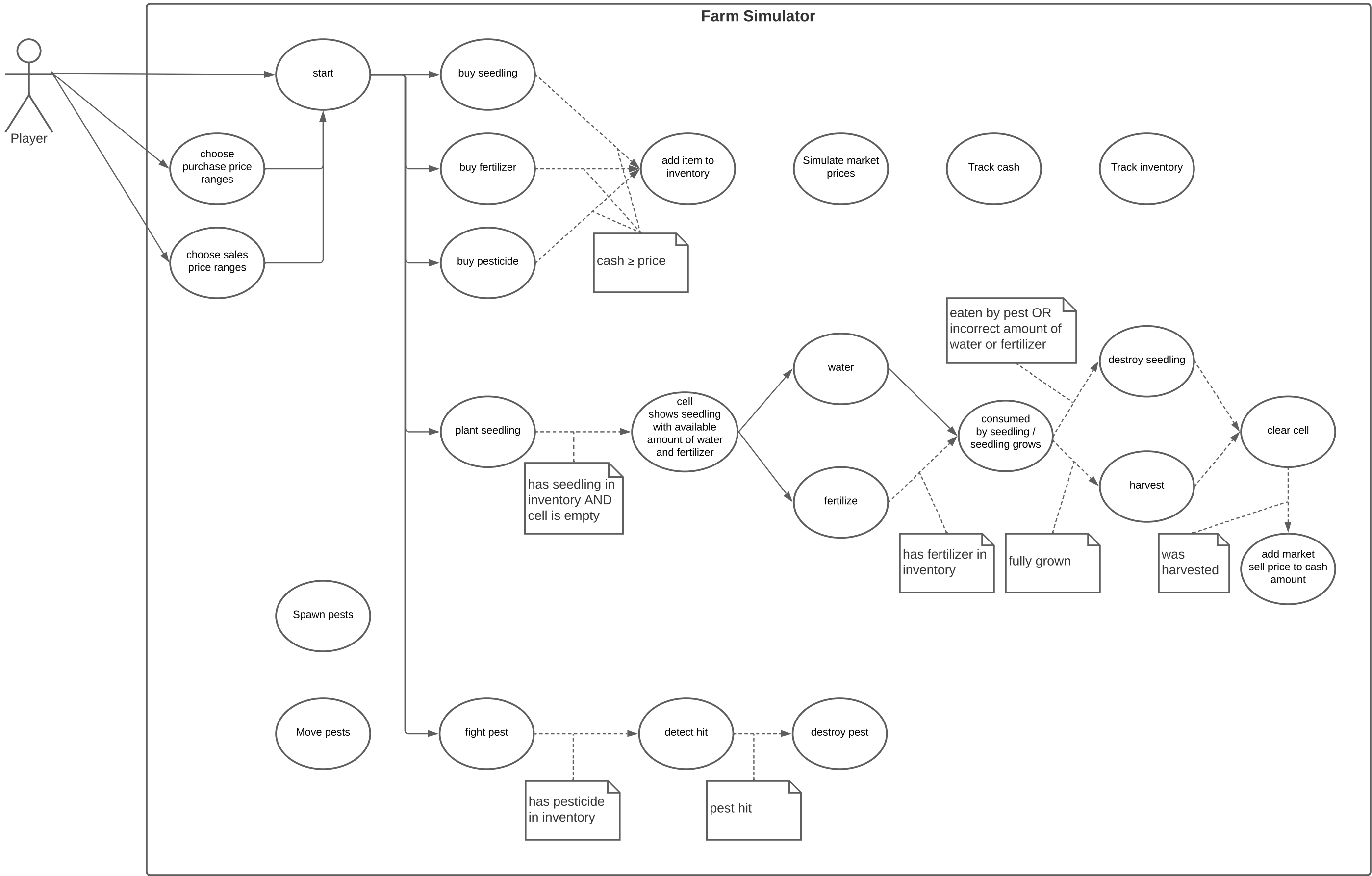


Design Document

EIA2 – Farm Simulator

Manuel Kasperzack

Use Case Diagram



User Interface Scribble

```
<button>  
type = "button"  
value = "start"  
id = "start"
```

click

Farm Simulator Settings

```
<h1>
```

Start simulation

Purchase prices (between 1 and 50)

```
<h3>
```

```
<div>  
id = "invalid-input"
```

```
<label>
```

Red cabbage

Min: Max:

Corn

Min: Max:

Carrot

Min: Max:

Tomato

Min: Max:

Onion

Min: Max:

Fertilizer

Min: Max:

Pesticide

Min: Max:

```
<input>  
type = "number"  
id = "purchase-redCabbage-max"  
min = "1"  
max = "50"  
value = "3"
```

```
<input>  
type = "number"  
id = "purchase-redCabbage-min"  
min = "1"  
max = "50"  
value = "1"
```

(Sales prices (between 1 and 50

```
<h3>
```

```
<label>
```

Red cabbage

Min: Max:

Corn

Min: Max:

Carrot

Min: Max:

Tomato

Min: Max:

Onion

Min: Max:

```
<div>  
id = "prices"
```

Note:

Always the same logic as above, only the values for the HTML attributes id and value change.

The remaining ids are:

"purchase-corn-min", "purchase-corn-max"

"purchase-carrot-min", "purchase-carrot-max"

"purchase-tomato-min", "purchase-tomato-max"

"purchase-onion-min", "purchase-onion-max"

"sell-redCabbage-min", "sell-redCabbage-max"

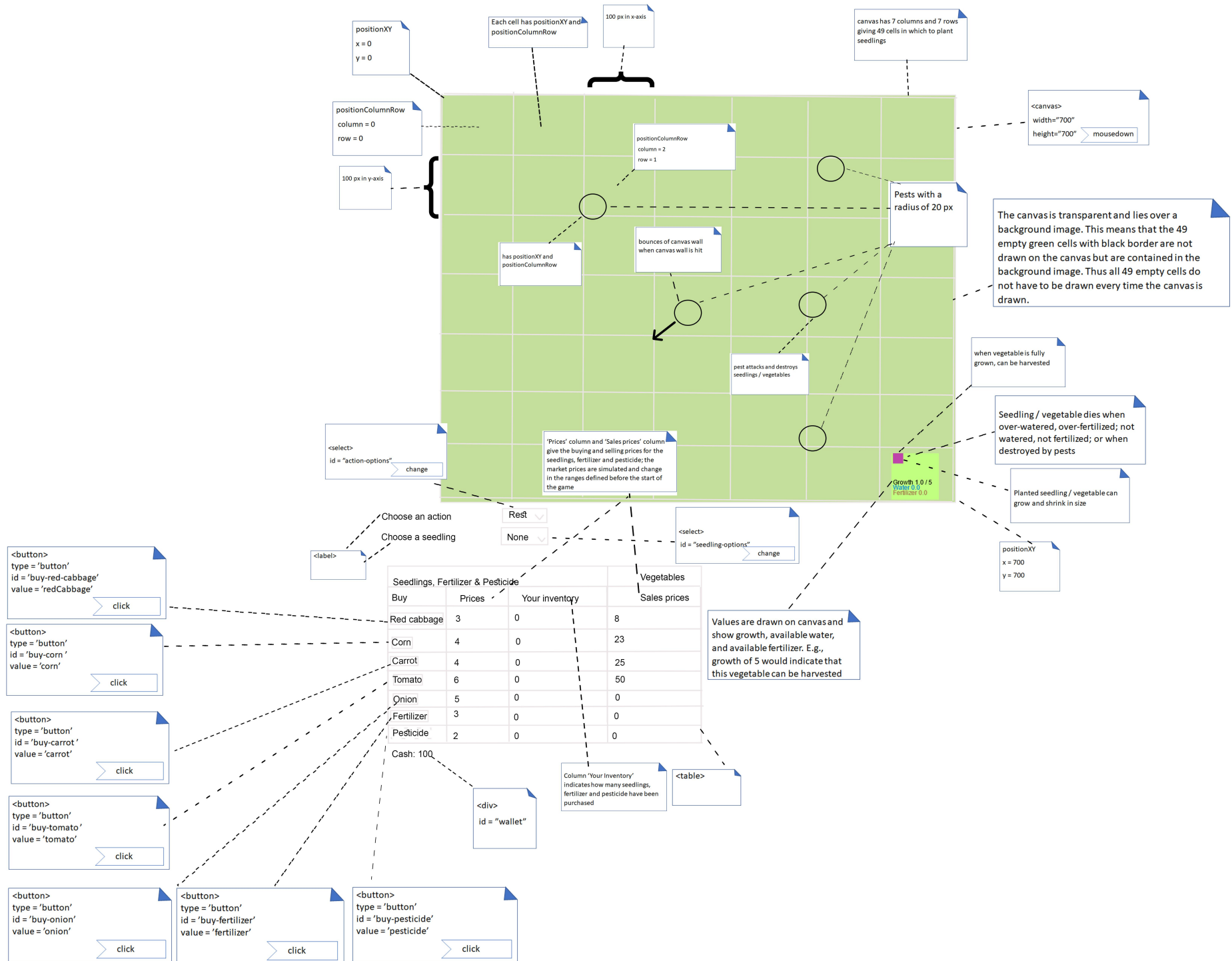
"sell-corn-min", "sell-corn-max"

"sell-carrot-min", "sell-carrot-max"

