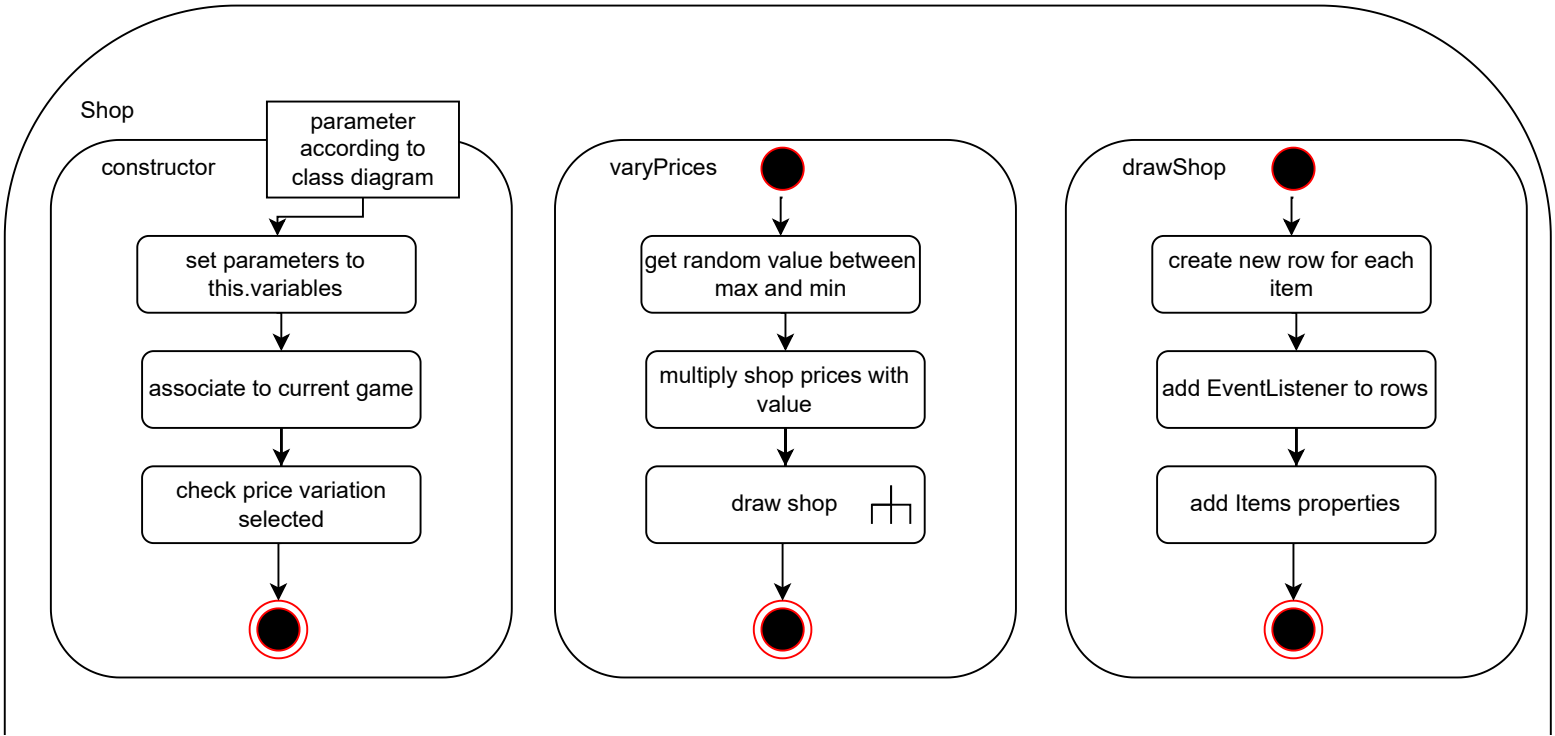

PlantItems
declare variables
constructor
parameter according to class diagram
set parameters to this.variables

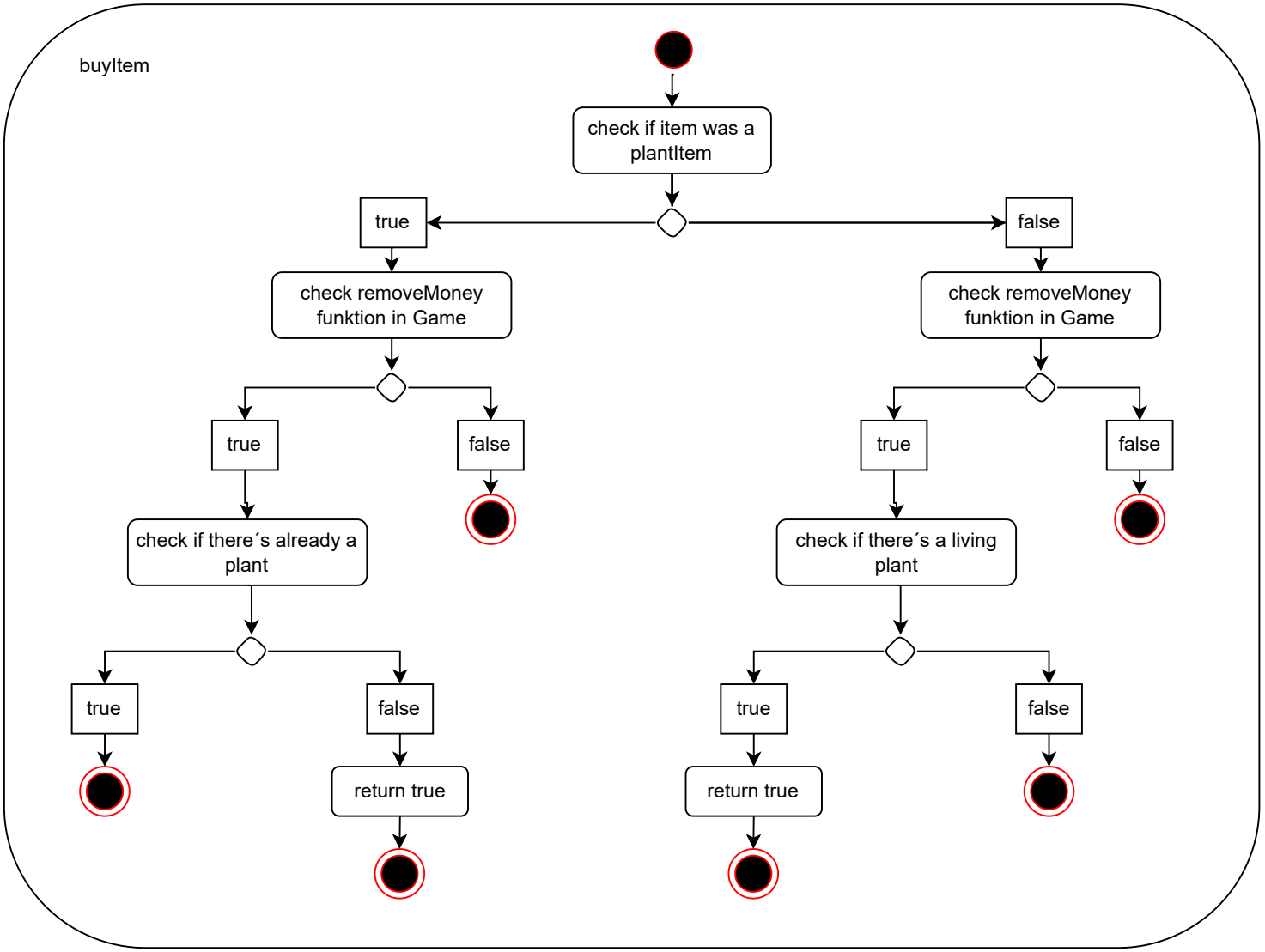

UtilityItems
declare variables
constructor
parameter according to class diagram
set parameters to this.variables
set effect on plant