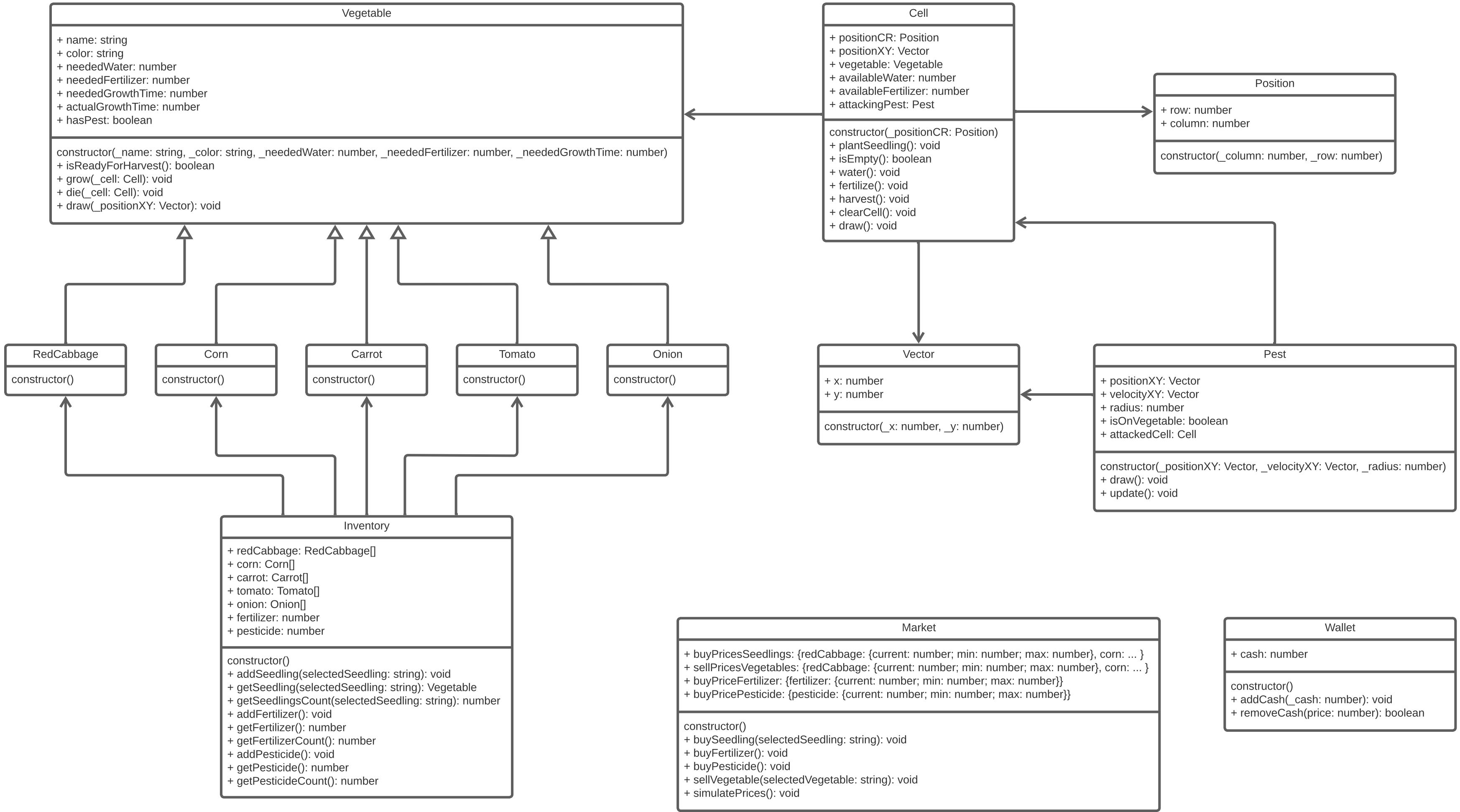
"sell-tomato-min", "sell-tomato-max"

"sell-onion-min", "sell-onion-max"

And the values for the HTML attribute value are the numbers written in the boxes.

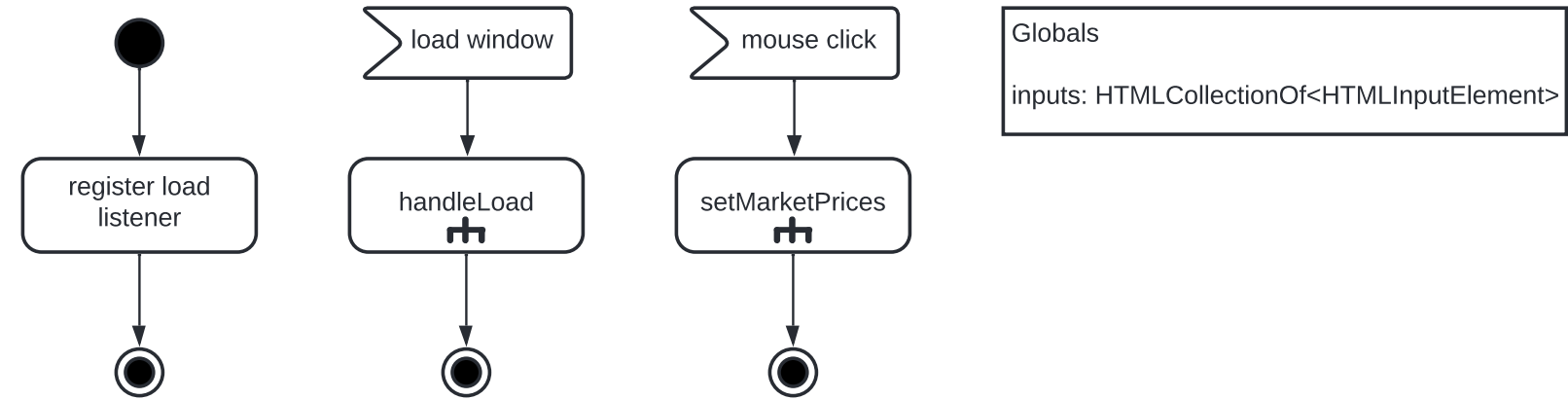


Class Diagram

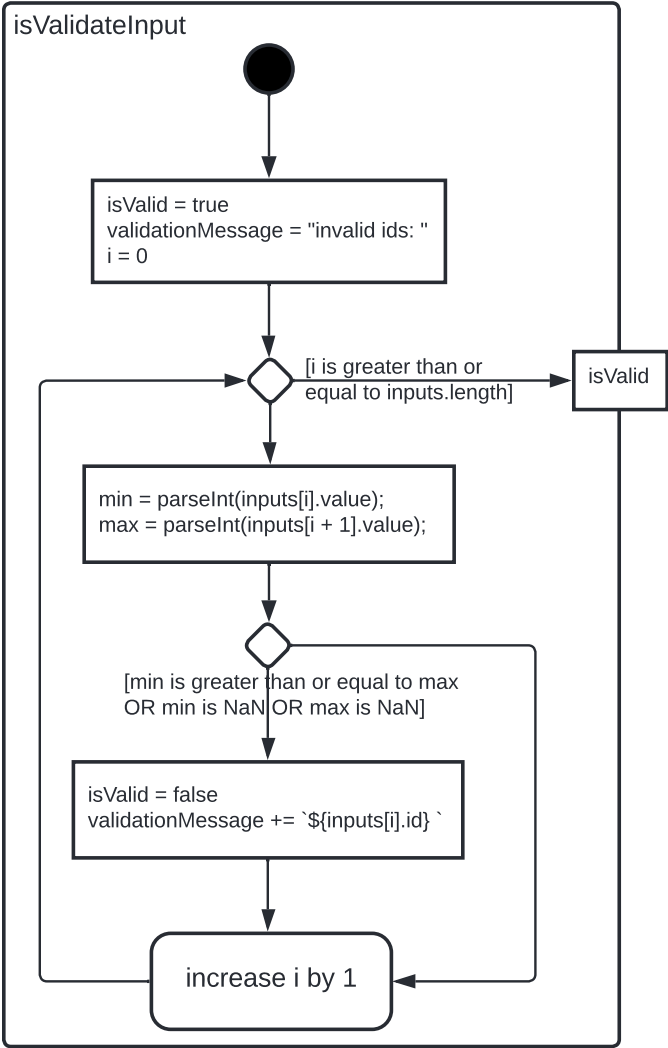
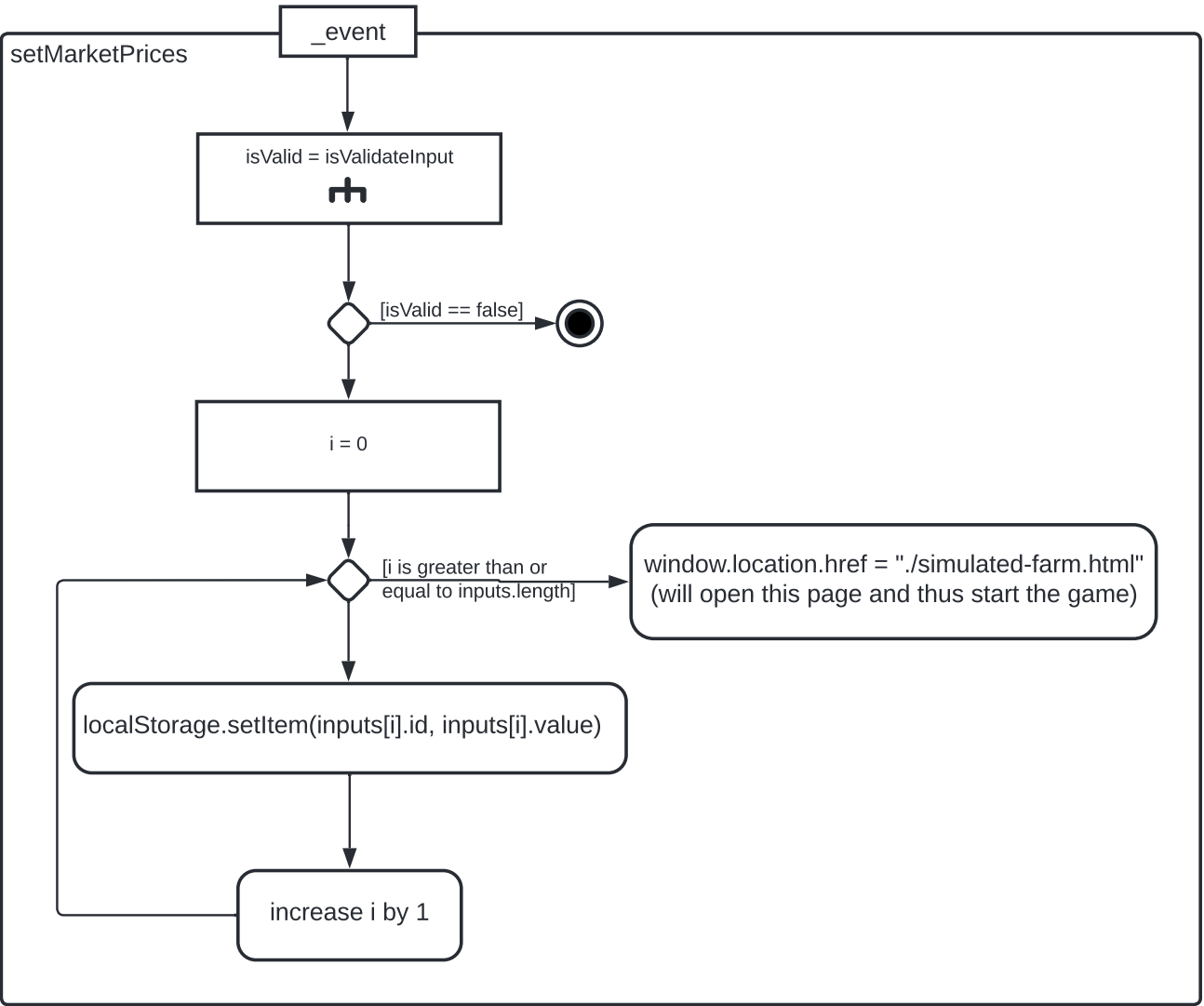
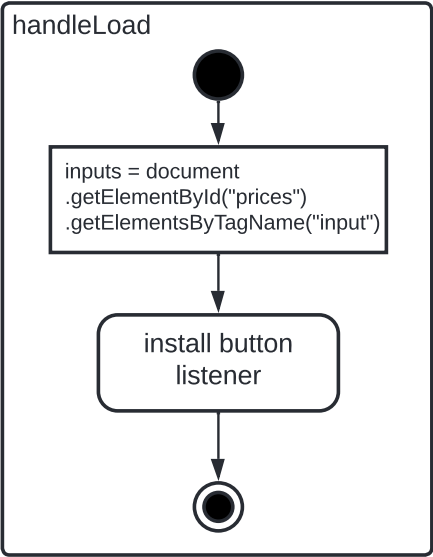


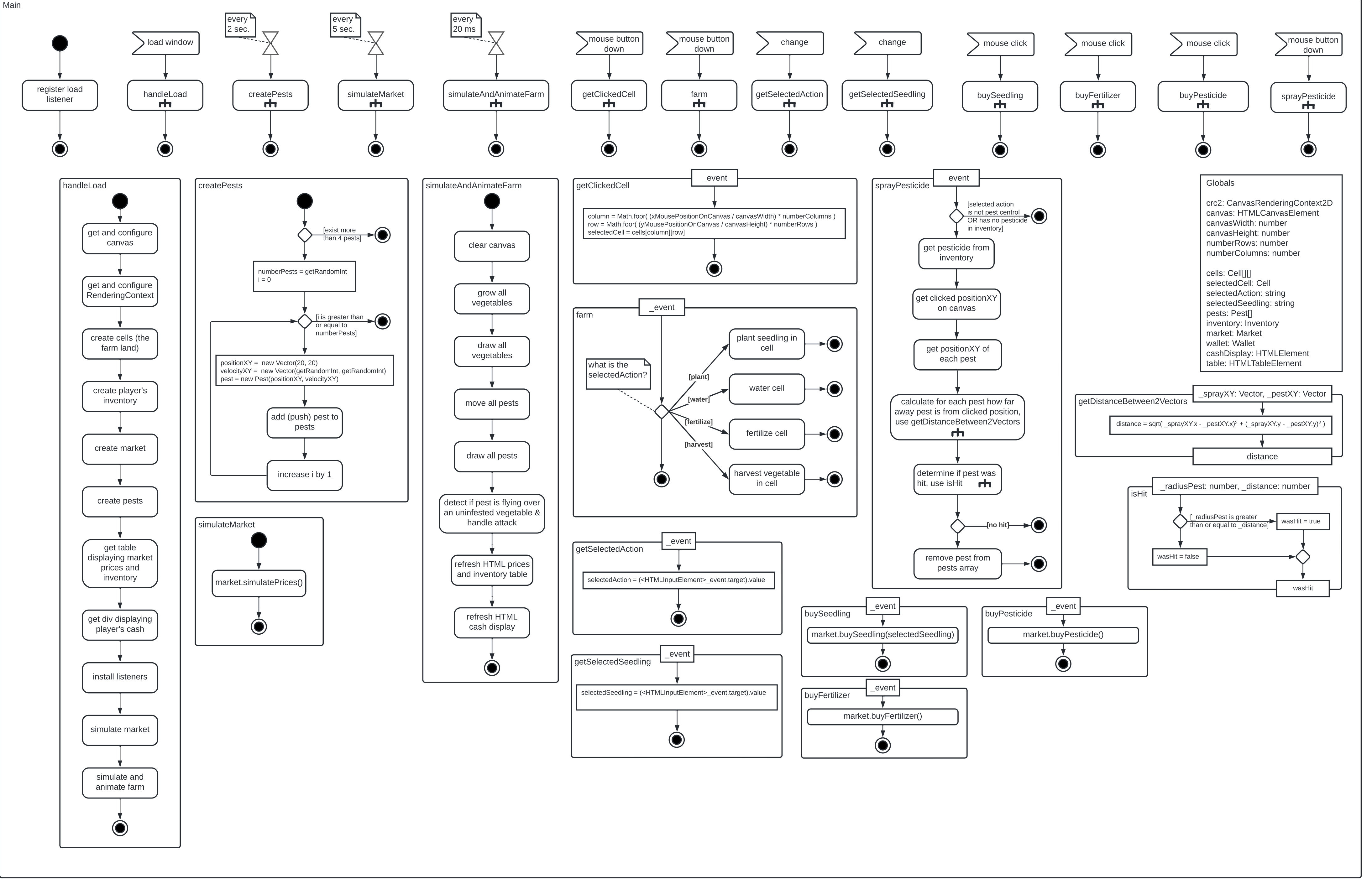
Activity Diagrams

Settings

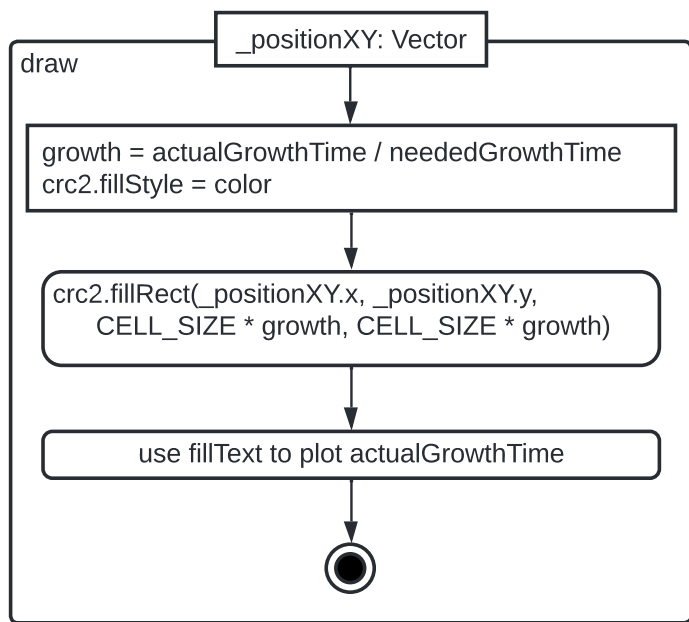
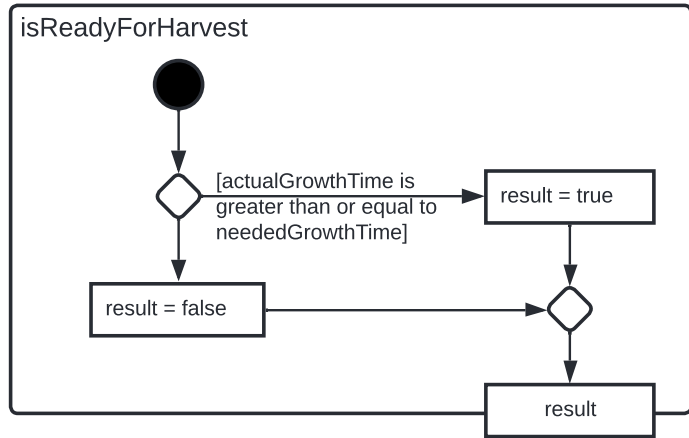
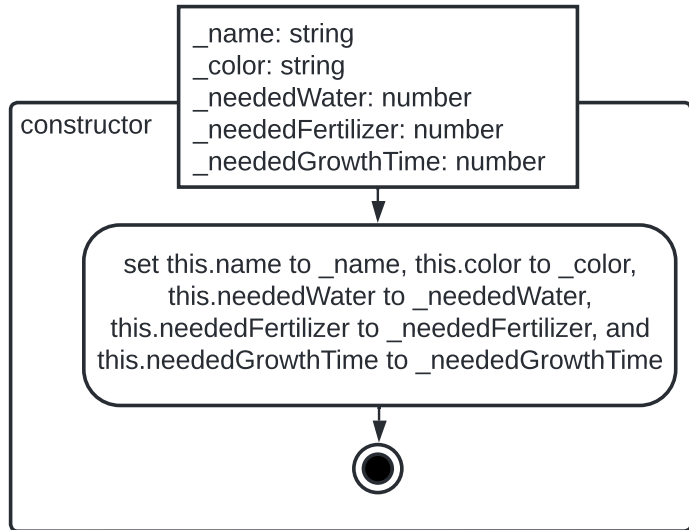