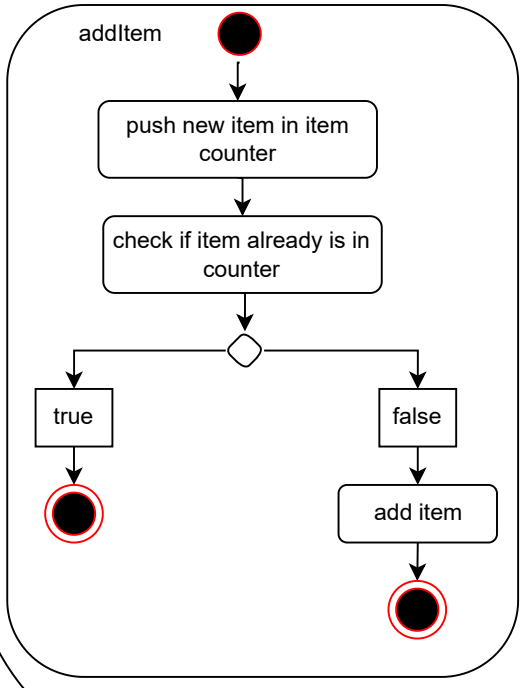

InterfaceItem
declare variables

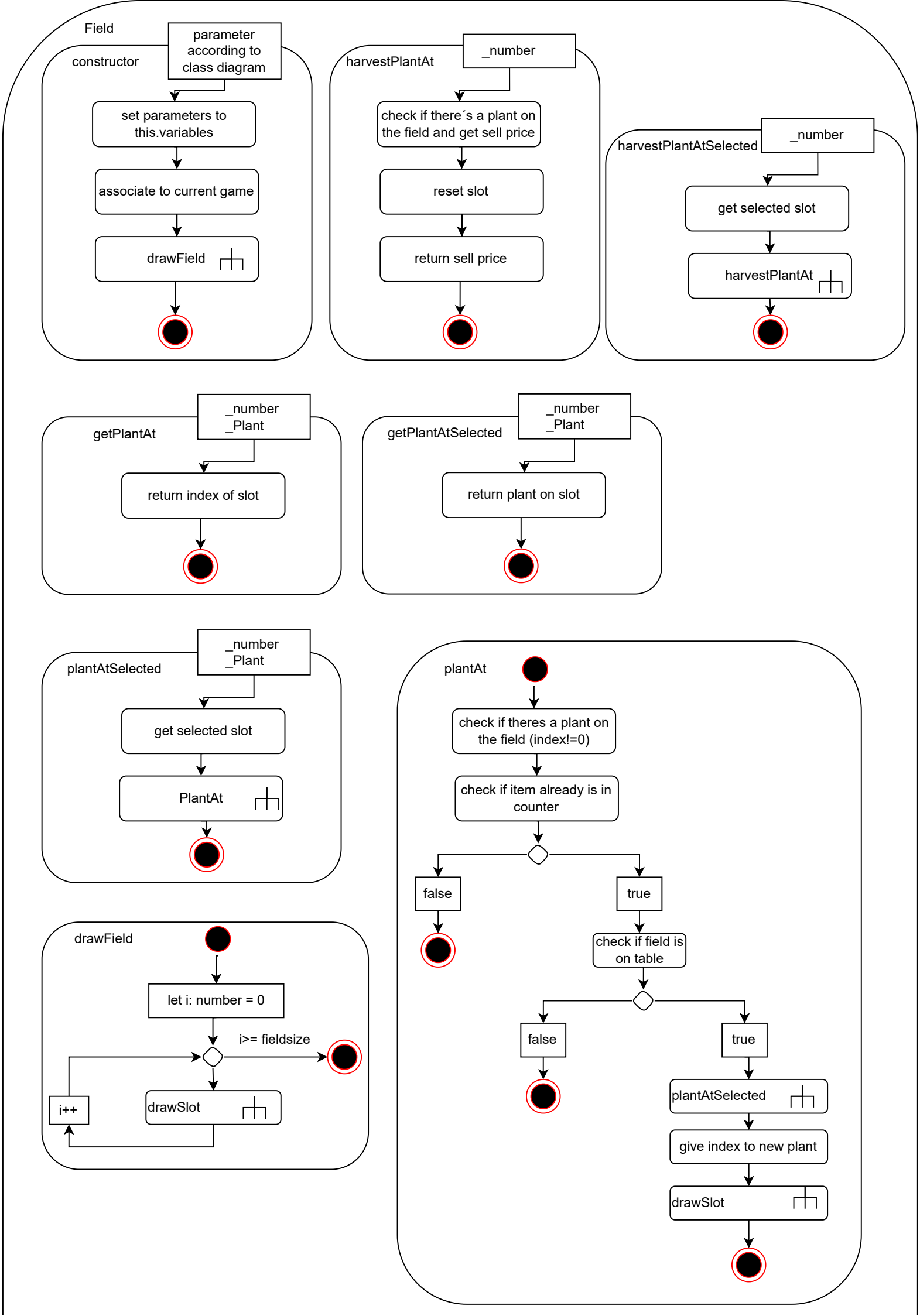

EnumEffects
declare variables

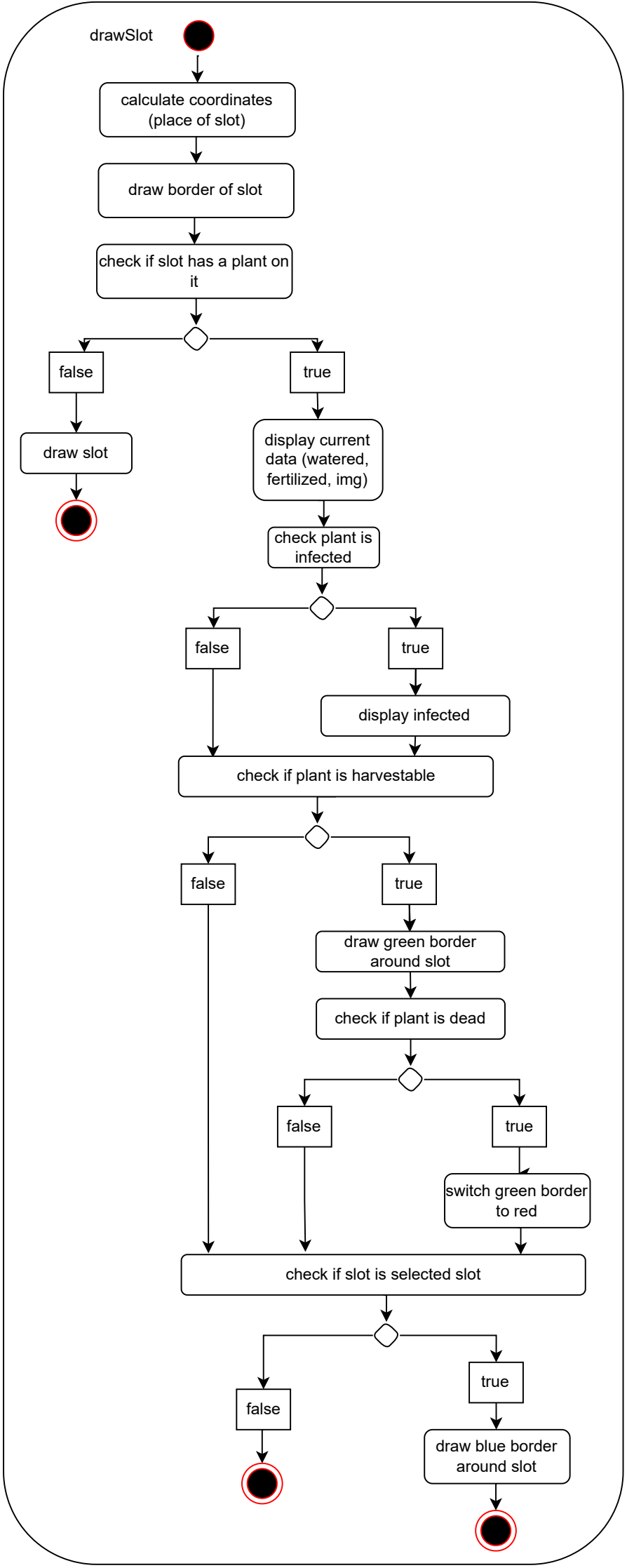

EnumGrowth
declare variables

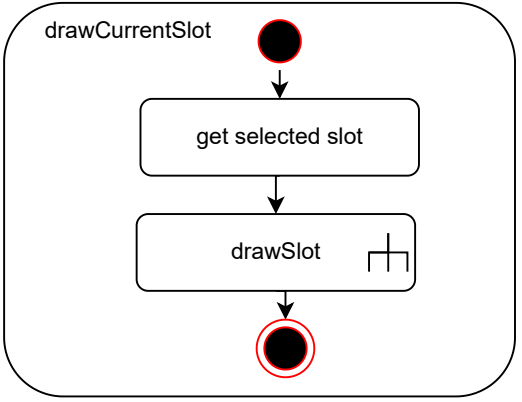

Shop
constructor
parameter according to class diagram
set parameters to this.variables
associate to current game
check price variation selected
varyPrices
get random value between max and min
multiply shop prices with value
draw shop
drawShop
create new row for each item
add EventListener to rows
add Items properties

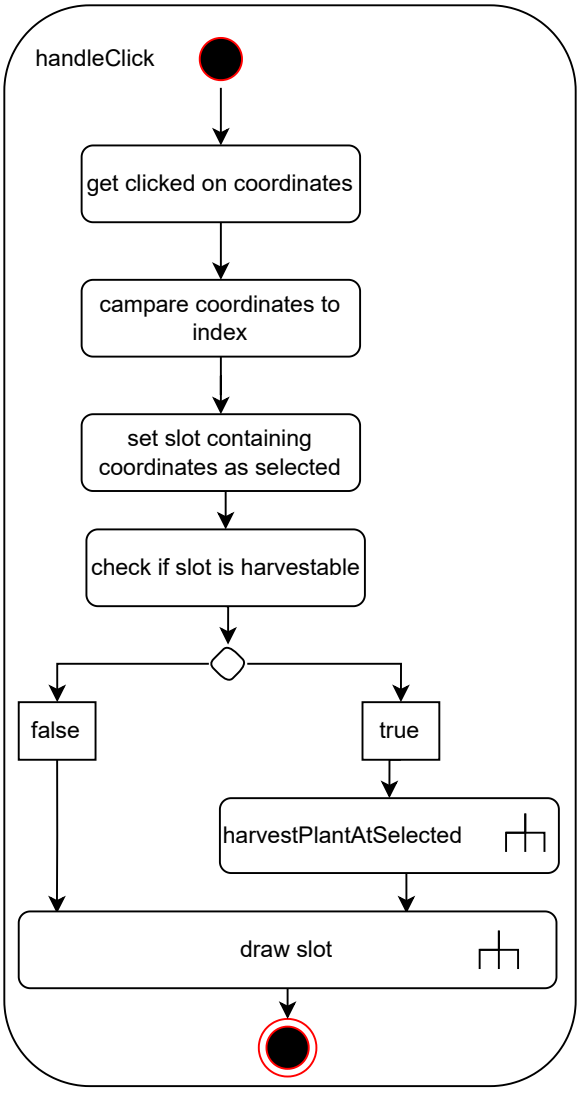

buyItem
check if item was a plantItem
true
false
check removeMoney funktion in Game
check removeMoney funktion in Game
true
false
true
false
check if there´s already a plant
check if there´s a living plant
true
false
true
false
return true
return true

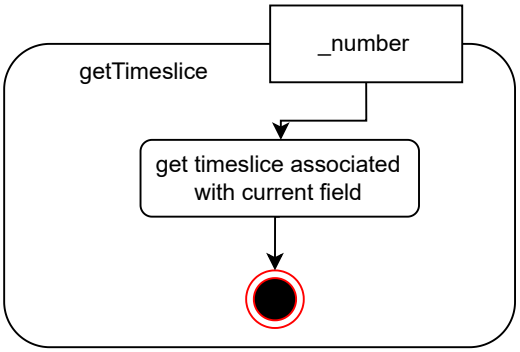

addItem
push new item in item counter
check if item already is in counter
true
false
add item