Globals
inputs: HTMLCollectionOf<HTMLInputElement>

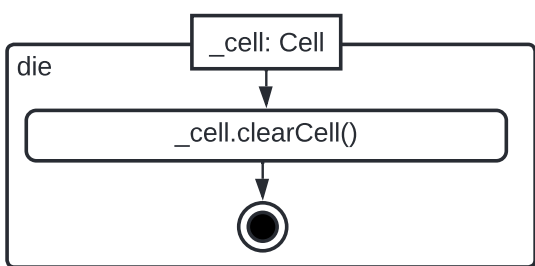
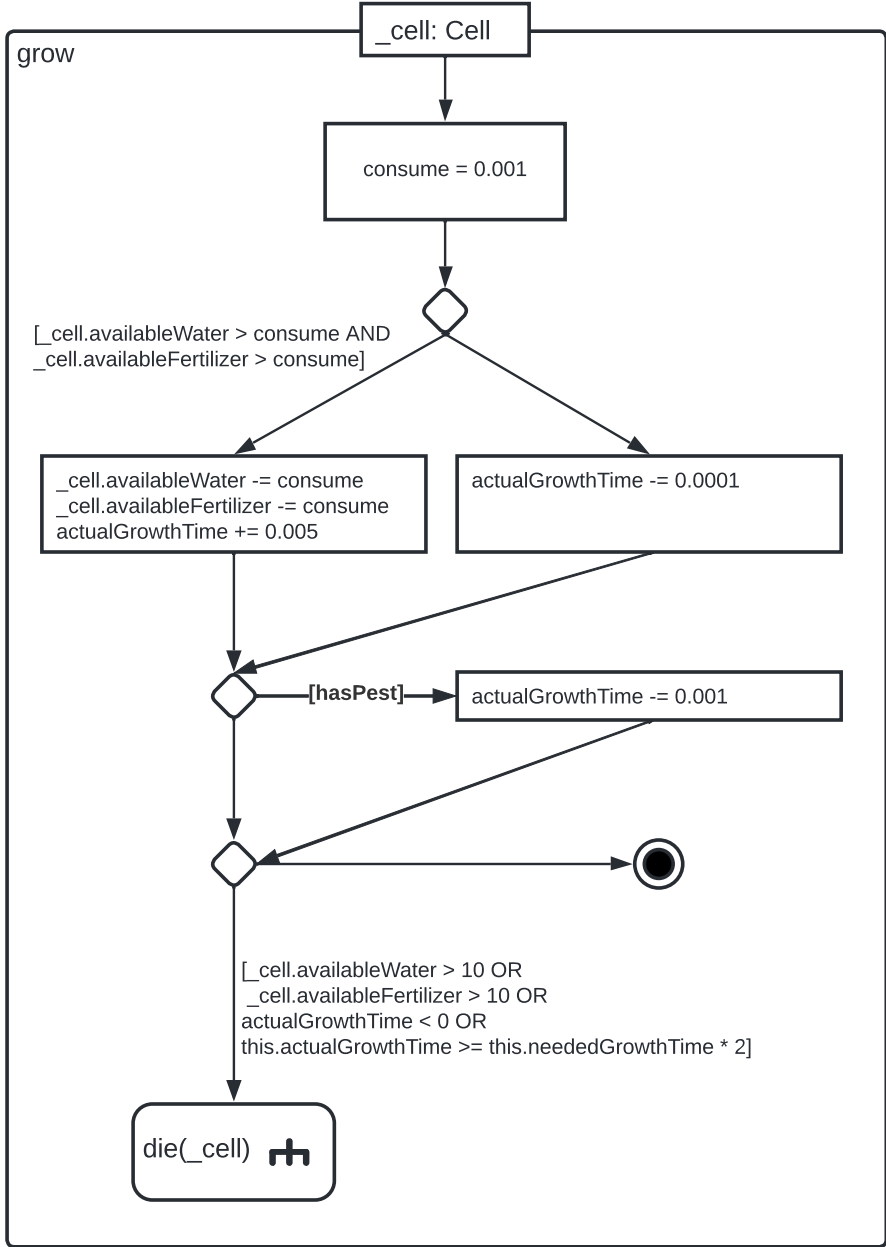