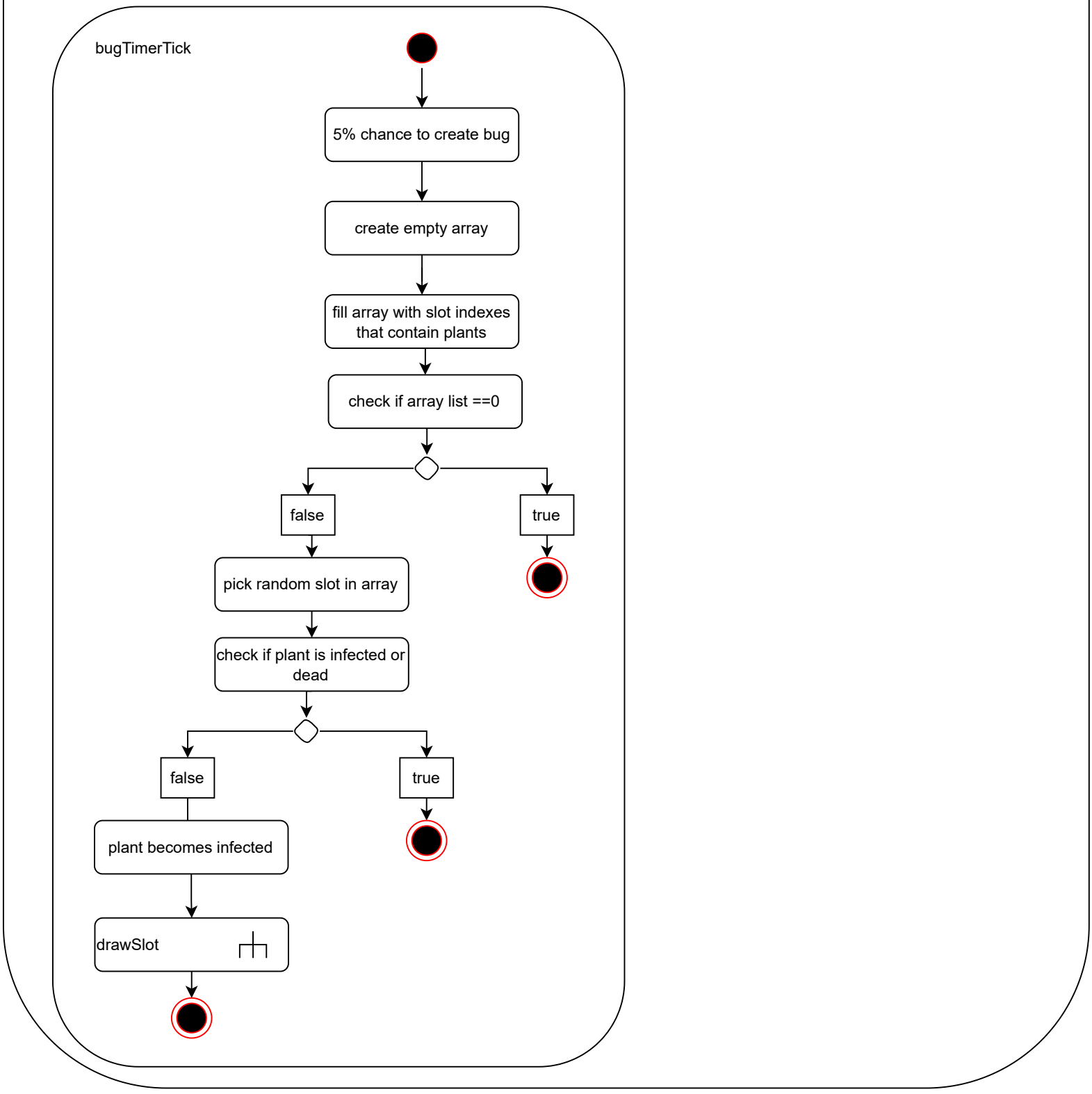

Field
constructor
parameter according to class diagram
set parameters to this.variables
associate to current game
drawField
harvestPlantAt
_number
check if there´s a plant on the field and get sell price
reset slot
return sell price
harvestPlantAtSelected
_number
get selected slot
harvestPlantAt
getPlantAt
_number
_Plant
return index of slot
getPlantAtSelected
_number
_Plant
return plant on slot
plantAtSelected
_number
_Plant
get selected slot
PlantAt
drawField
let i: number = 0
i>= fieldsize
i++
drawSlot
plantAt
check if theres a plant on the field (index!=0)
check if item already is in counter
false
true
check if field is on table
false
true
plantAtSelected
give index to new plant
drawSlot

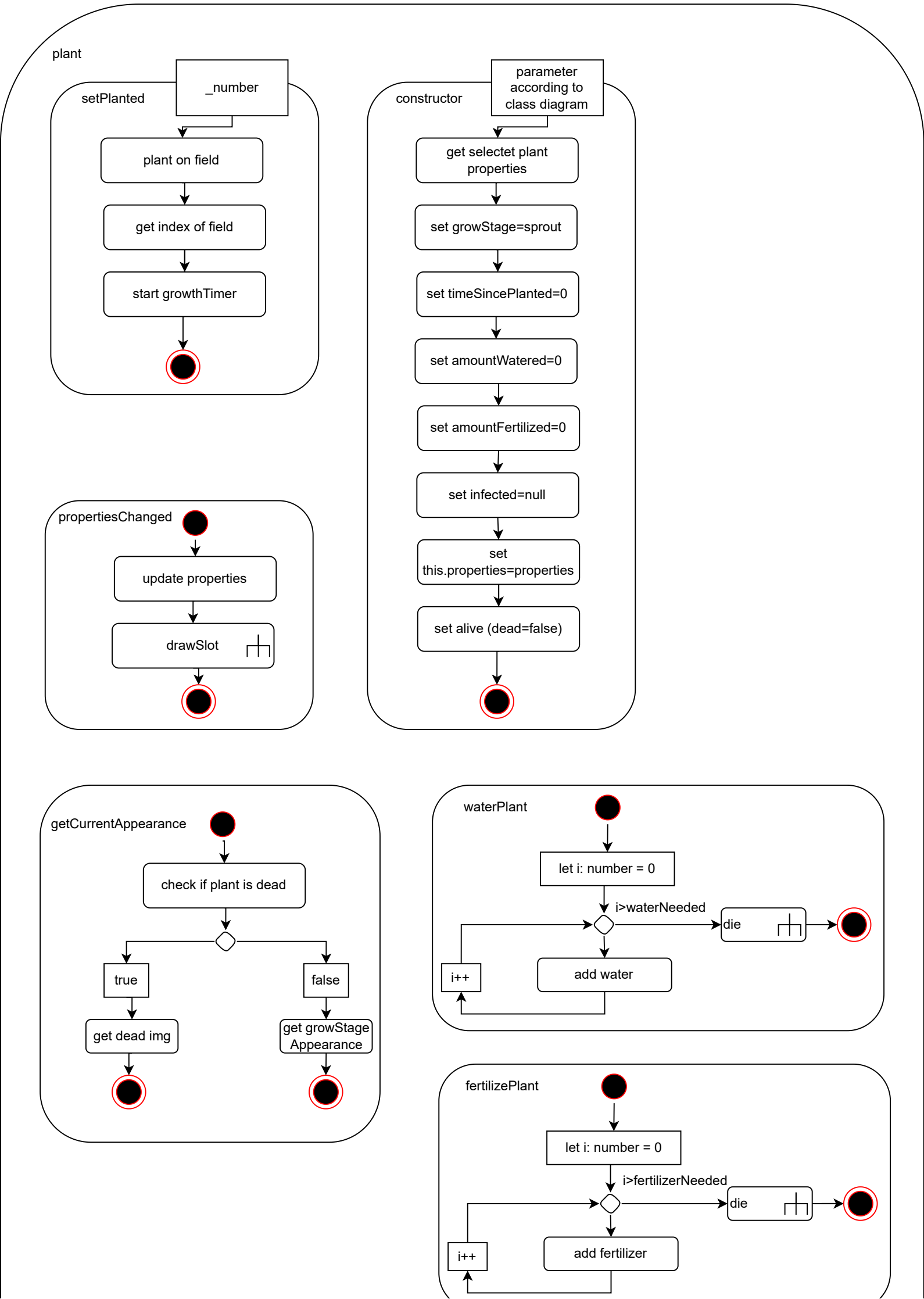

drawSlot
calculate coordinates (place of slot)
draw border of slot
check if slot has a plant on it
false
true
draw slot
display current data (watered, fertilized, img)
check plant is infected
false
true
display infected
check if plant is harvestable
false
true
draw green border around slot
check if plant is dead
false
true
switch green border to red
check if slot is selected slot
false
true
draw blue border around slot

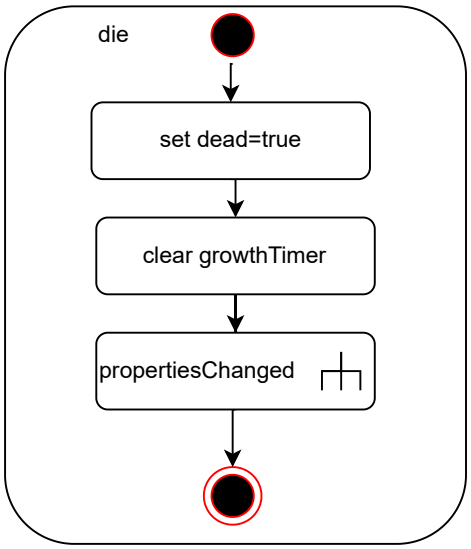

drawCurrentSlot
get selected slot
drawSlot

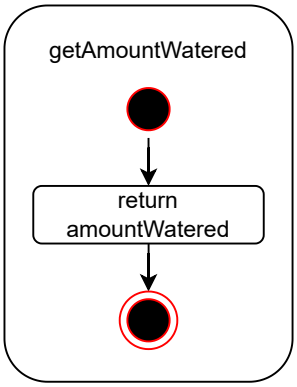

handleClick
get clicked on coordinates
campare coordinates to index
set slot containing coordinates as selected
check if slot is harvestable
false
true
harvestPlantAtSelected
draw slot

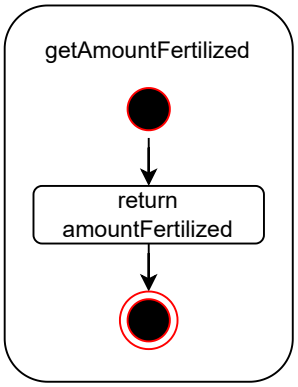

getTimeslice
_number
get timeslice associated with current field

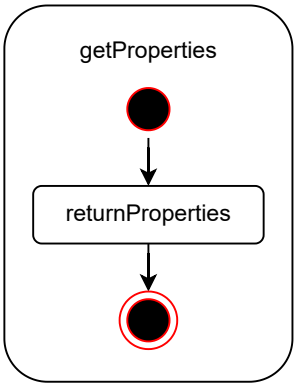

bugTimerTick
5% chance to create bug
create empty array
fill array with slot indexes that contain plants
check if array list ==0
false
true
pick random slot in array
check if plant is infected or dead
false
true
plant becomes infected
drawSlot

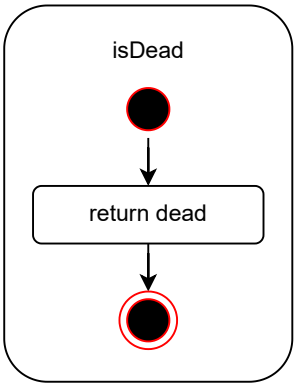

plant
setPlanted
_number
plant on field
get index of field
start growthTimer
constructor
parameter according to class diagram
get selectet plant properties
set growStage=sprout
set timeSincePlanted=0
set amountWatered=0
set amountFertilized=0
set infected=null
set this.properties=properties
set alive (dead=false)
propertiesChanged
update properties
drawSlot
getCurrentAppearance
check if plant is dead
true
false
get dead img
get growStage Appearance
waterPlant
let i: number = 0
i>waterNeeded
die
add water
i++
fertilizePlant
let i: number = 0
i>fertilizerNeeded
die
add fertilizer
i++