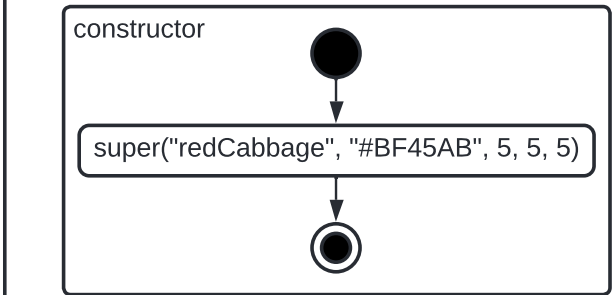
Vegetable



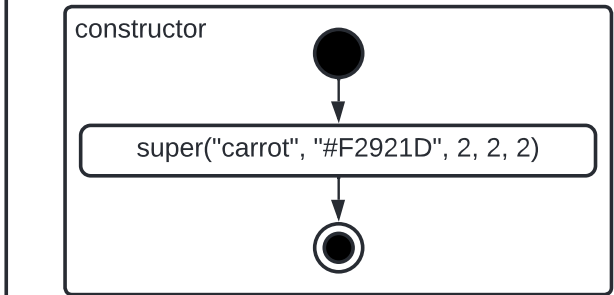
grow



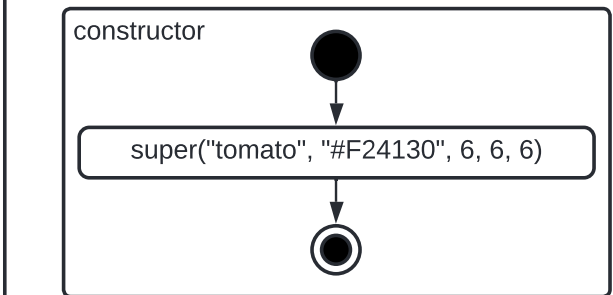
RedCabbage



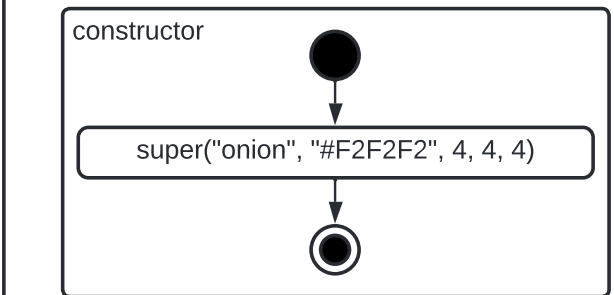
Carrot



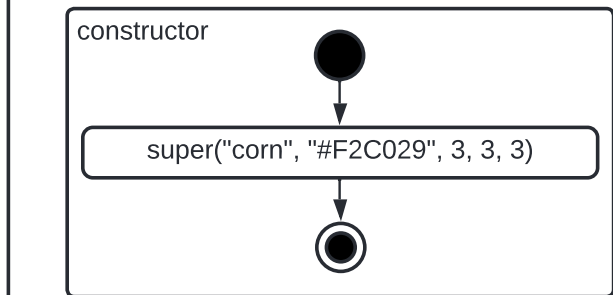
Tomato



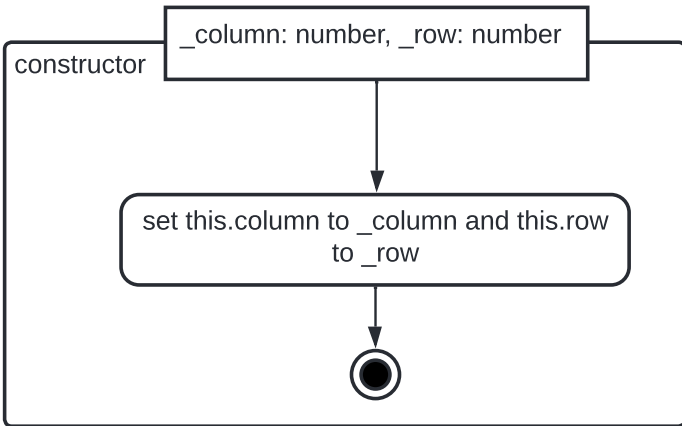
Onion



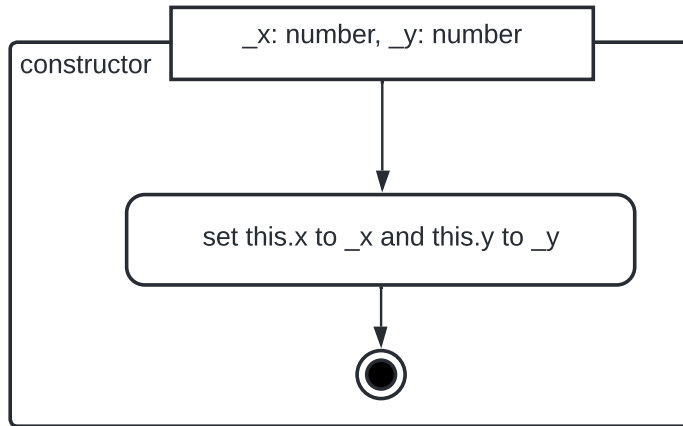
Corn

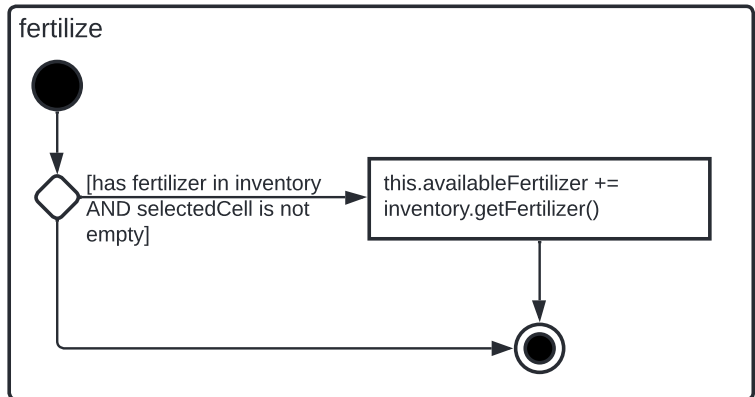
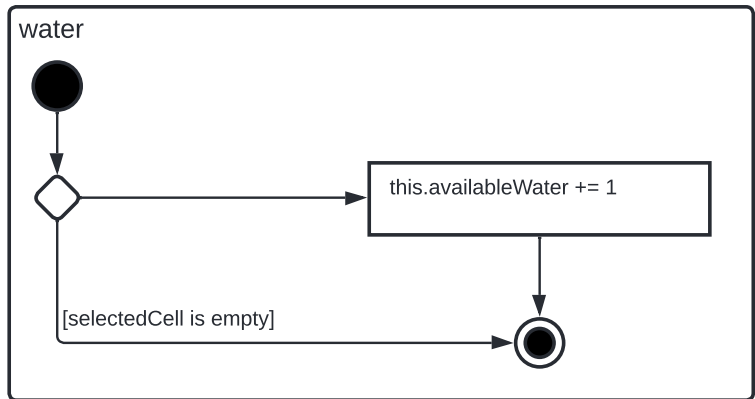
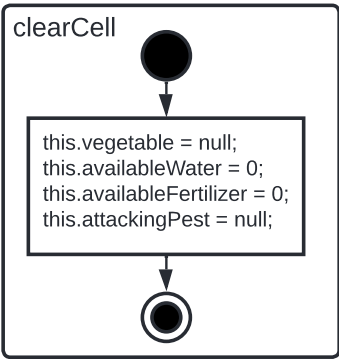
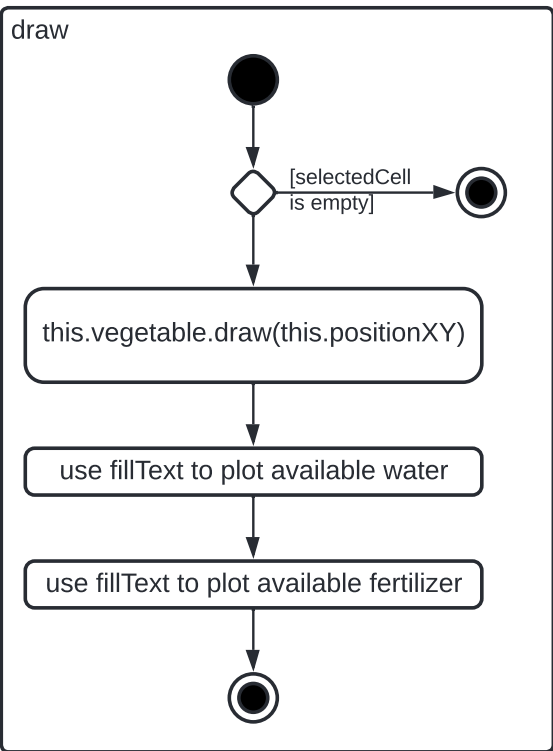
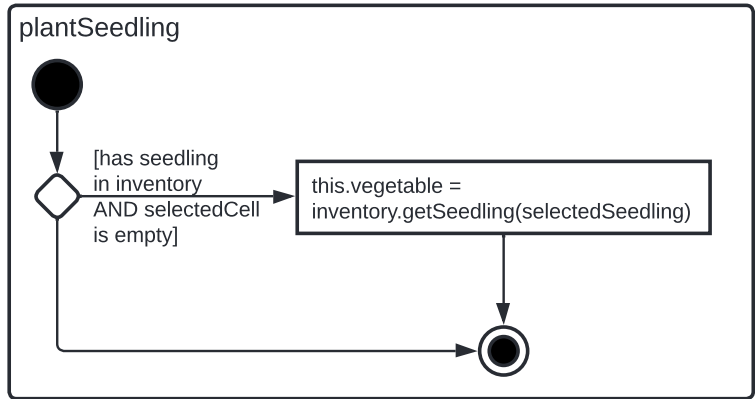
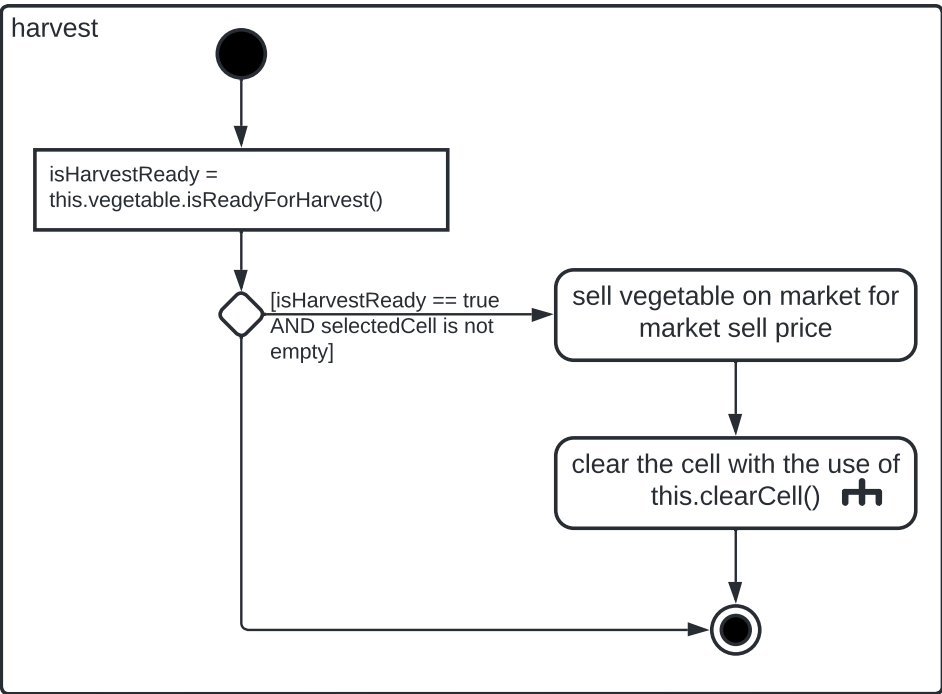
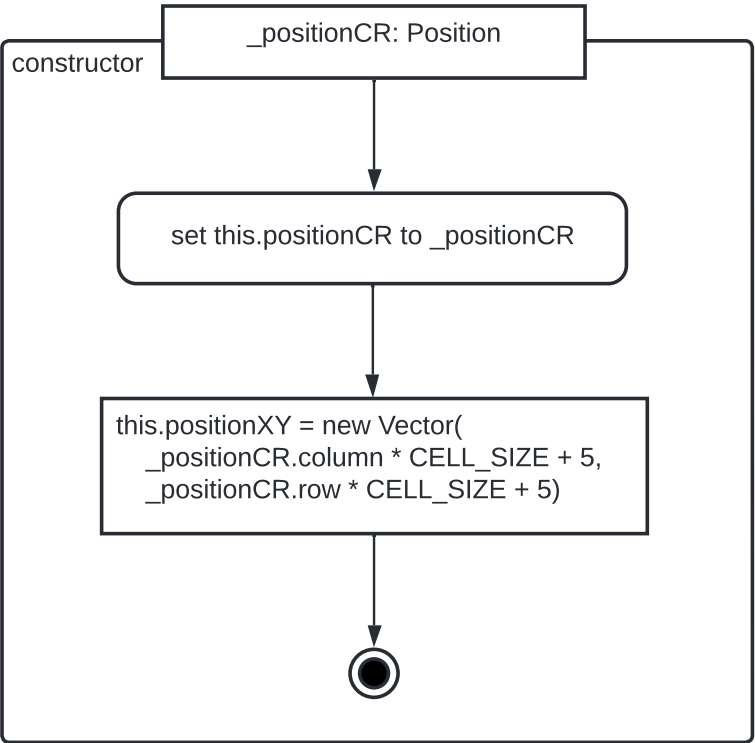