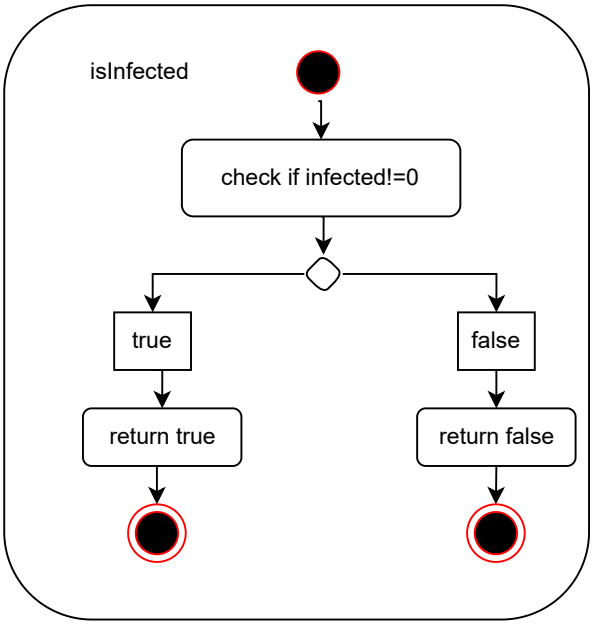

die
set dead=true
clear growthTimer
propertiesChanged

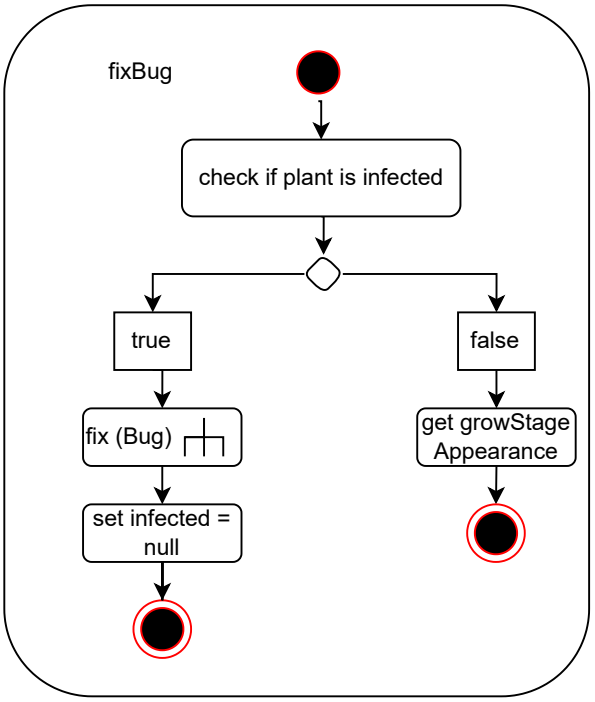

getAmountWatered
return amountWatered

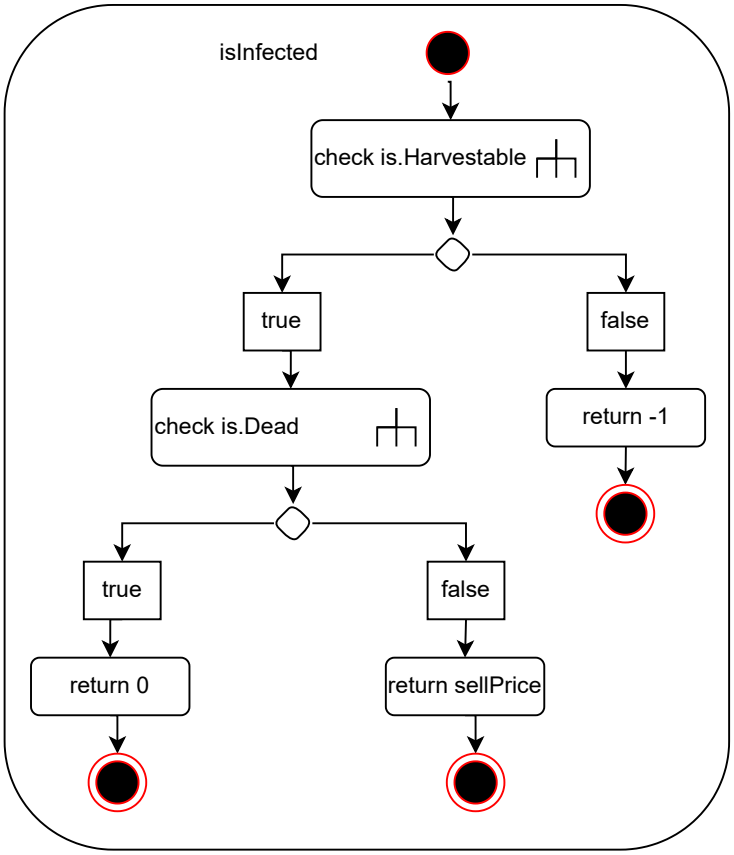

getAmountFertilized
return amountFertilized

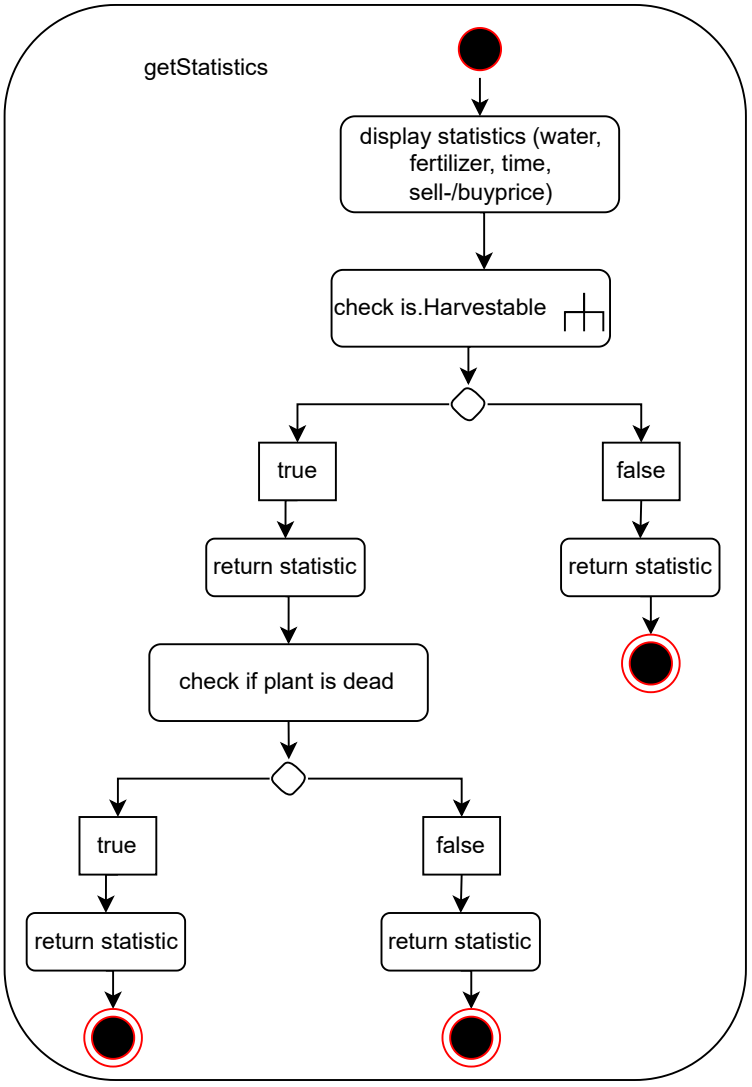

getProperties
returnProperties

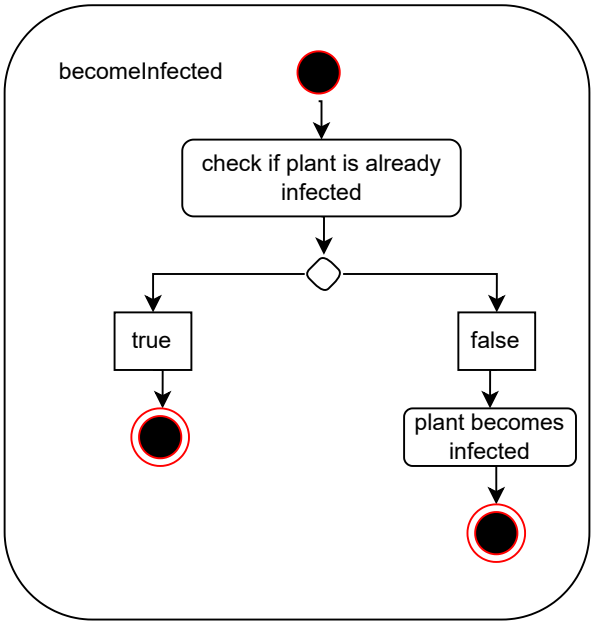

isDead
return dead

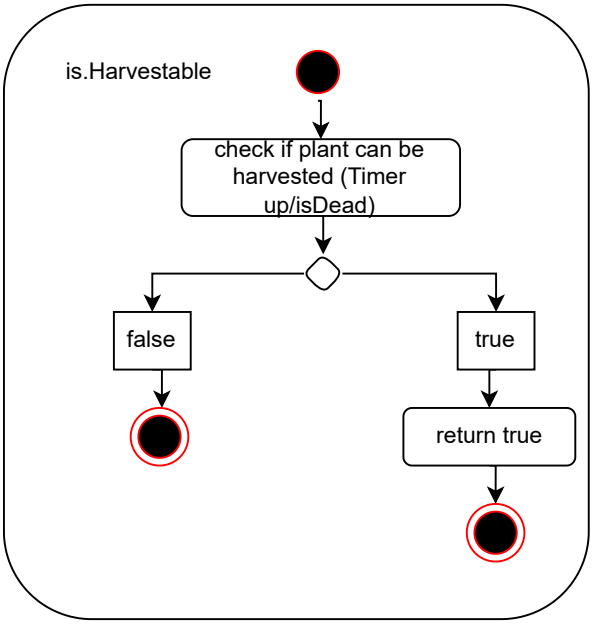

isInfected
check if infected!=0
true
false
return true
return false

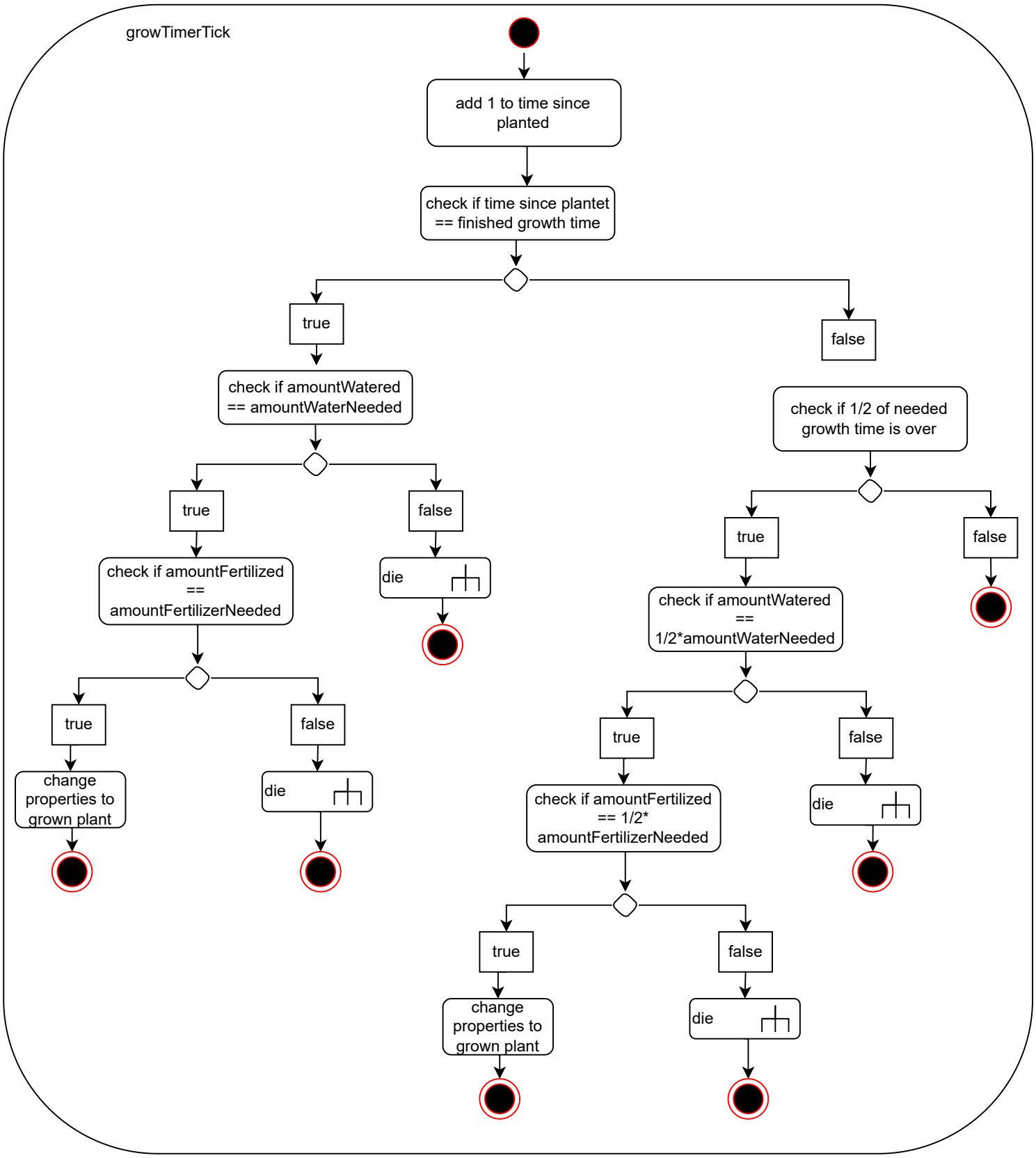

fixBug
check if plant is infected
true
false
fix (Bug)
get growStage Appearance
set infected = null