Position



Vector





Pest

constructor

_positionXY: Vector
_velocityXY: Vector
_radius: number

set this.positionXY to _positionXY,
this.velocityXY to _velocityXY, and
this.radius to _radius



draw



begin the draw path

create a whole arc (a
circle) with
this.positionXY and
this.radius

stroke the path



update



[this.positionXY.x + this.radius > CANVAS_WIDTH
OR this.positionXY.x - this.radius < 0]

this.velocityXY.x = -this.velocityXY.x

[this.positionXY.y + this.radius > CANVAS_HEIGHT
OR this.positionXY.y - this.radius < 0]

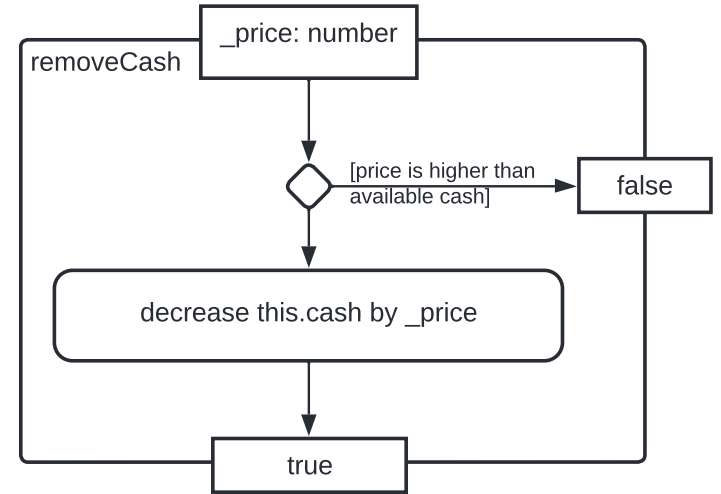
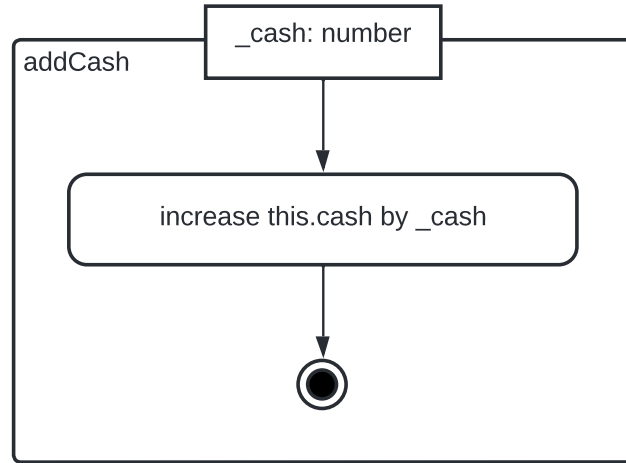
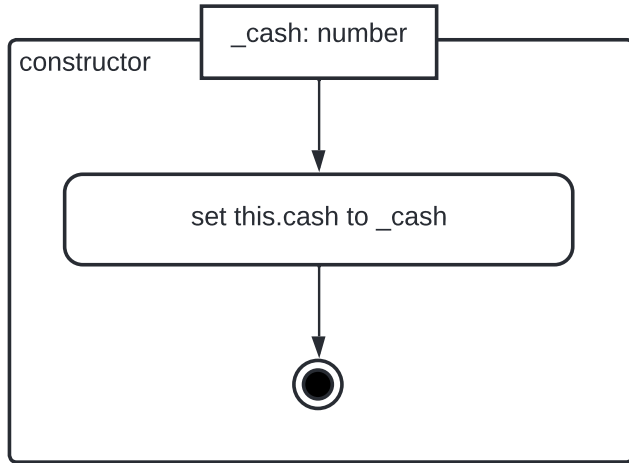
this.velocityXY.y = -this.velocityXY.y

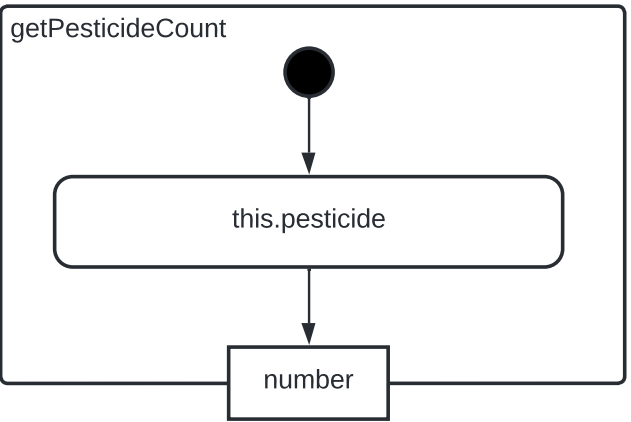
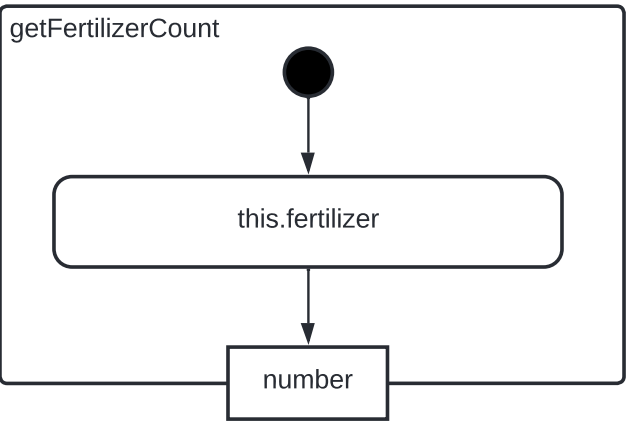
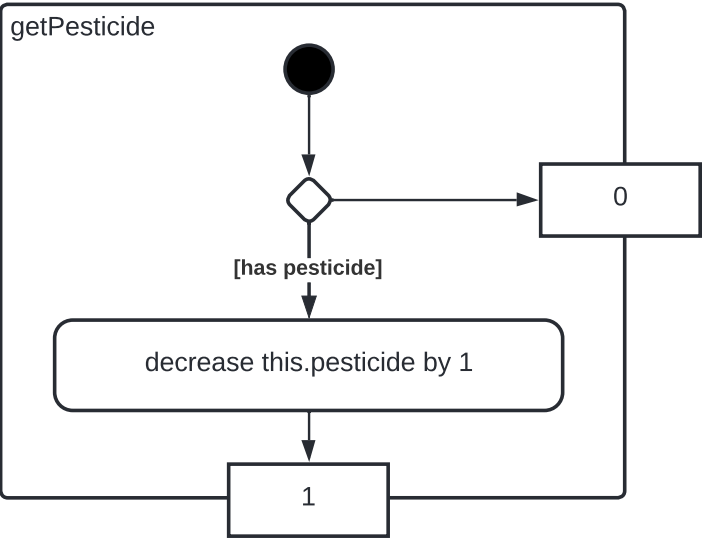
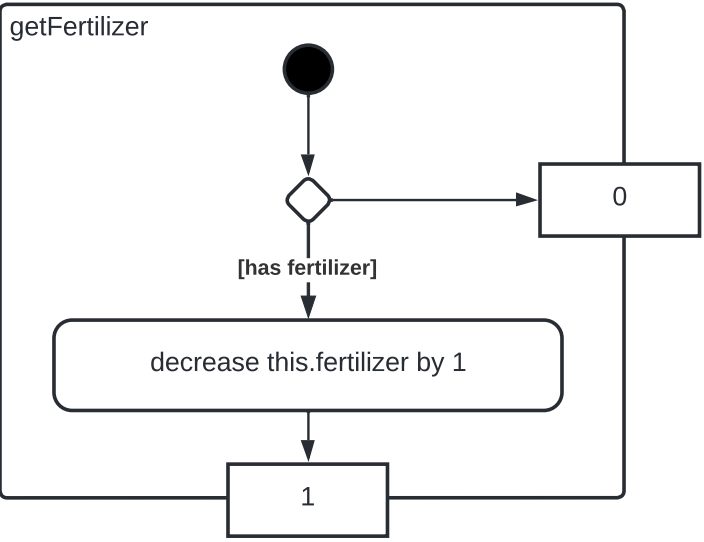
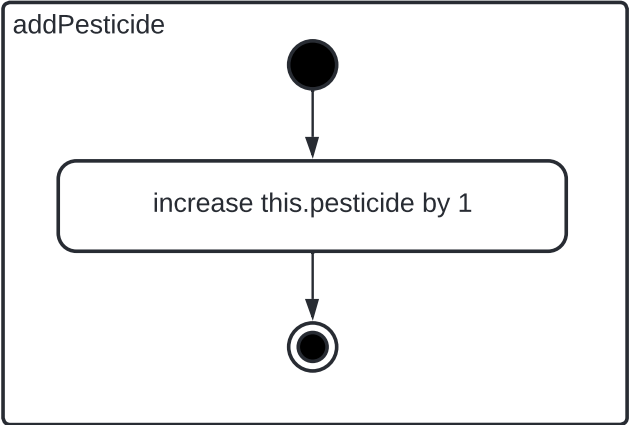
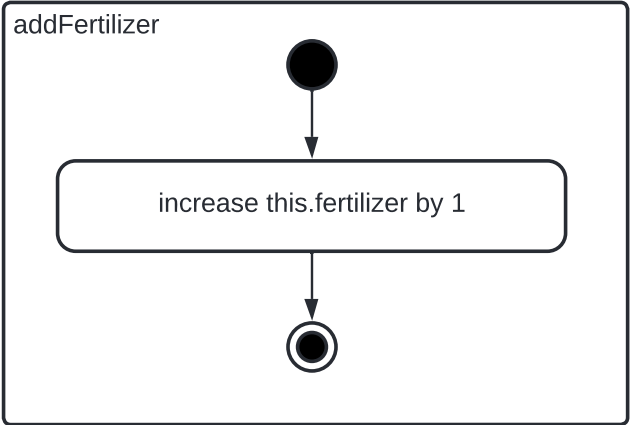
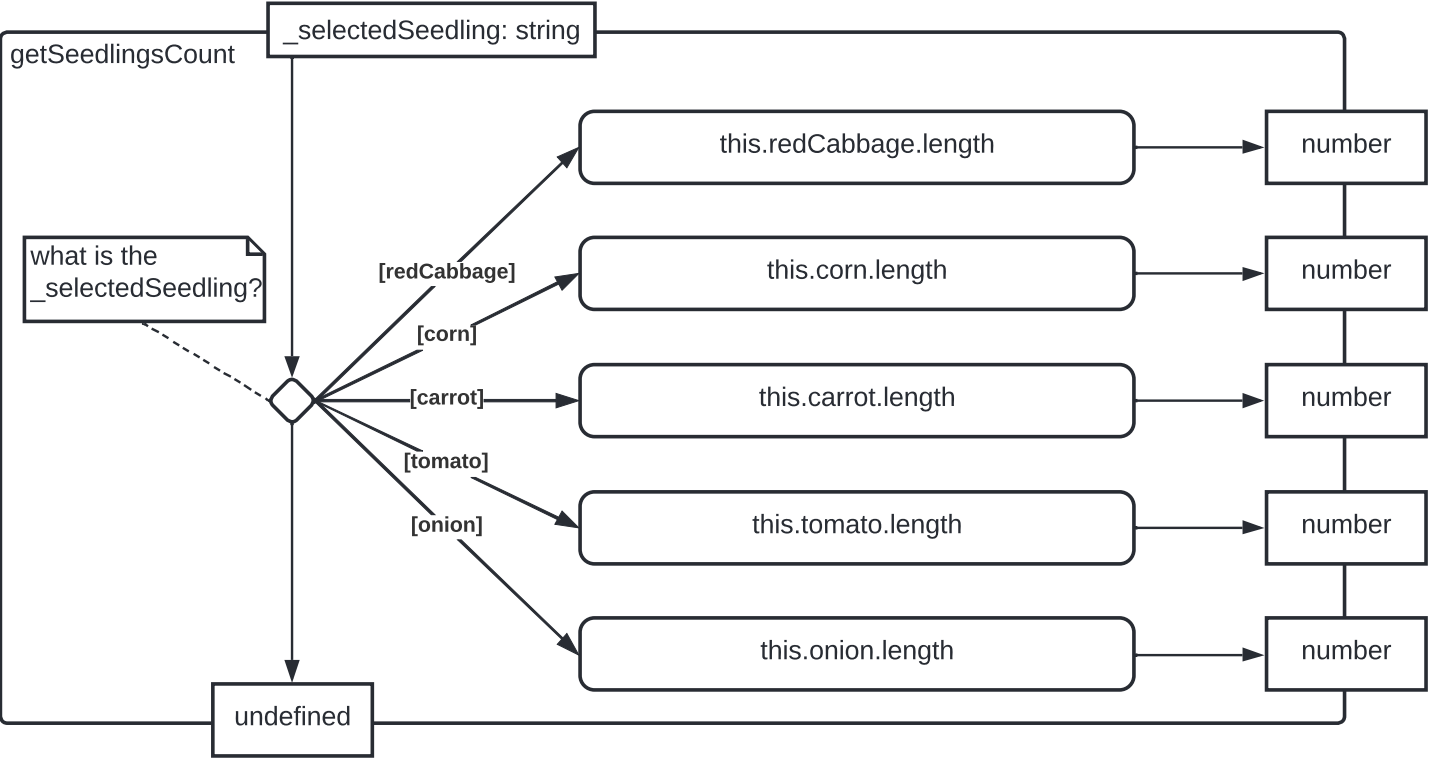
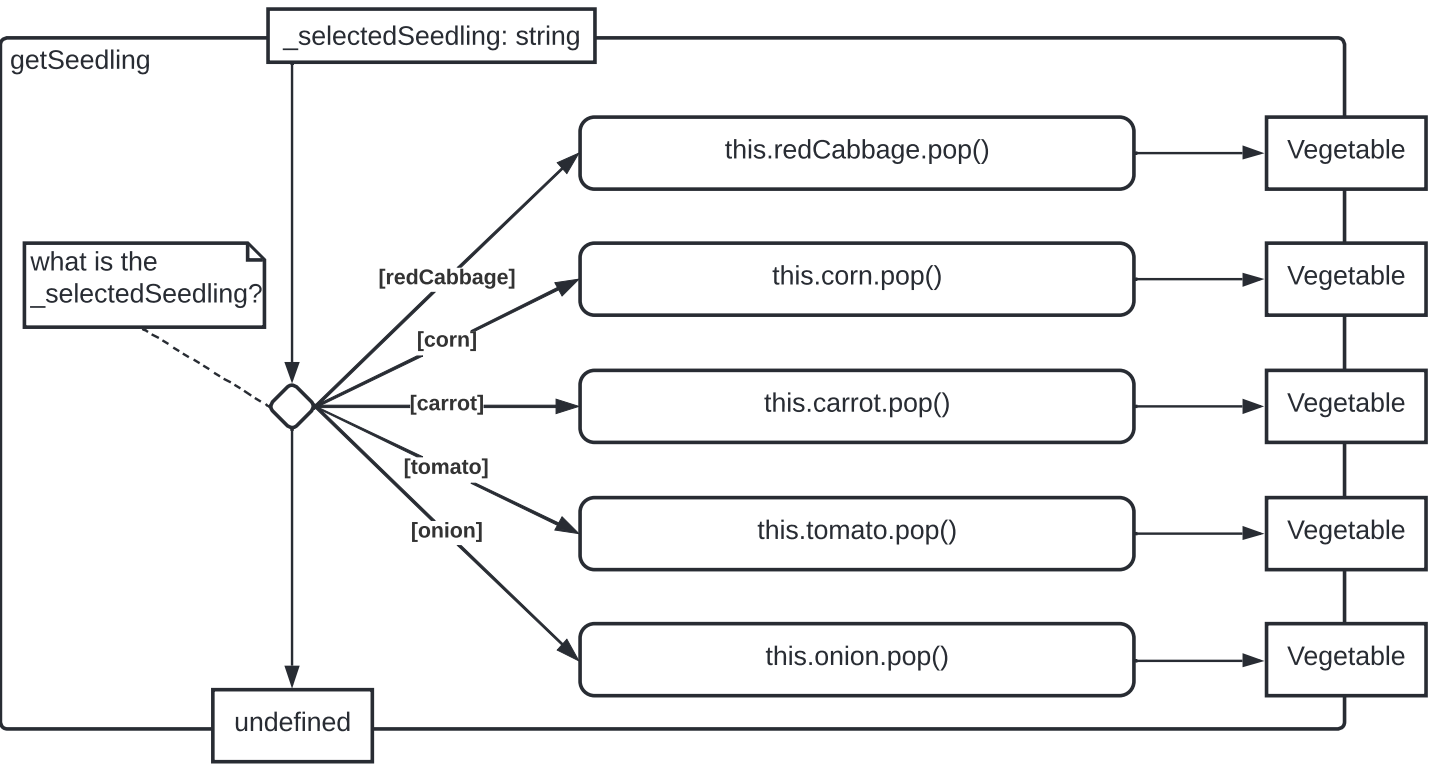
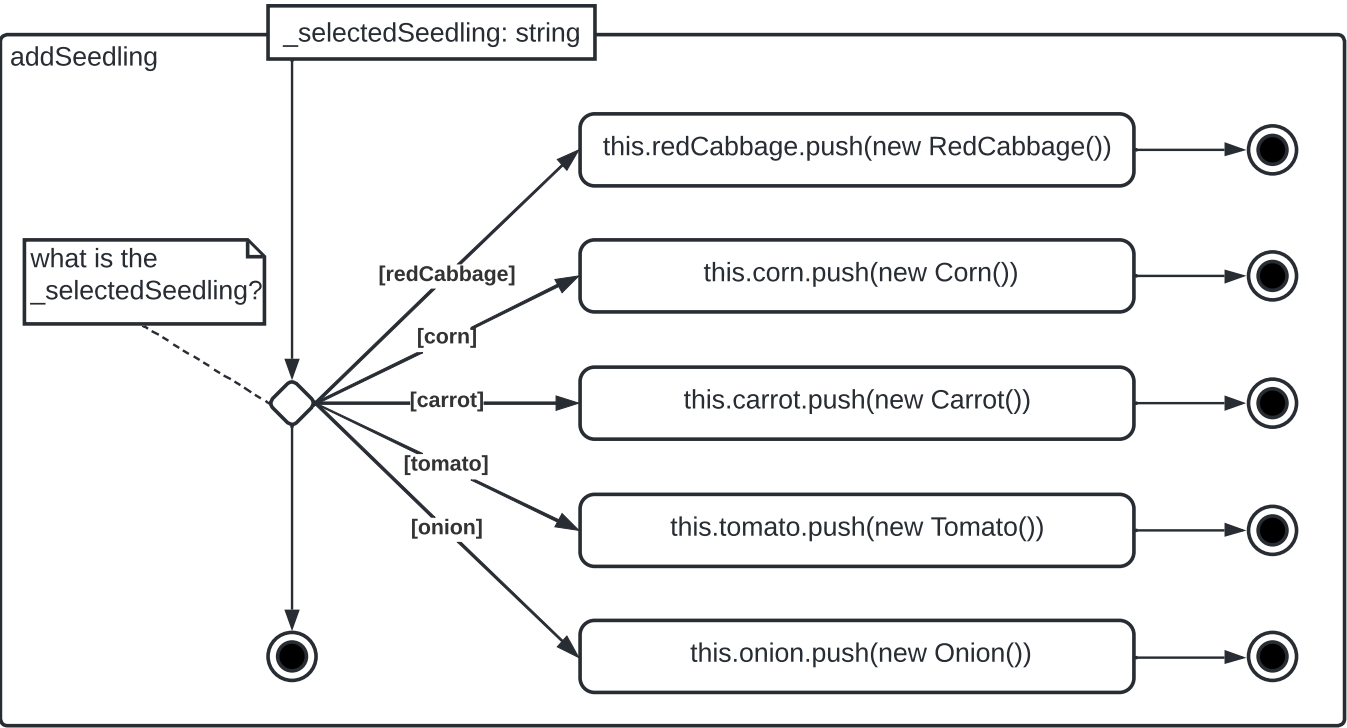
this.positionXY.x += this.velocityXY.x
this.positionXY.y += this.velocityXY.y