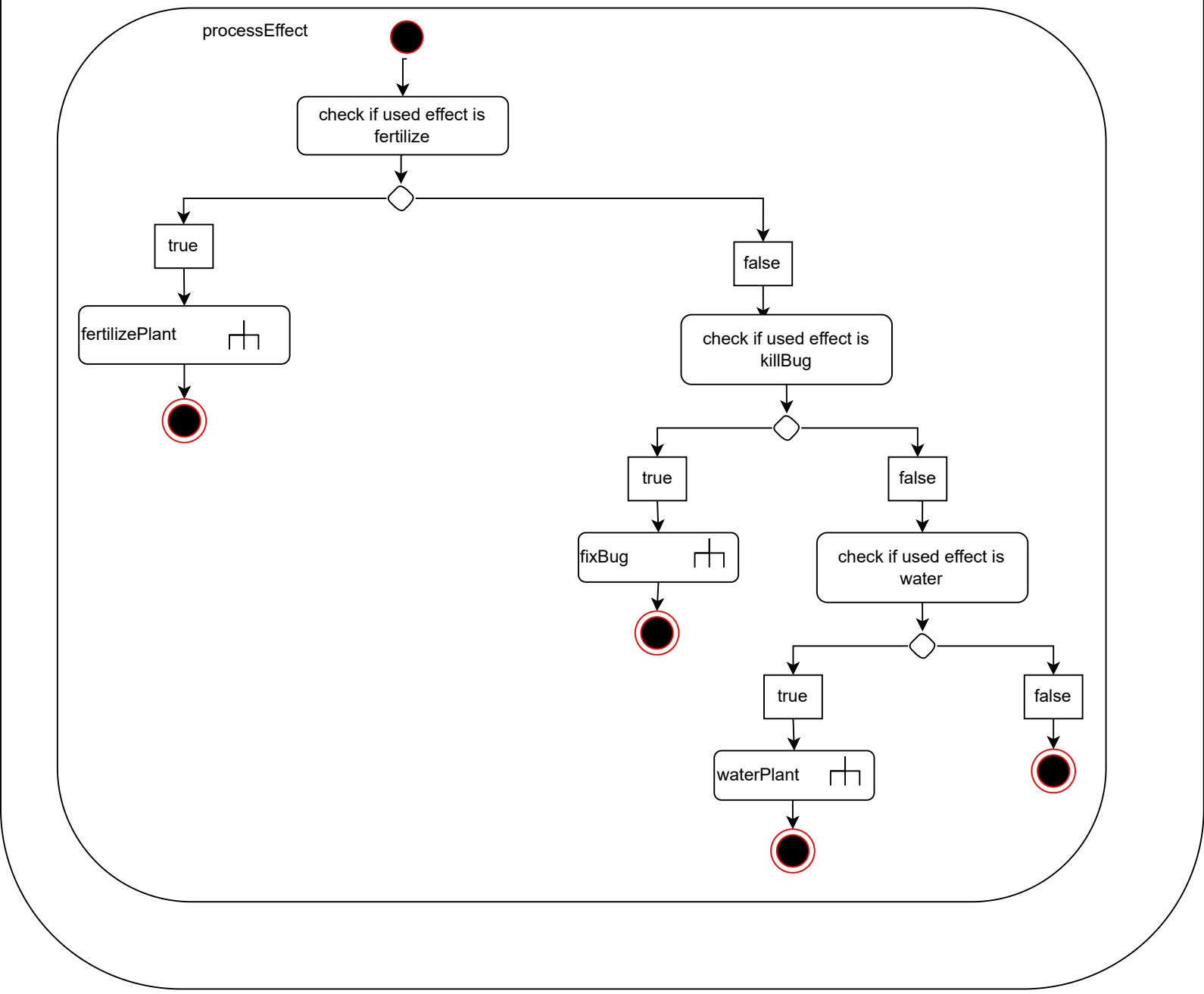

isInfected
check is.Harvestable
true
false
check is.Dead
return -1
true
false
return 0
return sellPrice

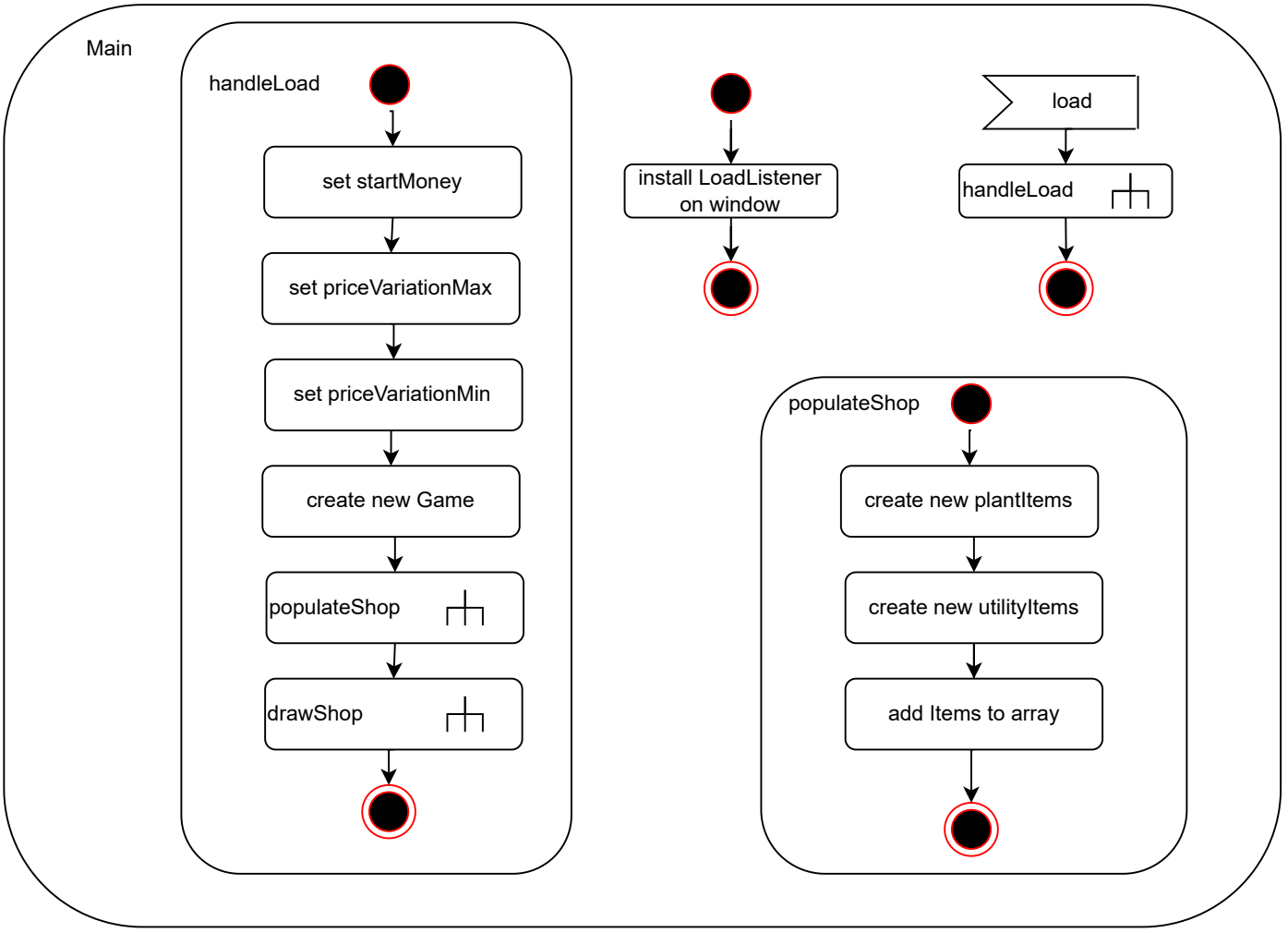

getStatistics
display statistics (water, fertilizer, time, sell-/buyprice)
check is.Harvestable
true
false
return statistic
return statistic
check if plant is dead
true
false
return statistic
return statistic


becomeInfected
check if plant is already infected
true
false
plant becomes infected


is.Harvestable
check if plant can be harvested (Timer up/isDead)
false
true
return true


growTimerTick
add 1 to time since planted
check if time since plantet == finished growth time
true
false
check if amountWatered == amountWaterNeeded
true
false
check if amountFertilized == amountFertilizerNeeded
die
true
false
change properties to grown plant
die
check if 1/2 of needed growth time is over
true
false
check if amountWatered == 1/2*amountWaterNeeded
true
false
check if amountFertilized == 1/2* amountFertilizerNeeded
die
true
false
change properties to grown plant
die


processEffect
check if used effect is fertilize
true
fertilizePlant
false
check if used effect is killBug
true
fixBug
false
check if used effect is water
true
waterPlant
false


Main
handleLoad
set startMoney
set priceVariationMax
set priceVariationMin
create new Game
populateShop
drawShop
install LoadListener
on window
load
handleLoad
populateShop
create new plantItems
create new utilityItems
add Items to array