ensures that pest
bounces off the
canvas edges



Wallet





constructor

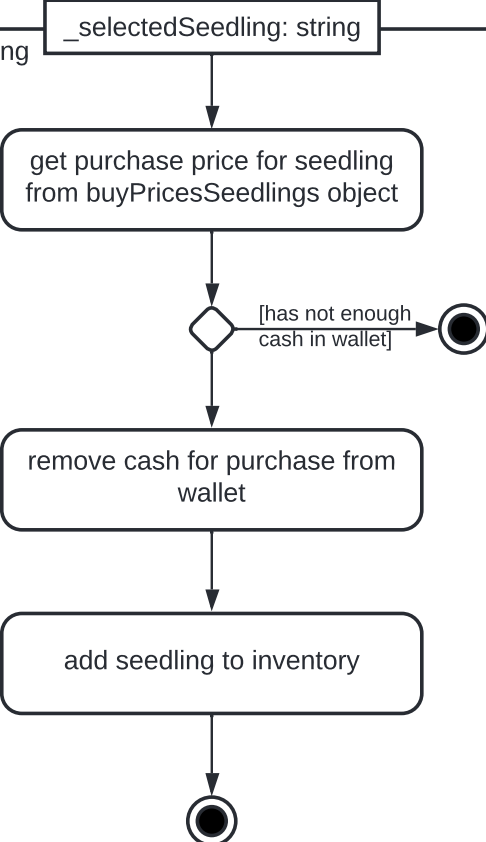
```
this.buyPricesSeedlings = {
  redCabbage: {
    current: 1,
    min: parseInt(localStorage.getItem("purchase-redCabbage-min")),
    max: parseInt(localStorage.getItem("purchase-redCabbage-max")),
  },
  corn: {
    current: 2,
    min: parseInt(localStorage.getItem("purchase-corn-min")),
    max: parseInt(localStorage.getItem("purchase-corn-max")),
  },
  carrot: {
    current: 3,
    min: parseInt(localStorage.getItem("purchase-carrot-min")),
    max: parseInt(localStorage.getItem("purchase-carrot-max")),
  },
  tomato: {
    current: 4,
    min: parseInt(localStorage.getItem("purchase-tomato-min")),
    max: parseInt(localStorage.getItem("purchase-tomato-max")),
  },
  onion: {
    current: 5,
    min: parseInt(localStorage.getItem("purchase-onion-min")),
    max: parseInt(localStorage.getItem("purchase-onion-max")),
  },
}
```

```
this.sellPricesVegetables = {
  redCabbage: {
    current: 10,
    min: parseInt(localStorage.getItem("sell-redCabbage-min")),
    max: parseInt(localStorage.getItem("sell-redCabbage-max")),
  },
  corn: {
    current: 20,
    min: parseInt(localStorage.getItem("sell-corn-min")),
    max: parseInt(localStorage.getItem("sell-corn-max")),
  },
  carrot: {
    current: 30,
    min: parseInt(localStorage.getItem("sell-carrot-min")),
    max: parseInt(localStorage.getItem("sell-carrot-max")),
  },
  tomato: {
    current: 40,
    min: parseInt(localStorage.getItem("sell-tomato-min")),
    max: parseInt(localStorage.getItem("sell-tomato-max")),
  },
  onion: {
    current: 50,
    min: parseInt(localStorage.getItem("sell-onion-min")),
    max: parseInt(localStorage.getItem("sell-onion-max")),
  },
}
```

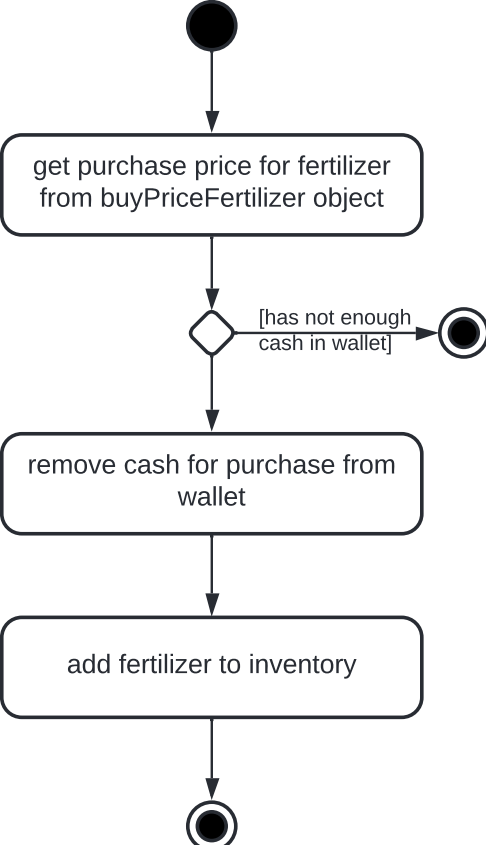
```
this.buyPriceFertilizer = {
  fertilizer: {
    current: 2,
    min: parseInt(localStorage.getItem("purchase-fertilizer-min")),
    max: parseInt(localStorage.getItem("purchase-fertilizer-max")),
  },
}
```

```
this.buyPricePesticide = {
  pesticide: {
    current: 2,
    min: parseInt(localStorage.getItem("purchase-pesticide-min")),
    max: parseInt(localStorage.getItem("purchase-pesticide-max")),
  },
}
```

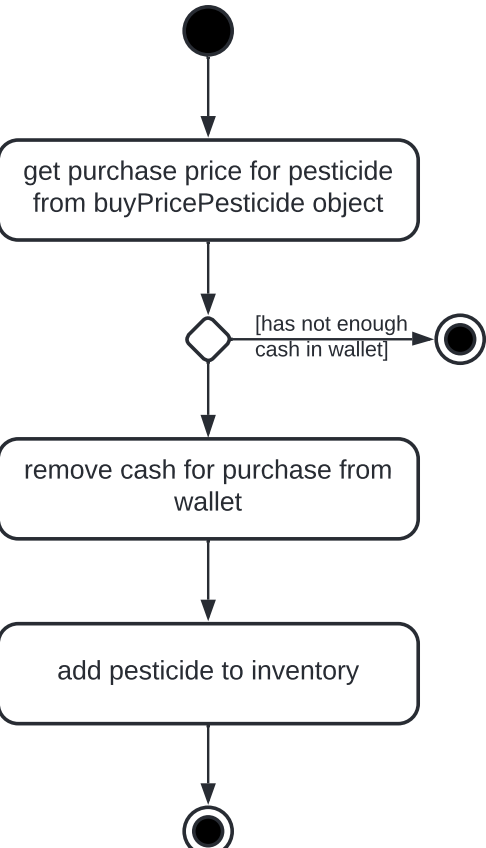
buySeedling



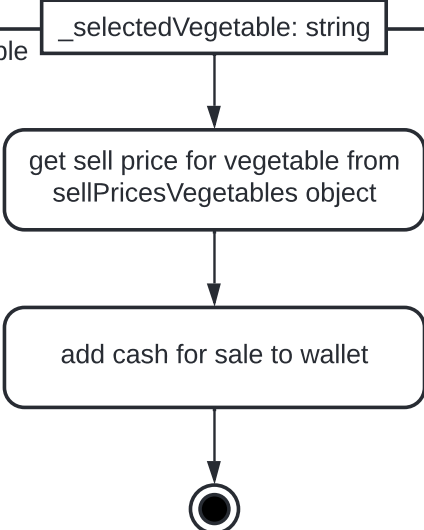
buyFertilizer



buyPesticide



sellVegetable



simulatePrices

